General Mathematics Units 3 & 4

Chapter 10 — Graphs, networks and trees

Aboriginal and Torres Strait Islander content has been excluded as accuracy cannot be guaranteed, and it is better to be respectful than cover the entire curriculum inaccurately.

Significant effort has been made to ensure this content is legal. Please send any areas of concern to admin@vwx.au with appropriate document/legal references so errors can be verified and changes be made.

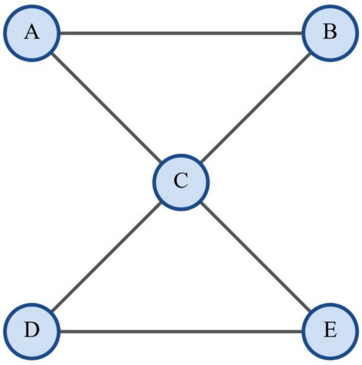
Contents

Theory

Many practical situations involve a set of objects and the connections between them: towns joined by roads, people who know one another, computers joined by cables, or tasks that must follow one another. A **graph** (or **network**) is a diagram that models this kind of information. A graph is made up of

- **vertices** (also called **nodes**) — dots that represent the objects, and
- **edges** — lines that join pairs of vertices to represent a connection.

Only *which* vertices are joined matters, not the length or shape of an edge or the exact position of a vertex. The same graph can therefore be drawn in many different ways.



The graph above has vertices A, B, C, D, E and edges AB, AC, BC, CD, CE and DE — six edges in all. Two vertices joined by an edge are said to be **adjacent**, and an edge is said to be **incident** with each of the two vertices it joins.

Degree of a vertex.

The **degree** of a vertex is the number of edge-ends meeting at it, written $deg(A)$. In the graph above $deg(C)=4$ because four edges meet at C, while $deg(A)=2$. To find a degree, simply count the edges leaving the vertex.

The handshake result.

Every edge has two ends, and each end adds 1 to the degree of the vertex it meets. Adding the degrees of all the vertices therefore counts every edge exactly twice:

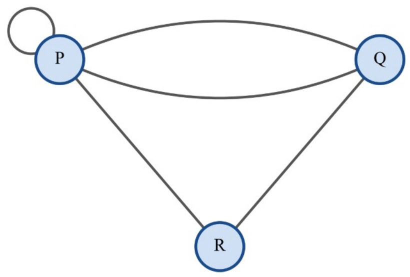
$$deg(v_1)+deg(v_2)+\cdots+deg(v_n)=2e,$$

where e is the number of edges. In words, **the sum of the degrees equals twice the number of edges**. (This is often called the *handshake* result: if several people shake hands, the total number of hands shaken is twice the number of handshakes.) For the graph above the degrees are $2, 2, 4, 2, 2$, which sum to $12=2\times 6$.

Two useful consequences follow. First, the sum of the degrees is always **even**. Second, the number of vertices of **odd degree** must be even, because the odd degrees have to pair up to give an even total.

Loops and multiple edges.

An edge that joins a vertex to **itself** is a **loop**; a loop adds 2 to the degree of its vertex, since both of its ends meet there. When two or more edges join the **same pair** of vertices they are called **multiple edges** (or parallel edges).



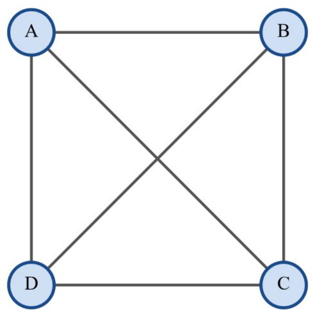
Here there is a loop at P and a double edge between P and Q. The degrees are $deg(P)=5$ (the loop counts twice, plus the two edges to Q and one to R), $deg(Q)=3$ and $deg(R)=2$, giving a sum of $10=2\times5$ for the five edges.

Types of graph.

- A **simple graph** has **no loops and no multiple edges** — every edge joins two *different* vertices, and each pair of vertices is joined by at most one edge. The first graph above is simple; the graph with the loop is not.
- A graph is **connected** if you can travel from any vertex to any other along edges. If a graph falls into separate pieces it is **disconnected**, and each separate piece is called a **connected component**.
- A **complete graph** is a simple graph in which **every** pair of vertices is joined by an edge. The complete graph on n vertices is written K_n; because each of the n vertices is joined to the other $n-1$ vertices, it has

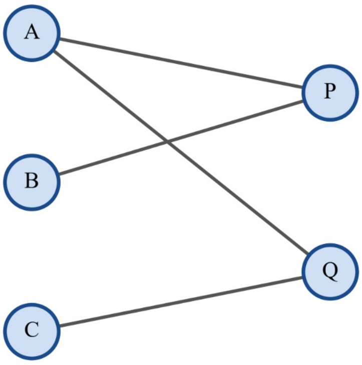
  $$e=\frac{n(n-1)}{2}\text{ edges,}$$

  and every vertex has degree $n-1$.



The graph shown is K_4: it has $\frac{4\times3}{2}=6$ edges and every vertex has degree 3.

- A graph is **bipartite** if its vertices can be split into **two groups** so that every edge joins a vertex in one group to a vertex in the other — no edge joins two vertices in the same group. Bipartite graphs model situations with two kinds of object, such as students and the subjects they take.

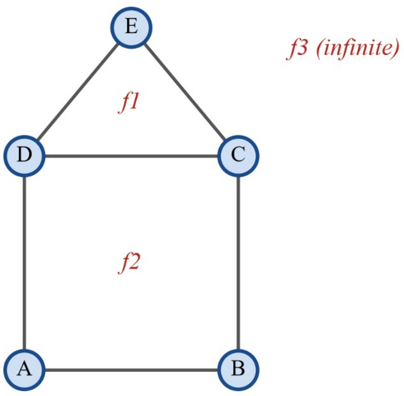


Here the two groups are $\{A, B, C\}$ and $\{P, Q\}$, and every edge crosses between them. A quick test is to try to colour the vertices with two colours so that no edge joins two vertices of the same colour; if this is possible the graph is bipartite.

- A **subgraph** of a graph G is a graph made up of **some of the vertices and some of the edges of** G (each chosen edge must have both of its end-vertices chosen as well). Deleting vertices or edges from a graph always leaves a subgraph.

Planar graphs, faces and Euler's formula.

A graph is **planar** if it can be drawn in the plane so that **no two edges cross** (except where they meet at a vertex). It does not matter that a *first* drawing has crossings; what matters is whether the graph *can* be redrawn without any. A planar drawing divides the plane into **faces** — the regions bounded by edges — and we always count the single unbounded region outside the graph, the **infinite face**, as one of the faces.



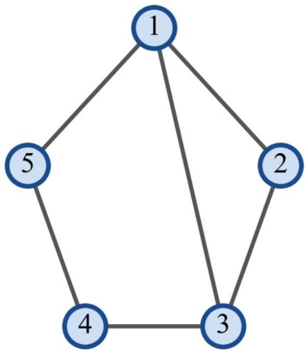
The connected planar graph above has $v=5$ vertices and $e=6$ edges. Its faces are the triangle f_1, the square f_2 and the infinite region f_3 outside — so $f=3$. These three numbers satisfy **Euler's formula**: for any **connected planar** graph

$$v-e+f=2.$$

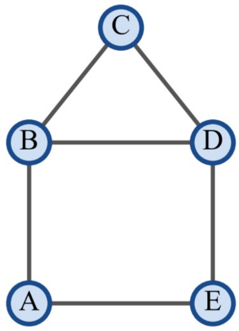
Checking, $v-e+f=5-6+3=2$. Euler's formula lets us find any one of v, e or f when the other two are known; for example a connected planar graph with $v=5$ and $e=6$ must have $f=2-v+e=3$ faces, however it is drawn.

Isomorphic graphs.

Because only *which* vertices are joined matters, the same graph can be drawn in many different ways — and two graphs that look quite different may in fact be the same graph in disguise. Two graphs are **isomorphic** if there is a **one-to-one correspondence** (a pairing) between their vertices that **preserves adjacency**: whenever two vertices are joined by an edge in one graph, the two corresponding vertices are joined by an edge in the other, and vice versa.



Graph 1



Graph 2

Graph 1 (a pentagon with one extra edge) and Graph 2 (drawn as a “house”) look nothing alike, yet they are isomorphic. The pairing

$$1 \leftrightarrow B, 2 \leftrightarrow C, 3 \leftrightarrow D, 4 \leftrightarrow E, 5 \leftrightarrow A$$

matches every edge of Graph 1 with an edge of Graph 2 — for instance edge 12 becomes BC, and the extra edge 13 becomes BD — so the two drawings really do show the same graph.

If two graphs are isomorphic then the correspondence pairs up their vertices and edges exactly, so they must share every structural feature. In particular:

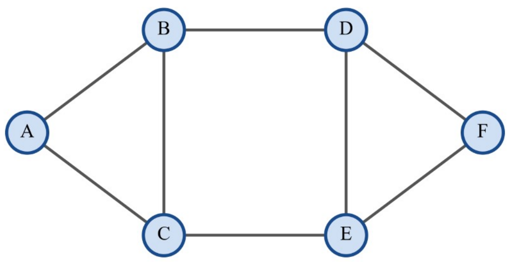
- they have the **same number of vertices**,
- they have the **same number of edges**, and
- they have the **same degree sequence** — the list of degrees is identical, and corresponding vertices have equal degree.

These three conditions are **necessary**: if any one of them fails the graphs *cannot* be isomorphic, and checking them is the quickest way to show two graphs are **not** isomorphic. They are **not sufficient**, however — two graphs can agree on all three and still fail to be isomorphic. To *confirm* that two graphs are isomorphic you must actually **exhibit the correspondence** and check that it preserves every edge.

Worked examples

Example 1 — scaffolded

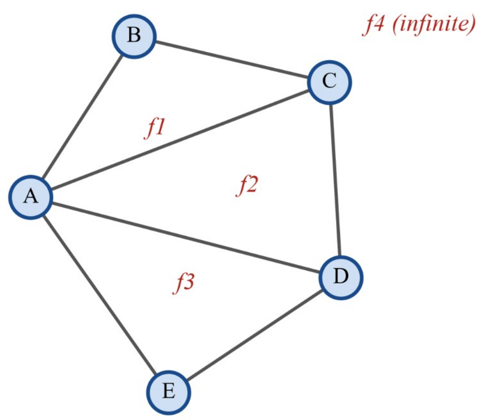
For the graph shown, find the degree of each vertex, verify the handshake result, and hence state the number of edges.



Working	Comment
$deg(A)=2$, $deg(B)=3$, $deg(C)=3$.	Count the edges meeting each vertex.
$deg(D)=3$, $deg(E)=3$, $deg(F)=2$.	B, C, D, E each have three edges.
Sum of degrees $=2+3+3+3+3+2=16$.	Add all the degrees.
Sum $=2e$, so $16=2e$ and $e=8$.	The handshake result: the total is twice the number of edges.
Counting the edges directly also gives 8.	Confirms the answer.

Example 2 — scaffolded

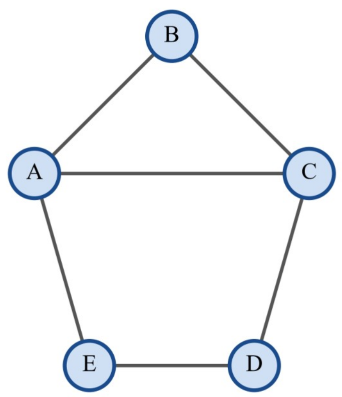
The graph shown is planar. Count the vertices, edges and faces (including the infinite face) and verify Euler's formula.



Working	Comment
Vertices A, B, C, D, E, so $v=5$.	Count the dots.
Edges $AB, BC, CD, DE, EA, AC, AD$, so $e=7$.	Count the lines.
Faces: f_1, f_2, f_3 inside and the infinite face outside, so $f=4$.	Always include the outside region.
$v-e+f=5-7+4=2$.	Euler's formula is satisfied.

Example 3 — unscaffolded

For the graph shown, state whether it is simple, whether it is connected, and write down its degree sequence. Is the graph complete? If not, how many edges must be added to make it complete?



The graph has no loops and no multiple edges, so it is **simple**. Any vertex can be reached from any other by travelling along edges, so it is **connected**.

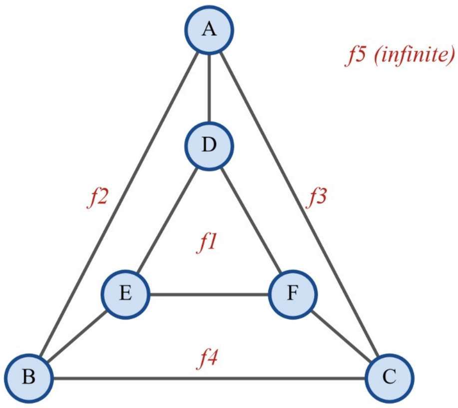
Counting the edges at each vertex gives $deg(A)=3$, $deg(B)=2$, $deg(C)=3$, $deg(D)=2$ and $deg(E)=2$, so the degree sequence (largest first) is $3,3,2,2,2$. The sum is $12=2\times 6$, confirming the graph has 6 edges.

A complete graph on 5 vertices, K_5, would have $\frac{5\times 4}{2}=10$ edges with every vertex of degree 4. This graph has only 6 edges, so it is **not** complete.

$\therefore$ the graph is simple and connected with degree sequence $3,3,2,2,2$; it is not complete, and $10-6=4$ further edges would be needed to make it K_5.

Example 4 — unscaffolded

The graph shown is planar (it is a triangular prism drawn with one triangle inside the other). Count v, e and f, verify Euler's formula, and confirm the handshake result.



There are six vertices, so $v=6$. The edges are the outer triangle AB, BC, CA, the inner triangle DE, EF, FD, and the three joining edges AD, BE, CF, giving $e=9$.

The planar drawing has the inner triangle f_1, the three regions f_2, f_3, f_4 between the triangles, and the infinite face f_5 outside, so $f=5$. Then

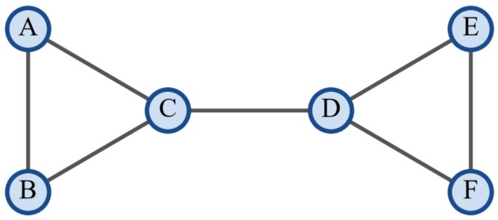
$$v-e+f=6-9+5=2,$$

so Euler's formula holds. Every vertex has three edges meeting it, so each degree is 3 and the degree sum is $6\times 3=18=2\times 9=2e$, which confirms the handshake result.

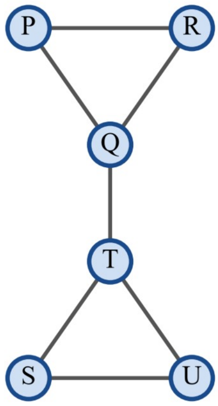
$\therefore$ $v=6$, $e=9$, $f=5$ with $v-e+f=2$, and the nine edges match the degree sum of 18.

Example 5 — scaffolded

Decide whether the two graphs shown are isomorphic. If they are, give a correspondence between their vertices and check that it preserves the edges.



Graph 1



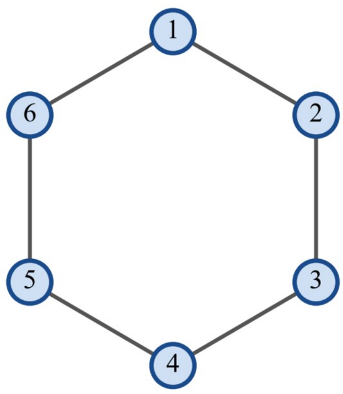
Graph 2

Working	Comment
Graph 1 has 6 vertices and 7 edges; Graph 2 has 6 vertices and 7 edges.	The vertex and edge counts agree, so isomorphism is still possible.
Degree sequence of Graph 1 is $3,3,2,2,2,2$, with $deg(C)=deg(D)=3$.	Count the edges meeting each vertex.
Degree sequence of Graph 2 is $3,3,2,2,2,2$, with $deg(Q)=deg(T)=3$.	The degree sequences match as well.
The two degree-3 vertices are joined in each graph (CD and QT), so pair $C \leftrightarrow Q$ and $D \leftrightarrow T$.	Match vertices of equal degree, using adjacency to decide which is which.
C's triangle $\{A,B\}$ pairs with Q's triangle $\{P,R\}$, and D's triangle $\{E,F\}$ with T's triangle $\{S,U\}$: take $A \leftrightarrow P$, $B \leftrightarrow R$, $E \leftrightarrow S$, $F \leftrightarrow U$.	Extend the correspondence to the remaining vertices.
Check: $AB \to PR$, $BC \to RQ$, $CA \to QP$, $CD \to QT$, $DE \to TS$, $EF \to SU$, $FD \to UT$ — all 7 are edges of Graph 2.	Every edge is preserved, so the correspondence is an isomorphism.

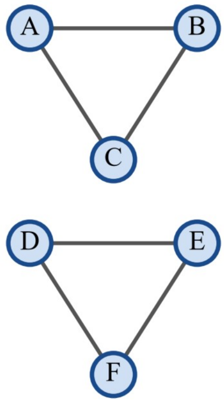
$\therefore$ the graphs are isomorphic, with correspondence $ABCDEF \leftrightarrow PRQTSU$.

Example 6 — unscaffolded

Each of the two graphs shown has 6 vertices and 6 edges, and every vertex has degree 2. Decide whether they are isomorphic.



Graph 1



Graph 2

Both graphs pass the first checks: each has 6 vertices, each has 6 edges, and each has degree sequence $2,2,2,2,2,2$. These conditions are necessary but not sufficient, so they do not settle the question — we must look at the actual pattern of connections.

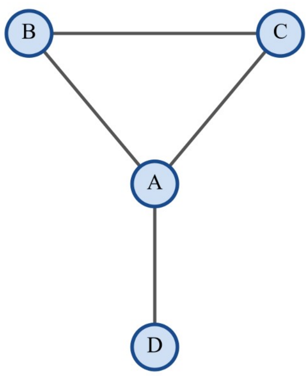
Graph 1 is a single six-sided cycle, so it is **connected**: you can travel from any vertex to any other along edges. Graph 2 is made of two separate triangles, so it is **disconnected** — it has two connected components, and there is no way to travel from a vertex of one triangle to a vertex of the other. A correspondence that preserved adjacency would have to turn a connected graph into a connected graph, so no such correspondence can exist. (Equivalently, Graph 2 contains triangles — cycles of length 3 — while the shortest cycle in Graph 1 has length 6, so the two cannot be matched.)

$\therefore$ the graphs are **not** isomorphic: they have the same degree sequence, but one is connected and the other is not.

Practice questions

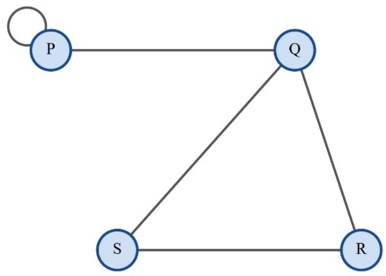
Scaffolded

Q1. For the graph shown:



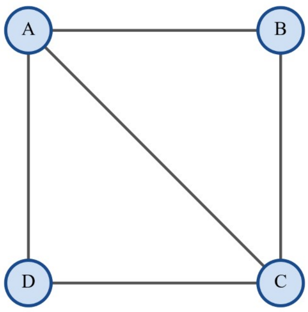
(a) Write down the vertices.
(b) Write down the edges.
(c) State the degree of each vertex.
(d) Find the sum of the degrees, and hence use the handshake result to state the number of edges.

Q2. The graph shown has a loop at P.



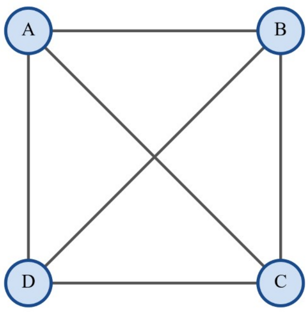
(a) State the degree of each vertex (remember a loop adds 2 to the degree).
(b) Find the sum of the degrees.
(c) How many edges does the graph have?
(d) Is the graph simple? Give a reason.

Q3. The graph shown is planar.



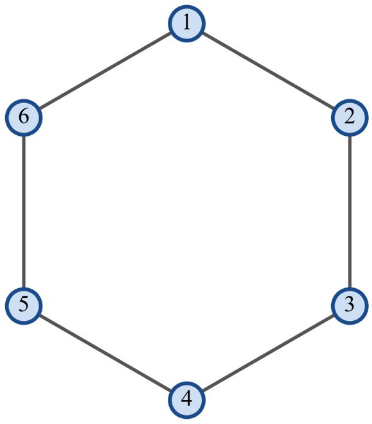
(a) Count the vertices v and the edges e.
(b) Count the faces f, including the infinite face.
(c) Verify Euler's formula $v-e+f=2$.

Q4. The graph shown is the complete graph K_4.



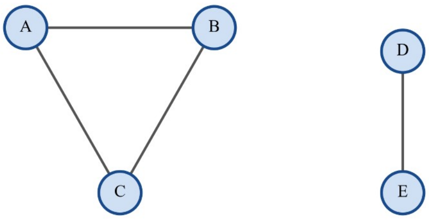
(a) How many edges does it have, and what is the degree of each vertex?
(b) Using $e=\dfrac{n(n-1)}{2}$, find the number of edges of the complete graph K_6.
(c) State the degree of each vertex in K_6.

Q5. A graph is bipartite if its vertices can be split into two groups so that every edge joins a vertex in one group to a vertex in the other.



(a) Colour the vertices of the graph shown using two colours so that no edge joins two vertices of the same colour.
(b) Write down the two groups.
(c) Hence state whether the graph is bipartite.

Q6. For the graph shown:



(a) Is the graph connected? Give a reason.
(b) How many connected components does it have?
(c) State the degree of each vertex and verify the handshake result.
(d) Write down the edges of the subgraph consisting of just the triangle on A, B and C.

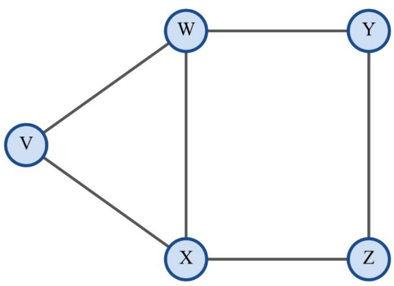
Unscaffolded

Q7. A graph has degree sequence $4, 3, 3, 2, 2$. How many edges does it have?

Q8. A graph has 6 vertices, each of degree 3. How many edges does it have?

Q9. Explain why no graph can have vertices with exactly the degrees $5, 3, 2, 2, 1$.

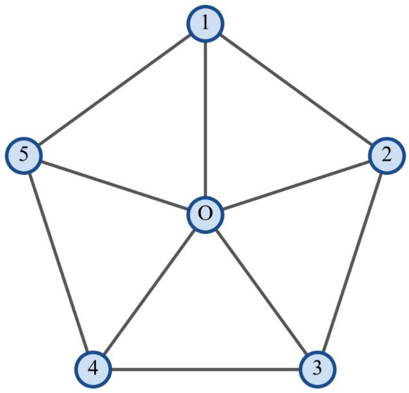
Q10. For the graph shown, state the degree of each vertex and use the handshake result to find the number of edges.



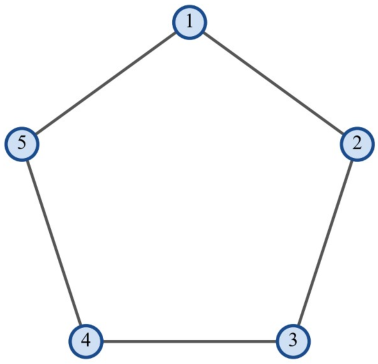
Q11. Eight tennis players enter a round-robin tournament in which every player plays every other player exactly once. Model the tournament as a complete graph and hence find the number of matches played.

Q12. A connected planar graph has 10 vertices and 15 edges. Use Euler's formula to find the number of faces.

Q13. The graph shown is planar. Count the vertices, edges and faces (including the infinite face) and verify Euler's formula.

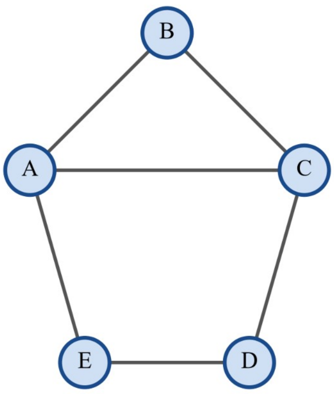


Q14. Show that the graph shown, a cycle on five vertices, is not bipartite.



Q15. Consider the complete bipartite graph $K_{3,4}$, in which each of 3 vertices in one group is joined to each of 4 vertices in the other group. (a) How many vertices does it have? (b) How many edges? (c) State the degree of each vertex.

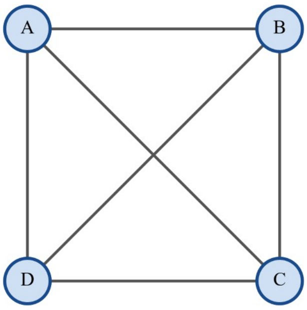
Q16. For the graph G shown, decide whether each of the following is a subgraph of G.



(i) The vertices $\{A, B, C\}$ with edges AB, BC, AC.
(ii) The vertices $\{A, B, C, D, E\}$ with edges AB, BC, CD, DE, EA, BD.

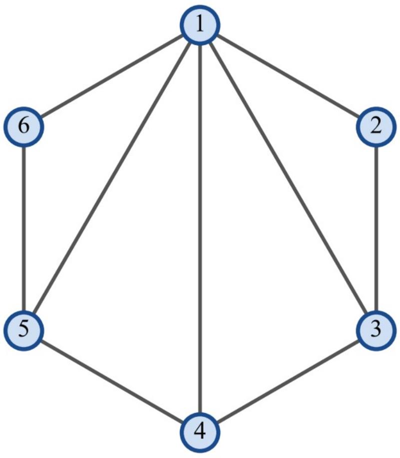
Q17. A connected planar graph has 6 faces and 8 edges. Use Euler's formula to find the number of vertices.

Q18. For the graph shown, state, with a reason in each case, whether it is (a) simple, (b) connected, (c) complete, (d) bipartite.



Q19. A graph is connected and has 7 vertices. What is the smallest possible number of edges? Explain your answer.

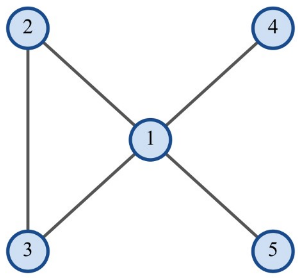
Q20. The graph shown is planar.



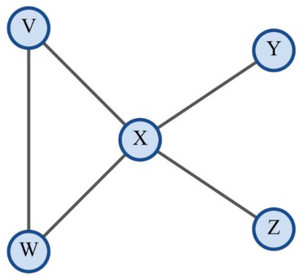
(a) State the degree of each vertex and verify the handshake result.
(b) Count the faces (including the infinite face) and verify Euler's formula.
(c) State, with reasons, whether the graph is simple, connected, complete and bipartite.

Isomorphic graphs

Q21. Decide whether the two graphs shown are isomorphic.



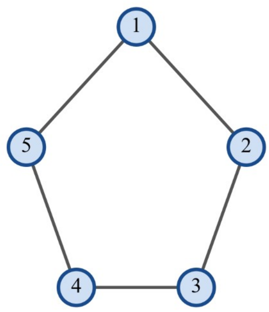
Graph 1



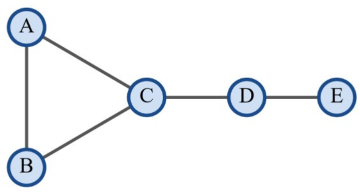
Graph 2

(a) Write down the number of vertices and the number of edges of each graph.
(b) Write down the degree sequence of each graph.
(c) Each graph has exactly one vertex of degree 4. Pair those two vertices, then extend to a full correspondence between the vertices.
(d) Check that your correspondence turns every edge of Graph 1 into an edge of Graph 2, and hence state whether the graphs are isomorphic.

Q22. Decide, giving a reason, whether the two graphs shown are isomorphic.

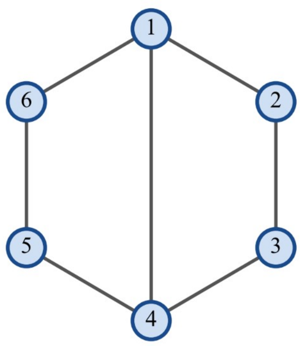


Graph 1

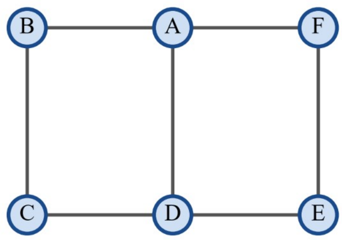


Graph 2

Q23. The two graphs shown are isomorphic. Give a correspondence between their vertices and verify that it preserves the edges.



Graph 1



Graph 2

Answers

Q1. Vertices A, B, C, D; edges AB, AC, AD, BC; degrees $A\,3$, $B\,2$, $C\,2$, $D\,1$; sum $8 = 2 \times 4$, so 4 edges. **Q2.** Degrees $P\,3$, $Q\,3$, $R\,2$, $S\,2$; sum 10; 5 edges; not simple, because it has a loop. **Q3.** $v=4$, $e=5$, $f=3$; $4-5+3=2$. **Q4.** K_4: 6 edges, each degree 3. K_6: 15 edges, each degree 5. **Q5.** Groups $\{1, 3, 5\}$ and $\{2, 4, 6\}$; the graph is bipartite. **Q6.** Not connected (it has 2 components); degrees $A\,2$, $B\,2$, $C\,2$, $D\,1$, $E\,1$, sum $8 = 2 \times 4$; triangle subgraph edges AB, BC, CA. **Q7.** Sum $=14$, so 7 edges. **Q8.** 9 edges. **Q9.** The degrees add to 13, which is odd; but the sum of the degrees must equal $2e$, an even number, so it is impossible. **Q10.** Degrees $V\,2$, $W\,3$, $X\,3$, $Y\,2$, $Z\,2$; sum 12, so 6 edges. **Q11.** K_8; $\frac{8 \times 7}{2} = 28$ matches. **Q12.** $f = 2 - v + e = 2 - 10 + 15 = 7$ faces. **Q13.** $v=6$, $e=10$, $f=6$; $6-10+6=2$. **Q14.** Two-colouring fails: colouring $1, 2, 3, 4, 5$ alternately forces 5 and 1 to share a colour, yet they are adjacent. It is an odd cycle, so it is not bipartite. **Q15.** (a) 7 vertices; (b) 12 edges; (c) the 3 vertices have degree 4 and the 4 vertices have degree 3. **Q16.** (i) Yes; (ii) no, because edge BD is not an edge of G. **Q17.** $v = 2 - f + e = 2 - 6 + 8 = 4$ vertices. **Q18.** (a) Simple — no loops or multiple edges; (b) connected — a path joins any two vertices; (c) complete — every pair of the four vertices is joined (it is K_4); (d) not bipartite — it contains a triangle. **Q19.** 6 edges; a connected graph on n vertices needs at least $n-1$ edges. **Q20.** Degrees $1{:}5$, $2{:}2$, $3{:}3$, $4{:}3$, $5{:}3$, $6{:}2$, sum $18 = 2 \times 9$ (9 edges); $v=6$, $e=9$, $f=5$, $6-9+5=2$; simple and connected, not complete, not bipartite. **Q21.** Both graphs have 5 vertices and 5 edges; both have degree sequence $4, 2, 2, 1, 1$; the correspondence $12345 \leftrightarrow XVWYZ$ turns every edge of Graph 1 into an edge of Graph 2, so the graphs **are** isomorphic. **Q22.** Not isomorphic: Graph 1 has degree sequence $2, 2, 2, 2, 2$ but Graph 2 has degree sequence $3, 2, 2, 2, 1$, so the degree sequences differ. **Q23.** Correspondence $123456 \leftrightarrow ABCDEF$; every edge of Graph 1 maps to an edge of Graph 2, including the long diagonal $14 \rightarrow AD$, so it is an isomorphism.

Fully-worked solutions

Q1. The dots are the vertices A, B, C, D. The lines are the edges AB, AC, AD and BC. Counting the edges at each vertex, $deg(A)=3$ (to B, C, D), $deg(B)=2$ (to A, C), $deg(C)=2$ (to A, B) and $deg(D)=1$ (to A). The sum of the degrees is $3+2+2+1=8$. By the handshake result the sum equals $2e$, so $8=2e$ and $e=4$, agreeing with the four edges listed.

Q2. A loop adds 2 to a degree. At P there is the loop (counting 2) and the edge to Q, so $deg(P)=3$. At Q the edges to P, R and S give $deg(Q)=3$; at R the edges to Q and S give $deg(R)=2$; at S the edges to R and Q give $deg(S)=2$. The sum is $3+3+2+2=10$, so $2e=10$ and $e=5$. The graph is **not** simple, because a simple graph is not allowed to contain a loop.

Q3. The graph has vertices A, B, C, D, so $v=4$, and edges AB, BC, CD, DA and the diagonal AC, so $e=5$. The diagonal splits the square into two triangular regions, and there is the infinite region outside, giving $f=3$ faces. Then $v-e+f=4-5+3=2$, so Euler's formula is verified.

Q4. In K_4 every pair of the four vertices is joined, giving $\frac{4 \times 3}{2}=6$ edges, and each vertex is joined to the other three, so every degree is 3. For K_6 the same rule gives $e=\frac{6 \times 5}{2}=15$ edges, and each vertex is joined to the other 5 vertices, so every degree is 5.

Q5. Colour vertex 1 black. Following the edges around the six-sided cycle, its neighbours 2 and 6 must be white, their neighbours 3 and 5 must be black, and 4 must be white. No edge then joins two vertices of the same colour, so the two groups are $\{1, 3, 5\}$ (black) and $\{2, 4, 6\}$ (white). Because such a two-colouring exists, the graph **is** bipartite.

Q6. The vertices A, B, C form a triangle, while D and E are joined only to each other. There is no edge between the two parts, so you cannot travel from A to D: the graph is **not connected**. It has **two** connected components, $\{A, B, C\}$ and $\{D, E\}$. The degrees are $deg(A)=deg(B)=deg(C)=2$ and $deg(D)=deg(E)=1$, with sum $2+2+2+1+1=8=2 \times 4$, confirming 4 edges. Choosing just the triangle gives the subgraph with vertices A, B, C and edges AB, BC, CA.

Q7. The sum of the degrees is $4+3+3+2+2=14$. By the handshake result this equals $2e$, so $e=\frac{14}{2}=7$ edges.

Q8. The sum of the degrees is $6\times3=18$, so $2e=18$ and $e=9$ edges.

Q9. The sum of the listed degrees is $5+3+2+2+1=13$, which is odd. But the handshake result says the sum of the degrees must equal $2e$, twice a whole number, and so must be even. An odd total is therefore impossible, so no such graph exists. (Equivalently, the number of odd-degree vertices — here three of them, 5, 3 and 1 — would have to be even.)

Q10. Counting edges at each vertex gives $deg(V)=2$, $deg(W)=3$, $deg(X)=3$, $deg(Y)=2$ and $deg(Z)=2$. The sum is $2+3+3+2+2=12$, so $2e=12$ and $e=6$ edges.

Q11. Represent each player by a vertex and each match by an edge joining the two players. Since every player meets every other exactly once, the graph is the complete graph K_8. The number of matches is the number of edges, $\frac{8\times7}{2}=28$.

Q12. Euler's formula $v-e+f=2$ rearranges to $f=2-v+e$. With $v=10$ and $e=15$, $f=2-10+15=7$ faces.

Q13. The graph is a wheel: a central vertex joined to five outer vertices that form a five-sided cycle. There are $v=6$ vertices. The edges are the five cycle edges and the five spokes, so $e=10$. The planar drawing has five triangular regions inside the cycle plus the infinite region outside, so $f=6$. Then $v-e+f=6-10+6=2$, as required.

Q14. Try to two-colour the cycle $1-2-3-4-5-1$. Colour 1 black; then 2 must be white, 3 black, 4 white and 5 black. But 5 and 1 are joined by an edge and are both black, so no valid two-colouring exists. A cycle with an odd number of vertices can never be two-coloured, so the graph is not bipartite.

Q15. In $K_{3,4}$ one group has 3 vertices and the other has 4, so there are $3+4=7$ vertices in all. Every one of the 3 vertices is joined to every one of the 4 vertices, giving $3\times4=12$ edges. Each of the 3 vertices is joined to all 4 of the other group, so those vertices have degree 4; each of the 4 vertices is joined to all 3 of the first group, so those vertices have degree 3. (Check: $3\times4+4\times3=24=2\times12$.)

Q16. The graph G has edges AB, BC, CD, DE, EA and AC. A subgraph may use only vertices and edges that are already in G. (i) The edges AB, BC and AC are all edges of G, and A, B, C are all vertices of G, so this **is** a subgraph. (ii) The edge BD is **not** an edge of G (there is no line joining B and D), so this collection is **not** a subgraph of G.

Q17. Euler's formula rearranges to $v=2-f+e$. With $f=6$ and $e=8$, $v=2-6+8=4$ vertices.

Q18. The graph is K_4. (a) It is **simple**: there are no loops and no two edges join the same pair of vertices. (b) It is **connected**: a path of edges joins any vertex to any other. (c) It is **complete**: each of the four vertices is joined to the other three, so every possible edge is present. (d) It is **not bipartite**: the vertices cannot be split into two groups with no edge inside a group, because any three mutually joined vertices (a triangle) would need three colours.

Q19. To keep 7 vertices connected you need enough edges that every vertex can be reached from every other. A connected graph on n vertices must have at least $n-1$ edges (this minimum is achieved by a tree). With $n=7$ the smallest number of edges is $7-1=6$.

Q20. (a) The graph is a hexagon $1-2-3-4-5-6$ with extra edges from vertex 1 to 3, 4 and 5. Counting edges at each vertex, $deg(1)=5$, $deg(2)=2$, $deg(3)=3$, $deg(4)=3$, $deg(5)=3$ and $deg(6)=2$. The sum is $5+2+3+3+3+2=18=2\times9$, so there are 9 edges. (b) The three extra edges cut the inside of the hexagon into four triangular regions, and there is the infinite region outside, so $f=5$. With $v=6$ and $e=9$, $v-e+f=6-9+5=2$, verifying Euler's formula. (c) The graph is **simple** (no loops or multiple edges) and **connected** (any vertex can be reached from any other). It is **not complete**, since K_6 would need $\frac{6\times5}{2}=15$ edges but this graph has only 9. It is **not bipartite**, because it contains triangles such as $1-2-3$ (an odd cycle cannot be two-coloured).

Q21. Graph 1 has vertices $1,2,3,4,5$ and edges $12,13,14,15,23$, so $v=5$ and $e=5$; Graph 2 has vertices V,W,X,Y,Z and edges XV,XW,XY,XZ,VW, so $v=5$ and $e=5$ as well. The degrees in Graph 1 are $deg(1)=4$, $deg(2)=2$, $deg(3)=2$, $deg(4)=1$, $deg(5)=1$, giving the sequence $4,2,2,1,1$; in Graph 2 they are $deg(X)=4$, $deg(V)=2$, $deg(W)=2$, $deg(Y)=1$, $deg(Z)=1$, the same sequence. The single degree-4 vertex must correspond, so $1 \leftrightarrow X$. The two degree-2 vertices are joined to each other in each graph (23 and VW), so $2 \leftrightarrow V$ and $3 \leftrightarrow W$; the two degree-1 vertices give $4 \leftrightarrow Y$ and $5 \leftrightarrow Z$. Checking the edges, $12 \rightarrow XV$, $13 \rightarrow XW$, $14 \rightarrow XY$, $15 \rightarrow XZ$ and $23 \rightarrow VW$ are all edges of Graph 2, so the correspondence preserves every edge and the graphs are isomorphic.

Q22. Count the degrees. Graph 1 is a five-sided cycle, so every vertex has degree 2 and the degree sequence is $2,2,2,2,2$. Graph 2 is a triangle ABC with a tail $C-D-E$ attached, so $deg(C)=3$, $deg(A)=deg(B)=deg(D)=2$ and $deg(E)=1$, giving the sequence $3,2,2,2,1$. Although both graphs have 5 vertices and 5 edges, their degree sequences are different — Graph 2 has a vertex of degree 3 and a vertex of degree 1, which Graph 1 does not. Equal degree sequences are a *necessary* condition for isomorphism, so the graphs are **not** isomorphic.

Q23. Graph 1 is a hexagon $1-2-3-4-5-6$ with the long diagonal 14 added, so $deg(1)=deg(4)=3$ and the other four vertices have degree 2. Graph 2 is drawn as two squares sharing the edge AD; its edges are AB,BC,CD,DE,EF,FA and AD, so $deg(A)=deg(D)=3$ and the others have degree 2. Both have 6 vertices, 7 edges and degree sequence $3,3,2,2,2,2$. Reading the outer six-cycle of Graph 1 around from vertex 1 and matching it to the six-cycle $A-B-C-D-E-F$ of Graph 2 gives $1 \leftrightarrow A$, $2 \leftrightarrow B$, $3 \leftrightarrow C$, $4 \leftrightarrow D$, $5 \leftrightarrow E$, $6 \leftrightarrow F$. Under this pairing the cycle edges $12,23,34,45,56,61$ become AB,BC,CD,DE,EF,FA, and the diagonal 14 becomes AD — all seven are edges of Graph 2. Since every edge is preserved, the correspondence is an isomorphism.

Theory

A graph is a picture, but it can also be stored exactly as a table of numbers. That table is the graph's **adjacency matrix**. It is how a graph is entered into a CAS or a computer, and — as we shall see — its **powers** count the number of ways of travelling through the network.

Definition. Label the vertices of a graph $A, B, C, \ldots$ and use the *same* order for the rows and the columns. The **adjacency matrix** A is the square matrix whose entry in row i, column j is the **number of edges joining vertex i to vertex j**. A graph with n vertices gives an $n \times n$ matrix.

Symmetry. In an **undirected** graph an edge between two vertices may be travelled either way, so the entry in row i, column j equals the entry in row j, column i. The adjacency matrix of an undirected graph is therefore **symmetric** about its leading (top-left to bottom-right) diagonal.

Multiple edges. If two vertices are joined by more than one edge, the entry records how many. A double edge between two vertices gives an entry of 2.

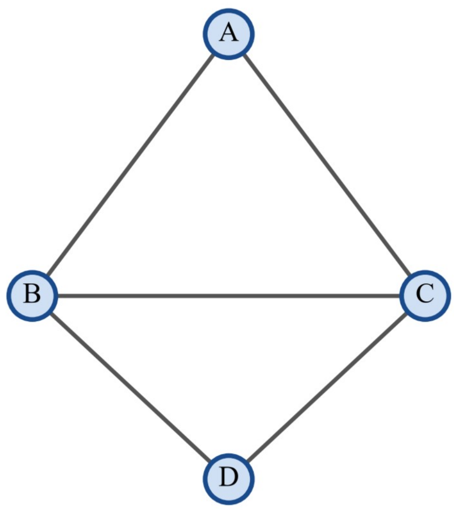
Loops. A **loop** is an edge that joins a vertex to itself, so it sits on the **leading diagonal**. The convention used here is to record a loop as **1** in that diagonal entry.

Reading degrees. The **degree** of a vertex is the number of edge-ends meeting it. For a graph with **no loops** this is exactly the **sum of the entries in that vertex's row** (equally, its column). A loop is different: both of its ends are at the same vertex, so a loop adds **2** to the degree while it is written as only 1 in the matrix. Hence, for a vertex carrying a loop,

$$\text{degree} = (\text{row sum}) + (\text{number of loops at that vertex}).$$

A useful check. Every ordinary edge adds 1 to two different rows, and every loop adds 2 to one degree, so the **sum of all the degrees is twice the number of edges** (the handshake result). For a loop-free graph this says: the sum of *all* the entries of the adjacency matrix equals twice the number of edges.

The graph below has its adjacency matrix written beside it.

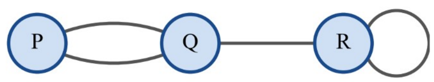


Taking the vertices in the order A, B, C, D,

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

The row sums $2, 3, 3, 2$ are the degrees of A, B, C, D; they add to 10, which is twice the number of edges (5). Reading the matrix the other way lets us **rebuild the graph**: a 1 in row A, column B tells us there is an edge AB, and so on.

Multiple edges and loops together. The small graph below has a double edge between P and Q, a single edge QR, and a loop at R.



With the vertices in the order P, Q, R,

$$\begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

The entry 2 records the double edge; the diagonal 1 records the loop. The row sums are $2, 3, 2$, but vertex R carries a loop, so its degree is $2+1=3$: the degrees are $2, 3, 3$, adding to $8 = 2 \times 4$ edges.

Directed graphs. For a **directed** graph (a digraph) the entry in row i, column j counts the arcs pointing *from* i *to* j. This matrix need **not** be symmetric. Each **row** sum is then an **out-degree** and each **column** sum an **in-degree**.

Powers count walks. A **walk** of length k is a sequence of k edges joined end to end; vertices and edges may be repeated. The adjacency matrix counts walks through its powers:

> The entry in row i, column j of A^k is the **number of walks of length** k from vertex i to vertex j.

To see why for $k=2$, use the rule for matrix multiplication:

$$\left(A^2\right)_{ij} = \sum_m A_{im} A_{mj}.$$

The product $A_{im} A_{mj}$ counts the walks $i \to m \to j$ through the middle vertex m (the number of first-step edges times the number of second-step edges). Adding over every possible middle vertex m counts **all** walks of length 2 from i to j. Multiplying by A once more extends each walk by one edge, so A^3 counts walks of length 3, and so on. In practice you raise the matrix to a power on your CAS and read off the entry you need.

For the graph above,

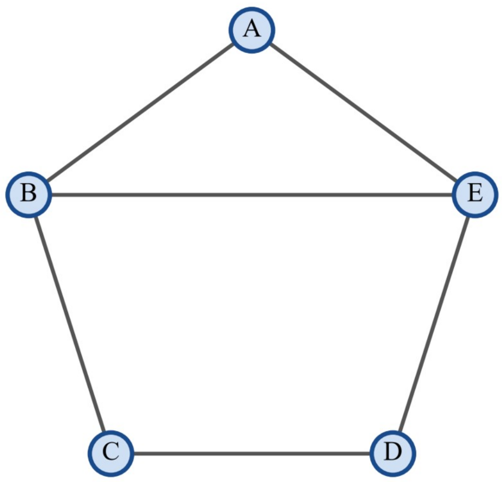
$$A^2 = \begin{bmatrix} 2 & 1 & 1 & 2 \\ 1 & 3 & 2 & 1 \\ 1 & 2 & 3 & 1 \\ 2 & 1 & 1 & 2 \end{bmatrix}.$$

The entry in row A, column D is 2, so there are two walks of length 2 from A to D (namely $A-B-D$ and $A-C-D$). Notice, too, that the leading-diagonal entries of A^2 are $2, 3, 3, 2$ — the **degrees**: a length-2 walk from a vertex back to itself simply goes out along an edge and straight back.

Worked examples

Example 1 — scaffolded

Write down the adjacency matrix of the graph below, using the vertex order A, B, C, D, E, and state the degree of each vertex.



Working	Comment
Edges: AB, AE, BC, BE, CD, DE.	List every edge once by reading the diagram.
Row A: joined to B and E $\Rightarrow 01001$.	Put a 1 in each column for a vertex A is joined to.
Row B: joined to A, C, E $\Rightarrow 10101$.	The matrix is symmetric, so row B mirrors column B.
$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$	Fill the remaining rows the same way; check it is symmetric.
Degrees = row sums $= 2, 3, 2, 2, 3$.	No loops, so each degree is simply the sum of its row.
Sum of degrees $= 12 = 2 \times 6$.	A check: there are 6 edges.

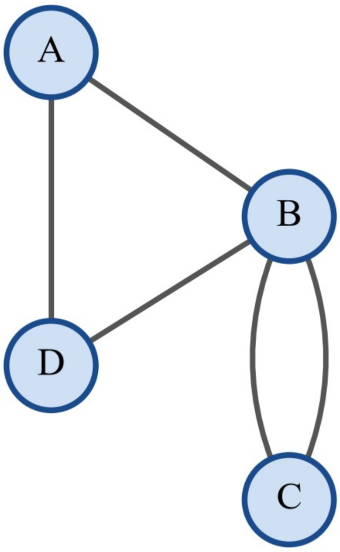
Example 2 — scaffolded

A graph has the adjacency matrix below, with vertices in the order A, B, C, D. Describe and draw the graph, and state the degree of each vertex.

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 2 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

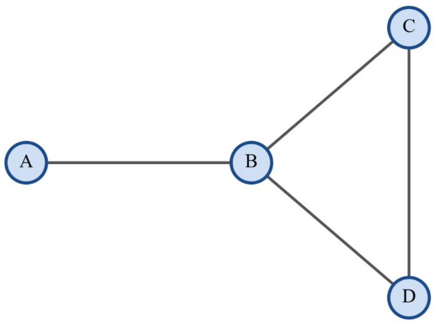
Working	Comment
The matrix is 4×4, so there are 4 vertices.	The order of the matrix gives the number of vertices.
Row A: $0101 \Rightarrow$ edges AB and AD.	Read the edges from the entries above (or on) the diagonal.
Row B: entry 2 in column C $\Rightarrow$ a **double edge** BC; also BD.	An entry of 2 means two edges join B and C.
Diagonal all 0, so there are **no loops**.	Loops would show as non-zero diagonal entries.
Degrees = row sums $= 2, 4, 2, 2$.	Vertex B has degree 4 (edges BA, BC, BC, BD).

The graph is drawn below; the double edge between B and C is shown as two separate arcs.



Example 3 — unscaffolded

The graph below has vertices A, B, C, D. Write its adjacency matrix A, then use A^2 and A^3 to find the number of walks of length 2 from A to C and the number of walks of length 3 from A to B.



With the vertices in the order A, B, C, D, the edges AB, BC, BD, CD give

$$A=\begin{bmatrix}0&1&0&0\\1&0&1&1\\0&1&0&1\\0&1&1&0\end{bmatrix}, A^2=\begin{bmatrix}1&0&1&1\\0&3&1&1\\1&1&2&1\\1&1&1&2\end{bmatrix}.$$

The entry of A^2 in row A, column C is 1, so there is exactly **one** walk of length 2 from A to C — the walk $A-B-C$.

Cubing the matrix,

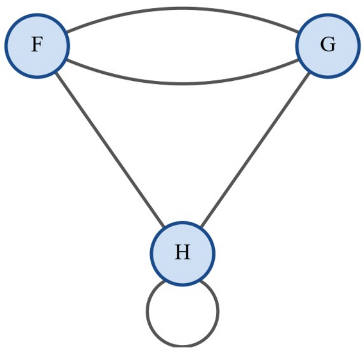
$$A^3 = \begin{bmatrix} 0 & 3 & 1 & 1 \\ 3 & 2 & 4 & 4 \\ 1 & 4 & 2 & 3 \\ 1 & 4 & 3 & 2 \end{bmatrix}.$$

The entry in row A, column B is 3, so there are **three** walks of length 3 from A to B: $A-B-A-B$, $A-B-C-B$ and $A-B-D-B$. (Walks may reuse vertices and edges, which is why the first of these is allowed.)

$\therefore$ there is 1 walk of length 2 from A to C and 3 walks of length 3 from A to B.

Example 4 — unscaffolded

A network has vertices F, G, H. There is a double edge between F and G, single edges FH and GH, and a loop at H. Write its adjacency matrix, find the degree of each vertex, and check the handshake result.



With the vertices in the order F, G, H, the double edge FG gives an entry 2, the single edges give entries 1, and the loop gives 1 on the diagonal at H:

$$A = \begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

The row sums are $3, 3, 3$. Vertices F and G have no loop, so their degrees are 3 and 3. Vertex H carries a loop, which counts twice, so its degree is (row sum) $+1 = 3+1 = 4$.

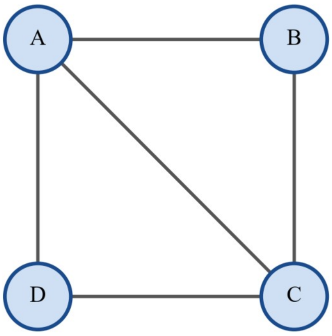
There are 5 edges (two FG, one FH, one GH, one loop), and the degrees $3, 3, 4$ add to $10 = 2 \times 5$.

$\therefore$ the degrees are $F:3$, $G:3$, $H:4$, and the sum of the degrees is twice the number of edges, as expected.

Practice questions

Scaffolded

Q1. For the graph below (vertices A, B, C, D):



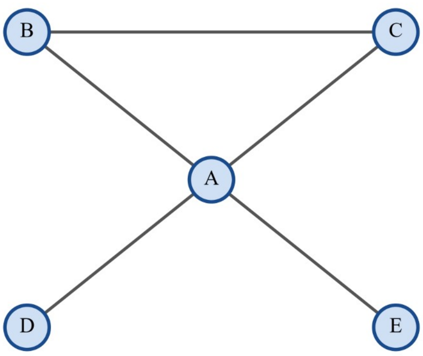
(a) Write down its adjacency matrix.
(b) Use the matrix to find the degree of each vertex.
(c) Check that the sum of the degrees is twice the number of edges.

Q2. A graph has the adjacency matrix below, with vertices in the order A, B, C, D.

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

(a) How many vertices does the graph have? (b) How many edges does it have? (c) Find the degree of each vertex. (d) Does the graph have any loops or multiple edges? Explain how you can tell.

Q3. For the graph below (vertices A, B, C, D, E):



(a) Complete its adjacency matrix.
(b) State the degree of each vertex.
(c) State the vertex of highest degree.

Q4. A graph has the adjacency matrix below, with vertices in the order A, B, C, D.

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Draw a graph that has this adjacency matrix.

Q5. A simple graph has the adjacency matrix A below, with vertices in the order A, B, C, D.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(a) Find A^2. (b) Use A^2 to write down the number of walks of length 2 from A to C. (c) State what the leading-diagonal entries of A^2 represent.

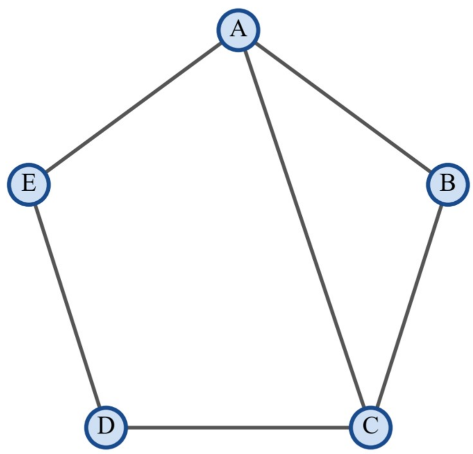
Q6. A graph has the adjacency matrix below, with vertices in the order A, B, C, D.

$$\begin{bmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

(a) Find the degree of each vertex. (b) State the number of edges. (c) Verify that the sum of the degrees is twice the number of edges.

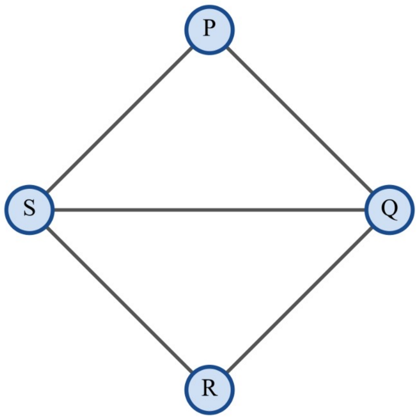
Unscaffolded

Q7. For the graph below (vertices A, B, C, D, E), write down its adjacency matrix and list its degrees in order from smallest to largest (its degree sequence).



Q8. A directed graph has the arcs $A \to B$, $A \to C$, $B \to C$ and $C \to A$. (a) Write its adjacency matrix, taking the rows as “from” and the vertices in the order A, B, C. (b) Is the matrix symmetric? What does your answer tell you about the graph? (c) State the out-degree and the in-degree of vertex C.

Q9. The network below shows the direct flights between four towns P, Q, R, S.



(a) Write down the adjacency matrix (order P, Q, R, S).

(b) Find A^2.

(c) How many routes with exactly one stopover connect P and R? Name them.

Q10. Using the flight network of the previous question, find A^3 and hence state the number of walks of length 3 from Q to S.

Q11. A graph on vertices A, B, C, D has a double edge between A and B, and single edges BC, CD and BD. (a) Write down its adjacency matrix (order A, B, C, D). (b) Find the degree of each vertex.

Q12. A graph has the adjacency matrix below, with vertices in the order A, B, C.

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

(a) Draw the graph. (b) State the degree of vertex A, explaining how the loop affects it.

Q13. An undirected graph has the partly-completed adjacency matrix below (vertices A, B, C, D); the entries marked * have been left out.

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ * & 0 & 0 & 1 \\ * & * & 0 & 1 \\ * & * & * & 0 \end{bmatrix}$$

(a) Use the symmetry of the matrix to complete it. (b) State the degree of each vertex.

Q14. (a) Explain why, for an undirected graph with no loops, the sum of all the entries of the adjacency matrix is always equal to twice the number of edges. (b) The entries of such a matrix add to 18. How many edges does the graph have?

Q15. A simple graph has the adjacency matrix A below, with vertices in the order A, B, C, D.

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(a) Find A^2. (b) Show that each leading-diagonal entry of A^2 equals the degree of the corresponding vertex, and explain why this must happen for any simple graph. (c) The entry in row C, column D of A^2 is 0, even though C and D are joined by an edge. Explain why.

Q16. A graph has adjacency matrix A and square A^2 (vertices A, B, C, D):

$$A=\begin{bmatrix}0&1&1&1\\1&0&1&1\\1&1&0&1\\1&1&1&0\end{bmatrix}, A^2=\begin{bmatrix}3&2&2&2\\2&3&2&2\\2&2&3&2\\2&2&2&3\end{bmatrix}.$$

(a) How many walks of length 2 start and finish at B? (b) How many walks of length 2 start at B in total? (c) List the walks of length 2 from B back to B.

Q17. A graph has the adjacency matrix below, with vertices in the order A, B, C, D.

$$\begin{bmatrix}0&1&2&0\\1&0&1&1\\2&1&0&0\\0&1&0&0\end{bmatrix}$$

(a) Write down its degree sequence. (b) How many edges does it have? (c) State whether it contains a loop or a multiple edge, quoting the entry that tells you.

Q18. A one-way road network has the arcs $V \to W$, $V \to X$, $W \to X$, $X \to Y$ and $Y \to V$. (a) Write its adjacency matrix, taking the rows as "from" and the vertices in the order V, W, X, Y. (b) State the out-degree of each vertex. (c) Which vertex can be reached directly from two others? Explain how the matrix shows this.

Q19. A graph has adjacency matrix A (vertices A, B, C, D, E), with

$$A^2=\begin{bmatrix}3&1&1&2&0\\1&2&1&1&1\\1&1&3&0&2\\2&1&0&2&0\\0&1&2&0&2\end{bmatrix}, A^3=\begin{bmatrix}2&4&6&1&5\\4&2&4&2&2\\6&4&2&5&1\\1&2&5&0&4\\5&2&1&4&0\end{bmatrix}.$$

Use these to find: (a) the number of walks of length 2 from A to D; (b) the number of walks of length 3 from A to C; (c) the degree of A, read from the diagonal of A^2.

Q20. Five students A, B, C, D, E have the friendships A–B, A–C, B–C, B–D, C–D and D–E. (a) Write down the adjacency matrix (order A, B, C, D, E). (b) Find A^2. (c) How many friends do A and D have in common, and who are they? (Use the entry of A^2 in row A, column D.)

Answers

Q1. (a) $\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$; (b) degrees $3, 2, 3, 2$; (c) sum $= 10 = 2 \times 5$ edges. **Q2.** (a) 4; (b) 5; (c) degrees $2, 3, 2, 3$; (d) a loop at D (diagonal entry 1); no multiple edges (no entry exceeds 1). **Q3.** (a) $\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$; (b) degrees $4, 2, 2, 1, 1$; (c) vertex A. **Q4.** A 4-cycle $A–B–C–D–A$ (a square); each vertex has degree 2. **Q5.** (a) $A^2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$; (b) 1; (c) the degree of each vertex. **Q6.** (a) degrees $3, 3, 2, 2$; (b) 5 edges; (c) $3 + 3 + 2 + 2 = 10 = 2 \times 5$. **Q7.** $\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$; degree sequence $2, 2, 2, 3, 3$. **Q8.** (a) $\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$; (b) not symmetric, so the graph is directed; (c) out-degree 1, in-degree 2. **Q9.** (a) $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$; (b) $A^2 = \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 3 & 1 & 2 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 1 & 3 \end{bmatrix}$; (c) 2 — $P–Q–R$ and $P–S–R$. **Q10.** $A^3 = \begin{bmatrix} 2 & 5 & 2 & 5 \\ 5 & 4 & 5 & 5 \\ 2 & 5 & 2 & 5 \\ 5 & 5 & 5 & 4 \end{bmatrix}$; 5 walks of length 3 from Q to S. **Q11.** (a) $\begin{bmatrix} 0 & 2 & 0 & 0 \\ 2 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$; (b) degrees $2, 4, 2, 2$. **Q12.** (a) a loop at A with edges AB and BC; (b) degree of A is 3 (the loop counts twice: 2 for the loop plus 1 for edge AB). **Q13.** (a) $\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$; (b) degrees $3, 2, 2, 3$. **Q14.** (a) each edge joins two vertices and so contributes 1 to two entries (by symmetry), i.e. 2 to the total; (b) $18 \div 2 = 9$ edges. **Q15.** (a) $A^2 = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 3 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$; (b) diagonal $2, 2, 3, 1 =$ degrees, because $\left(A^2\right)_{ii}$ sums $A_{im}^2 = A_{im}$, i.e. counts the vertices joined to i; (c) a walk of length 2 from C to D needs a middle vertex joined to both, but the only vertex joined to D is C, which has no loop. **Q16.** (a) 3; (b) 9; (c) $B–A–B$, $B–C–B$, $B–D–B$. **Q17.** (a) $1, 3, 3, 3$; (b) 5 edges; (c) a multiple (double) edge between A and C — the entry 2; no loop (diagonal all 0). **Q18.** (a) $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$; (b) out-degrees

$V:2$, $W:1$, $X:1$, $Y:1$; (c) X — its column has two 1s (in-degree 2), reached from V and W. **Q19.** (a) 2; (b) 6; (c) 3.

Q20. (a) $\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$; (b) $A^2 = \begin{bmatrix} 2 & 1 & 1 & 2 & 0 \\ 1 & 3 & 2 & 1 & 1 \\ 1 & 2 & 3 & 1 & 1 \\ 2 & 1 & 1 & 3 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$; (c) 2 common friends, B and C.

Fully-worked solutions

Q1. Reading the diagram, the edges are AB, AD, AC, BC and CD. Placing a 1 wherever two vertices are joined gives

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

(b) The row sums are $3, 2, 3, 2$, and there are no loops, so these are the degrees of A, B, C, D. (c) The degrees add to $3+2+3+2=10$, and the graph has 5 edges, so the sum of the degrees is $2 \times 5 = 10$, as required.

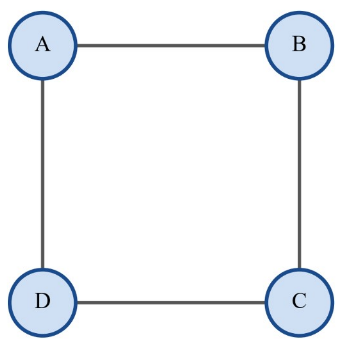
Q2. (a) The matrix is 4×4, so there are 4 vertices. (c) The row sums are $2, 3, 2, 2$, but the diagonal entry in row D is 1, showing a loop at D; a loop counts twice towards the degree, so the degree of D is $2+1=3$ and the degrees are $2, 3, 2, 3$. (b) The degrees add to 10, so there are $10 \div 2 = 5$ edges (the edges AB, AC, BC, BD and the loop at D). (d) There is a loop at D (the non-zero diagonal entry) and no multiple edges, because no entry is greater than 1.

Q3. (a) The spokes AB, AC, AD, AE and the extra edge BC give

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

(b) The row sums are $4, 2, 2, 1, 1$, which are the degrees (there are no loops). (c) Vertex A has the highest degree, 4.

Q4. The 1s off the diagonal give the edges AB, AD, BC and CD, so the graph is the four-cycle A–B–C–D–A (a square). Every row sum is 2, so each vertex has degree 2.



Q5. (a) Multiplying A by itself,

$$A^2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

(b) The entry in row A, column C is 1, so there is 1 walk of length 2 from A to C (the walk A–B–C). (c) The leading-diagonal entries $1, 2, 2, 1$ are the degrees of A, B, C, D: a length-2 walk from a vertex to itself goes out along an edge and back.

Q6. (a) The row sums are $3, 3, 2, 2$; there are no loops, so these are the degrees. (b) The degrees add to 10, so there are $10 \div 2 = 5$ edges (a double edge AB, and single edges AC, BD, CD). (c) $3+3+2+2=10=2\times5$, confirming the handshake result.

Q7. The edges AB, AC, AE, BC, CD and DE give

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The row sums are $3, 2, 3, 2, 2$, so the degree sequence, from smallest to largest, is $2, 2, 2, 3, 3$.

Q8. (a) Each arc puts a 1 in the "from" row and "to" column, giving

$$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

(b) The matrix is not symmetric — for example the entry in row A, column C is 1 but the entry in row C, column A is also 1, yet row A, column B is 1 while row B, column A is 0. A non-symmetric adjacency matrix means the graph is **directed**. (c) Row C sums to 1 (the single arc $C \to A$), so the out-degree of C is 1; column C sums to 2 (the arcs $A \to C$ and $B \to C$), so the in-degree of C is 2.

Q9. (a) The direct flights PQ, QR, RS, PS and QS give

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

(b) Squaring,

$$A^2 = \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 3 & 1 & 2 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 1 & 3 \end{bmatrix}.$$

(c) A route with exactly one stopover between P and R is a walk of length 2; the entry in row P, column R of A^2 is 2, so there are 2 such routes, namely P–Q–R and P–S–R.

Q10. Multiplying once more,

$$A^3 = \begin{bmatrix} 2 & 5 & 2 & 5 \\ 5 & 4 & 5 & 5 \\ 2 & 5 & 2 & 5 \\ 5 & 5 & 5 & 4 \end{bmatrix}.$$

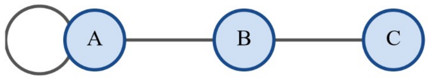
The entry in row Q, column S is 5, so there are 5 walks of length 3 from Q to S.

Q11. (a) The double edge AB gives an entry 2; the single edges BC, CD, BD give entries 1:

$$\begin{bmatrix} 0 & 2 & 0 & 0 \\ 2 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

(b) The row sums are $2, 4, 2, 2$, and there are no loops, so these are the degrees. (Vertex B has degree 4: the two edges to A plus the edges to C and D.)

Q12. (a) The diagonal entry 1 in row A is a loop at A; the off-diagonal 1s give edges AB and BC.



(b) The row sum for A is 2, but A carries a loop, which contributes 2 (not 1) to the degree. Its degree is therefore (row sum) $+1 = 2+1 = 3$: the loop counts twice and the edge AB once.

Q13. (a) The adjacency matrix of an undirected graph is symmetric, so each missing lower entry equals the entry mirrored across the diagonal:

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

(b) The row sums are $3, 2, 2, 3$, which are the degrees of A, B, C, D.

Q14. (a) In an undirected graph with no loops, each edge joins two different vertices i and j and so contributes a 1 to the entry in row i, column j **and** a 1 to the entry in row j, column i (the matrix is symmetric). Every edge therefore adds 2 to the total of all the entries, so that total is $2 \times (\text{number of edges})$. (b) If the entries add to 18 then the number of edges is $18 \div 2 = 9$.

Q15. (a) Squaring,

$$A^2 = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 3 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

(b) The leading-diagonal entries are $2, 2, 3, 1$. The row sums of A are also $2, 2, 3, 1$, so the diagonal entries of A^2 equal the degrees of A, B, C, D. This must happen for any simple graph: every entry of A is then 0 or 1, so

$$(A^2)_{ii} = \sum_m A_{im} A_{mi} = \sum_m A_{im}^2 = \sum_m A_{im},$$

which is just the number of vertices joined to i — that is, the degree of i. (c) A walk of length 2 from C to D has the form C–m–D, where the middle vertex m must be joined to both C and D. Row D of A shows that the only vertex joined to D is C itself, and C carries no loop, so no such m exists and the entry is 0. The edge CD gives a walk of length 1, not one of length 2.

Q16. (a) The entry in row B, column B of A^2 is 3, so 3 walks of length 2 start and finish at B. (b) The total number of walks of length 2 starting at B is the sum of row B of A^2, namely $2+3+2+2=9$. (c) The three closed walks of length 2 go out to a neighbour of B and back; row B of A shows that B is joined to A, C and D, so they are B–A–B, B–C–B and B–D–B.

Q17. (a) The row sums are $3, 3, 3, 1$, so the degree sequence is $1, 3, 3, 3$. (b) The degrees add to 10, so there are $10 \div 2 = 5$ edges. (c) The entry 2 in row A, column C (and row C, column A) shows a double edge between A and C; every diagonal entry is 0, so there is no loop.

Q18. (a) Each arc puts a 1 in its "from" row and "to" column:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

(b) The row sums give the out-degrees: $V:2$, $W:1$, $X:1$, $Y:1$. (c) The column for X contains two 1s (from rows V and W), so X has in-degree 2 and is reached directly from both V and W; no other column has more than one 1.

Q19. (a) The entry in row A, column D of A^2 is 2, so there are 2 walks of length 2 from A to D. (b) The entry in row A, column C of A^3 is 6, so there are 6 walks of length 3 from A to C. (c) The diagonal entry in row A of A^2 is 3, which is the degree of A.

Q20. (a) The friendships give

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

(b) Squaring,

$$A^2 = \begin{bmatrix} 2 & 1 & 1 & 2 & 0 \\ 1 & 3 & 2 & 1 & 1 \\ 1 & 2 & 3 & 1 & 1 \\ 2 & 1 & 1 & 3 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

(c) The entry in row A, column D of A^2 is 2: a walk of length 2 from A to D goes A–(mutual friend)–D, so A and D have 2 friends in common. Their common neighbours are B and C.

Theory

Many practical questions about a network fall into one of two families. In an **exploring** problem we must travel along *every connection* — every street, pipe or cable — as when a postie delivers to every street or an inspector checks every power line. In a **travelling** problem we must visit *every location* — every town, depot or customer — as when a salesperson calls on every client, or a tour visits every attraction. In graph language, an exploring problem is about using every **edge**, and a travelling problem is about visiting every **vertex**. This section develops the language of routes through a graph and the two problems built on it.

Walks, trails, paths, circuits and cycles.

A **walk** is any route that moves from vertex to vertex along edges of the graph. In a walk, edges *and* vertices may be repeated. Three special kinds of walk are named according to what may *not* be repeated:

- a **trail** is a walk with **no repeated edge** (a vertex may still be repeated);
- a **path** is a walk with **no repeated vertex** (so no edge can be repeated either);
- a route is **closed** if it starts and finishes at the *same* vertex.

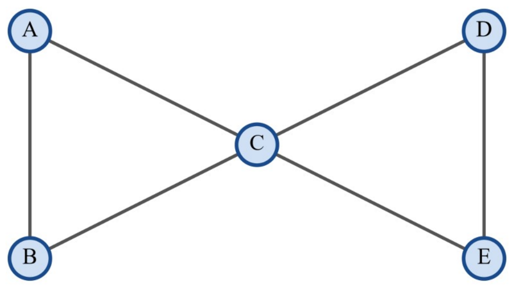
The two closed routes are:

- a **circuit** is a **closed trail** — it uses no edge more than once and returns to its starting vertex;
- a **cycle** is a **closed path** — it uses no vertex more than once (apart from the start, which is also the finish) and returns to its starting vertex.

Every path is a trail, and every cycle is a circuit, so when describing a route we give the **most specific** name that fits.

Route	Repeated edges?	Repeated vertices?	Closed (start = finish)?
walk	allowed	allowed	not necessarily
trail	no	allowed	not necessarily
path	no	no	no
circuit	no	allowed	yes
cycle	no	no (except start = finish)	yes

The graph below has two triangles joined at C. Reading routes from it:



Route	Most specific name	Reason
$A-B-C-D-C-E$	walk	the edge $C-D$ is used twice
$A-B-C-D-E-C$	trail	no edge repeats, but vertex C repeats
$A-B-C-D-E$	path	no vertex (and so no edge) repeats
$A-B-C-D-E-C-A$	circuit	closed, no edge repeats, but vertex C repeats

Route	Most specific name	Reason
$A-B-C-A$	cycle	closed, and no vertex repeats apart from A

Eulerian trails and circuits — the exploring problem.

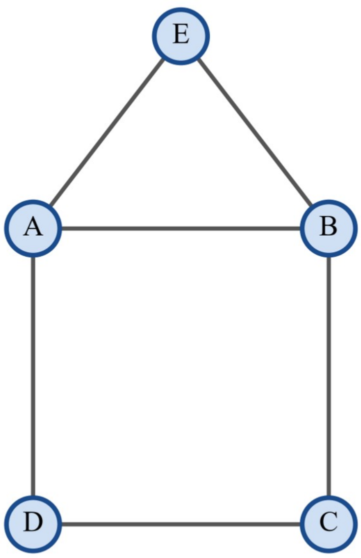
An **Eulerian trail** is a trail that uses **every edge of the graph exactly once**. An **Eulerian circuit** is a *closed* Eulerian trail — it uses every edge exactly once and returns to its starting vertex. These are exactly the routes needed to solve an exploring problem.

Whether they exist is decided by a single, easily checked feature: the number of vertices of **odd degree** (recall the degree of a vertex is the number of edges meeting it). For a **connected** graph:

- an **Eulerian circuit** exists **if and only if every vertex has even degree** (zero vertices of odd degree);
- an **Eulerian trail** exists **if and only if exactly** 0 **or** 2 **vertices have odd degree**. If there are two odd vertices, every Eulerian trail must **start at one of them and finish at the other**;
- if **more than two** vertices have odd degree, **neither** exists.

The reason is short. Each time a trail passes *through* a vertex it arrives on one edge and leaves on another, using edges two at a time; so every vertex the trail only passes through must have an even number of edges. Only the start and finish of an *open* trail can be left with an odd count — which is why at most two odd vertices are allowed, and none at all if the trail is to close up into a circuit.

The graph below (a square with an apex) has degrees $A\,3, B\,3, C\,2, D\,2, E\,2$. Exactly two vertices, A and B, are odd, so it has an Eulerian trail but no Eulerian circuit. One such trail, starting at A and finishing at B, is $A-D-C-B-E-A-B$.

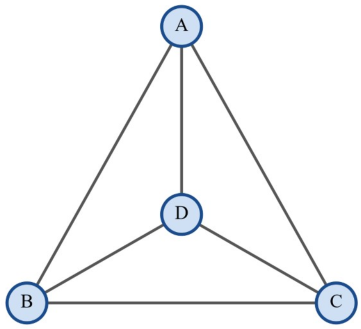


By contrast, in the two-triangle graph above every vertex is even ($A\,2, B\,2, C\,4, D\,2, E\,2$), so it has an Eulerian circuit, for example $A-B-C-D-E-C-A$. Typical exploring applications are mail delivery along every street, inspecting every pipe or power line, and street-sweeping, gritting or garbage routes that must cover every road.

Hamiltonian paths and cycles — the travelling problem.

A **Hamiltonian path** visits **every vertex of the graph exactly once**. A **Hamiltonian cycle** visits every vertex exactly once and **returns to the start**. These solve a travelling problem, where each location must be visited.

Unlike the Eulerian case, there is **no simple degree test** for Hamiltonian routes: to decide whether one exists you generally have to search for it. The two problems are also independent of each other. The graph below is the complete graph on four vertices; every vertex has degree 3, so all four are odd and it has *no* Eulerian trail — yet it has the Hamiltonian cycle $A-B-C-D-A$.



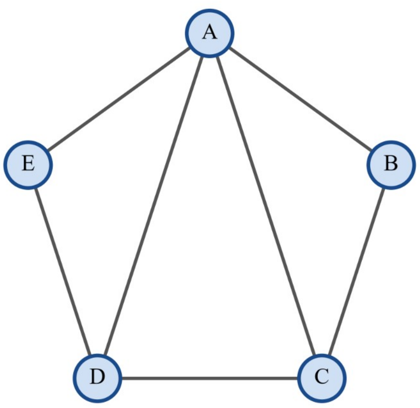
The reverse can also happen. The two-triangle graph has an Eulerian circuit but **no** Hamiltonian cycle: every route reaching both triangles must pass through the shared vertex C twice. (It does have a Hamiltonian *path*, $A-B-C-D-E$.) The square-with-apex graph, on the other hand, has *both* an Eulerian trail and the Hamiltonian cycle $E-A-D-C-B-E$. Typical travelling applications are a salesperson calling on every town, a delivery van visiting each address once, and a tour that stops at every attraction.

Problem	Requirement	Route needed	Test
exploring	use every **edge** once	Eulerian trail / circuit	count odd-degree vertices (0 or 2)
travelling	visit every **vertex** once	Hamiltonian path / cycle	no simple test — search

Worked examples

Example 1 — scaffolded

For the graph shown, give the most specific description (walk, trail, path, circuit or cycle) of each route.



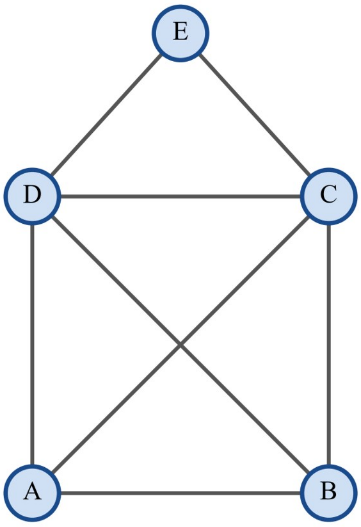
(a) $A-B-C-D-E$ (b) $A-C-D-A$
(b) $A-B-C-A-D$ (d) $A-B-C-A-C-D$
(c) $A-B-C-A-D-E-A$

Working	Comment
(a) Edges AB, BC, CD, DE used; vertices A, B, C, D, E all different.	No repeats at all and not closed, so it is a **path**.
(b) Edges AC, CD, DA; returns to A; no vertex	A closed route with no repeated vertex is a **cycle**.

Working	Comment
repeats except A.	
(c) Edges AB, BC, CA, AD all different, but vertex A repeats; ends at D.	No repeated edge (so a trail) but a vertex repeats, so it is a **trail**, not a path.
(d) Edge $A-C$ is used twice ($C-A$ then $A-C$).	A repeated edge means it is only a **walk**.
(e) Edges AB, BC, CA, AD, DE, EA all different; returns to A; vertex A repeats.	Closed with no repeated edge but a repeated vertex, so a **circuit** (not a cycle).

Example 2 — scaffolded

A security guard must walk along **every corridor** of the building shown exactly once. Each vertex is a junction and each edge a corridor. Determine whether this is possible, and if so give a suitable route.

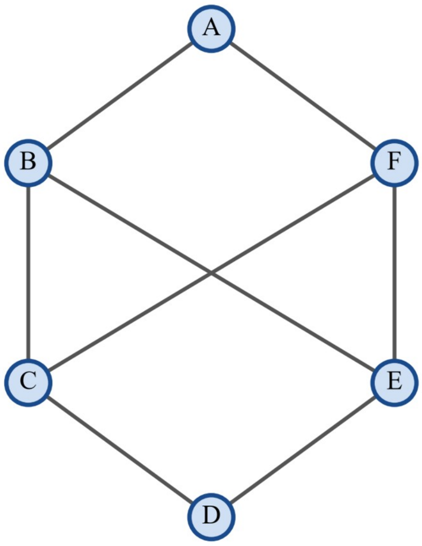


Working	Comment
Degrees: $A3, B3, C4, D4, E2$.	Count the edges meeting each vertex.
Odd-degree vertices: A and B — exactly two.	Compare with the Eulerian conditions.
Two odd vertices ⇒ an Eulerian trail exists, but **no** Eulerian circuit.	Walking every corridor once = an Eulerian trail.
The trail must start at one odd vertex and finish at the other, i.e. between A and B.	The odd vertices are forced to be the endpoints.
A suitable route: $A-D-E-C-D-B-A-C-B$.	Check: all 8 corridors used exactly once, starting at A, finishing at B.

So the guard can walk every corridor exactly once, provided they begin at junction A and end at junction B (or the reverse).

Example 3 — unscaffolded

A courier must visit each of the six depots A–F **exactly once**, starting and finishing at depot B, travelling only along the roads shown. Find a suitable route.



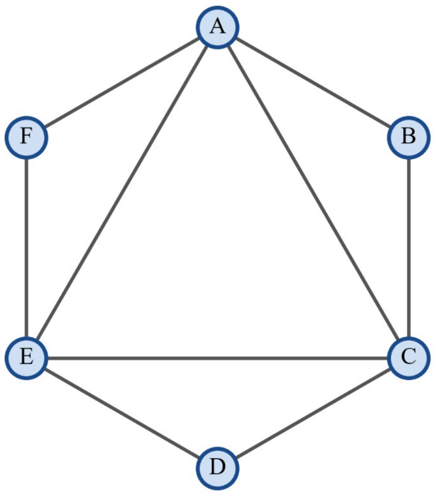
Visiting every depot once and returning to the start is a travelling problem, so we need a **Hamiltonian cycle** through B. There is no degree test, so we reason about the edges. Depot A lies on only two roads, $A-B$ and $A-F$, so any cycle must use both of them; that fixes the piece $B-A-F$ of the route. From F the only new depots reachable are C and E; choosing $F-E$ leads to $E-D-C$, and C joins back to B.

Following this through gives $B-A-F-E-D-C-B$. Checking: the six depots B, A, F, E, D, C are each visited once and the route returns to B, using the roads BA, AF, FE, ED, DC, CB, all of which exist.

$\therefore$ a suitable route is $B-A-F-E-D-C-B$.

Example 4 — unscaffolded

The graph shows the paths (edges) joining the features (vertices) of a park. (a) A gardener wants to inspect **every path** exactly once and return to the depot at A. (b) A visitor wants to walk to **every feature** exactly once and return to A. Determine whether each is possible and give a route where it is.



The degrees are $A\,4, B\,2, C\,4, D\,2, E\,4, F\,2$, so **every** vertex is even.

(a) Inspecting every path once and returning to the start is an exploring problem needing an **Eulerian circuit**. Because every vertex has even degree, an Eulerian circuit exists. Tracing every edge once from A: $A-B-C-D-E-F-A-C-E-A$, which uses all nine paths and returns to A.

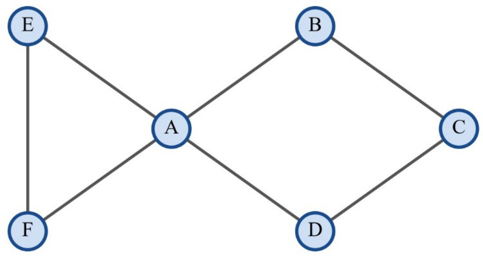
(b) Visiting every feature once and returning is a travelling problem needing a **Hamiltonian cycle**. The six features lie on the outer six-sided loop, so $A-B-C-D-E-F-A$ visits each feature exactly once and returns to A.

$\therefore$ the gardener can follow the Eulerian circuit $A-B-C-D-E-F-A-C-E-A$ (every path once), and the visitor can follow the Hamiltonian cycle $A-B-C-D-E-F-A$ (every feature once).

Practice questions

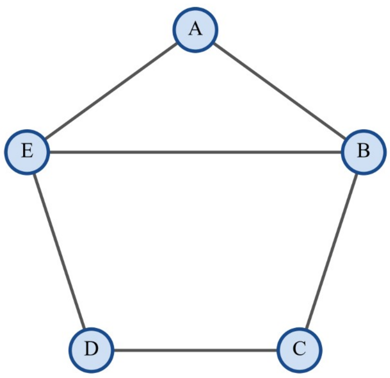
Scaffolded

Q1. For the graph shown, give the most specific description (walk, trail, path, circuit or cycle) of each route.



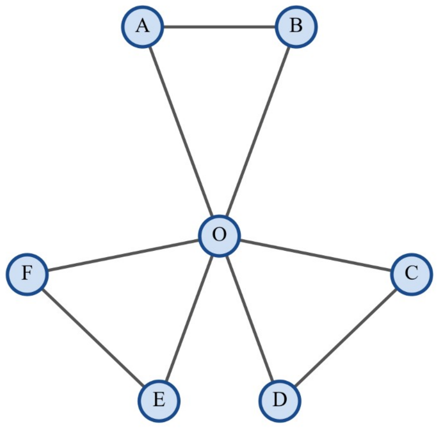
(a) $A-B-C-D-A-B$
(b) $E-F-A-B$
(c) $A-E-F-A$
(d) $A-B-C-D-A-E-F-A$
(e) $A-B-C-D-A-E$

Q2. Use the graph shown to answer the following.



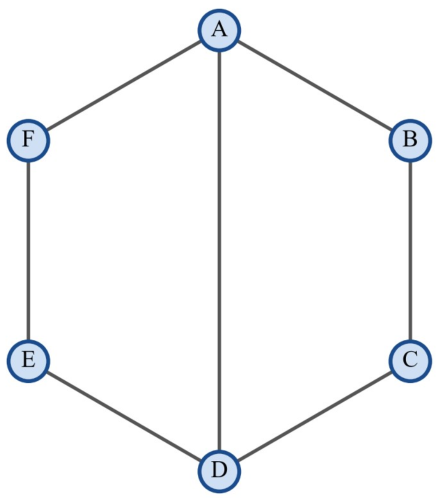
(a) Write down the degree of each vertex.
(b) How many vertices have odd degree?
(c) Does the graph have an Eulerian trail? Give a reason.
(d) Does the graph have an Eulerian circuit? Give a reason.
(e) Write down an Eulerian trail.

Q3. Use the graph shown to answer the following.



(a) Write down the degree of each vertex.
(b) Explain why the graph has an Eulerian circuit.
(c) Write down an Eulerian circuit.

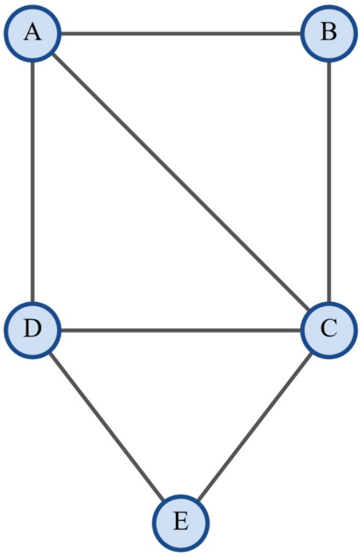
Q4. Use the graph shown to answer the following.



(a) Find a Hamiltonian path.
(b) Find a Hamiltonian cycle.
(c) Does the graph have an Eulerian circuit? Explain.

Q5. A connected graph has six vertices, with degrees $2,2,2,2,3$ and 3. (a) How many vertices have odd degree? (b) State whether the graph has an Eulerian trail, and whether it has an Eulerian circuit. (c) A route uses every edge exactly once. Between which two vertices must it start and finish?

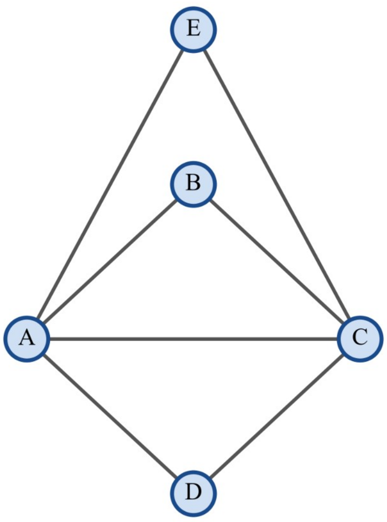
Q6. The graph shows the streets of a small estate; each edge is a street. A postie must walk along **every** street exactly once.



(a) Explain whether this is possible.
(b) If it is, between which two corners must the walk start and finish?
(c) Write down a suitable route.

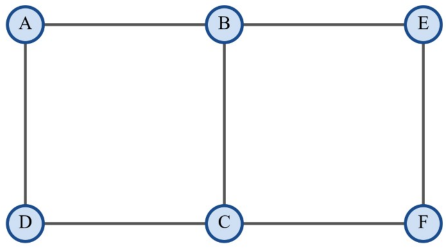
Unscaffolded

Q7. For the graph shown, give the most specific description (walk, trail, path, circuit or cycle) of each route.

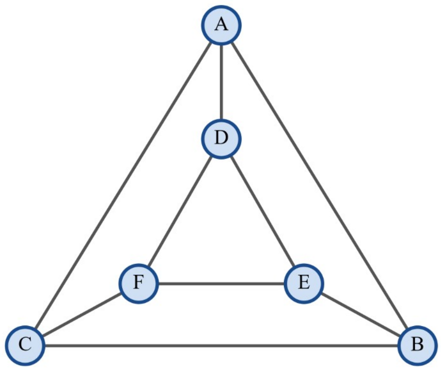


(a) $A - C - D - A$
(b) $B - A - D - C - A - E$
(c) $A - B - C - D - A - E - C - A$
(d) $E - A - B - C - D$
(e) $D - C - A - C - E$

Q8. Determine whether the graph shown has an Eulerian trail, an Eulerian circuit, or neither. Give a reason, and write down a suitable route where one exists.

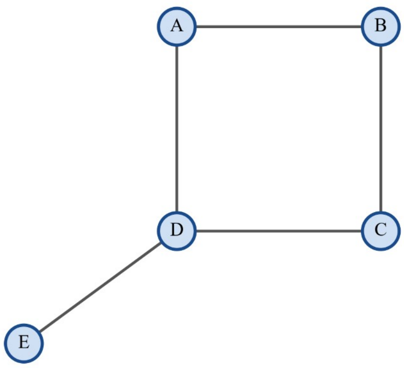


Q9. The graph shows the walking tracks joining six lookouts. A bushwalker wants to visit every lookout exactly once and return to the start.



(a) Is this an exploring or a travelling problem?
(b) Find a suitable route.
(c) Does the graph have an Eulerian circuit? Explain.

Q10. For the graph shown:

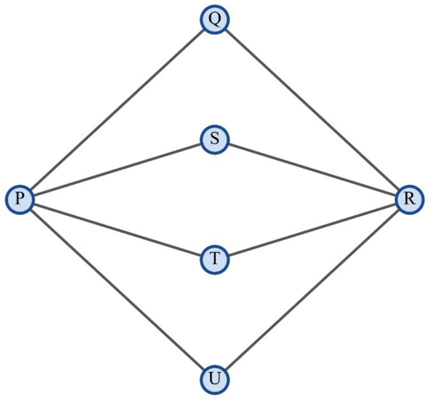


(a) Does it have an Eulerian trail? If so, give one.
(b) Does it have a Hamiltonian cycle? Explain your answer.

Q11. Each of three *connected* graphs is described by the degrees of its vertices. For each, state whether it has an Eulerian trail, an Eulerian circuit, or neither. (a) Every vertex has even degree. (b) Exactly two vertices have odd degree. (c) Four vertices have odd degree.

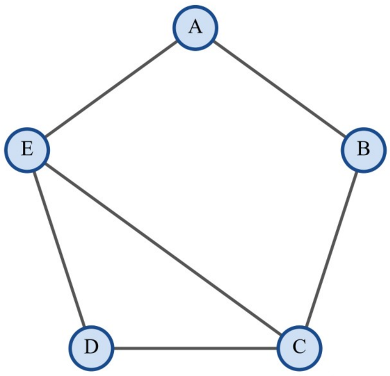
Q12. Explain the difference between: (a) a trail and a path; (b) an Eulerian trail and a Hamiltonian path; (c) the exploring problem and the travelling problem.

Q13. The graph shows the paths in a garden; P is the gate and R is the pond. A gardener starts at P, sweeps **every path** exactly once, and wants to return to P.



(a) Explain whether this is possible.
(b) Write down a suitable route.

Q14. Five attractions are joined by paths as shown. A visitor wants to see **every attraction** exactly once, starting and finishing at A.

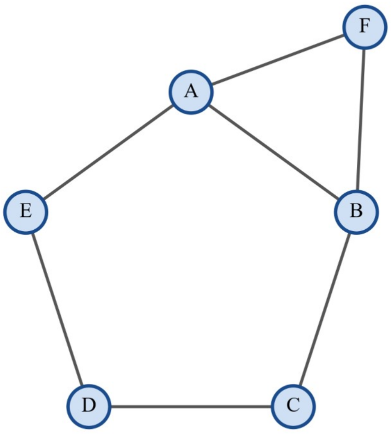


(a) What kind of route is required?
(b) Find one.
(c) Could the visitor instead walk along every path exactly once and return to A? Explain.

Q15. A connected graph has exactly four vertices of odd degree. (a) Explain why it has no Eulerian trail. (b) What is the least number of edges that must be added so that an Eulerian trail becomes possible, and where should they be placed?

Q16. The seven bridges of Königsberg join four land masses. Represented as a graph, the four vertices have degrees $3, 3, 3$ and 5. Explain, using the Eulerian condition, why no walk can cross every bridge exactly once.

Q17. For the graph shown, state whether it has each of the following, giving a route wherever one exists.

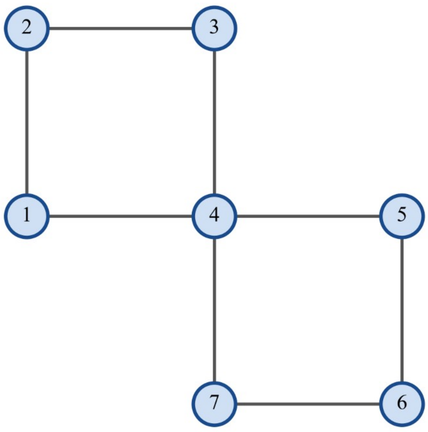


(a) an Eulerian trail (b) an Eulerian circuit (c) a Hamiltonian path (d) a Hamiltonian cycle

Q18. A graph has a vertex of degree 1 (a **pendant** vertex). (a) Explain why the graph cannot have a Hamiltonian cycle. (b) Can it still have an Eulerian trail? Explain.

Q19. In any graph, the sum of the degrees of all the vertices equals twice the number of edges. (a) Explain why this sum is always an even number. (b) Hence explain why the number of vertices of odd degree must be even. (c) Explain why a graph with exactly one vertex of odd degree cannot exist.

Q20. The graph shows the roads of a village; junction 4 is the depot.



(a) A snow-plough must clear **every road** exactly once and return to the depot. Is this possible? Give a route if it is.
(b) A school bus must stop at **every junction** exactly once and return to the depot. Is this possible? Explain.

Answers

Q1. (a) walk (b) path (c) cycle (d) circuit (e) trail. **Q2.** (a) $A2, B3, C2, D2, E3$; (b) two (B and E); (c) yes — exactly two odd vertices; (d) no — not every vertex is even; (e) e.g. $B-A-E-D-C-B-E$. **Q3.** (a) $O6$ and each of A, B, C, D, E, F is 2; (b) every vertex is even; (c) e.g. $O-A-B-O-C-D-O-E-F-O$. **Q4.** (a) e.g. $A-B-C-D-E-F$; (b) e.g. $A-B-C-D-E-F-A$; (c) no — A and D are odd, so only an Eulerian trail (from A to D) is possible. **Q5.** (a) two; (b) an Eulerian trail but no Eulerian circuit; (c) the two odd-degree (degree-3) vertices. **Q6.** (a) yes — exactly two corners (A and D) have odd degree; (b) A and D; (c) e.g. $A-B-C-A-D-E-C-D$. **Q7.** (a) cycle (b) trail (c) circuit (d) path (e) walk. **Q8.** Eulerian trail only (B and C are the two odd vertices); e.g. $B-A-D-C-F-E-B-C$. **Q9.** (a) travelling (a Hamiltonian cycle); (b) e.g. $A-B-C-F-E-D-A$; (c) no — every vertex is odd (degree 3), so it has no Eulerian trail or circuit. **Q10.** (a) yes — D and E are odd; e.g. $E-D-A-B-C-D$; (b) no — E has degree 1 and cannot lie on any cycle (a Hamiltonian *path* $E-D-A-B-C$ does exist). **Q11.** (a) an Eulerian circuit (also an Eulerian trail); (b) an Eulerian trail only; (c) neither. **Q12.** See solutions. **Q13.** (a) yes — every vertex is even (P and R have degree 4, the other three degree 2), so an Eulerian circuit exists; (b) e.g. $P-Q-R-S-P-T-R-U-P$. **Q14.** (a) a Hamiltonian cycle; (b) e.g. $A-B-C-D-E-A$; (c) no — C and E are odd, so there is no Eulerian circuit (only an Eulerian trail from C to E). **Q15.** (a) an Eulerian trail needs 0 or 2 odd vertices, and four is too many; (b) one edge, joining two of the four odd-degree vertices (leaving exactly two odd vertices). **Q16.** All four vertices are odd; an Eulerian trail needs at most two odd vertices, so no route crossing every bridge once exists. **Q17.** (a) yes, e.g. $A-F-B-C-D-E-A-B$; (b) no; (c) yes, e.g. $A-F-B-C-D-E$; (d) yes, e.g. $A-F-B-C-D-E-A$. **Q18.** (a) a Hamiltonian cycle must enter and leave every vertex, needing degree ≥ 2; a degree-1 vertex cannot be both entered and left; (b) yes — an Eulerian trail only needs 0 or 2 odd vertices, and the pendant vertex can be one endpoint. **Q19.** (a) each edge adds 2 to the total (one at each end), so the sum is $2 \times (\text{edges})$; (b) an odd number of odd degrees would make the total odd; (c) one odd vertex is an odd count, which is impossible. **Q20.** (a) yes — every junction is even, so an Eulerian circuit exists, e.g. $1-2-3-4-5-6-7-4-1$; (b) no — every route visiting all junctions must pass through junction 4 twice, so no Hamiltonian cycle exists.

Fully-worked solutions

Q1. (a) $A-B-C-D-A-B$ repeats the edge $A-B$, so it is only a **walk**. (b) $E-F-A-B$ uses edges EF, FA, AB and repeats no vertex; it is open, so it is a **path**. (c) $A-E-F-A$ returns to A with no repeated vertex apart from A, so it is a **cycle**. (d) $A-B-C-D-A-E-F-A$ is closed, repeats no edge, but repeats vertex A, so it is a **circuit** (in fact an Eulerian circuit, as it uses all seven edges once). (e) $A-B-C-D-A-E$ uses edges AB, BC, CD, DA, AE (all different) but repeats vertex A and is open, so it is a **trail**.

Q2. (a) Counting edges at each vertex: $A2, B3, C2, D2, E3$. (b) The odd-degree vertices are B and E, so there are **two**. (c) A connected graph with exactly two odd vertices has an Eulerian trail, so **yes**, and it must run between B and E. (d) An Eulerian circuit needs every vertex even; since B and E are odd, there is **no** Eulerian circuit. (e) Starting at B and finishing at E: $B-A-E-D-C-B-E$ uses each of the six edges exactly once.

Q3. (a) The centre O meets six edges, so O has degree 6; every outer vertex A, B, C, D, E, F meets two edges, so each has degree 2. (b) Every vertex has even degree and the graph is connected, so an Eulerian circuit exists. (c) Taking each triangle in turn from O: $O-A-B-O-C-D-O-E-F-O$ uses all nine edges once and returns to O.

Q4. (a) The six vertices in order round the outside give the Hamiltonian path $A-B-C-D-E-F$. (b) Closing this loop, $A-B-C-D-E-F-A$ visits every vertex once and returns to A, a Hamiltonian cycle. (c) The degrees are $A3, B2, C2, D3, E2, F2$, so A and D are odd. With two odd vertices the graph has an Eulerian trail (from A to D) but **no** Eulerian circuit.

Q5. (a) The degrees $2, 2, 2, 2, 3, 3$ include two odd values, so **two** vertices have odd degree. (b) A connected graph with exactly two odd vertices has an Eulerian trail but no Eulerian circuit. (c) An Eulerian trail (using every edge once) must begin and end at the two odd-degree vertices, i.e. the two vertices of degree 3.

Q6. (a) The degrees are $A3, B2, C4, D3, E2$, so exactly two corners, A and D, are odd. A connected graph with two odd vertices has an Eulerian trail, so walking every street once is possible. (b) The trail must start and finish at the odd corners A and D. (c) One route is $A-B-C-A-D-E-C-D$, which uses each of the seven streets exactly once, starting at A and ending at D.

Q7. (a) $A-C-D-A$ is closed with no repeated vertex apart from A — a **cycle**. (b) $B-A-D-C-A-E$ uses edges BA, AD, DC, CA, AE (all different) but repeats vertex A and is open — a **trail**. (c) $A-B-C-D-A-E-C-A$ is closed, repeats no edge, but repeats the vertices A and C — a **circuit** (an Eulerian circuit, using all seven edges). (d) $E-A-B-C-D$ has no repeats and is open — a **path**. (e) $D-C-A-C-E$ uses the edge $A-C$ twice ($C-A$ then $A-C$) — only a **walk**.

Q8. The degrees are $A2, B3, C3, D2, E2, F2$, so exactly two vertices, B and C, are odd. A connected graph with two odd vertices has an **Eulerian trail** (but no Eulerian circuit); the trail must run between B and C. One such trail is $B-A-D-C-F-E-B-C$, using each of the seven edges once.

Q9. (a) Visiting every lookout (vertex) once and returning is a **travelling** problem, requiring a Hamiltonian cycle. (b) Taking the outer triangle to the inner triangle: $A-B-C-F-E-D-A$ visits all six lookouts once and returns to A. (c) Every vertex has degree 3, so all six are odd. Because more than two vertices are odd, the graph has neither an Eulerian trail nor an Eulerian circuit.

Q10. (a) The degrees are $A2, B2, C2, D3, E1$, so D and E are the two odd vertices and an Eulerian trail exists between them; for example $E-D-A-B-C-D$ uses all five edges once. (b) Vertex E has degree 1. A cycle must enter and leave each of its vertices, which needs at least two edges there, so a degree-1 vertex can never lie on a cycle. Hence there is **no** Hamiltonian cycle. (A Hamiltonian *path* $E-D-A-B-C$, visiting every vertex once, does exist.)

Q11. (a) A connected graph in which every vertex is even has an **Eulerian circuit** (which is also an Eulerian trail). (b) Exactly two odd vertices gives an **Eulerian trail only** (it starts and finishes at those two vertices). (c) Four odd vertices is more than two, so the graph has **neither** an Eulerian trail nor an Eulerian circuit.

Q12. (a) A **trail** may not repeat an *edge* but may pass through the same vertex more than once; a **path** may not repeat a *vertex* (and so cannot repeat an edge either). Every path is a trail, but not every trail is a path. (b) An **Eulerian trail** uses every *edge* exactly once (an exploring route) and may revisit vertices; a **Hamiltonian path** visits every *vertex* exactly once (a travelling route) and need not use every edge. (c) The **exploring** problem asks for a route along every *edge*/connection (solved with Eulerian trails and circuits, tested by counting odd vertices); the **travelling** problem asks for a route visiting every *vertex*/location (solved with Hamiltonian paths and cycles, for which there is no simple test).

Q13. (a) The degrees are $P4, Q2, R4, S2, T2, U2$, so every vertex is even. A connected graph with all vertices even has an Eulerian circuit, so the gardener can sweep every path once and return to P. (b) Walk out to the pond through Q and back through S, then out again through T and back through U: $P-Q-R-S-P-T-R-U-P$ uses all eight paths once and finishes at the gate P.

Q14. (a) Seeing every attraction (vertex) once and returning requires a **Hamiltonian cycle**. (b) The outer five-sided loop gives $A-B-C-D-E-A$, visiting each attraction once and returning to A. (c) The degrees are $A2, B2, C3, D2, E3$, so C and E are odd. With two odd vertices there is an Eulerian trail (from C to E) but **no** Eulerian circuit, so the visitor could *not* walk every path once and return to A.

Q15. (a) An Eulerian trail exists only when a connected graph has 0 or 2 odd-degree vertices; here there are four, which is too many, so no Eulerian trail exists. (b) Adding a single edge between two of the four odd-degree vertices increases each of their degrees by 1, making both even. That leaves exactly two odd-degree vertices, and a connected graph with two odd vertices does have an Eulerian trail. So the least number of edges to add is **one**, joining two of the odd-degree vertices.

Q16. Crossing every bridge exactly once corresponds to an Eulerian trail on the graph, which requires at most two vertices of odd degree. Here all four vertices have odd degree $(3, 3, 3, 5)$, which is more than two, so no Eulerian trail (or circuit) exists — it is impossible to cross every bridge exactly once.

Q17. The degrees are $A3, B3, C2, D2, E2, F2$. (a) Two odd vertices (A and B) give an **Eulerian trail**, and it must run between them; for example $A-F-B-C-D-E-A-B$, using all seven edges once. (b) Since A and B

are odd, not every vertex is even, so there is **no** Eulerian circuit. (c) A **Hamiltonian path** visiting each vertex once is $A-F-B-C-D-E$. (d) Closing that loop, $A-F-B-C-D-E-A$ is a **Hamiltonian cycle**.

Q18. (a) A Hamiltonian cycle passes through every vertex and, being a cycle, must arrive at and leave each vertex on different edges — so every vertex on it needs degree at least 2. A pendant vertex has only one edge, so it cannot be both entered and left; therefore no Hamiltonian cycle can exist. (b) It still can. An Eulerian trail only requires the connected graph to have 0 or 2 odd-degree vertices. A pendant vertex has degree 1 (odd), so it can serve as one of the two allowed odd vertices — an endpoint of the trail.

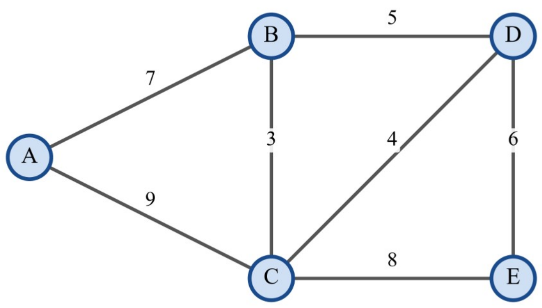
Q19. (a) Every edge has two ends and so contributes 1 to the degree of each of its two vertices, i.e. 2 to the total. Hence the sum of all degrees is $2\times(\text{number of edges})$, which is even. (b) Split the vertices into those of even degree and those of odd degree. The even-degree vertices contribute an even total; for the grand total to be even (part (a)), the odd-degree vertices must also contribute an even total, and a sum of odd numbers is even only when there is an even number of them. So the number of odd-degree vertices is even. (c) "Exactly one odd vertex" would be an *odd* number of odd-degree vertices, contradicting (b); so no such graph can exist.

Q20. (a) The degrees are $12, 22, 32, 44, 52, 62, 72$, so every junction is even. A connected graph with all vertices even has an Eulerian circuit, so the snow-plough can clear every road once and return to the depot; one route is $1-2-3-4-5-6-7-4-1$. (b) Junction 4 is the only link between the two loops of roads, so any route reaching every junction must pass through 4 twice. A Hamiltonian cycle visits each junction exactly once, so no such route exists — the school bus cannot visit every junction once and return to the depot.

Theory

In the earlier sections of this chapter a graph recorded only *which* vertices were joined by an edge. In most real situations each connection also carries a **number** that measures something about it — the length of a road in kilometres, the time to drive it in minutes, or the cost of using it in dollars. A graph in which every edge is labelled with such a number is a **weighted graph**, and the number on an edge is its **weight**. When a weighted graph models a practical situation such as a road system, a pipe network or a set of cable links, it is usually called a **network**.

The network below shows the distances, in kilometres, along the roads joining five towns A, B, C, D and E. The label on each edge is the weight of that edge: for example, the road from A to C is 9 km long, and the road from C to E is 8 km long.



Routes, paths and length. To travel from one vertex to another you follow a **route** — a sequence of edges leading from the first vertex to the second. A route that never returns to a vertex it has already visited is called a **path**. The **length** (or **weight**) of a route is found by adding the weights of the edges along it:

$$\text{length of a route} = \text{the sum of the weights of the edges in the route}.$$

For the network above, the path A–B–D has length $7+5=12$ km, and the path A–C–E has length $9+8=17$ km. Note that "length" here means **total weight**, not the number of edges: a path with many edges can be shorter than a path with few edges if its edges have small weights.

The shortest path problem. In applications we usually want to travel between two vertices as cheaply, as quickly, or over as short a distance as possible. The **shortest path** between two vertices is a path joining them whose total weight is as small as possible. Because "shortest" refers to the total weight, a shortest path may actually use *more* edges than some longer paths.

Finding a shortest path by inspection. For a small network you can find the shortest path simply by **listing the sensible routes** between the two vertices, working out the length of each, and choosing the smallest. Three ideas keep the listing short and reliable:

- start at the first vertex and, at each step, branch out to a neighbouring vertex you have not already visited, continuing until you reach the destination;
- a route that returns to a vertex it has already visited can be ignored: the edges travelled between the two visits form a closed detour of positive weight, so cutting that detour out leaves a shorter route between the same two vertices. Only paths need to be checked;
- in a larger network you need not write out every path — as soon as a partly built route already totals more than a complete route you have found, abandon it, because adding further edges can only make it longer.

Setting the routes and their lengths out in a table makes sure none is missed and makes the comparison easy. For the network above, the seven paths from A to E are:

Route	Length (km)
A–C–E	$9+8=17$
A–B–C–E	$7+3+8=18$
A–B–D–E	$7+5+6=18$
A–C–D–E	$9+4+6=19$
A–B–C–D–E	$7+3+4+6=20$
A–C–B–D–E	$9+3+5+6=23$
A–B–D–C–E	$7+5+4+8=24$

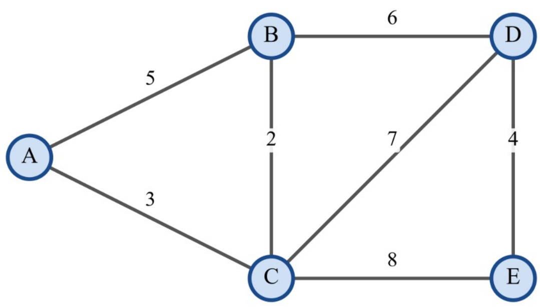
The smallest total is 17 km, so the shortest path from A to E is A–C–E, of length 17 km. Sometimes two or more routes share the smallest total; then the network has **more than one shortest path**, all of the same length.

Listing every route is practical only when the network is small. For a larger network the number of routes grows quickly, and a systematic method — **Dijkstra's algorithm** — is developed later in this chapter. This section concentrates on reading a weighted network, finding the length of a given path, and finding a shortest path in a small network by inspection.

Worked examples

Example 1 — scaffolded

The network shows the distances (in kilometres) along the roads joining five towns. Find the shortest path from A to E, and state its length.

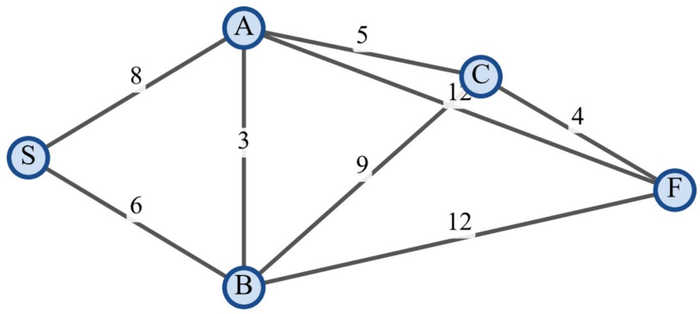


Working	Comment
List every route from A to E that uses each town at most once.	Start at A and branch to each neighbour; a route that revisits a town can be ignored.
A–C–E: $3+8=11$	Add the weights of the edges along the route.
A–C–D–E: $3+7+4=14$	
A–B–D–E: $5+6+4=15$	
A–B–C–E: $5+2+8=15$	
A–C–B–D–E: $3+2+6+4=15$	Branching from A to C first still allows a detour through B.
A–B–C–D–E: $5+2+7+4=18$	
A–B–D–C–E: $5+6+7+8=26$	Seven routes in all — none is missed.
The smallest total is 11 km, for A–C–E.	Compare the lengths and choose the least.

The shortest path from A to E is A–C–E, of length 11 km.

Example 2 — scaffolded

The network shows the driving times, in minutes, between a depot S and a delivery point F through suburbs A, B and C. Find the quickest route from S to F and the time it takes.

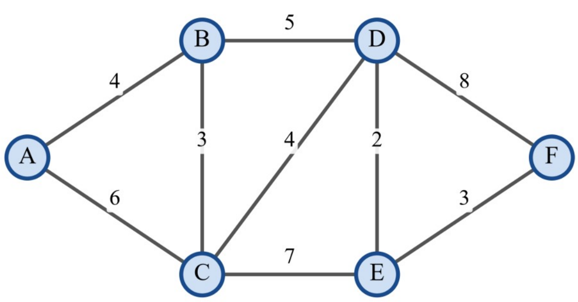


Working	Comment
List the shorter routes from S to F and add the times along each.	The weights are times, so the “shortest” path is the quickest one.
S–A–C–F: $8+5+4=17$	
S–B–A–C–F: $6+3+5+4=18$	
S–B–F: $6+12=18$	
S–B–C–F: $6+9+4=19$	
S–A–F: $8+12=20$	
Every other route from S to F takes more than 20 minutes.	None of them can be the quickest, so they need not be listed.
The least time is 17 minutes, for S–A–C–F.	Choose the route with the smallest total.

The quickest route is S–A–C–F, taking 17 minutes.

Example 3 — unscaffolded

For the network shown, the weights are distances in kilometres. (a) Find the length of the path A–C–E–F. (b) Find the shortest path from A to F, and state its length.



(a) The path A–C–E–F uses the edges A–C, C–E and E–F, so its length is

$$6+7+3=16\,\text{km}.$$

(b) Listing the shorter routes from A to F and their lengths:

Route	Length (km)
A–B–D–E–F	$4+5+2+3=14$

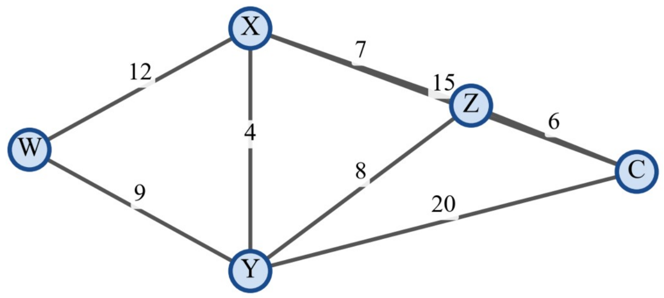
Route	Length (km)
A–C–D–E–F	$6+4+2+3=15$
A–B–C–D–E–F	$4+3+4+2+3=16$
A–C–E–F	$6+7+3=16$
A–B–C–E–F	$4+3+7+3=17$
A–B–D–F	$4+5+8=17$
A–C–D–F	$6+4+8=18$

Every other route from A to F is longer than 18 km, so none of them can be the shortest. The smallest total is 14 km. Notice that this route uses four edges, yet it is shorter than the three-edge route A–C–E–F; the number of edges does not decide the length.

∴ the shortest path from A to F is A–B–D–E–F, of length 14 km.

Example 4 — unscaffolded

A courier must drive from a warehouse W to a customer C. The network shows the cost, in dollars, of using each road (fuel plus tolls). Find the cheapest route and its cost.



The "shortest" path is now the one of least total cost. Listing the cheaper routes:

Route	Cost ($)
W–Y–Z–C	$9+8+6=23$
W–X–Z–C	$12+7+6=25$
W–Y–X–Z–C	$9+4+7+6=26$
W–X–C	$12+15=27$
W–Y–X–C	$9+4+15=28$
W–Y–C	$9+20=29$

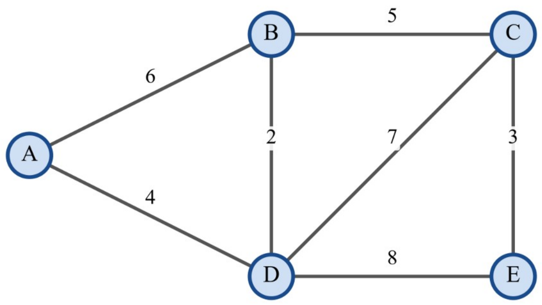
Every other route from W to C costs more than $29, so none of them can be the cheapest. The least total cost is $23, for the route W–Y–Z–C.

∴ the cheapest route is W–Y–Z–C, costing $23.

Practice questions

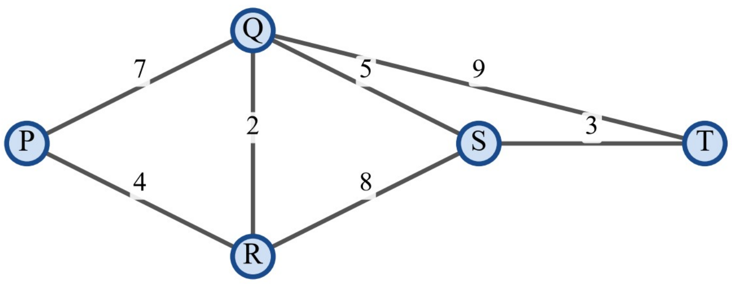
Scaffolded

Q1. The network shows the distances (km) between five towns.



(a) State the weight of the edge joining B and D.
(b) List the towns joined to D by a single road, with the length of each road.
(c) Find the length of the path A–B–C–E.
(d) Find the length of the path A–D–E.
(e) Find the length of the path A–D–C–E.

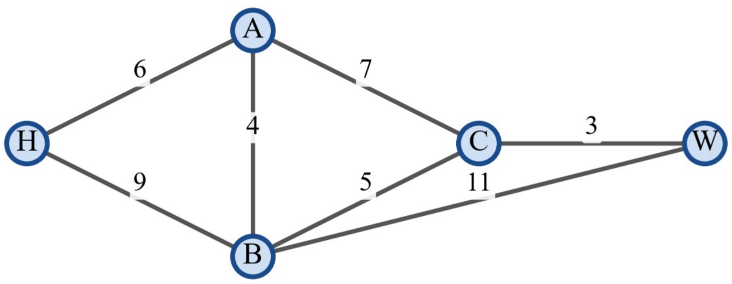
Q2. For the network shown, find the shortest path from P to T by completing the table of routes.



Route	Length
P–Q–T	
P–Q–S–T	
P–R–S–T	
P–R–Q–T	
P–R–Q–S–T	

(a) Complete the length of each route.
(b) State the shortest path and its length.

Q3. The network shows the travel times, in minutes, between a home H and a workplace W.

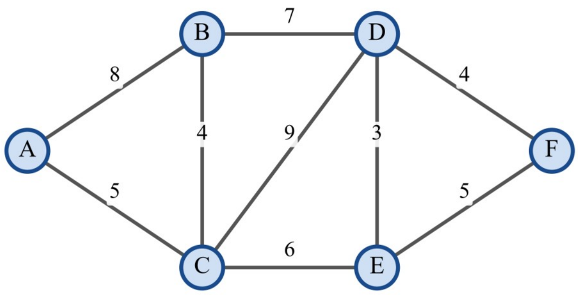


(a) Find the time taken on the route H–A–C–W.

(b) Find the time taken on the route H–B–W.

(c) Find the quickest route from H to W and state its time.

Q4. For the network shown, find the shortest path from A to F.

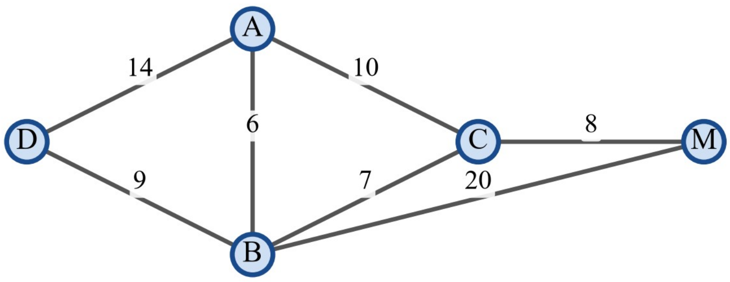


(a) Complete the length of each route below.

Route	Length
A–B–D–F	
A–C–D–F	
A–C–E–F	
A–C–E–D–F	

(b) State the shortest path and its length.

Q5. The network shows the cost, in dollars, of transporting goods along each road from a depot D to a market M.



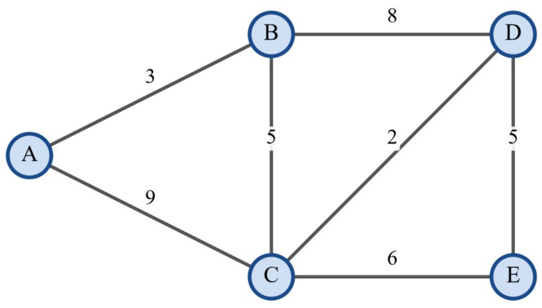
(a) Find the cost of the route D–A–C–M.

(b) Find the cost of the route D–B–C–M.

(c) Find the cost of the route D–B–M.

(d) State the cheapest route and its cost.

Q6. For the network shown, the weights are distances in kilometres.



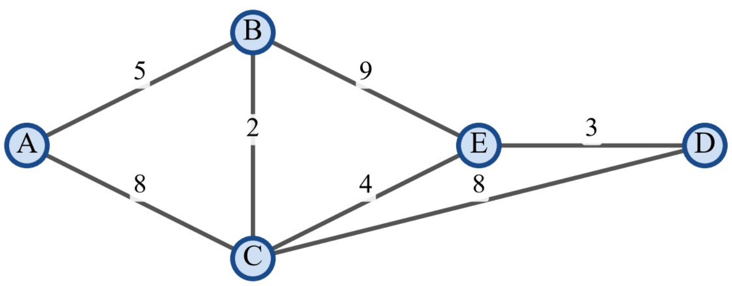
(a) List all the routes from A to E that do not revisit a vertex.

(b) Find the length of each route.

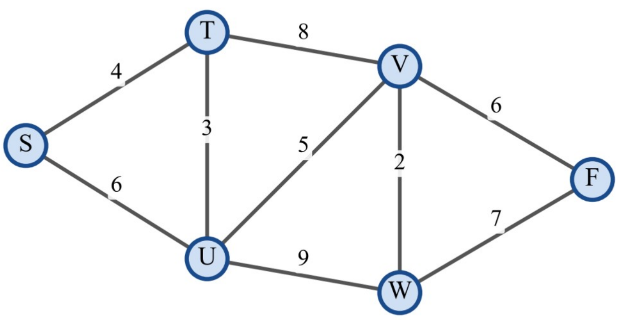
(c) State the shortest path from A to E and its length.

Unscaffolded

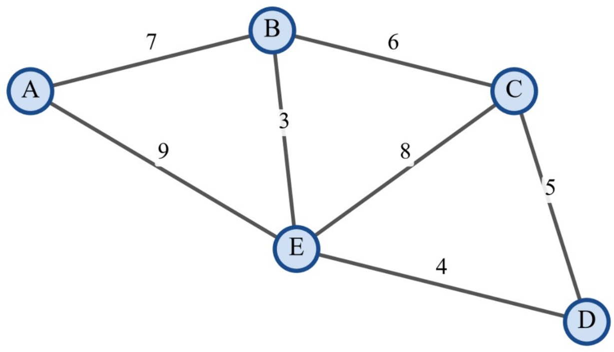
Q7. Find the shortest path from A to D in the network shown, and state its length. Comment on how many edges it uses compared with the route A–C–D.



Q8. Find the shortest path from S to F in the network shown, and state its length.



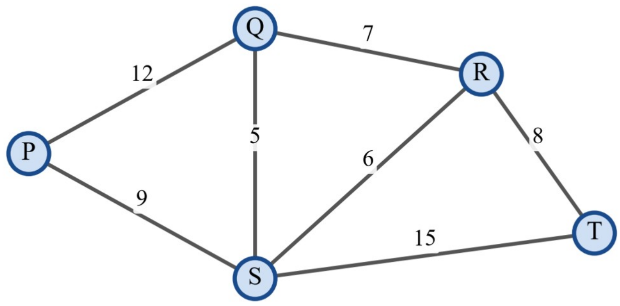
Q9. For the network shown, the weights are distances in kilometres.



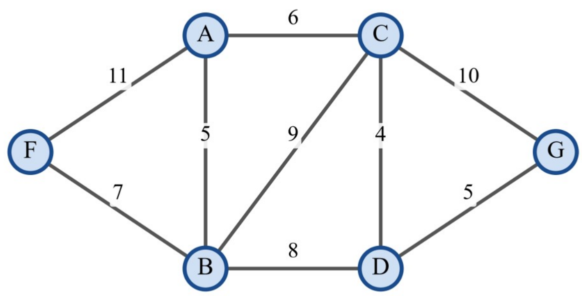
(a) Find the length of the path A–B–C–D.

(b) Find the shortest path from A to D, and state its length.

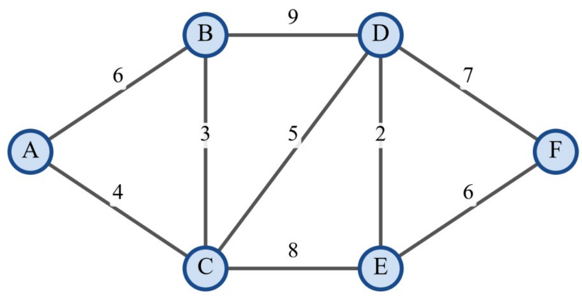
Q10. The network shows the distances (km) between five locations. Find the shortest path from P to T and its length.



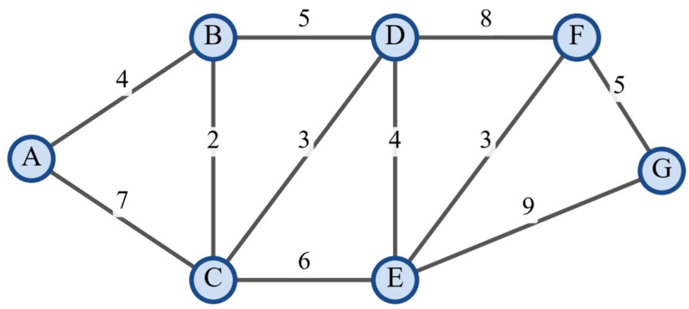
Q11. The network shows the cost, in dollars, of freighting a container along each route from a factory F to a port G. Find the cheapest route and its cost.



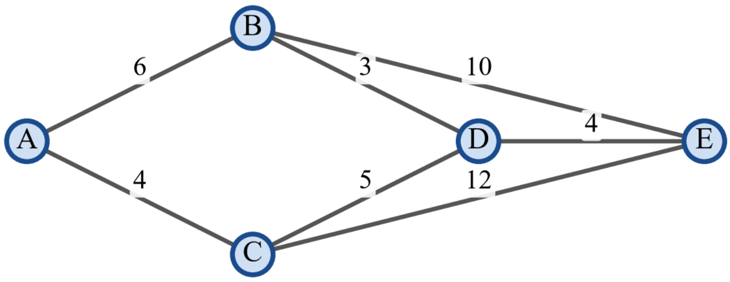
Q12. Find the shortest path from A to F in the network shown, and state its length.



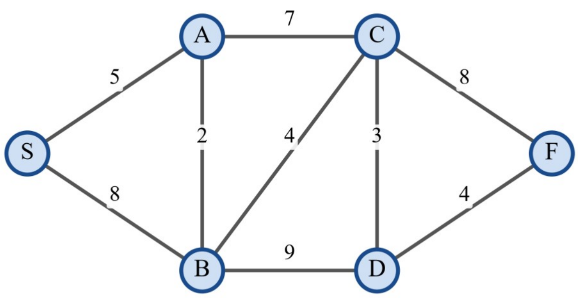
Q13. The network shows the distances (km) between seven towns. Find the shortest path from A to G, and state its length.



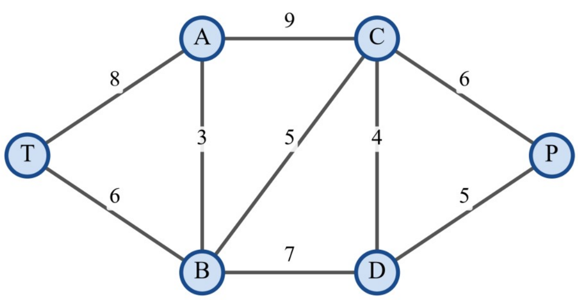
Q14. For the network shown, show that there are two different shortest paths from A to E, and state their common length.



Q15. Find the shortest path from S to F in the network shown, and state its length.



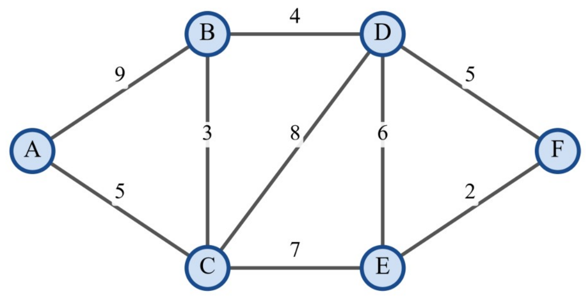
Q16. A tourist travels from a hotel T to a peak P. The network shows the walking times, in minutes, along the mountain tracks. Find the quickest route from T to P and the time it takes.



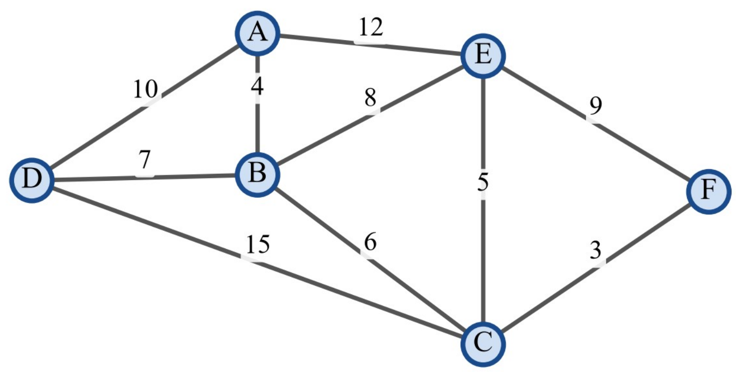
Q17. (a) Explain what is meant by the *length* of a path in a weighted network. (b) A student claims that the shortest path between two towns must always be the one that passes through the fewest towns. Explain, with reference to your answer to an earlier question, why this claim is false.

Q18. (a) A *route* through a network is allowed to pass through the same vertex more than once. In a network in which every edge weight is positive, explain why such a route can never be the shortest way of travelling between two vertices. (b) Explain how a network can have more than one shortest path between the same two vertices.

Q19. Find the shortest path from A to F in the network shown, and state its length.



Q20. A delivery van travels from a depot D to a store F. The network shows the distance, in kilometres, of each road.



(a) Find the shortest path from D to F, and state its length.

(b) Roadworks close the road that directly joins D and C. Explain why the length of the shortest path from D to F is unchanged.

Answers

Q1. (a) 2; (b) $A(4)$, $B(2)$, $C(7)$, $E(8)$; (c) 14; (d) 12; (e) 14. **Q2.** (a) 16, 15, 15, 15, 14; (b) $P–R–Q–S–T$, length 14. **Q3.** (a) 16 min; (b) 20 min; (c) $H–A–C–W$, 16 min. **Q4.** (a) 19, 18, 16, 18; (b) $A–C–E–F$, length 16. **Q5.** (a) \$32; (b) \$24; (c) \$29; (d) $D–B–C–M$, \$24. **Q6.** (a) see solutions; (b) see solutions; (c) $A–B–C–E$, length 14. **Q7.** $A–B–C–E–D$, length 14; it uses four edges, more than the two-edge route $A–C–D$ (length 16), yet is shorter. **Q8.** $S–U–V–F$, length 17. **Q9.** (a) 18; (b) $A–E–D$, length 13. **Q10.** $P–S–R–T$, length 23. **Q11.** $F–B–D–G$, cost \$20. **Q12.** $A–C–D–F$, length 16. **Q13.** $A–B–C–E–F–G$, length 20. **Q14.** $A–B–D–E$ and $A–C–D–E$, both length 13. **Q15.** $S–A–B–C–D–F$, length 18. **Q16.** $T–B–C–P$, 17 min. **Q17.** (a) The sum of the weights of the edges that make up the path. (b) A route with more edges can have a smaller total weight (e.g. Q7). **Q18.** (a) Revisiting a vertex adds a closed detour of positive weight, so cutting it out gives a shorter route between the same two vertices. (b) Two different routes can have the same least total weight. **Q19.** $A–C–E–F$, length 14. **Q20.** (a) $D–B–C–F$, length 16; (b) that path does not use the road $D–C$, so closing it does not change the shortest path.

Fully-worked solutions

Q1. (a) The edge joining B and D is labelled 2. (b) The roads at D are $A–D(4)$, $B–D(2)$, $C–D(7)$ and $D–E(8)$, so D is joined to A, B, C and E. (c) $A–B–C–E=6+5+3=14$. (d) $A–D–E=4+8=12$. (e) $A–D–C–E=4+7+3=14$.

Q2. Adding the weights along each route:

Route	Length
$P–Q–T$	$7+9=16$
$P–Q–S–T$	$7+5+3=15$
$P–R–S–T$	$4+8+3=15$
$P–R–Q–T$	$4+2+9=15$
$P–R–Q–S–T$	$4+2+5+3=14$

The smallest total is 14, so the shortest path is $P–R–Q–S–T$, of length 14.

Q3. (a) $H–A–C–W=6+7+3=16$ minutes. (b) $H–B–W=9+11=20$ minutes. (c) The seven routes from H to W are $H–A–C–W=16$, $H–B–C–W=9+5+3=17$, $H–A–B–C–W=6+4+5+3=18$, $H–B–W=20$, $H–A–B–W=6+4+11=21$, $H–B–A–C–W=9+4+7+3=23$ and $H–A–C–B–W=6+7+5+11=29$. The quickest is $H–A–C–W$, taking 16 minutes.

Q4. (a) $A–B–D–F=8+7+4=19$; $A–C–D–F=5+9+4=18$; $A–C–E–F=5+6+5=16$; $A–C–E–D–F=5+6+3+4=18$. (b) The smallest total is 16, so the shortest path is $A–C–E–F$, of length 16.

Q5. (a) $D–A–C–M=14+10+8=32$, so \$32. (b) $D–B–C–M=9+7+8=24$, so \$24. (c) $D–B–M=9+20=29$, so \$29. (d) Comparing these with any other route (for example $D–A–B–C–M=14+6+7+8=35$), the least cost is \$24, for the route $D–B–C–M$.

Q6. (a), (b) Branching out from A, first through B and then through C, gives seven routes to E that do not revisit a vertex:

Route	Length (km)
$A–B–C–E$	$3+5+6=14$
$A–B–C–D–E$	$3+5+2+5=15$
$A–C–E$	$9+6=15$
$A–C–D–E$	$9+2+5=16$
$A–B–D–E$	$3+8+5=16$

Route	Length (km)
A–B–D–C–E	$3+8+2+6=19$
A–C–B–D–E	$9+5+8+5=27$

(c) The smallest total is 14 km, so the shortest path is A–B–C–E, of length 14 km.

Q7. The seven routes from A to D are A–B–C–E–$D=5+2+4+3=14$, A–B–C–$D=5+2+8=15$, A–C–E–$D=8+4+3=15$, A–C–$D=8+8=16$, A–B–E–$D=5+9+3=17$, A–C–B–E–$D=8+2+9+3=22$ and A–B–E–C–$D=5+9+4+8=26$. The shortest is A–B–C–E–D, of length 14. It uses four edges — more than the two-edge route A–C–D (length 16) — which shows that the shortest path need not be the one with the fewest edges.

Q8. The routes from S to F include S–U–V–$F=6+5+6=17$, S–T–V–$F=4+8+6=18$, S–T–U–V–$F=4+3+5+6=18$, S–U–V–W–$F=6+5+2+7=20$ and S–U–W–$F=6+9+7=22$; every route not listed is 21 or more. The shortest is S–U–V–F, of length 17.

Q9. (a) A–B–C–$D=7+6+5=18$ km. (b) The seven routes from A to D are A–E–$D=9+4=13$, A–B–E–$D=7+3+4=14$, A–B–C–$D=18$, A–E–C–$D=9+8+5=22$, A–B–E–C–$D=7+3+8+5=23$, A–E–B–C–$D=9+3+6+5=23$ and A–B–C–E–$D=7+6+8+4=25$. The shortest is A–E–D, of length 13 km — much shorter than the "outer" route A–B–C–D found in part (a).

Q10. The seven routes from P to T are P–S–R–$T=9+6+8=23$, P–S–$T=9+15=24$, P–Q–R–$T=12+7+8=27$, P–S–Q–R–$T=9+5+7+8=29$, P–Q–S–R–$T=12+5+6+8=31$, P–Q–S–$T=12+5+15=32$ and P–Q–R–S–$T=12+7+6+15=40$. The shortest is P–S–R–T, of length 23.

Q11. The routes from F to G include F–B–D–$G=7+8+5=20$, F–B–C–D–$G=7+9+4+5=25$, F–A–C–D–$G=11+6+4+5=26$, F–B–C–$G=7+9+10=26$ and F–A–C–$G=11+6+10=27$; every route not listed costs \$27 or more. The cheapest is F–B–D–G, costing \$20.

Q12. The routes from A to F include A–C–D–$F=4+5+7=16$, A–C–D–E–$F=4+5+2+6=17$, A–C–E–$F=4+8+6=18$, A–B–C–D–$F=6+3+5+7=21$ and A–B–D–$F=6+9+7=22$; every route not listed is 21 or more. The shortest is A–C–D–F, of length 16.

Q13. Working from A, the shorter routes to G include A–B–C–E–F–$G=4+2+6+3+5=20$, A–B–C–E–$G=4+2+6+9=21$, A–C–E–F–$G=7+6+3+5=21$, A–B–D–E–F–$G=4+5+4+3+5=21$ and A–B–C–D–F–$G=4+2+3+8+5=22$; every route not listed is 21 or more. The smallest total is 20, so the shortest path is A–B–C–E–F–G, of length 20.

Q14. The six routes from A to E are A–B–D–$E=6+3+4=13$, A–C–D–$E=4+5+4=13$, A–B–$E=6+10=16$, A–C–$E=4+12=16$, A–C–D–B–$E=4+5+3+10=22$ and A–B–D–C–$E=6+3+5+12=26$. Exactly two of them, A–B–D–E and A–C–D–E, share the smallest total, so there are two shortest paths, each of length 13.

Q15. The routes from S to F include S–A–B–C–D–$F=5+2+4+3+4=18$, S–A–B–C–$F=5+2+4+8=19$, S–A–C–D–$F=5+7+3+4=19$, S–B–C–D–$F=8+4+3+4=19$ and S–A–C–$F=5+7+8=20$; every route not listed is 20 or more. The smallest total is 18, so the shortest path is S–A–B–C–D–F, of length 18. Once again a five-edge route is shorter than every route with fewer edges.

Q16. The routes from T to P include T–B–C–$P=6+5+6=17$, T–B–D–$P=6+7+5=18$, T–B–C–D–$P=6+5+4+5=20$, T–A–B–C–$P=8+3+5+6=22$ and T–A–C–$P=8+9+6=23$; every route not listed takes 23 minutes or more. The quickest is T–B–C–P, taking 17 minutes.

Q17. (a) The length of a path is the sum of the weights of the edges that make it up; it is the total distance, time or cost of travelling along the path, not the number of edges. (b) The claim is false. In Q7 the shortest path A–B–C–E–D uses four edges but has length 14, while the route A–C–D uses only two edges yet has the greater length 16. A route with more edges can be shorter overall when its edges have small weights, so the fewest towns does not guarantee the shortest path.

Q18. (a) Suppose a route from one vertex to another passes through some vertex twice. The edges travelled between the two visits form a closed loop, and because every edge weight is positive, that loop adds a positive amount to the total. Cutting the loop out leaves a shorter route with the same start and end, so the original route could not have been the shortest. Hence the shortest route between two vertices never repeats a vertex — it is always a path. (b) If two different routes between the same two vertices happen to have exactly the same total weight, and that weight is the smallest possible, then both routes are shortest paths. A network can therefore have more than one shortest path between a given pair of vertices (as in Q14).

Q19. The routes from A to F include $A–C–E–F = 5+7+2 = 14$, $A–C–B–D–F = 5+3+4+5 = 17$, $A–B–D–F = 9+4+5 = 18$ and $A–C–D–F = 5+8+5 = 18$; every route not listed is 20 or more. The shortest is $A–C–E–F$, of length 14.

Q20. (a) The routes from D to F include $D–B–C–F = 7+6+3 = 16$, $D–C–F = 15+3 = 18$, $D–A–B–C–F = 10+4+6+3 = 23$, $D–B–E–C–F = 7+8+5+3 = 23$ and $D–B–E–F = 7+8+9 = 24$; every route not listed is 24 km or more. The shortest is $D–B–C–F$, of length 16 km. (b) The shortest path $D–B–C–F$ travels D to B to C to F and never uses the road directly joining D and C. Closing that road therefore removes only routes that were already longer than 16 km, so the shortest path — and its length of 16 km — is unchanged.

Theory

In an earlier section the edges of a network were given **weights** — numbers that record a distance, a travel time, or a cost along each edge. A natural question is then the **shortest path problem**: of all the paths joining two vertices, which has the smallest total weight? For a small network the answer can often be found **by inspection** — list a few sensible paths, add their weights, and compare. As a network grows this quickly becomes unreliable, because the number of paths explodes and it is easy to overlook the best one.

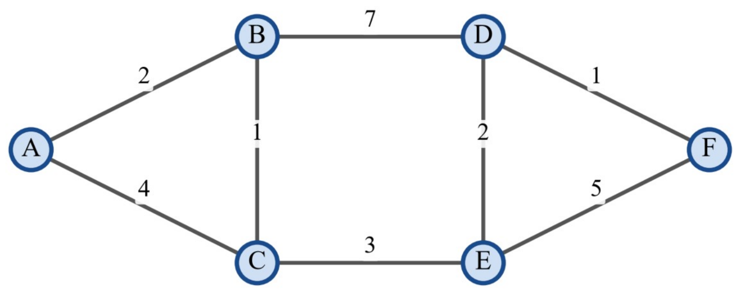
Dijkstra's algorithm is a systematic procedure that removes the guesswork. Starting from one chosen vertex, it finds the shortest distance from that vertex to **every** other vertex in the network. The idea is to give each vertex a **label** — a running record of the shortest distance found *so far* from the start. A label is at first only **tentative** (it may still be improved); once we are certain it cannot be beaten it is made **permanent**.

The algorithm.

1. Give the start vertex the permanent label 0. Give every other vertex the tentative label ∞, meaning "no path found yet".
2. From the vertex most recently made permanent (the **current** vertex), look at each neighbour that is *not yet permanent*. Add the current vertex's permanent label to the weight of the connecting edge. If this total is **smaller** than the neighbour's tentative label, replace the tentative label with the smaller value.
3. Among all vertices that are not yet permanent, choose the one with the **smallest** tentative label and make it permanent. (No later path can improve it, because every edge weight is positive.)
4. Repeat steps 2 and 3, each time working from the vertex just made permanent, until the destination vertex — or every vertex — is permanent.

The permanent label of a vertex is then the **shortest distance** to it from the start.

Recording the working — the label table. The steps are set out in a **Dijkstra label table**. Each row corresponds to making one vertex permanent, and the columns are the vertices. In each row the newly permanent value is written in **bold**; a vertex made permanent in an *earlier* row is shown as a dash; and every other cell holds that vertex's current tentative label. Consider the network below, with start vertex A.



Step	Made permanent	A	B	C	D	E	F
1	$A=0$	**0**	2	4	∞	∞	∞
2	$B=2$	—	**2**	3	9	∞	∞
3	$C=3$	—	—	**3**	9	6	∞
4	$E=6$	—	—	—	8	**6**	11
5	$D=8$	—	—	—	**8**	—	9
6	$F=9$	—	—	—	—	—	**9**

Reading across the table: from A we reach B (distance 2) and C (distance 4). The smallest tentative label is $B=2$, so B becomes permanent. Working from B, the label of C improves from 4 to $2+1=3$ and D is first reached at $2+7=9$. Next $C=3$ is made permanent, then $E=6$, then $D=8$ (its label improving from 9 to $6+2=8$), and finally $F=9$ (improving from 11 to $8+1=9$).

Back-tracking to find the path. The table gives the shortest *distance* to F (namely 9) but not yet the *route*. To recover it, start at the destination and step backwards: a vertex u lies on the shortest path just before v when

$$(\text{permanent label of } u)+(\text{weight of edge } uv)=(\text{permanent label of } v).$$

Here $F=9$: the edge DF gives $8+1=9$ ✓ (while EF gives $6+5=11$ ✗), so step back to D. Then $D=8$ comes from E ($6+2=8$), $E=6$ from C ($3+3=6$), $C=3$ from B ($2+1=3$), and $B=2$ from A. Reversing gives the shortest path

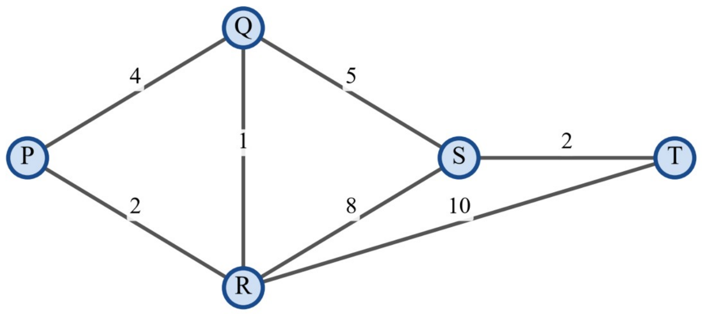
$$A \to B \to C \to E \to D \to F, \text{ length } 9.$$

Two remarks. First, Dijkstra's algorithm applies unchanged to a **directed network** (a digraph): in step 2 we simply follow only the arrows leaving the current vertex. Second, because the algorithm labels every vertex, a single run gives the shortest distance from the start to *all* other vertices at once, not merely to one destination.

Worked examples

Example 1 — scaffolded

For the network shown, use Dijkstra's algorithm to find the shortest path from P to T, and state its length.



Working	Comment
Label $P=0$; every other vertex is ∞.	The start is 0 away from itself.
From P: $Q=0+4=4$, $R=0+2=2$.	Update each neighbour of P.
Smallest tentative label is $R=2$, so make R permanent.	Choose the least tentative value.
From R: $Q=min(4,2+1)=3$, $S=2+8=10$, $T=2+10=12$.	The label of Q improves from 4 to 3.
Smallest tentative label is $Q=3$; make Q permanent.	
From Q: $S=min(10,3+5)=8$.	S improves from 10 to 8.
Smallest tentative label is $S=8$; make S permanent.	
From S: $T=min(12,8+2)=10$.	T improves from 12 to 10.
Only T remains; make $T=10$ permanent.	The shortest distance to T is 10.

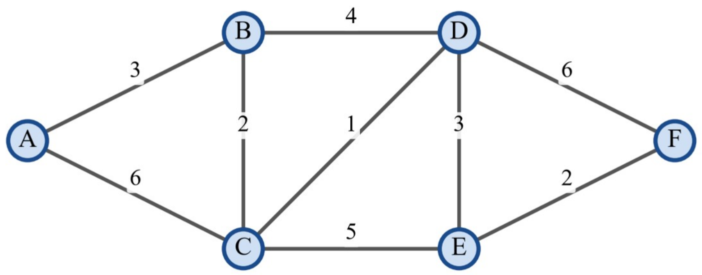
The completed label table is:

Step	Made permanent	P	Q	R	S	T
1	$P=0$	**0**	4	2	∞	∞
2	$R=2$	—	3	**2**	10	12
3	$Q=3$	—	**3**	—	8	12
4	$S=8$	—	—	—	**8**	10
5	$T=10$	—	—	—	—	**10**

Back-tracking from T: $T=10$ comes from S ($8+2=10$), $S=8$ from Q ($3+5=8$), $Q=3$ from R ($2+1=3$), and $R=2$ from P. The shortest path is $P\rightarrow R\rightarrow Q\rightarrow S\rightarrow T$, of length 10.

Example 2 — scaffolded

The network below shows the times (in minutes) to walk along paths between six landmarks. Find the quickest route from A to F.



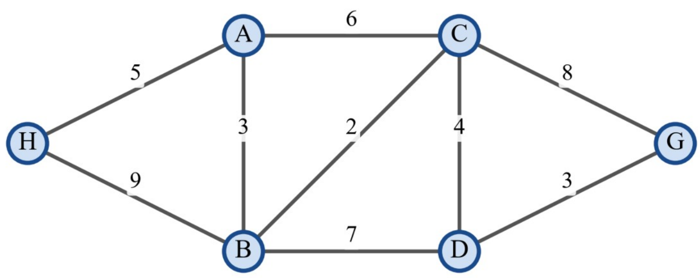
Working	Comment
Label $A=0$; all other labels ∞.	
From A: $B=3$, $C=6$. Make $B=3$ permanent (smallest).	
From B: $C=min(6,3+2)=5$, $D=3+4=7$. Make $C=5$ permanent.	C improves 6 → 5.
From C: $D=min(7,5+1)=6$, $E=5+5=10$. Make $D=6$ permanent.	D improves 7 → 6.
From D: $E=min(10,6+3)=9$, $F=6+6=12$. Make $E=9$ permanent.	E improves 10 → 9.
From E: $F=min(12,9+2)=11$. Make $F=11$ permanent.	F improves 12 → 11.

Step	Made permanent	A	B	C	D	E	F
1	$A=0$	**0**	3	6	∞	∞	∞
2	$B=3$	—	**3**	5	7	∞	∞
3	$C=5$	—	—	**5**	6	10	∞
4	$D=6$	—	—	—	**6**	9	12
5	$E=9$	—	—	—	—	**9**	11
6	$F=11$	—	—	—	—	—	**11**

Back-tracking from F: $F=11$ from E $(9+2)$, $E=9$ from D $(6+3)$, $D=6$ from C $(5+1)$, $C=5$ from B $(3+2)$, $B=3$ from A. The quickest route is $A \to B \to C \to D \to E \to F$, taking 11 minutes.

Example 3 — unscaffolded

The network below gives the distances, in kilometres, along roads connecting a farmhouse H to a grain depot G through several intermediate towns. Determine the shortest route from H to G and its length.



Applying Dijkstra's algorithm from H, the vertices become permanent in the order

$$H=0, A=5, B=8, C=10, D=14, G=17.$$

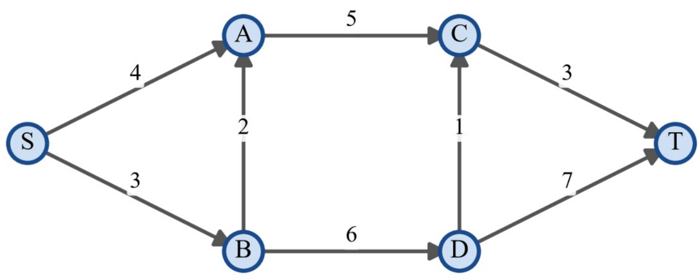
The label of B improves from its first value 9 (edge HB) to 8 (via A, since $5+3=8$); C settles at $8+2=10$ (via B); D improves to $10+4=14$ (via C); and G improves from $10+8=18$ (edge CG) to $14+3=17$ (via D).

Back-tracking from G: $G=17$ comes from D $(14+3=17)$, $D=14$ from C $(10+4)$, $C=10$ from B $(8+2)$, $B=8$ from A $(5+3)$, and $A=5$ from H.

$\therefore$ the shortest route is $H \to A \to B \to C \to D \to G$, a distance of 17 km.

Example 4 — unscaffolded

A courier must travel through a system of one-way streets from depot S to destination T. The directed network shows the travel time (in minutes) along each street. Find the quickest route and its time.



Because the streets are one-way, at each stage we follow only the arrows *leaving* the current vertex. Running Dijkstra's algorithm from S, the permanent labels are

$$S=0, B=3, A=4, C=9, D=9, T=12.$$

From S we reach A $(=4)$ and B $(=3)$; making B permanent offers $A=3+2=5$, which does **not** improve the existing 4, so A stays 4 (reached directly from S). From A the label $C=4+5=9$ is set, and T finally settles at $9+3=12$ via C.

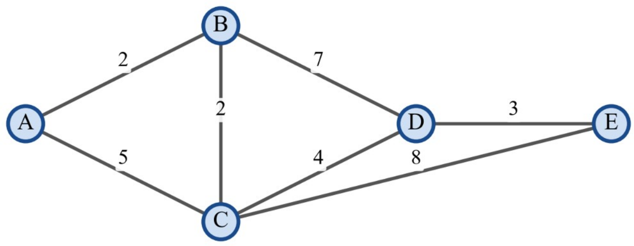
Back-tracking from T: $T=12$ comes from C $(9+3=12$, whereas D gives $9+7=16)$, $C=9$ from A $(4+5=9)$, and $A=4$ directly from S.

$\therefore$ the quickest route is $S \to A \to C \to T$, taking 12 minutes.

Practice questions

Scaffolded

Q1. Use Dijkstra's algorithm on the network shown to find the shortest path from A to E.



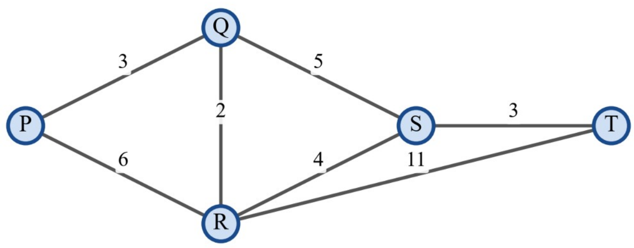
(a) Label $A = 0$ and update its neighbours B and C.

(b) Make the vertex with the smallest tentative label permanent, then update *its* neighbours. Continue until E is permanent, recording your working in a label table.

(c) State the shortest distance from A to E.

(d) Back-track to write down the shortest path.

Q2. For the network shown, find the shortest path from P to T.

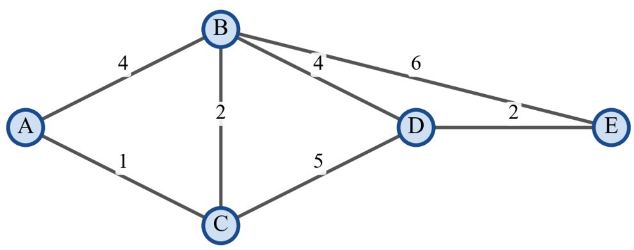


(a) Complete a Dijkstra label table, starting with $P = 0$.

(b) State the shortest distance from P to T.

(c) Back-track to state the shortest path.

Q3. For the network shown, find the shortest path from A to E.

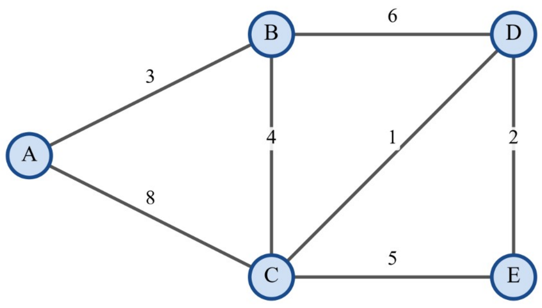


(a) Show that after C is made permanent, the label of B becomes 3.

(b) Complete the algorithm and state the shortest distance from A to E.

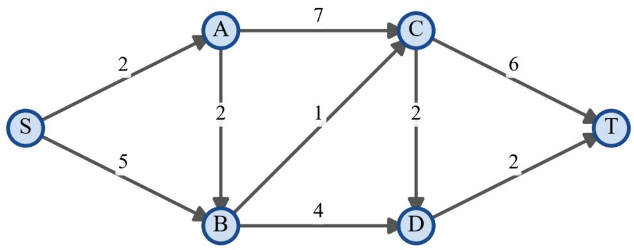
(c) Write down the shortest path.

Q4. The network shows the costs (in dollars) of cables between five junctions.



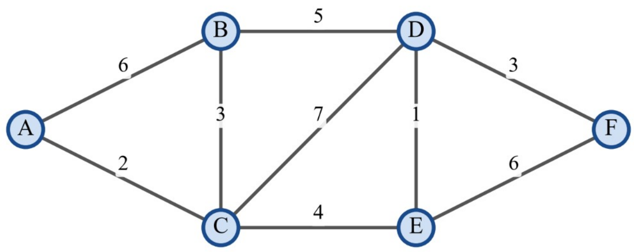
(a) Apply Dijkstra's algorithm from A and record the permanent label of every vertex.
(b) Hence write down the shortest distance from A to each of B, C, D and E.
(c) State the shortest path from A to E.

Q5. The directed network shows one-way streets with travel times in minutes. Find the shortest path from S to T.



(a) Explain why, when working from a vertex, you follow only the arrows leaving it.
(b) Complete a label table and state the shortest distance from S to T.
(c) Back-track to state the shortest route.

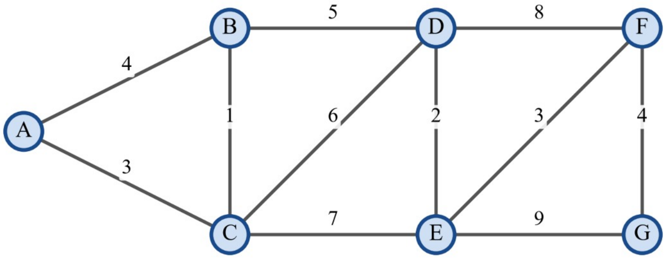
Q6. For the network shown, find the shortest path from A to F.



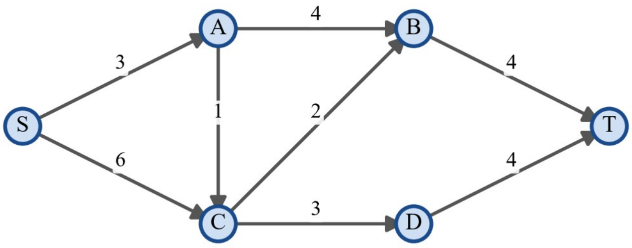
(a) Make the vertices permanent one at a time, recording a label table.
(b) State the shortest distance from A to F.
(c) Write down the shortest path.

Unscaffolded

Q7. For the network shown, find the shortest path from A to G and its length.



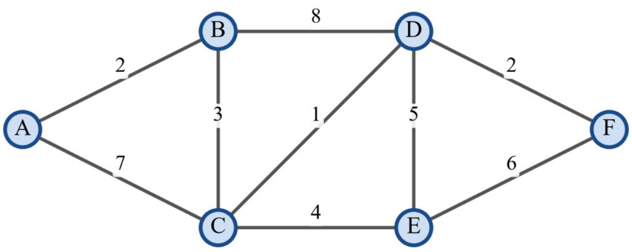
Q8. The directed network shows travel times (in minutes) through a one-way system. Find the shortest path from S to T and its length.



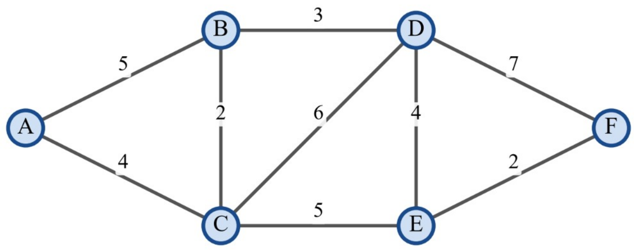
Q9. The table gives the road distances (in km) between towns that are directly connected; a blank entry means there is no direct road. Using Dijkstra's algorithm, find the shortest distance from town A to town D and state the route.

	A	B	C	D	E
A	—	8	5		
B	8	—	2	6	
C	5	2	—	9	4
D		6	9	—	3
E			4	3	—

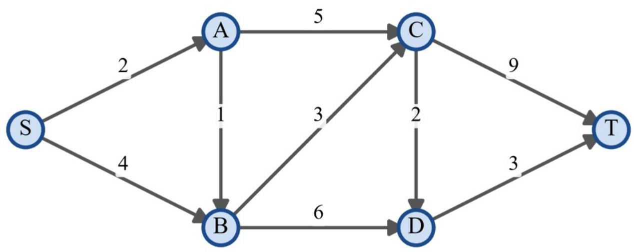
Q10. For the network shown, use one run of Dijkstra's algorithm from A to find the shortest distance from A to *every* other vertex, and state the shortest path to F.



Q11. For the network shown, Ravi claims that the shortest path from A to F is $A \to B \to D \to F$, of length 15. Use Dijkstra's algorithm to find the true shortest path and its length, and hence state whether Ravi is correct.



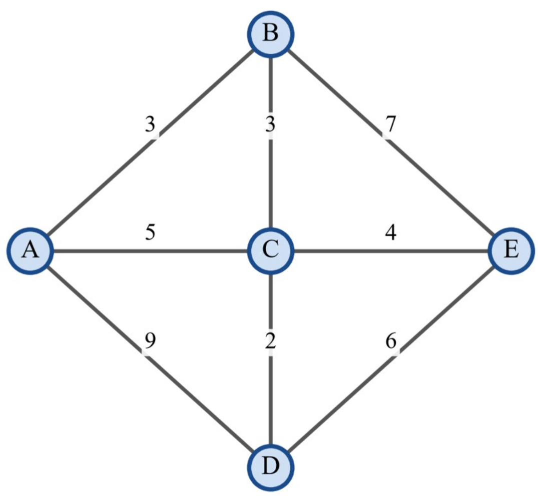
Q12. The directed network shows one-way roads with distances in km. Find the shortest path from S to T and its length.



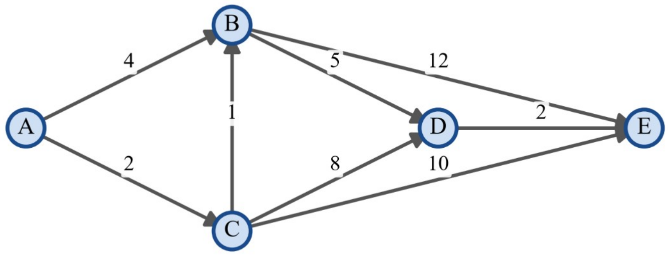
Q13. The table gives the distances (in km) along the roads that directly connect six locations. Find the shortest distance from P to U and state the route.

	P	Q	R	S	T	U
P	—	7	9			
Q	7	—	3	5		
R	9	3	—	8	4	
S		5	8	—	6	9
T			4	6	—	2
U				9	2	—

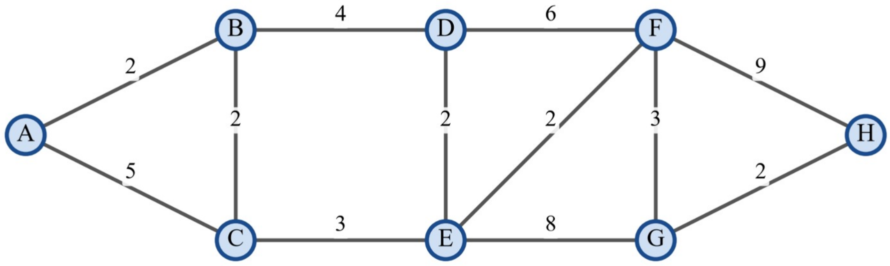
Q14. For the network shown, apply Dijkstra's algorithm from A. (a) State the shortest distance from A to each other vertex. (b) Which vertex is farthest from A, and by which path?



Q15. The directed network shows one-way links with weights. Find the shortest path from A to E and its length.

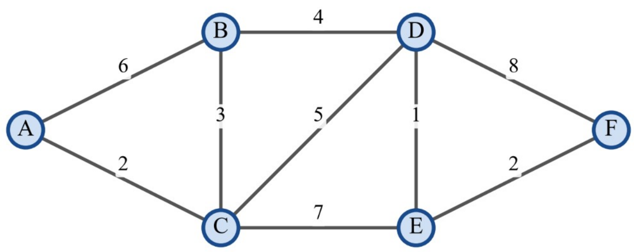


Q16. For the network shown, find the shortest path from A to H and its length.

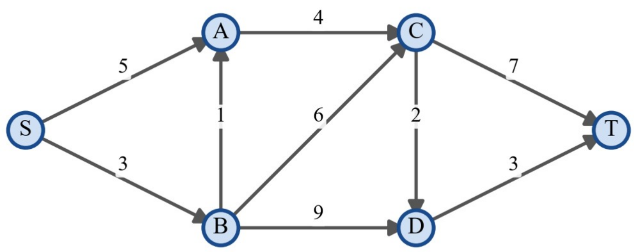


Q17. Explain, in the context of Dijkstra's algorithm: (a) what the label 0 on the start vertex represents; (b) why, at each stage, the vertex with the *smallest* tentative label may safely be made permanent; (c) how the shortest *path* (not merely its length) is recovered once the labels are complete.

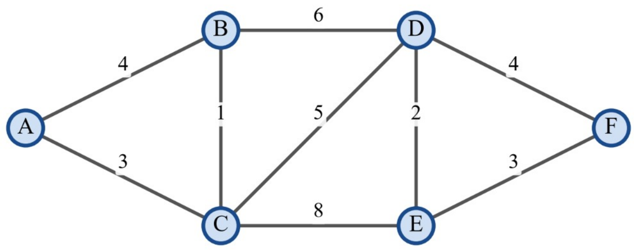
Q18. For the network shown, use a single run of Dijkstra's algorithm from A to find the shortest path from A to D and the shortest path from A to F, with their lengths.



Q19. A delivery van travels through the one-way network shown, where each weight is a travel time in minutes. Find the quickest route from S to T and its time.



Q20. For the network shown, weights are distances in km.



(a) Find the shortest path from A to F and its length.

(b) Road works close the direct road DF. Find the new shortest path from A to F and its length.

Answers

Q1. (a) see solutions; (b) see solutions; (c) shortest distance 11; (d) $A \to B \to C \to D \to E$. **Q2.** (a) see solutions; (b) shortest distance 11; (c) $P \to Q \to S \to T$. **Q3.** (a) $C=1$, then $B=min(4,1+2)=3$; (b) shortest distance 8; (c) $A \to C \to D \to E$. **Q4.** (a) $A=0$, $B=3$, $C=7$, $D=8$, $E=10$; (b) B \$3, C \$7, D \$8, E \$10; (c) $A \to B \to C \to D \to E$. **Q5.** (a) A one-way street can be travelled only in the direction of its arrow, so from the current vertex only the arrows leaving it lead anywhere new; (b) shortest distance 9; (c) $S \to A \to B \to C \to D \to T$. **Q6.** (a) see solutions; (b) shortest distance 10; (c) $A \to C \to E \to D \to F$. **Q7.** $A \to C \to E \to F \to G$, length 17. **Q8.** $S \to A \to C \to B \to T$, length 10. **Q9.** $A \to C \to E \to D$, distance 12 km. **Q10.** $B=2$, $C=5$, $D=6$, $E=9$, $F=8$; shortest path to F is $A \to B \to C \to D \to F$, length 8. **Q11.** True shortest path $A \to C \to E \to F$, length 11; Ravi is **incorrect** (his route is 15). **Q12.** $S \to A \to B \to C \to D \to T$, length 11. **Q13.** $P \to R \to T \to U$, distance 15 km. **Q14.** (a) $B=3$, $C=5$, $D=7$, $E=9$; (b) E is farthest (9), by $A \to C \to E$. **Q15.** $A \to C \to B \to D \to E$, length 10. **Q16.** $A \to B \to C \to E \to F \to G \to H$, length 14. **Q17.** (a) The shortest distance from the start to itself; (b) no not-yet-permanent vertex has a smaller label, and all weights are positive, so no later path can reach it more cheaply; (c) back-track from the destination, at each step choosing a predecessor u with (label of u) + (edge weight) = (label of current vertex), until the start is reached, then reverse. **Q18.** $A \to C \to D$ (length 7) to D; $A \to C \to D \to E \to F$ (length 10) to F. **Q19.** $S \to B \to A \to C \to D \to T$, 13 minutes. **Q20.** (a) $A \to C \to D \to F$, length 12; (b) with DF closed, $A \to C \to D \to E \to F$, length 13.

Fully-worked solutions

Q1. From $A=0$: $B=2$, $C=5$. Make $B=2$ permanent; then $C=min(5,2+2)=4$ and $D=2+7=9$. Make $C=4$ permanent; then $D=min(9,4+4)=8$ and $E=4+8=12$. Make $D=8$ permanent; then $E=min(12,8+3)=11$. Make $E=11$ permanent. Back-tracking: $E=11$ from D $(8+3)$, $D=8$ from C $(4+4)$, $C=4$ from B $(2+2)$, $B=2$ from A. Shortest path $A \to B \to C \to D \to E$, length 11.

Q2. From $P=0$: $Q=3$, $R=6$. Make $Q=3$ permanent; then $R=min(6,3+2)=5$ and $S=3+5=8$. Make $R=5$ permanent; then $S=min(8,5+4)=8$ (no change) and $T=5+11=16$. Make $S=8$ permanent; then $T=min(16,8+3)=11$. Make $T=11$ permanent. Back-tracking: $T=11$ from S $(8+3)$, $S=8$ from Q $(3+5)$, $Q=3$ from P. Shortest path $P \to Q \to S \to T$, length 11.

Q3. From $A=0$: $B=4$, $C=1$. The smallest tentative label is $C=1$, so C becomes permanent, and updating from C gives $B=min(4,1+2)=3$ and $D=1+5=6$. Make $B=3$ permanent; from B, $D=min(6,3+4)=6$ (no change) and $E=3+6=9$. Make $D=6$ permanent; then $E=min(9,6+2)=8$. Make $E=8$ permanent. Back-tracking: $E=8$ from D $(6+2)$, $D=6$ from C $(1+5)$, $C=1$ from A. Shortest path $A \to C \to D \to E$, length 8.

Q4. From $A=0$: $B=3$, $C=8$. Make $B=3$ permanent; then $C=min(8,3+4)=7$ and $D=3+6=9$. Make $C=7$ permanent; then $D=min(9,7+1)=8$ and $E=7+5=12$. Make $D=8$ permanent; then $E=min(12,8+2)=10$. Make $E=10$ permanent. The permanent labels are $A=0$, $B=3$, $C=7$, $D=8$, $E=10$, so the cheapest connections from A cost \$3, \$7, \$8 and \$10 respectively. Back-tracking from E: $E=10$ from D $(8+2)$, $D=8$ from C $(7+1)$, $C=7$ from B $(3+4)$, $B=3$ from A, giving $A \to B \to C \to D \to E$.

Q5. (a) The streets are one-way, so a vertex can only be reached by travelling *along* an arrow into it; from the current vertex, only the arrows leaving it lead to new vertices. Running the algorithm from $S=0$: $A=2$, $B=5$. Make $A=2$ permanent; then $B=min(5,2+2)=4$ and $C=2+7=9$. Make $B=4$ permanent; then $C=min(9,4+1)=5$ and $D=4+4=8$. Make $C=5$ permanent; then $D=min(8,5+2)=7$ and $T=5+6=11$. Make $D=7$ permanent; then $T=min(11,7+2)=9$. Make $T=9$ permanent. Back-tracking: $T=9$ from D $(7+2)$, $D=7$ from C $(5+2)$, $C=5$ from B $(4+1)$, $B=4$ from A $(2+2)$, $A=2$ from S. Shortest route $S \to A \to B \to C \to D \to T$, length 9.

Q6. From $A=0$: $B=6$, $C=2$. Make $C=2$ permanent; then $B=min(6,2+3)=5$, $D=2+7=9$, $E=2+4=6$. Make $B=5$ permanent (no improvement to D). Make $E=6$ permanent; then $D=min(9,6+1)=7$ and $F=6+6=12$. Make $D=7$ permanent; then $F=min(12,7+3)=10$. Make $F=10$ permanent. Back-tracking: $F=10$ from D $(7+3)$, $D=7$ from E $(6+1)$, $E=6$ from C $(2+4)$, $C=2$ from A. Shortest path $A \to C \to E \to D \to F$, length 10.

Q7. From $A=0$: $B=4$, $C=3$. Permanent order and labels: $C=3$, then $B=4$ (the value $3+1=4$ ties the direct AB, so B stays 4), then $D=9$ (via B, $4+5$), then $E=10$ (via C, $3+7$), then $F=13$ (via E, $10+3$), then $G=17$ (via F, $13+4$). Back-track: $G=17$ from F ($13+4$), $F=13$ from E ($10+3$), $E=10$ from C ($3+7$), $C=3$ from A. Shortest path $A\rightarrow C\rightarrow E\rightarrow F\rightarrow G$, length 17.

Q8. Following only the arrows: from $S=0$, $A=3$, $C=6$. Make $A=3$ permanent; then $C=min(6,3+1)=4$ and $B=3+4=7$. Make $C=4$ permanent; then $B=min(7,4+2)=6$ and $D=4+3=7$. Make $B=6$ permanent; then $T=6+4=10$. Make $D=7$ permanent; then $T=min(10,7+4)=10$ (no change). Make $T=10$ permanent. Back-track: $T=10$ from B ($6+4$), $B=6$ from C ($4+2$), $C=4$ from A ($3+1$), $A=3$ from S. Shortest path $S\rightarrow A\rightarrow C\rightarrow B\rightarrow T$, length 10.

Q9. Reading the edges from the table ($AB=8$, $AC=5$, $BC=2$, $BD=6$, $CD=9$, $CE=4$, $DE=3$) and running Dijkstra's algorithm from $A=0$: $B=8$, $C=5$. Make $C=5$ permanent; then $B=min(8,5+2)=7$, $D=5+9=14$, $E=5+4=9$. Make $B=7$ permanent; then $D=min(14,7+6)=13$. Make $E=9$ permanent; then $D=min(13,9+3)=12$. Make $D=12$ permanent. Back-track: $D=12$ from E ($9+3$), $E=9$ from C ($5+4$), $C=5$ from A. Shortest distance 12 km by $A\rightarrow C\rightarrow E\rightarrow D$.

Q10. From $A=0$: $B=2$, $C=7$. Make $B=2$ permanent; then $C=min(7,2+3)=5$ and $D=2+8=10$. Make $C=5$ permanent; then $D=min(10,5+1)=6$ and $E=5+4=9$. Make $D=6$ permanent; then $E=min(9,6+5)=9$ (no change) and $F=6+2=8$. Make $F=8$ permanent, then $E=9$ permanent. The permanent labels are $B=2$, $C=5$, $D=6$, $E=9$, $F=8$. Back-track to F: $F=8$ from D ($6+2$), $D=6$ from C ($5+1$), $C=5$ from B ($2+3$), $B=2$ from A. Shortest path to F is $A\rightarrow B\rightarrow C\rightarrow D\rightarrow F$, length 8.

Q11. From $A=0$: $B=5$, $C=4$. Make $C=4$ permanent; then $D=4+6=10$ and $E=4+5=9$ (and $B=min(5,4+2)=5$ is unchanged). Make $B=5$ permanent; then $D=min(10,5+3)=8$. Make $D=8$ permanent; then $E=min(9,8+4)=9$ (no change) and $F=8+7=15$. Make $E=9$ permanent; then $F=min(15,9+2)=11$. Make $F=11$ permanent. Back-track: $F=11$ from E ($9+2$), $E=9$ from C ($4+5$), $C=4$ from A. The true shortest path is $A\rightarrow C\rightarrow E\rightarrow F$, length 11. Since $11<15$, Ravi is **incorrect**.

Q12. Following the arrows from $S=0$: $A=2$, $B=4$. Make $A=2$ permanent; then $B=min(4,2+1)=3$ and $C=2+5=7$. Make $B=3$ permanent; then $C=min(7,3+3)=6$ and $D=3+6=9$. Make $C=6$ permanent; then $D=min(9,6+2)=8$ and $T=6+9=15$. Make $D=8$ permanent; then $T=min(15,8+3)=11$. Make $T=11$ permanent. Back-track: $T=11$ from D ($8+3$), $D=8$ from C ($6+2$), $C=6$ from B ($3+3$), $B=3$ from A ($2+1$), $A=2$ from S. Shortest path $S\rightarrow A\rightarrow B\rightarrow C\rightarrow D\rightarrow T$, length 11.

Q13. Reading the edges ($PQ=7$, $PR=9$, $QR=3$, $QS=5$, $RS=8$, $RT=4$, $ST=6$, $TU=2$, $SU=9$) and running Dijkstra's algorithm from $P=0$: $Q=7$, $R=9$. Make $Q=7$ permanent; then $R=min(9,7+3)=9$ (no change) and $S=7+5=12$. Make $R=9$ permanent; then $T=9+4=13$ (and S is unchanged, since $9+8=17>12$). Make $S=12$ permanent; then $U=12+9=21$ (and T unchanged). Make $T=13$ permanent; then $U=min(21,13+2)=15$. Make $U=15$ permanent. Back-track: $U=15$ from T ($13+2$), $T=13$ from R ($9+4$), $R=9$ from P. Shortest distance 15 km by $P\rightarrow R\rightarrow T\rightarrow U$.

Q14. From $A=0$: $B=3$, $C=5$, $D=9$. Make $B=3$ permanent; then $C=min(5,3+3)=5$ (no change) and $E=3+7=10$. Make $C=5$ permanent; then $D=min(9,5+2)=7$ and $E=min(10,5+4)=9$. Make $D=7$ permanent (no improvement to E). Make $E=9$ permanent. The permanent labels are $B=3$, $C=5$, $D=7$, $E=9$, so the farthest vertex from A is E at distance 9. Back-track: $E=9$ from C ($5+4$), $C=5$ from A, giving the path $A\rightarrow C\rightarrow E$.

Q15. Following the arrows from $A=0$: $B=4$, $C=2$. Make $C=2$ permanent; then $B=min(4,2+1)=3$, $D=2+8=10$, $E=2+10=12$. Make $B=3$ permanent; then $D=min(10,3+5)=8$ (and $E=min(12,3+12)=12$ is unchanged). Make $D=8$ permanent; then $E=min(12,8+2)=10$. Make $E=10$ permanent. Back-track: $E=10$ from D ($8+2$), $D=8$ from B ($3+5$), $B=3$ from C ($2+1$), $C=2$ from A. Shortest path $A\rightarrow C\rightarrow B\rightarrow D\rightarrow E$, length 10.

Q16. From $A=0$: $B=2$, $C=5$. Make $B=2$ permanent; then $C=min(5,2+2)=4$ and $D=2+4=6$. Make $C=4$ permanent; then $E=4+3=7$. Make $D=6$ permanent; then $E=min(7,6+2)=7$ (no change) and $F=6+6=12$.

Make $E=7$ permanent; then $F=min(12,7+2)=9$ and $G=7+8=15$. Make $F=9$ permanent; then $G=min(15,9+3)=12$ and $H=9+9=18$. Make $G=12$ permanent; then $H=min(18,12+2)=14$. Make $H=14$ permanent. Back-track: $H=14$ from G $(12+2)$, $G=12$ from F $(9+3)$, $F=9$ from E $(7+2)$, $E=7$ from C $(4+3)$, $C=4$ from B $(2+2)$, $B=2$ from A. Shortest path $A \to B \to C \to E \to F \to G \to H$, length 14.

Q17. (a) The label 0 is the shortest distance from the start vertex to itself, which is trivially zero. (b) All edge weights are positive, so any *other* route to that vertex would first have to pass through a not-yet-permanent vertex whose own label is already at least as large; adding a further positive edge could only increase the total. Hence no future step can produce a smaller distance, and the label is safe to fix. (c) Starting at the destination, repeatedly find a predecessor u for which (permanent label of u) + (weight of the edge to the current vertex) equals the current vertex's permanent label; move to u and repeat until the start is reached. Reversing the list of vertices gives the shortest path.

Q18. From $A=0$: $B=6$, $C=2$. Make $C=2$ permanent; then $B=min(6,2+3)=5$, $D=2+5=7$, $E=2+7=9$. Make $B=5$ permanent; then $D=min(7,5+4)=7$ (no change). Make $D=7$ permanent; then $E=min(9,7+1)=8$ and $F=7+8=15$. Make $E=8$ permanent; then $F=min(15,8+2)=10$. Make $F=10$ permanent. Back-track to D: $D=7$ from C $(2+5)$, $C=2$ from A, giving $A \to C \to D$ (length 7). Back-track to F: $F=10$ from E $(8+2)$, $E=8$ from D $(7+1)$, $D=7$ from C $(2+5)$, $C=2$ from A, giving $A \to C \to D \to E \to F$ (length 10).

Q19. Following the arrows from $S=0$: $A=5$, $B=3$. Make $B=3$ permanent; then $A=min(5,3+1)=4$, $C=3+6=9$, $D=3+9=12$. Make $A=4$ permanent; then $C=min(9,4+4)=8$. Make $C=8$ permanent; then $D=min(12,8+2)=10$ and $T=8+7=15$. Make $D=10$ permanent; then $T=min(15,10+3)=13$. Make $T=13$ permanent. Back-track: $T=13$ from D $(10+3)$, $D=10$ from C $(8+2)$, $C=8$ from A $(4+4)$, $A=4$ from B $(3+1)$, $B=3$ from S. Quickest route $S \to B \to A \to C \to D \to T$, taking 13 minutes.

Q20. (a) From $A=0$: $B=4$, $C=3$. Make $C=3$ permanent; then $B=min(4,3+1)=4$ (unchanged), $D=3+5=8$, $E=3+8=11$. Make $B=4$ permanent (no improvement to D). Make $D=8$ permanent; then $E=min(11,8+2)=10$ and $F=8+4=12$. Make $E=10$ permanent (no improvement to F). Make $F=12$ permanent. Back-track: $F=12$ from D $(8+4)$, $D=8$ from C $(3+5)$, $C=3$ from A. Shortest path $A \to C \to D \to F$, length 12. (b) With DF removed, the only remaining way into F is the edge EF. The labels up to E are unchanged ($C=3$, $D=8$, $E=10$), and now $F=E+3=10+3=13$. Back-track: $F=13$ from E $(10+3)$, $E=10$ from D $(8+2)$, $D=8$ from C $(3+5)$, $C=3$ from A. The new shortest path is $A \to C \to D \to E \to F$, length 13.

Theory

A recurring practical problem is to connect a set of locations — towns, computers, reservoirs — as cheaply as possible. The mathematics needed to solve it uses a special kind of graph called a **tree**.

Trees.

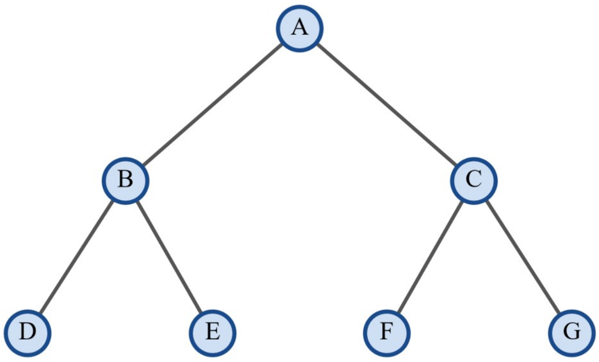
A **tree** is a connected graph that contains no cycles (no circuits). Because a tree is connected there is a path between every pair of vertices, and because it has no cycles that path is **unique**: there is exactly one way to travel between any two vertices.

The single most useful fact about trees is a count of their edges:

- a tree with n vertices has exactly $n-1$ edges;
- conversely, a **connected** graph with n vertices and $n-1$ edges must be a tree.

Two further properties follow. **Removing** any edge of a tree disconnects it (the tree falls into two pieces), and **adding** any new edge between two of its vertices creates exactly one cycle. A vertex of degree 1 is called a **leaf**; every tree with at least two vertices has at least two leaves.

The graph below is a tree: it is connected and has no cycles. It has 7 vertices and $7-1=6$ edges, and its leaves are D, E, F and G.



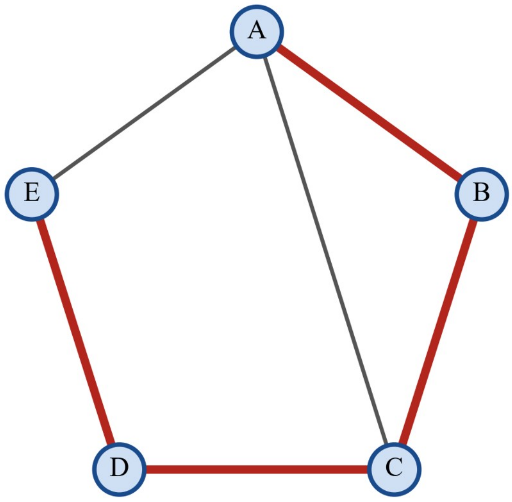
Spanning trees.

A **spanning tree** of a connected graph is a subgraph that

- includes **every** vertex of the graph, and
- is a tree (connected, no cycles).

A spanning tree therefore holds all n vertices together using exactly $n-1$ of the graph's edges. A connected graph usually has **many** different spanning trees. To produce one, delete edges to break every cycle while keeping the graph connected. If the graph has n vertices and m edges, a spanning tree keeps $n-1$ edges, so $m-(n-1)$ edges must be removed.

In the network below the highlighted edges form one spanning tree: they reach all five vertices, use $5-1=4$ edges and contain no cycle.



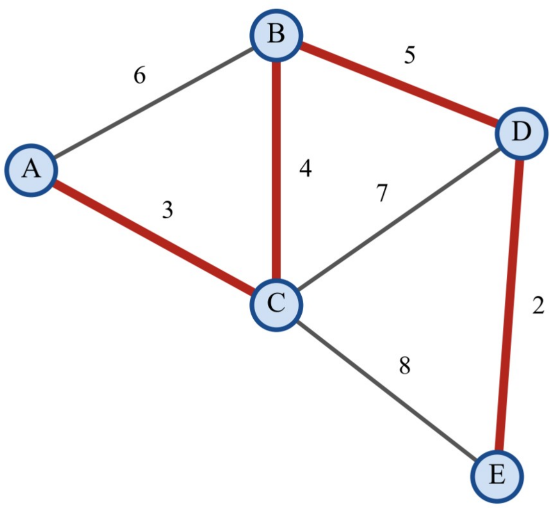
Weighted graphs and the minimum spanning tree.

In a **weighted graph** (a **network**) each edge carries a number, its **weight** — a length, cost or time. The **weight of a spanning tree** is the sum of the weights of its edges. The **minimum spanning tree** (MST), also called the **minimum connector**, is a spanning tree of **smallest possible total weight**. Its weight is the least total edge weight needed to connect every vertex of the network.

Two cautions:

- an MST need not be unique — if some edges have equal weight, a network may have more than one MST — but the **minimum total weight is always unique**;
- the MST is **not** the same as a collection of shortest paths. It minimises the total weight used to connect **all** vertices, not the distance between any particular pair.

The MST of the network below (highlighted) has weight $3+4+5+2=14$.



Prim's algorithm.

For a small network the MST can often be found by **inspection**. **Prim's algorithm** is a systematic method that grows the tree outward from a starting vertex:

1. Choose any vertex to start; it forms the initial tree.

2. Find the **smallest-weight edge** that joins a vertex **already in** the tree to a vertex **not yet in** the tree. Add that edge and the new vertex to the tree. (Never add an edge whose two ends are both already in the tree — it would create a cycle.)
3. Repeat step 2 until every vertex has been included.

When all n vertices are in the tree it has $n-1$ edges and is a minimum spanning tree. The **starting vertex does not change the final total weight**, so any vertex may be used.

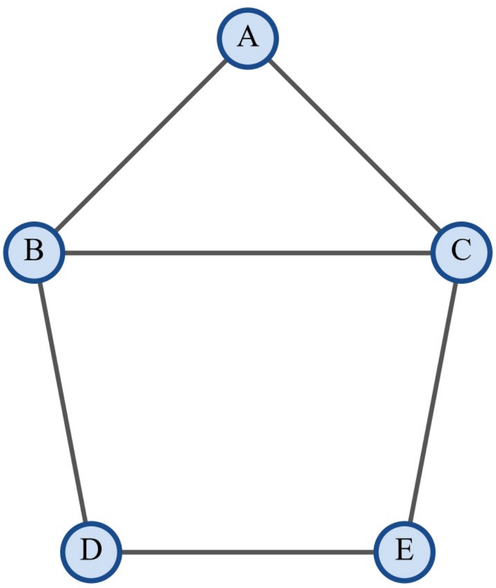
Minimum connector problems.

Many practical problems ask for the cheapest way to join a set of locations: laying fibre-optic cable between exchanges, building roads or pipelines, or wiring components. Model each location as a vertex and each possible link as a weighted edge (its cost or length). The cheapest connection is the **minimum spanning tree**, and its weight is the least total cost. Because a spanning tree uses only $n-1$ links, the extra links in the network are never needed once an MST has been chosen.

Worked examples

Example 1 — scaffolded

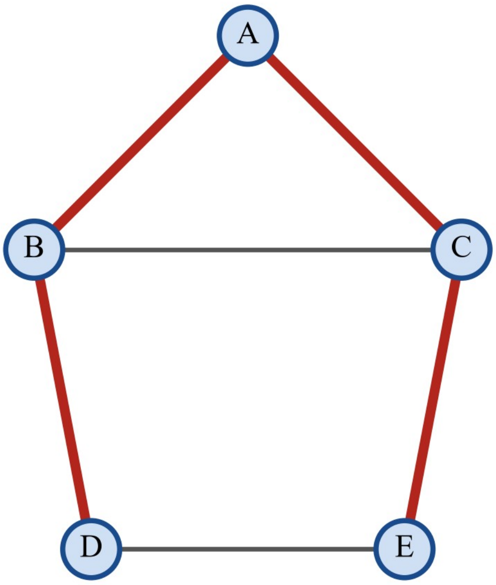
The network below shows possible tracks between five camp sites A–E.



Answer the following. (a) Explain why the network is not a tree. (b) How many edges must a spanning tree of this network have? (c) How many edges must be removed to leave a spanning tree? (d) Find a spanning tree of the network.

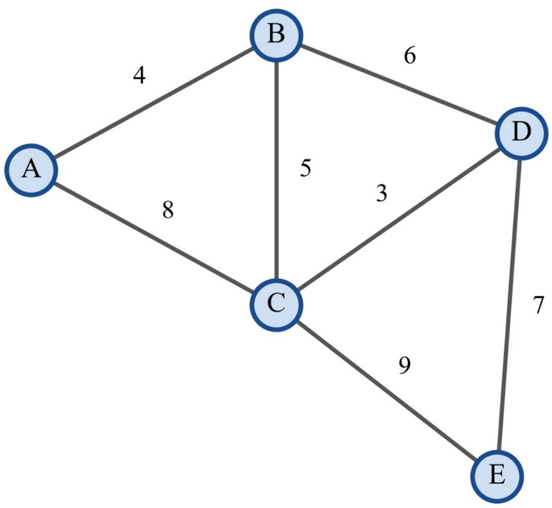
Working	Comment
The network contains the cycle A–B–C–A, so it is not a tree.	A tree has **no** cycles.
A spanning tree has $n-1$ edges. Here $n=5$, so it has $5-1=4$ edges.	The spanning tree must reach all 5 vertices.
The network has $m=6$ edges, so $6-4=2$ edges must be removed.	Remove one edge from each independent cycle.
Remove BC and DE. The edges AB, AC, BD, CE remain.	They connect all 5 vertices with no cycle.

The four highlighted edges below form one spanning tree.



Example 2 — scaffolded

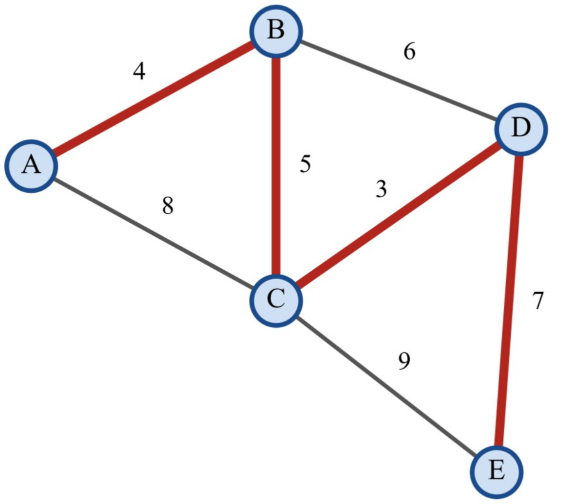
Use Prim's algorithm, starting at vertex A, to find a minimum spanning tree of the network below, and state its weight.



Working	Comment
Start at A. Tree $=\{A\}$.	Prim's may begin at any vertex.
Edges to a new vertex: $AB=4$, $AC=8$. Add AB. Tree $=\{A,B\}$.	Choose the shortest edge leaving the tree.
Edges to a new vertex: $AC=8$, $BC=5$, $BD=6$. Add BC. Tree $=\{A,B,C\}$.	5 is smallest and reaches new vertex C.
Edges to a new vertex: $BD=6$, $CD=3$, $CE=9$. Add CD. Tree $=\{A,B,C,D\}$.	$CD=3$ is smallest; D is new.
Edges to a new vertex: $CE=9$, $DE=7$. Add DE. Tree $=\{A,B,C,D,E\}$.	All 5 vertices are now connected — stop.

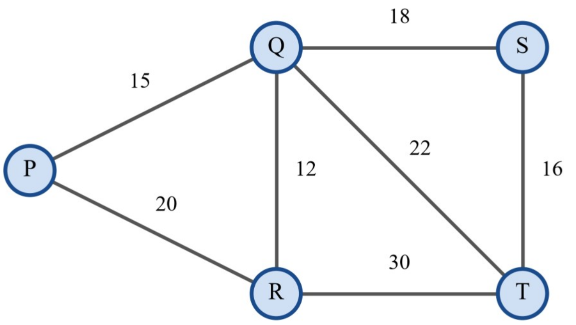
Working	Comment
MST edges AB, BC, CD, DE; weight $4+5+3+7=19$.	$n-1=4$ edges, as required.

The minimum spanning tree is highlighted below; its weight is 19.



Example 3 — unscaffolded

The network shows the possible cable routes between five towns P–T, with distances in kilometres. Find the least total length of cable needed to connect all five towns.

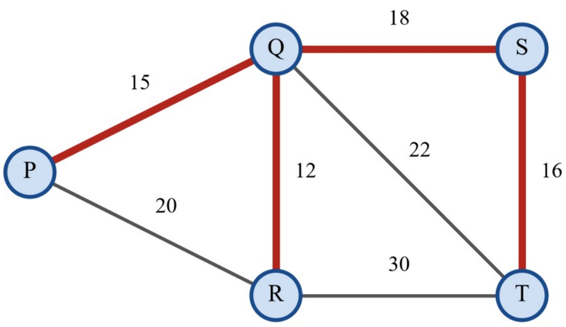


Connecting all five towns at the least total length is a minimum connector problem, so a minimum spanning tree is required. Apply Prim's algorithm starting at P.

Begin with $\{P\}$ and add the shortest edge to a new town at each stage. From P the shortest edge is $PQ=15$, giving $\{P,Q\}$. From this tree the shortest edge to a new town is $QR=12$, giving $\{P,Q,R\}$. The shortest edge to a remaining town is now $QS=18$ (shorter than $QT=22$ and $RT=30$), giving $\{P,Q,R,S\}$. Finally the shortest edge to the last town T is $ST=16$ (shorter than $QT=22$ and $RT=30$).

The minimum spanning tree uses PQ, QR, QS and ST, of total length

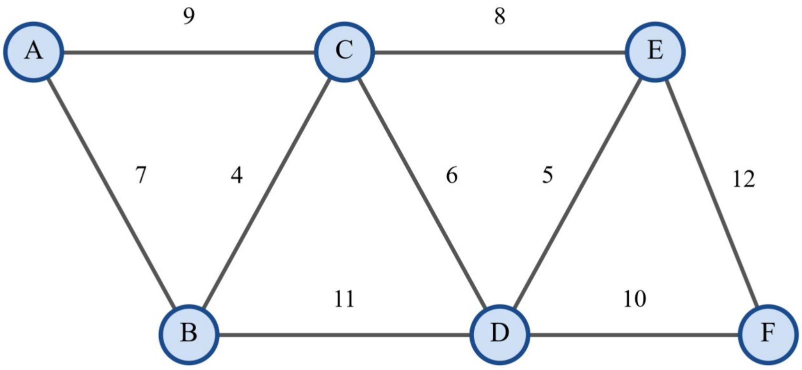
$$15+12+18+16=61.$$



$\therefore$ the minimum length of cable needed to connect all five towns is 61 km.

Example 4 — unscaffolded

Find the minimum spanning tree of the network below and state its weight.

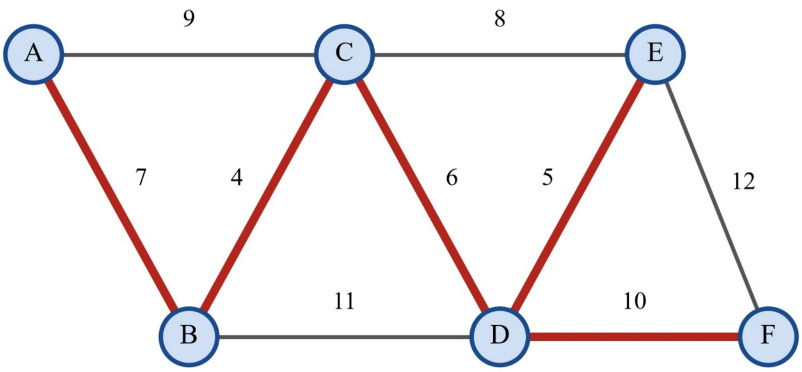


Apply Prim's algorithm from A. Start with $\{A\}$; the shortest edge is $AB=7$, giving $\{A,B\}$. The shortest edge to a new vertex is $BC=4$ (shorter than $AC=9$), giving $\{A,B,C\}$. Next $CD=6$ (shorter than $BD=11$ and $CE=8$) adds D, giving $\{A,B,C,D\}$. Then $DE=5$ (shorter than $CE=8$ and $DF=10$) adds E. Finally $DF=10$ (shorter than $EF=12$) adds the last vertex F.

The minimum spanning tree uses AB, BC, CD, DE and DF, of weight

$$7+4+6+5+10=32.$$

The same total is obtained no matter which vertex Prim's algorithm starts from.



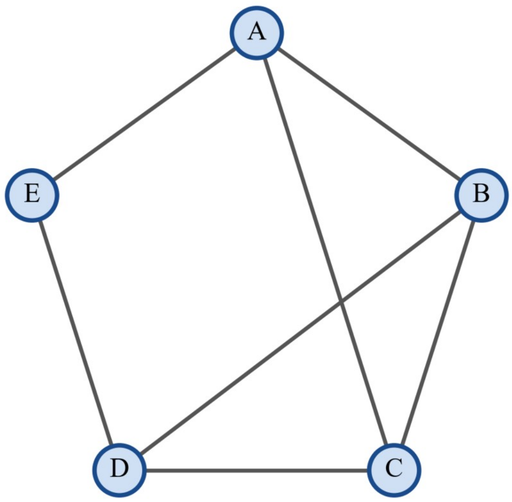
$\therefore$ the minimum spanning tree has weight 32.

Practice questions

Scaffolded

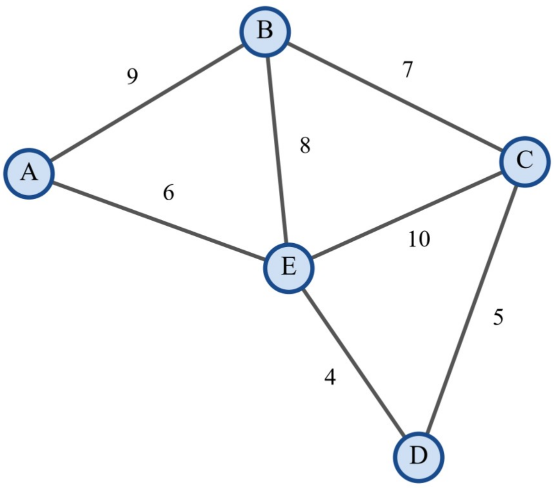
Q1. (a) How many edges does a tree with 12 vertices have? (b) A tree has 20 edges. How many vertices does it have? (c) A connected graph has 10 vertices and 9 edges. Explain why it must be a tree. (d) Can a graph with 7 vertices and 5 edges be a tree? Explain.

Q2. The network below shows possible walking paths between five landmarks.



(a) How many vertices and how many edges does the network have?
(b) How many edges must a spanning tree of this network have?
(c) How many edges must be removed to obtain a spanning tree?
(d) By removing suitable edges, find a spanning tree of the network and list its edges.

Q3. Use Prim's algorithm, starting at A, to find a minimum spanning tree of the network below.



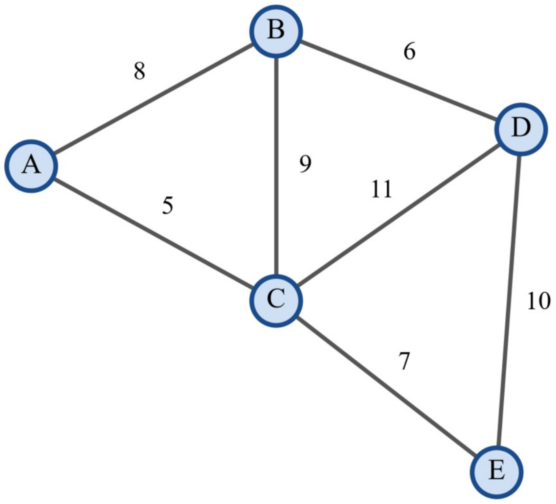
(a) Copy and complete the table.

Step	Edge added	Weight	Vertices in tree
1	(start)	—	A
2	AE	6	A, E
3			

Step	Edge added	Weight	Vertices in tree
4			
5			

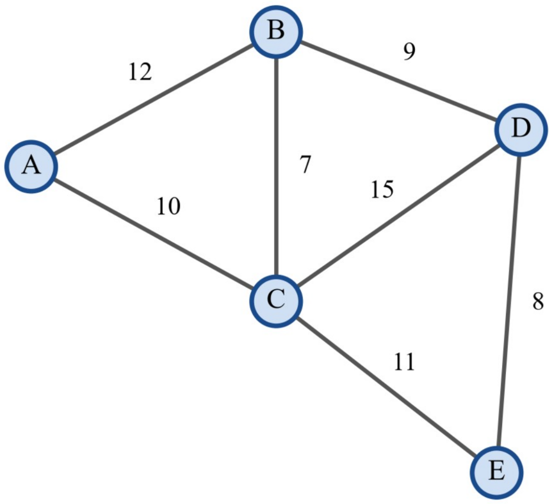
(b) List the edges of the minimum spanning tree and state its total weight.

Q4. The network shows the cost (in $'000) of building each possible road between five towns.



(a) How many roads must be built to connect all five towns?
(b) Use Prim's algorithm from A (or inspection) to find the minimum spanning tree.
(c) State the least total cost of connecting all five towns.

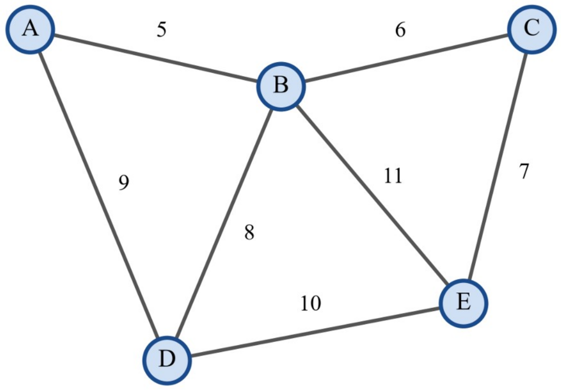
Q5. Find the minimum spanning tree of the network shown and state its weight.



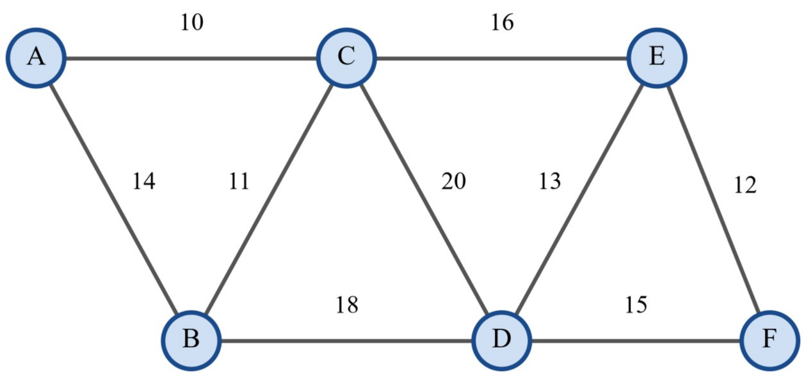
Q6. (a) A tree has 6 vertices. How many edges does it have? (b) One extra edge is added to that tree, joining two of its vertices. How many cycles does the new graph contain? (c) A connected network has 8 vertices and 13 edges. How many edges must be removed to obtain a spanning tree?

Unscaffolded

Q7. Find the minimum spanning tree of the network below and its total weight.

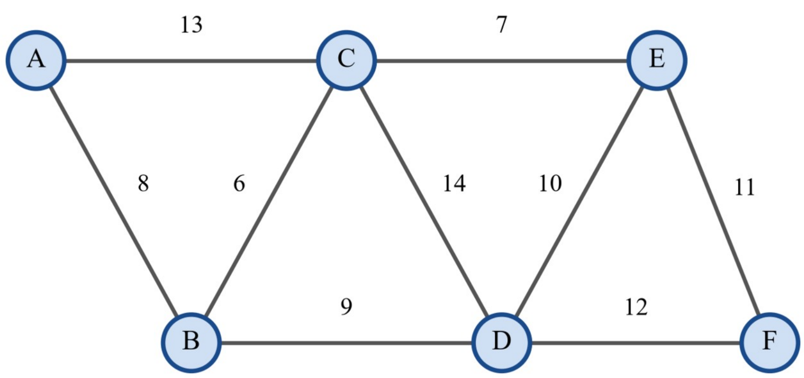


Q8. A telecommunications company will lay optical fibre to connect six exchanges A–F. The network shows the cost (in $'000) of each possible link. Find the least cost of connecting all six exchanges, and list the links used.

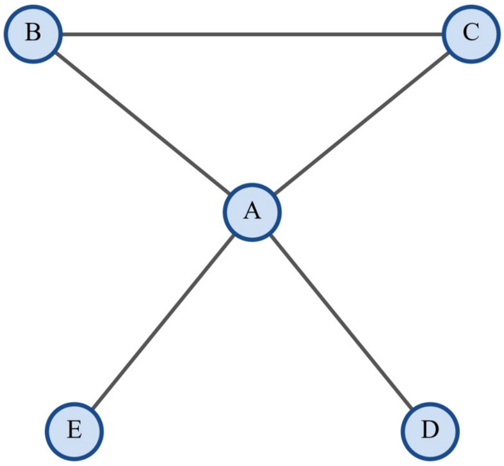


Q9. (a) Explain why every tree with more than one vertex has at least one vertex of degree 1. (b) A tree has 10 vertices. State the sum of the degrees of all its vertices.

Q10. Find a minimum spanning tree of the network below and state its weight.

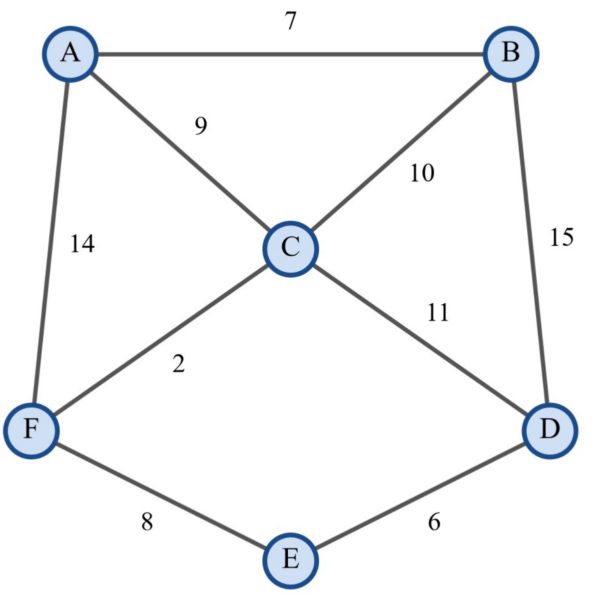


Q11. The network shows five sites, with the edge BC and the four edges from A.

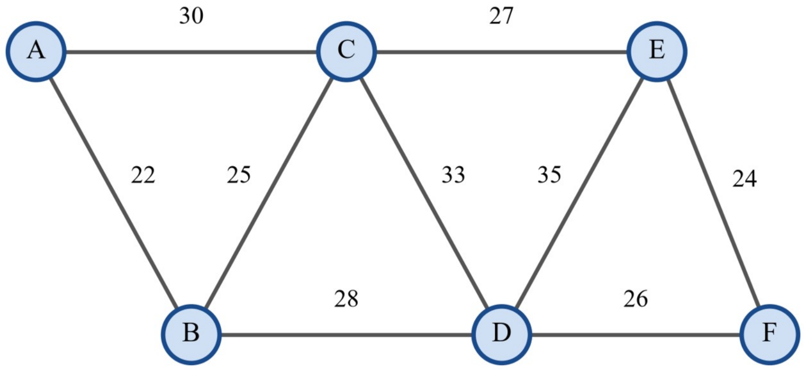


(a) How many edges does a spanning tree of this network have?
(b) The network has exactly one cycle. State the vertices of that cycle.
(c) List the edges of a spanning tree of the network.

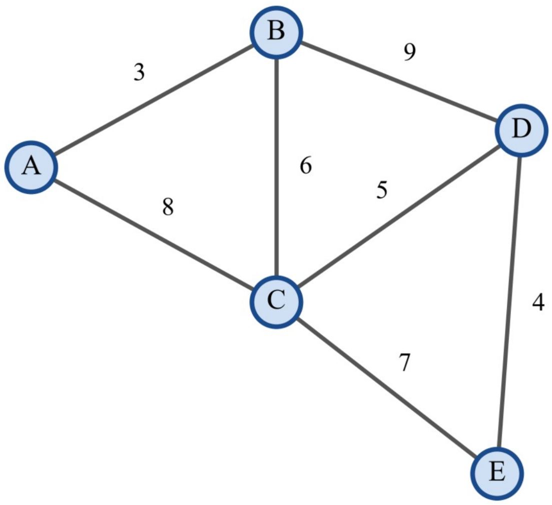
Q12. Find the minimum spanning tree of the network below and its weight.



Q13. A water authority will lay pipes to connect six reservoirs. The network shows the length (km) of each possible pipe. Find the minimum total length of pipe needed to connect all six reservoirs.

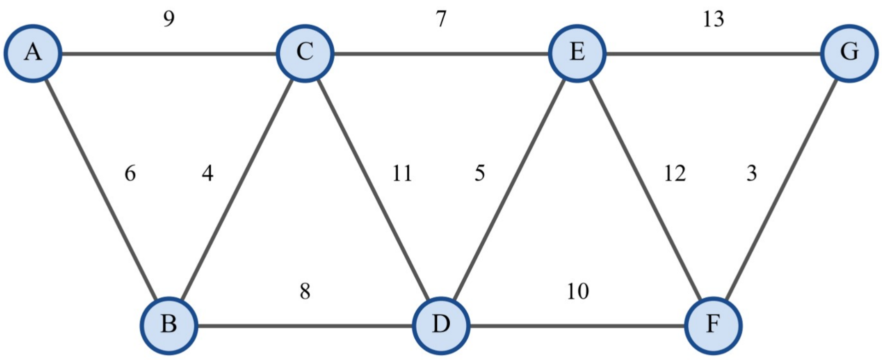


Q14. The network shows the cost (in $'000) of cabling between five buildings.



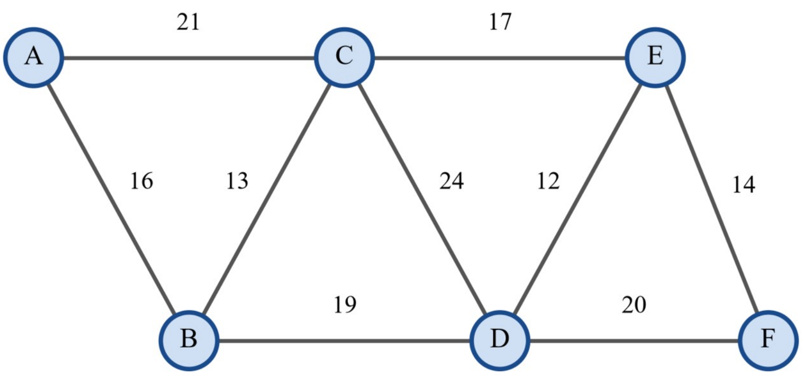
(a) Find the minimum spanning tree and its weight.
(b) Explain what the weight of the minimum spanning tree represents.
(c) What is the total cost of all the cables shown, and why is it not necessary to build them all?

Q15. Find a minimum spanning tree of the network below and state its total weight.



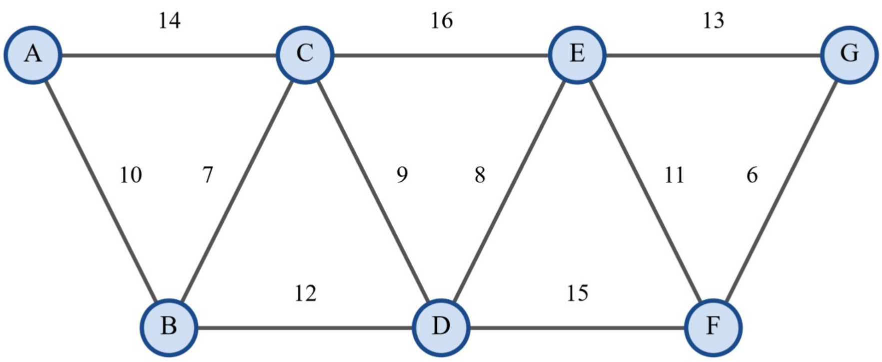
Q16. (a) Is the minimum spanning tree of a network always unique? Explain. (b) A student claims that the minimum spanning tree gives the shortest path between every pair of vertices. Explain why this is false.

Q17. Six towns are to be connected by sealed roads. The network shows the cost (in $'000) of sealing each possible road. Determine the minimum cost of connecting all towns and state which roads should be sealed.

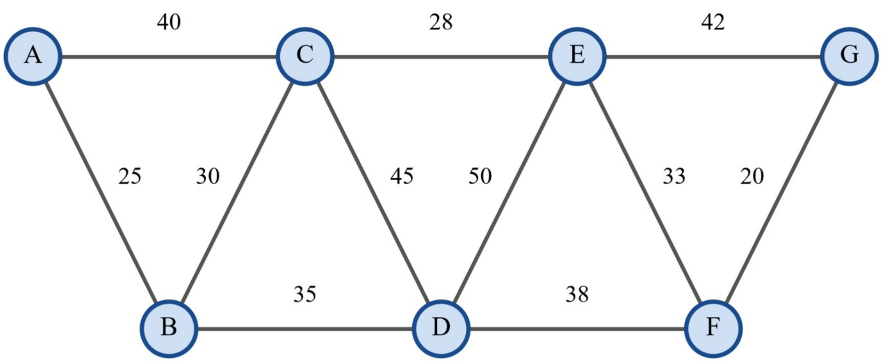


Q18. A connected graph has 9 vertices and 12 edges. (a) How many edges must be removed to produce a spanning tree? (b) After removing those edges, how many edges does the spanning tree have?

Q19. Find the minimum spanning tree of the network below and its weight.



Q20. An electricity company must connect seven substations. The network shows the cost (in $’000) of each possible cable.



(a) Find the minimum spanning tree and the least total cost of connecting all seven substations.

(b) How many cables are used, and how does this compare with the number of possible cables shown?

Answers

Q1. (a) 11; (b) 21; (c) it is connected with $n-1=9$ edges, so it is a tree; (d) no — a tree with 7 vertices needs 6 edges, so 5 edges cannot connect all 7. **Q2.** (a) 5 vertices, 7 edges; (b) 4; (c) 3; (d) e.g. AB, BC, CD, DE (remove EA, AC, BD). **Q3.** (a) Step 3 $ED=4$; Step 4 $DC=5$; Step 5 $CB=7$. (b) AE, ED, DC, CB; weight 22. **Q4.** (a) 4; (b) AC, CE, AB, BD; (c) $26,000. **Q5.** AC, CB, BD, DE; weight 34. **Q6.** (a) 5; (b) exactly 1; (c) 6. **Q7.** AB, BC, CE, BD; weight 26. **Q8.** AC, BC, CE, EF, DE; least cost $62,000. **Q9.** (a) see solution; (b) 18. **Q10.** AB, BC, CE, BD, EF; weight 41. **Q11.** (a) 4; (b) A, B, C; (c) e.g. AB, AC, AD, AE (remove BC). **Q12.** CF, DE, AB, EF, AC; weight 32. **Q13.** AB, BC, CE, EF, DF; length 124 km. **Q14.** (a) AB, DE, CD, BC; weight 18; (b) the least total cost to cable all five buildings together; (c) all cables total 42, i.e. $42,000, but only 4 cables are needed to connect five buildings, so the rest are unnecessary. **Q15.** AB, BC, CE, DE, DF, FG; weight 35. **Q16.** (a) no — equal edge weights can give more than one MST, though the minimum total weight is unique; (b) the MST minimises the total connecting weight, not the distance between a chosen pair, so a path through the MST may be longer than a direct edge. **Q17.** AB, BC, CE, DE, EF; least cost $72,000. **Q18.** (a) 4; (b) 8. **Q19.** AB, BC, CD, DE, EF, FG; weight 51. **Q20.** (a) AB, BC, BD, CE, EF, FG; least cost $171,000; (b) 6 cables (that is $n-1$) out of the 11 possible cables shown.

Fully-worked solutions

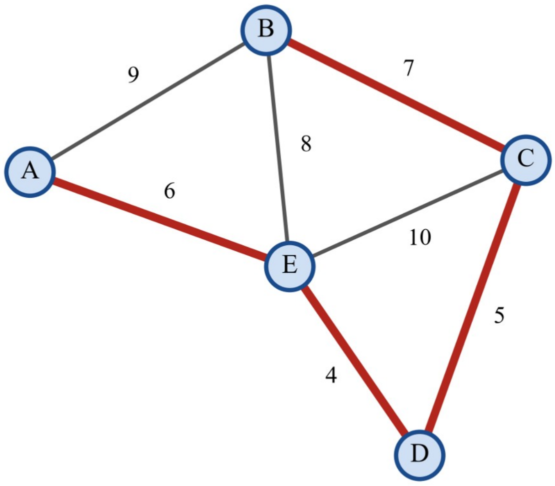
Q1. (a) A tree with n vertices has $n-1$ edges, so $12-1=11$. (b) If $n-1=20$ then $n=21$ vertices. (c) A connected graph with n vertices and exactly $n-1$ edges is a tree; here $n=10$ and $n-1=9$, so the graph is a tree. (d) No. A tree on 7 vertices must have $7-1=6$ edges; with only 5 edges the graph cannot be connected, so it cannot be a tree.

Q2. (a) Counting from the diagram, there are 5 vertices and 7 edges. (b) A spanning tree has $n-1=5-1=4$ edges. (c) It must therefore lose $7-4=3$ edges. (d) The network contains the outer cycle A–B–C–D–E–A and the chords AC and BD. Removing AC, BD and EA leaves AB, BC, CD, DE — a path through all five vertices, which is a spanning tree. (Other answers are possible.)

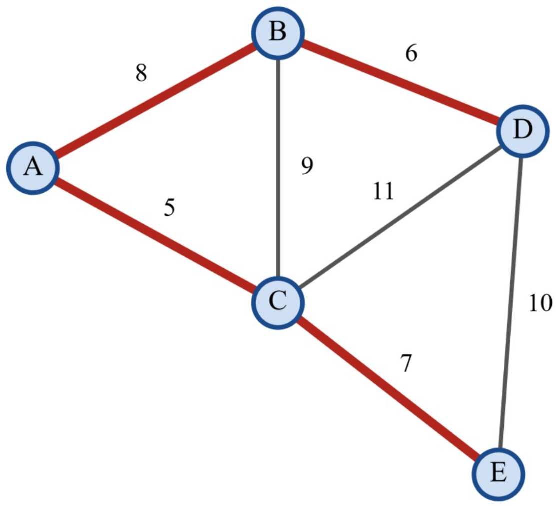
Q3. Prim's algorithm from A. Start with $\{A\}$. The shortest edge from the tree is $AE=6$, adding E. From $\{A,E\}$ the shortest edge to a new vertex is $ED=4$ (shorter than $EB=8$, $EC=10$, $AB=9$), adding D. From $\{A,E,D\}$ the shortest is $DC=5$, adding C. From $\{A,E,D,C\}$ the shortest edge to the last vertex is $CB=7$ (shorter than $EB=8$ and $AB=9$). The completed table is:

Step	Edge added	Weight	Vertices in tree
1	(start)	—	A
2	AE	6	A,E
3	ED	4	A,E,D
4	DC	5	A,E,D,C
5	CB	7	A,E,D,C,B

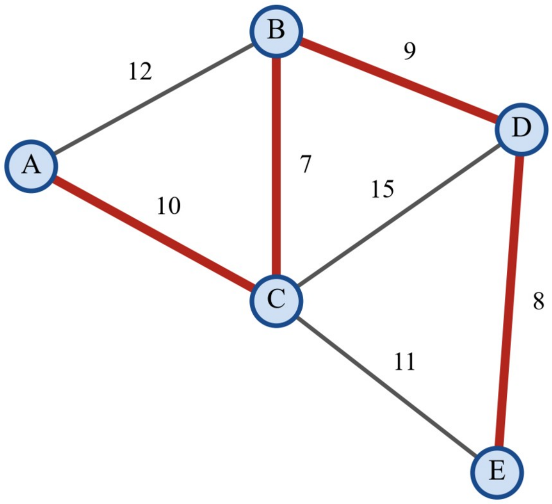
The minimum spanning tree is AE, ED, DC, CB, of weight $6+4+5+7=22$.



Q4. (a) Five towns need $5-1=4$ roads to be connected. (b) Prim's from A: start $\{A\}$; add $AC=5$ to get $\{A,C\}$; add $CE=7$ (shortest to a new town) to get $\{A,C,E\}$; add $AB=8$ to get $\{A,C,E,B\}$; add $BD=6$ to get all five. The MST is AC, CE, AB, BD. (c) Total cost $=5+7+8+6=26$, that is \$26,000.

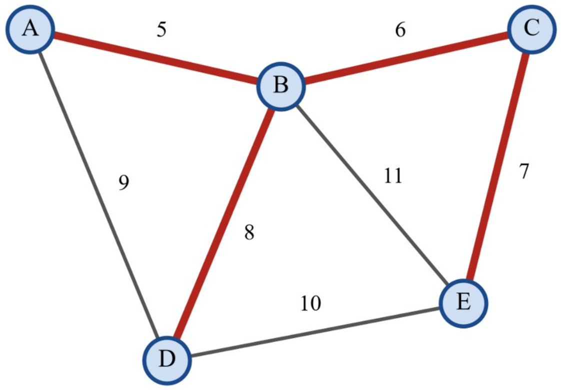


Q5. Prim's from A: start $\{A\}$; add $AC=10$; add $CB=7$ (shortest to a new vertex); add $BD=9$; add $DE=8$. The minimum spanning tree is AC, CB, BD, DE, of weight $10+7+9+8=34$.



Q6. (a) A tree with 6 vertices has $6-1=5$ edges. (b) Adding one edge to a tree creates exactly one cycle. (c) A spanning tree of an 8-vertex network has $8-1=7$ edges, so $13-7=6$ edges must be removed.

Q7. Prim's from A: start $\{A\}$; add $AB=5$; add $BC=6$ (shortest to a new vertex); add $CE=7$; add $BD=8$. The minimum spanning tree is AB, BC, CE, BD, of weight $5+6+7+8=26$.



Q8. Connecting six exchanges is a minimum connector problem. Prim's from A: start $\{A\}$; add $AC=10$; add $BC=11$; add $CE=16$; add $EF=12$; add $DE=13$. The MST uses AC, BC, CE, EF, DE, of total cost $10+11+16+12+13=62$, that is \$62,000.

Q9. (a) Suppose, for a contradiction, every vertex had degree 2 or more. Then starting at any vertex you could always leave along a new edge, and because the graph is finite you would eventually return to a vertex already visited, forming a cycle. A tree has no cycles, so at least one vertex must have degree 1. (b) The tree has $10-1=9$ edges. The sum of all vertex degrees equals twice the number of edges, $2\times 9=18$.

Q10. Prim's from A: start $\{A\}$; add $AB=8$; add $BC=6$; add $CE=7$; add $BD=9$; add $EF=11$. The minimum spanning tree is AB, BC, CE, BD, EF, of weight $8+6+7+9+11=41$.

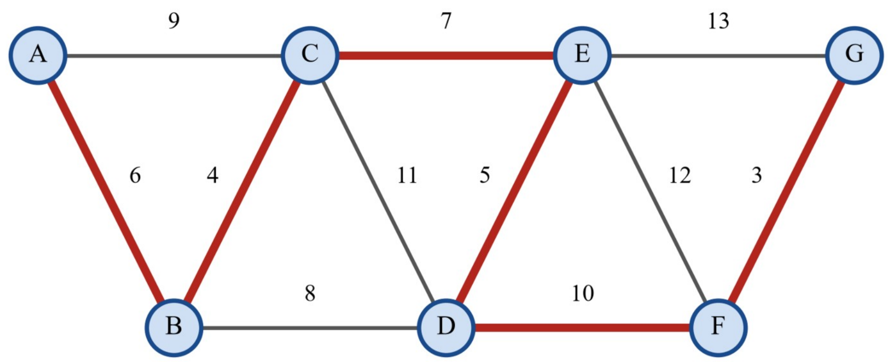
Q11. (a) A spanning tree has $5-1=4$ edges. (b) The only cycle is A–B–C–A, using vertices A, B and C. (c) Removing one edge of that cycle breaks it while keeping the graph connected; removing BC leaves AB, AC, AD, AE, a spanning tree. (Removing AB or AC instead also works.)

Q12. Prim's from A: start $\{A\}$; add $AB=7$; add $AC=9$; add $CF=2$; add $EF=8$; add $DE=6$. The minimum spanning tree is AB, AC, CF, EF, DE, of weight $7+9+2+8+6=32$.

Q13. Connecting six reservoirs is a minimum connector problem. Prim's from A: start $\{A\}$; add $AB=22$; add $BC=25$; add $CE=27$; add $EF=24$; add $DF=26$. The MST uses AB, BC, CE, EF, DF, of total length $22+25+27+24+26=124$ km.

Q14. (a) Prim's from A: start $\{A\}$; add $AB=3$; add $BC=6$; add $CD=5$; add $DE=4$. The MST is AB, BC, CD, DE, of weight $3+6+5+4=18$. (b) The weight 18 (that is $18,000) is the least total cost of cabling that connects all five buildings together. (c) The seven cables shown total $3+8+6+9+5+7+4=42$, that is $42,000. Only $n-1=4$ cables are needed to connect five buildings, and any extra cable would complete a cycle, so building them all is unnecessary.

Q15. Prim's from A: start $\{A\}$; add $AB=6$; add $BC=4$; add $CE=7$; add $DE=5$; add $DF=10$; add $FG=3$. The minimum spanning tree is AB, BC, CE, DE, DF, FG, of weight $6+4+7+5+10+3=35$.



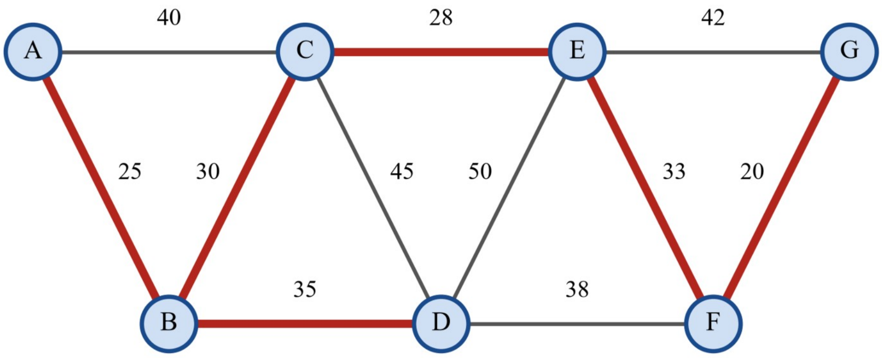
Q16. (a) No. If two or more edges share the same weight, a network can have several minimum spanning trees; however every one of them has the same (minimum) total weight, so the minimum weight itself is unique. (b) The minimum spanning tree is chosen to make the **total** connecting weight as small as possible, not to make the journey between any two particular vertices short. The unique path between two vertices within the MST may be longer than a direct edge joining them, so the MST does not give shortest paths.

Q17. Connecting six towns is a minimum connector problem. Prim's from A: start $\{A\}$; add $AB=16$; add $BC=13$; add $CE=17$; add $DE=12$; add $EF=14$. The MST uses AB, BC, CE, DE, EF, of total cost $16+13+17+12+14=72$, that is $72,000.

Q18. A spanning tree of a 9-vertex network has $9-1=8$ edges. (a) The graph must lose $12-8=4$ edges. (b) The spanning tree then has 8 edges.

Q19. Prim's from A: start $\{A\}$; add $AB=10$; add $BC=7$; add $CD=9$; add $DE=8$; add $EF=11$; add $FG=6$. The minimum spanning tree is AB, BC, CD, DE, EF, FG, of weight $10+7+9+8+11+6=51$.

Q20. (a) Prim's from A: start $\{A\}$; add $AB=25$; add $BC=30$; add $CE=28$; add $EF=33$; add $FG=20$; add $BD=35$. The minimum spanning tree is AB, BC, CE, EF, FG, BD, of total cost $25+30+28+33+20+35=171$, that is $171,000. (b) Seven substations are connected by $7-1=6$ cables, whereas the network shows 11 possible cables, so only 6 of the 11 are built.


40
28
42
A
C
E
G
25
30
45
50
33
20
35
38
B
D
F