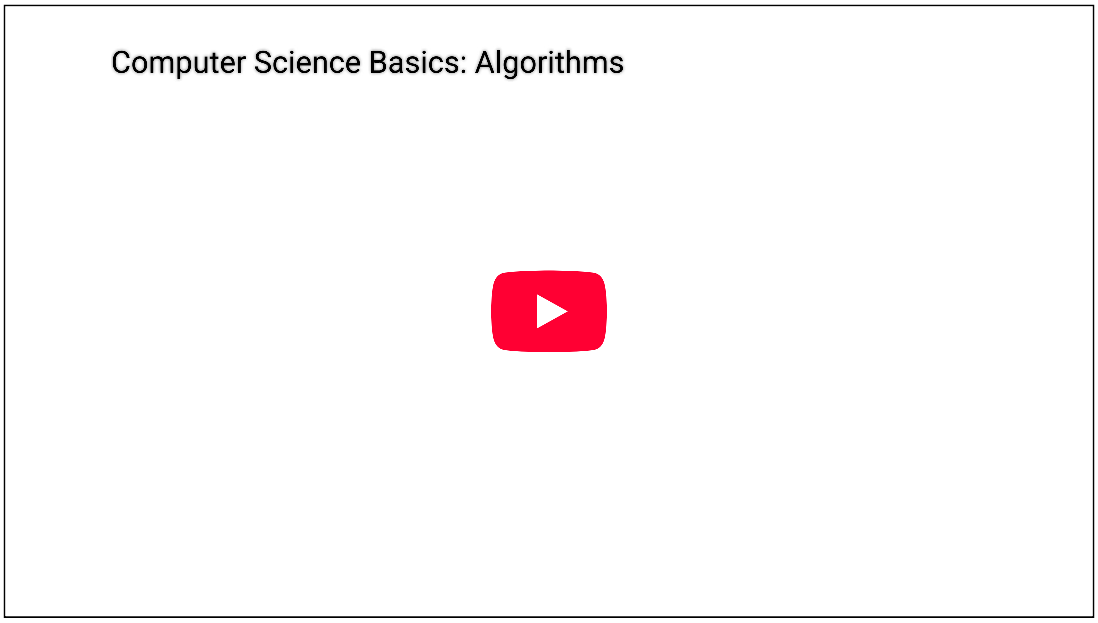
Index

The big picture

From automated coffee machines to self-driving cars, computers are being developed to perform more and more tasks for us. For this reason, it is important to understand how a computer 'thinks'.



As you saw in the video above, algorithms are step-by-step instructions that are used to perform a specific task. Algorithms are the backbone of computer processing and, as a result, are integral to our computer-driven world.

One specific area where computer algorithms are particularly important is the logistics of delivery and supply systems. For companies like Amazon and Alibaba, it is crucial that the products sold on their websites are delivered in a timely manner. This requires a highly sophisticated delivery system, which is discussed in the following video.

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

What is Vehicle Routing Problem (VRP)?

Concept

Complex real-world problems can be modelled and solved using graph theory algorithms.

Theory of Knowledge

Application of knowledge is a key feature within all areas of knowledge, including mathematics. When discussing and contemplating application, the question of ethics arises. For example, mathematical knowledge is necessary to create weapons of mass destruction; should such mathematics be pursued and shared?

Knowledge Question: Is all knowledge value neutral?

3. Geometry and trigonometry / 3.16 Graph algorithms

Moving around a graph

Student
view

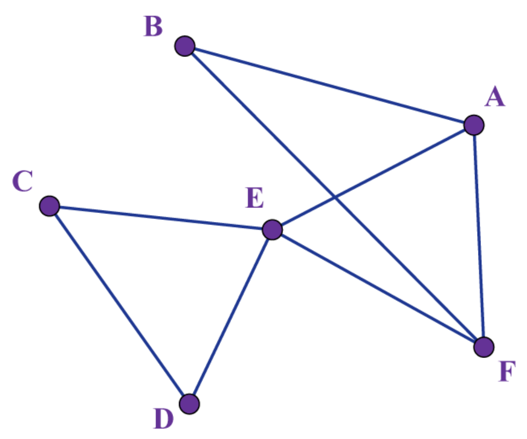
Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

Paths and trails

Making connections

Recall from section 3.15.2 (/study/app/math-ai-hl/sid-132-cid-761618/book/walks-id-28235/) that a walk is a sequence of connected edges in a graph.

Consider the graph below. Can you find a walk that begins at vertex A, includes at least five vertices in total and ends at vertex D?



More information

The image shows a graph with six vertices labeled A, B, C, D, E, and F. The vertices are connected by lines representing edges. Vertex A is connected to vertices B, F, and E. Vertex B is connected to vertices A, E, and C. Vertex F is connected to vertices A, E, and D. Vertex E is connected to vertices A, B, F, and C. Vertex C is connected to vertices B, E, and D. Vertex D is connected to vertices C and F. This graph demonstrates possible walks starting from vertex A and ending at vertex D, such as AE, EF, FB, BA, AE, and so on, incorporating at least five vertices in a walk.

[Generated by AI]

Student view

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

There are many different walks that satisfy the conditions given above, including $\mathrm{AEFBAED}$, AFBAED, and ABFED. When considering different possible walks on a graph, sometimes there is also the limitation that a vertex cannot be included multiple times. Such walks are called paths.

Section Student... (0/0) Feedback Print (/study/app/math-ai-hl/sid-132-cid-761618/book/the-big-picture-id-28238/print/) Assign

Important

A path is a walk on a graph that has no repeated vertices.

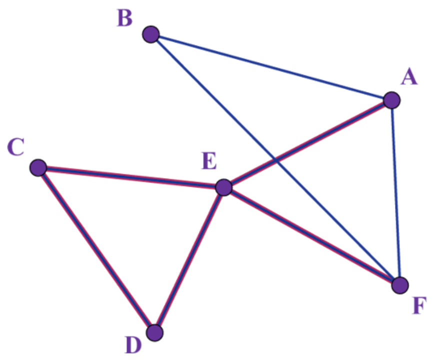
Example 1

★☆☆

For the graph shown above, determine whether there is a path that travels from A to C and finally ends at F.

Student view

Overview (/study/app/math-ai-hl/sid-132-cid-761618/ov

Steps	Explanation
One possible option of a walk from A to C that ends at F is AECDEF, which is shown below:  However, as this walk includes vertex E twice, it cannot be considered a path. In fact, all the walks that go from A to C and end at F must go through vertex E more than once. Therefore, there are no paths that meet the requirements of the problem.	Remember that path cannot include any repeated vertices.

Consider the walk shown in the answer to **Example 1.** The walk was not a path since it contain vertex Etwice. However, it is an example of a trail, since it does not contain any repeated edge

Important

A trail is a walk on a graph that has no repeated edges.

Exam tip

The notation for a walk, path, or trail can simply be the **ordered** list of the included vertices.

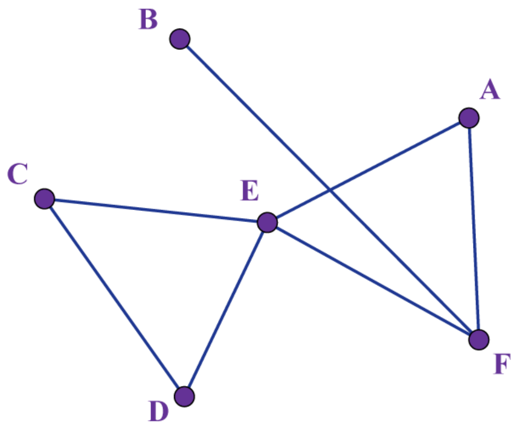
Student view

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

Example 2

★★☆

For the graph shown below, determine whether there is a trail that travels from A to B and ends at D.



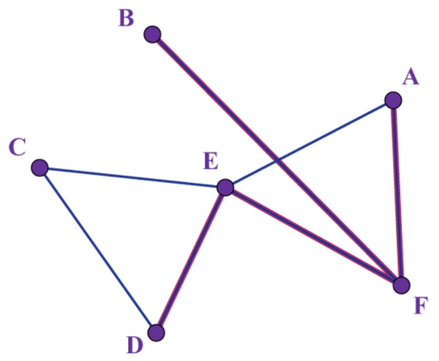
More information

The image is a graph consisting of six nodes labeled A, B, C, D, E, and F. The nodes are connected by edges enabling paths between them. Node A is connected to nodes E and F. Node B connects to node E. Node C is connected to nodes D and E. Node D connects exclusively with node C. Node E connects to nodes A, B, C, and F. Finally, node F connects with nodes A and E. The task is to determine whether there is a trail that starts at node A, goes to node B, and ends at node D.

[Generated by AI]

Student view

Overview (/study/app/math-ai-hl/sid-132-cid-761618/ov

Steps	Explanation
One possible option of a walk from A to B that ends at D is AFBFED, which is shown below:  However, as this walk includes edge BF twice, it cannot be considered a trail. In fact, all the trails that can go from A to B and end at D use edge BF more than once. Therefore, there are no trails that meet the requirements of the problem.	Remember that a trail cannot include any repeated edges.

Cycles and circuits

✓ **Important**

A cycle on a graph is a walk that begins and ends at the same vertex, but otherwise there are no repeated vertices. The length of the cycle is the total number of included edges. It is also described as a closed path.

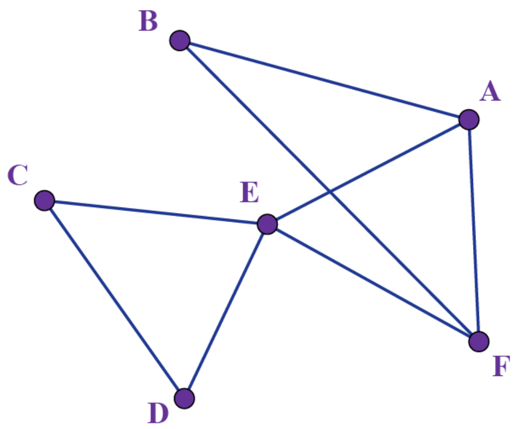
Example 3

★★☆

Student view

List all cycles in the graph below that begin at vertex E and state their length.

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov



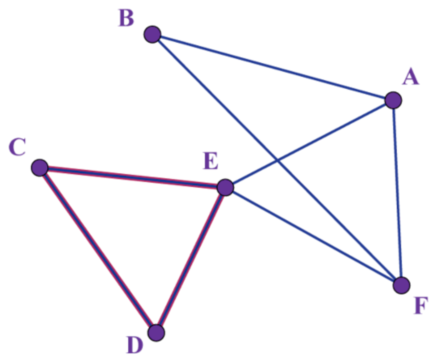
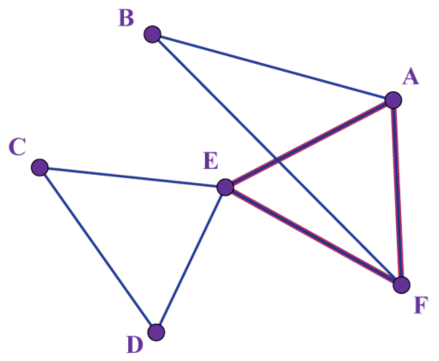
More information

The graph comprises six vertices labeled A, B, C, D, E, and F. Vertex E is central, connecting to vertices A, B, C, and F. Vertex A connects to vertices B and F. Vertex B connects to vertex E. Vertex C connects to vertices D and E. Vertex D connects to vertex C. Vertex F connects to vertices A and E. The task is to identify cycles beginning at vertex E and note their lengths.

[Generated by AI]

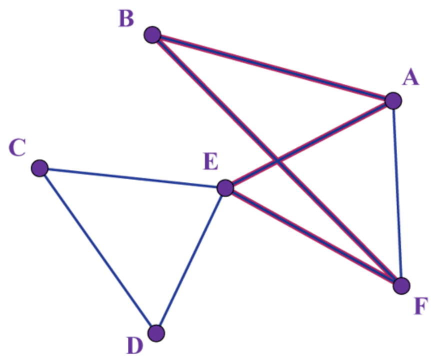
Student
view

Overview (/study/app/math-ai-hl/sid-132-cid-761618/ov

Steps	Explanation
One possible cycle beginning at vertex E is ECDE (or EDCE). This cycle has a length of 3. 	Remember that a cycl cannot include any repeated vertices and must begin and end a the same vertex.
Another possible cycle that begins at vertex E is EAFE (or EFAE). This cycle has a length of 3. 	
The third possible cycle that begins at vertex E is EABFE (or EFBAE). This cycle has a length of 4.	

Student view

Overview (/study/app/math-ai-hl/sid-132-cid-761618/ov

Steps	Explanation
	

Be aware

When stating the length of a cycle, you do not count the initial vertex twice. Therefore, the cycle ECDE has a length of 3 and the cycle EABFE has a length of 4.

Important

A circuit on a graph is a trail that begins and ends at the same vertex. The length of the circuit is the total number of included edges. It is also described as a closed trail.

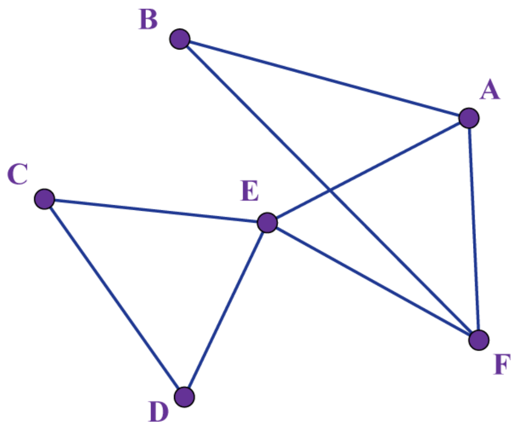
Example 4

★★☆

State a circuit in the graph shown below that begins at vertex F and travels through vertex C.

Student view

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov



More information

The image is a graph with six vertices labeled A, B, C, D, E, and F. The vertices are connected by edges in the following manner:

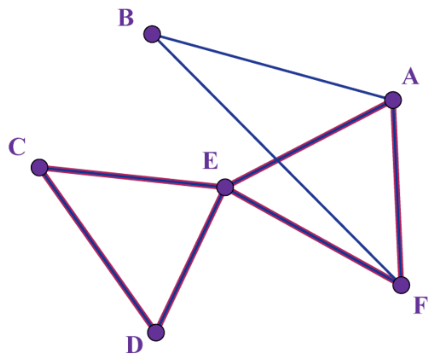
- Vertex A is connected to B, E, and F.
- Vertex B is connected to A, E, and C.
- Vertex C is connected to B, E, and D.
- Vertex D is connected to C and E.
- Vertex E is connected to A, B, C, D, and F.
- Vertex F is connected to A and E.

To find a circuit starting at vertex F and passing through vertex C, one possible path is: F to A, A to B, B to C, C to D, D to E, and E back to F. This forms a loop starting and ending at vertex F.

[Generated by AI]

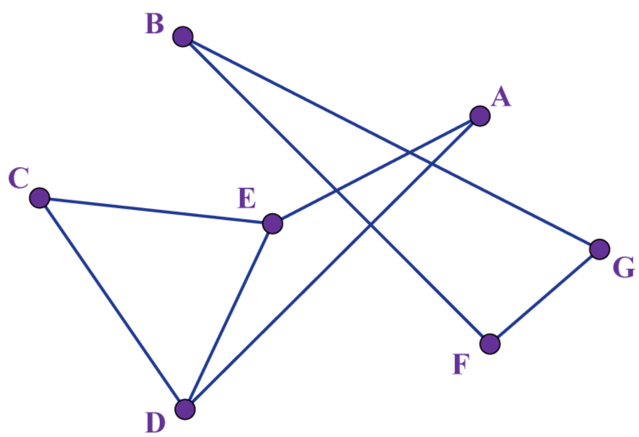
Student view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

Steps	Explanation
The walk shown below meets the requirements of being a circuit. One possible way of notating the circuit is $\mathrm{FECDEAF}$.  Notice that this walk cannot be considered a cycle since vertex E is used more than once.	Remember that a circ begins and ends at th same vertex and canr include any repeated edges.

Connected graphs

Consider the graph below. Can you find a path from vertex A to vertex B?



Student
view

More information

This is a graph depicted with vertices labeled as A, B, C, D, E, F, and G. Lines or edges connect the vertices as follows: A is connected to E, F, and G. B is connected to E. C is connected to E and D. D is connected to E and F. E also connects to F. The connections form a network where some vertices can reach each other through one or more edges, but there is no continuous path from vertex A to vertex B, making the graph disconnected. The layout is such that vertex E acts as a central node connecting multiple edges, while vertex G only connects to A, isolating it from other components except through vertex A.

[Generated by AI]

By looking at all the possible edges in the graph, you see that it is not possible to travel from vertex A to vertex B. For this reason, the graph is considered a disconnected graph.

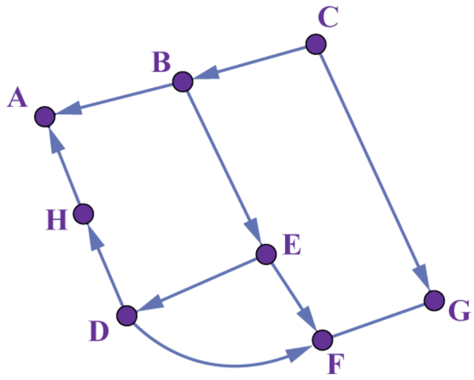
Important

A connected graph contains a path from every vertex to every other vertex in the graph.

Let us now consider a directed graph from section 3.14.3 (/study/app/math-ai-hl/sid-132-cid-761618/book/weighted-and-directed-graphs-id-28229/), which is shown below. Is it possible to start at vertex F and travel to any other vertex in the graph?

As you can see from the diagram, the directed edges of the graph limit your ability to start at F and travel to vertices other than G. In fact, the only vertex that you can start at and still reach any other vertex is vertex C. This is another example of a disconnected graph.

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov



More information

This image shows a directed graph with eight vertices labeled from A to H. Directed edges connect the vertices as follows:

- A points to B and H.
- B points to C.
- C points to E and G.
- D points to A and F.
- E points to B and F.
- F points to G and D.
- G has no outgoing edges.
- H points to D.

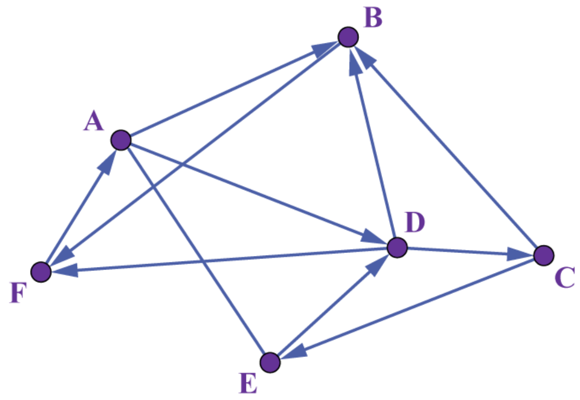
The diagram demonstrates how the directed edges limit movement from vertex F, where you can only reach vertex G. Conversely, starting at vertex C allows reaching all other vertices, illustrating the concept of a disconnected graph.

[Generated by AI]

In comparison, now consider the directed graph below. Verify that it is possible to reach every vertex from every other vertex in the graph.

Student
view

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov



More information

The image is a directed graph with six vertices labeled A, B, C, D, E, and F. Each vertex is a point, and they are connected by arrows that indicate the direction of travel from one vertex to another. The connections are as follows:

- A is connected to B, D, and F.
- B is connected to C and D.
- C is connected to D.
- D is connected to B and E.
- E is connected to A and F.
- F is connected to A.

The graph is laid out in an irregular shape, with the vertices and arrows creating multiple pathways. The task is to verify reachability across all vertices, ensuring that every vertex can be reached from every other vertex.

[Generated by AI]

Important

A strongly connected graph is a **directed graph** that contains a path from every vertex to every other vertex in the graph.

3 section questions

Student view

Minimum spanning trees

Trees

Making connections

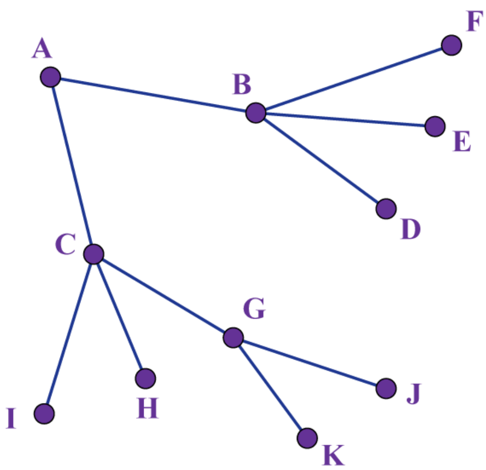
Recall from the previous section (/study/app/math-ai-hl/sid-132-cid-761618/book/moving-around-a-graph-id-28369/) that a cycle is a path that begins and ends at the same vertex.

Important

A tree is a **connected** graph that contains no cycles.

Activity

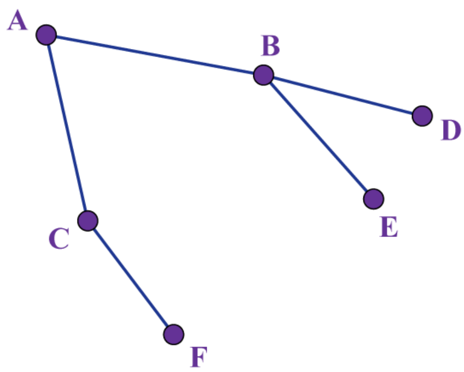
Consider the three graphs shown below.



More information

The diagram represents a network of interconnected nodes labeled from A to K. Node A connects to nodes B and C. Node B connects to nodes F, E, D, and C. Node C connects to nodes I, H, and G. Node G connects to nodes J and K. Arrows indicate the direction of the connections, forming a tree-like structure with multiple branching paths from the central nodes A, B, and C.

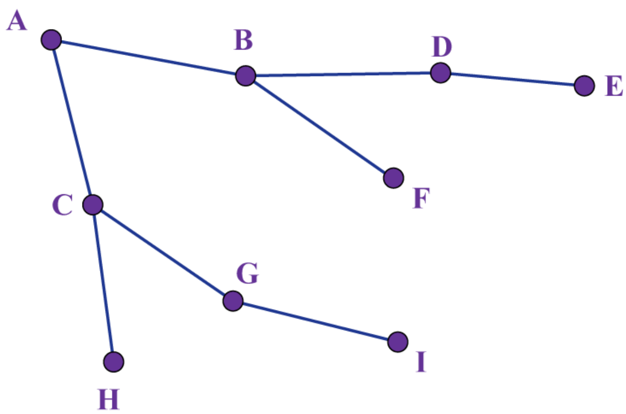
[Generated by AI]



More information

The image displays a diagram of a connected graph with six nodes labeled A, B, C, D, E, and F. Node A is connected to nodes B and C. Node B connects to nodes D and E, while node C is linked to node F. The graph showcases lines connecting these nodes to portray a network structure.

[Generated by AI]



Student
view

More information

The image is a diagram, specifically depicting a network of nodes connected by lines. There are nine nodes labeled from A to I. Node A connects to nodes B and C. Node B further connects to nodes D and F. Node D connects to node E. Node C connects to nodes G and H. Node G connects to node I. The layout represents a tree-like structure with branching paths among the nodes. Each connection is represented by a line indicating the relationship between the nodes. The nodes are distributed throughout the diagram with their specific labels visible.

[Generated by AI]

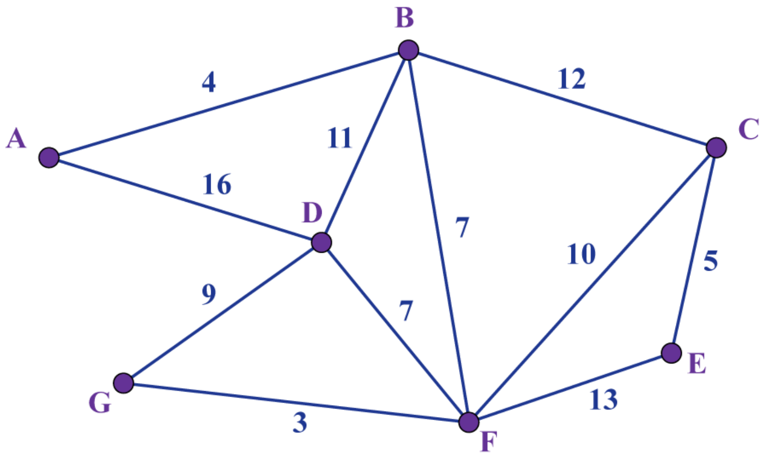
1. Count the number of vertices and edges in each of the graphs.
2. Propose a relationship based on the pattern in the numbers.
3. Draw a few more trees of different sizes. Does your relationship continue to hold true?
4. Explain why the relationship is true.
5. Note that each of the trees shown above are drawn so that vertex A is the root of the tree. An interesting fact about trees is that any of the vertices can be used as the root. Verify this by redrawing each tree with vertex B as the root.

Important

A tree that contains n vertices will contain $n-1$ edges.

Minimum spanning trees

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov



More information

The image is a weighted graph representing a network of villages, labeled as vertices A, B, C, D, E, F, and G. The graph is connected by edges, each with a specific numerical value representing the cost of connecting the respective villages. The edges and their weights are as follows:

- A to B: 4
- A to D: 16
- A to G: 9
- B to C: 12
- B to D: 11
- B to F: 7
- D to F: 7
- C to E: 5
- E to F: 13
- F to G: 3

The illustration appears to show a method for connecting the villages to the electric grid at minimal cost, with understanding of the edge weights being crucial for determining the shortest paths.

[Generated by AI]

Student
view

Imagine that vertices in the weighted graph shown above represent the villages within a small region of a country. The weights of each edge represent the cost of connecting each village to the electric grid. How can the local government connect all the villages to the grid

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

for the minimum cost?

To answer the question posed above, you need to find the minimum spanning tree for the graph shown above.

Important

A minimum spanning tree is the subgraph of an undirected weighted graph that has the following three attributes:

- It is the subgraph with the lowest weight.
- It connects all the vertices in the graph.
- It does not include any cycles.

Activity

Interactive 1. Demonstrating Kruskal's Algorithm and Prim's Algorithm.
Credit: GeoGebra (https://www.geogebra.org/m/QVJP48tR) slik

More information for interactive 1

This interactive visualizes and compares two key algorithms for constructing a Minimum Spanning Tree (MST) in a weighted graph: Kruskal's algorithm and Prim's algorithm. The screen displays a connected graph made up of seven vertices labeled A through G, forming a hexagonal network. Each pair of connected vertices has a numerical label showing the weight (or cost) of the edge

Student view

Overview (/study/app/math-ai-hl/sid-132-cid-761618/ov

between them. Users can toggle between Kruskal's and Prim's algorithms using a switch at the top of the screen. Once an algorithm is selected, pressing the Play button initiates a step-by-step animation of the selected method.

For Kruskal's algorithm, the edges are first sorted in increasing order of weight. The algorithm then selects the shortest edge that doesn't form a cycle, progressively building the MST. For example, the algorithm might start with the edge EG (weight 13), then AC (15), followed by AB (18), DE (19), BE (21), and finally FG (27), yielding a minimum spanning tree with a total length of 113. The interactive highlights each selected edge in orange and displays it in a list on the right side. Each step is also labeled along the left margin (Step 1 through Step 7), providing a clear breakdown of the MST formation.

When Prim's algorithm is selected, the process begins at a specific vertex (default is A). The algorithm grows the MST by repeatedly adding the smallest edge that connects the current tree to a new vertex. As with Kruskal's, the selected edges are highlighted, listed, and the total MST weight is calculated. The algorithm may follow a different order of adding edges, but the final tree will still have the same total weight.

Additional features include a Reset button to start over and replay the algorithm with a fresh perspective, a green text box at the bottom that gives step-by-step textual feedback (e.g., "AC is the next shortest"), and dynamic highlighting of included edges. This hands-on tool gives learners an intuitive understanding of MST construction, how different strategies select edges, and how both algorithms achieve the same optimal result through different approaches.

The applet demonstrates Kruskal's algorithm and Prim's algorithm . These are the two algorithms for finding a minimum spanning tree that you will need to know for this course.

1. Begin by moving the slider at the top right to Kruskal's and press play.
2. From the explanations that appear in the green box as the animation runs, list the steps in the algorithm. If necessary, reset the applet and run the animation again.
3. Note the final tree that Kruskal's algorithm creates and its length.
4. Next, move the slider to Prim's and again list the steps in the algorithm.
5. How does the final tree created by Prim's algorithm compare to that for Kruskal's? Consider both the shape of the tree and its length.
6. Which of the two algorithms do you prefer to use? Can you think of situations where one algorithm would be better to use than the other?

Kruskal's algorithm

The following steps of Kruskal's algorithm find a minimum spanning tree of a graph:

1. List all the edges in order by weight from the smallest to the largest.

Student view

Overview (/study/app/math-ai-hl/sid-132-cid-761618/ov

2. Begin choosing edges for the minimum spanning tree by selecting the edge with the smallest weight. If there is more than one edge with the smallest weight, choose any of them.
3. Select the edge with the next smallest weight that does not create a cycle. If there is more than one option for the next smallest edge, choose any of them.
4. Repeat Step 3 until all the vertices of the graph have been included in the tree.

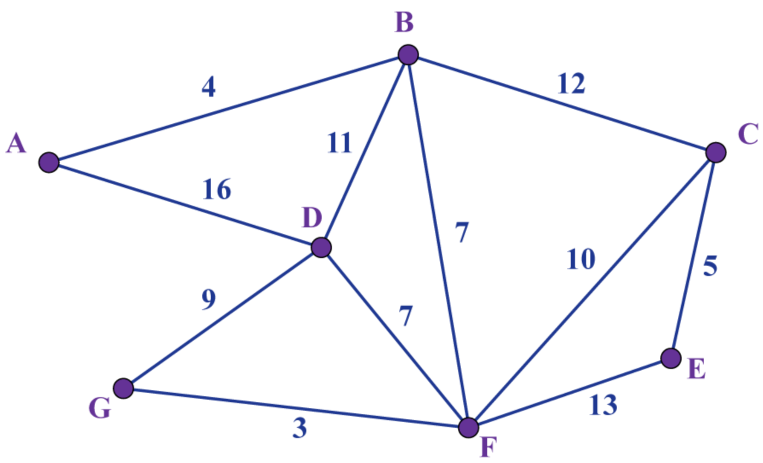
Exam tip

A graph may have several minimum spanning trees. However, each of them will have the same overall length.

Example 1

★★☆

The villages in a small region of a country are represented as the vertices in the weighted graph below. The weight of each edge represents the cost, in thousands of euros, of laying a power line between two villages. Use Kruskal's algorithm to determine the minimum cost needed to ensure that all the villages are connected to the electric grid.



More information

Student view

The image is a weighted graph representing the cost of laying power lines between seven villages labeled A through G. The vertices represent the villages, and the edges are marked with weights indicating costs in thousands of euros.

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

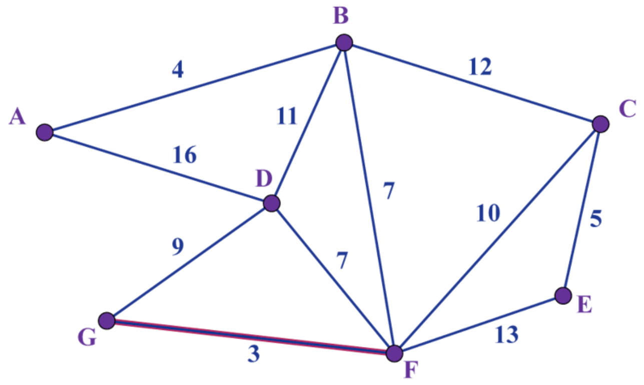
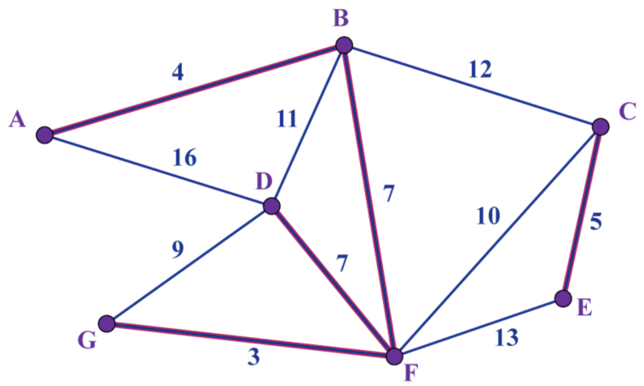
- The edges and their respective weights are:
- Between A and B: 4
- Between A and D: 16
- Between A and G: 9
- Between B and C: 12
- Between B and D: 11
- Between B and F: 7
- Between C and E: 5
- Between D and E: 10
- Between D and F: 7
- Between E and F: 13
- Between F and G: 3

These values indicate the cost of connecting each pair of villages with a power line, and the task is to find the minimum spanning tree using Kruskal's algorithm to ensure all villages are connected with minimal cost.

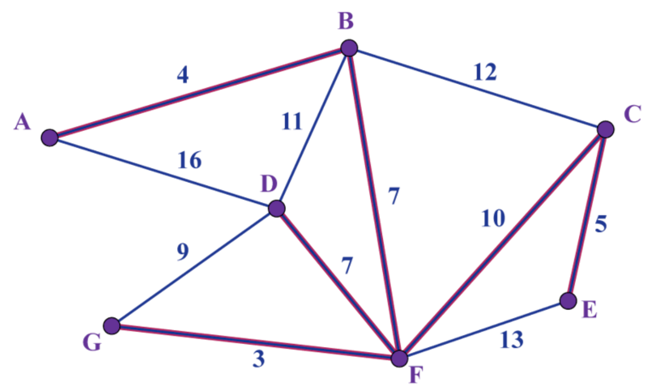
[Generated by AI]

Steps	Explanation
$\text{FG} - 3$ $\text{AB} - 4$ $\text{CE} - 5$ $\text{BF} - 7$ $\text{DF} - 7$ $\text{DG} - 9$ $\text{CF} - 10$ $\text{BD} - 11$ $\text{BC} - 12$ $\text{EF} - 13$ $\text{AD} - 16$	Step 1: List all the edges in the graph i order of weight, beginning with the smallest weight.

Student view

Steps	Explanation
	Step 2: Select the edge with the smallest weight.
 Notice that at this point the next smallest edge is DG . However, selecting DG would create a cycle. Therefore, it is omitted.	Step 3: Continue by choosing edges witl the next smallest weight.

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

Steps	Explanation
	Step 4: Repeat Step until all vertices hav been included in the tree.
$\text{Total cost} = 3 + 4 + 5 + 7 + 7 + 10 = 36$. $\therefore$ the minimum cost to connect all the villages in the region to the electric grid is $36\,000$ euros.	State the minimum cost by summing th weights of all the edges in the tree.

Exam tip

When building a minimum spanning tree, you will need to show each edge in the order you considered it and whether it was included or skipped.

International Mindedness

Section Student... (0/0) Feedback Print (/study/app/math-ai-hl/sid-132-cid-761618/book/moving-around-a-graph-id-28369/print/) Assign

Reliable access to electricity may be something that you take for granted. However, according to the World Bank, more than 11% of the world's population still did not have access to electricity in 2017. That means over $800\,000\,000$ people do have not electricity. A list of the percentage of each country's population with access to electricity can be found here (https://data.worldbank.org/indicator/eg.elc.accs.zs?end=2017&start=1990).

With all the technological developments occurring in the world today, why is it so difficult to bring electricity to everyone? An article that discusses some of the problems faced by a country trying to bring electricity to its entire population can

Student view

Overview (/study/app/math-ai-hl/sid-132-cid-761618/ov

be found here (https://aif.org/rural-electrification-in-india-problems-progress-and-the-power-of-citizens-media/).

Prim's algorithm

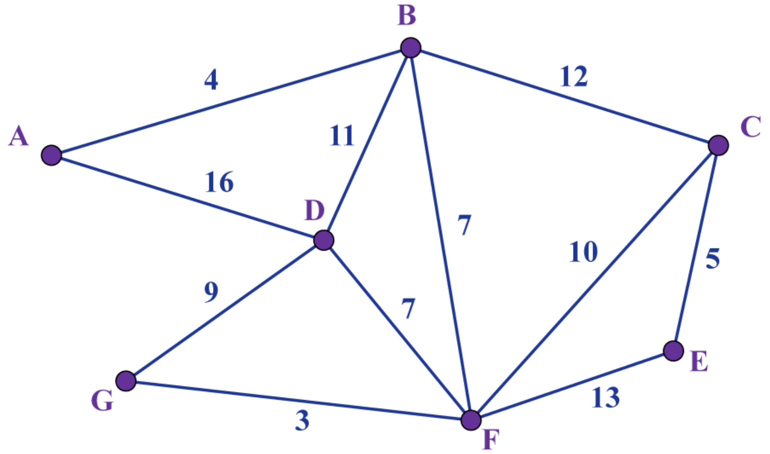
The following steps of Prim's algorithm are followed to find a minimum spanning tree of a graph:

1. Choose any vertex in the graph (sometimes this will be stated in the problem).
2. Consider all the vertices adjacent to the vertex chosen in Step 1. Choose the one whose edge has the lowest weight and also choose the edge.
3. Consider all the vertices adjacent to the vertices already chosen in previous steps. Choose the one whose edge has the lowest weight. Do not choose a vertex more than once. Choose also the corresponding edge.
4. Repeat Step 3 until all the vertices in the graph are included in the tree.

Example 2

★★☆

The graph below shows the villages in a small region of a country again. Given that the power lines will begin at the village represented by vertex A, use Prim's algorithm to determine the minimum cost needed to ensure that all the villages are connected to the electric grid.



Student view

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

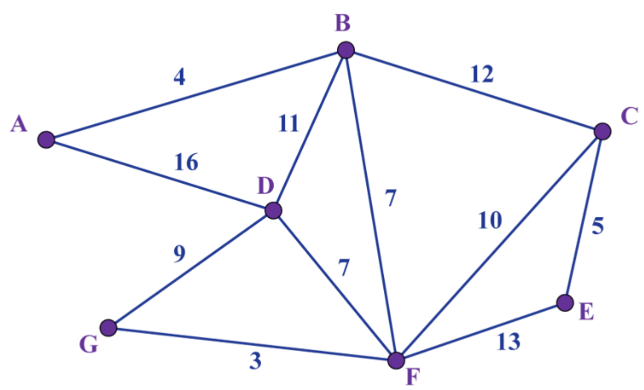
More information

The image is a graph consisting of seven vertices labeled A, B, C, D, E, F, and G. The vertices are connected by edges that have weights assigned to them, representing distances or costs.

- Vertex A is connected to vertex B with an edge weight of 4, to vertex D with an edge weight of 16, and to vertex G with an edge weight of 9.
- Vertex B is connected to vertex C with an edge weight of 12, to vertex D with an edge weight of 11, and to vertex F with an edge weight of 7.
- Vertex C is connected to vertex E with an edge weight of 5 and to vertex F with an edge weight of 10.
- Vertex D is connected to vertex F with an edge weight of 7.
- Vertex E is connected to vertex F with an edge weight of 13.
- Vertex F is connected to vertex G with an edge weight of 3.

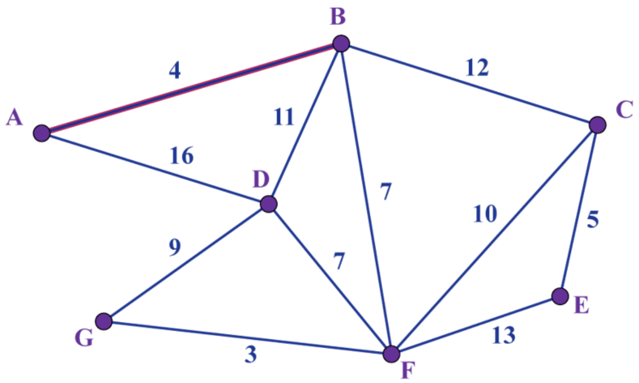
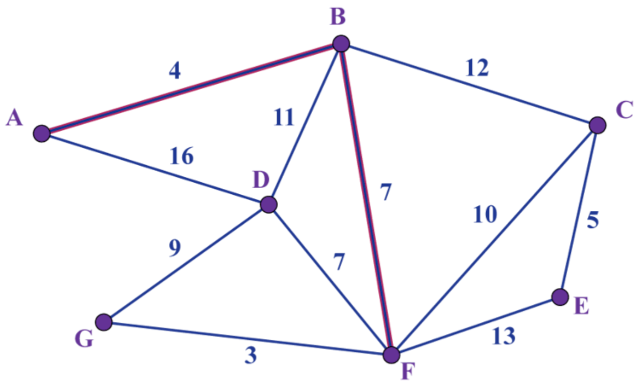
The task is to use Prim's algorithm starting at vertex A to find the minimum cost of connecting all the villages.

[Generated by AI]

Steps	Explanation
	Step 1: Begin with vertex A as stated i the problem.

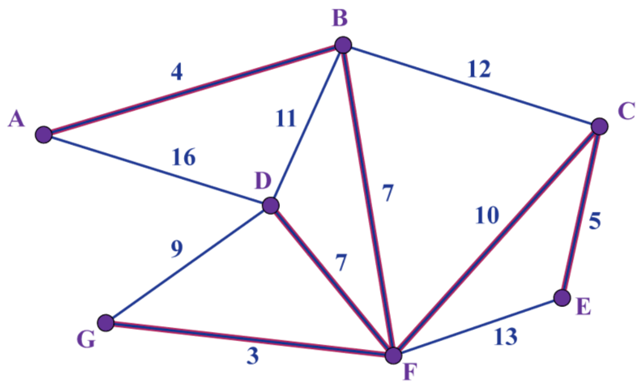
Student view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

<table>
<tr><th>Steps</th><th>Explanation</th></tr>
<tr><td>Vertices B and D are adjacent to vertex A. Vertex B is chosen since the edge connecting it to A has a weight of 4, which is less than the weight of the edge connecting vertex D to A.

</td><td>Step 2: Consider al the vertices adjacer to vertex A. Choos the one with the lowest weight edge and mark the edge.</td></tr>
<tr><td>You now have three options: vertices C, D, or F. Of the four possible edges that could be chosen (AD, BD, BF, or BC), the one with the lowest weight is BF, so it is chosen.

</td><td>Step 3: Consider all other vertices adjacent to vertices A and B. Choose tl vertex with the lowe weight edge and mark the corresponding edg</td></tr>
</table>

Student
view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

Steps	Explanation
	Step 4: Repeat Step 3 until all vertices have been included in the tree.
$\text{Total cost} = 3 + 4 + 5 + 7 + 7 + 10 = 36.$ $\therefore$ the minimum cost to connect all the villages in the region to the electric grid is $36\,000 \text{ euros}.$	State the minimum cost by summing the weights of all the edges in the tree.

Prim's algorithm with adjacency tables

Making connections

Recall from section 3.15.1 (/study/app/math-ai-hl/sid-132-cid-761618/book/adjacency-tables-id-28234/) that weighted adjacency tables include the weight for each edge between two vertices rather than the number of edges connecting the vertices.

One significant advantage that Prim's algorithm has over Kruskal's algorithm is that it can be used with the weighted adjacency table for a large graph (or network) and there is no need to look at the graph. Here are the steps in Prim's algorithm for a weighted adjacency table:

Student view

1. Cross out the row in the table for the initial vertex and write the number 1 above the column for that vertex.

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

2. Circle the lowest non-zero value in the numbered column of the table and write down the edge represented by the circled value in the edge list for the minimum spanning tree being created.
3. Cross out the row in the table that contains the newly selected value and number the column in the table that corresponds with the newly selected vertex.
4. Circle the lowest non-zero value from the numbered columns of the table that has not been crossed out and add the edge represented by the circled value to the edge list.
5. Repeat Steps 3 and 4 until all rows have been crossed out.

Example 3

★★★

The table shown below is the weighted adjacency table for the villages in the small region of a country. The weight of each edge represents the cost, in thousands of euros, of laying power lines between two villages. Given that the power lines will begin at the village represented by vertex A, use Prim's algorithm with the weighted adjacency table to determine the minimum cost needed to ensure that all the villages are connected to the electric grid.

	A	B	C	D	E	F	G
A	0	4	0	16	0	0	0
B	4	0	12	11	0	7	0
C	0	12	0	0	5	10	0
D	16	11	0	0	0	7	9
E	0	0	5	0	0	13	0
F	0	7	10	7	13	0	3
G	0	0	0	9	0	3	0

Student view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

Steps	Explanation

	1						
	A	B	C	D	E	F	G
~~A~~	~~0~~	~~4~~	~~0~~	~~16~~	~~0~~	~~0~~	~~0~~
B	4	0	12	11	0	7	0
C	0	12	0	0	5	10	0
D	16	11	0	0	0	7	9
E	0	0	5	0	0	13	0
F	0	7	10	7	13	0	3
G	0	0	0	9	0	3	0

Step 1: Cross out th row that correspon to vertex A, the starting vertex, and write the number 1 above column A.

	1						
	A	B	C	D	E	F	G
~~A~~	~~0~~	~~4~~	~~0~~	~~16~~	~~0~~	~~0~~	~~0~~
B	4	0	12	11	0	7	0
C	0	12	0	0	5	10	0
D	16	11	0	0	0	7	9
E	0	0	5	0	0	13	0
F	0	7	10	7	13	0	3
G	0	0	0	9	0	3	0

Step 2: Circle the lowest non-zero val in the numbered column and write down the relevant edge in the list for t minimum spanning tree.

AB

Student
view

Steps	Explanation

	1	2					
	A	B	C	D	E	F	G
~~A~~	~~0~~	~~4~~	~~0~~	~~16~~	~~0~~	~~0~~	~~0~~
~~B~~	~~4~~	~~0~~	~~12~~	~~11~~	~~0~~	~~7~~	~~0~~
C	0	12	0	0	5	10	0
D	16	11	0	0	0	7	9
E	0	0	5	0	0	13	0
F	0	7	10	7	13	0	3
G	0	0	0	9	0	3	0

Step 3: Cross out th
row that correspond
to the new vertex, B
and write the numb
2 above column B.

	1	2					
	A	B	C	D	E	F	G
~~A~~	~~0~~	~~4~~	~~0~~	~~16~~	~~0~~	~~0~~	~~0~~
~~B~~	~~4~~	~~0~~	~~12~~	~~11~~	~~0~~	~~7~~	~~0~~
C	0	12	0	0	5	10	0
D	16	11	0	0	0	7	9
E	0	0	5	0	0	13	0
F	0	7	10	7	13	0	3
G	0	0	0	9	0	3	0

Step 4: Circle the
lowest non-zero val
in the numbered
columns that has no
been crossed out ar
write down the
relevant edge in the
list for the minimum
spanning tree.

BF

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

<table>
<tr><th>Steps</th><th>Explanation</th></tr>
<tr><td>
<table>
<tr><td></td><td>1</td><td>2</td><td>6</td><td>5</td><td></td><td>3</td><td>4</td></tr>
<tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td></tr>
<tr><td>A</td><td>0</td><td>4</td><td>0</td><td>16</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>B</td><td>4</td><td>0</td><td>12</td><td>11</td><td>0</td><td>7</td><td>0</td></tr>
<tr><td>C</td><td>0</td><td>12</td><td>0</td><td>0</td><td>5</td><td>10</td><td>0</td></tr>
<tr><td>D</td><td>16</td><td>11</td><td>0</td><td>0</td><td>0</td><td>7</td><td>9</td></tr>
<tr><td>E</td><td>0</td><td>0</td><td>5</td><td>0</td><td>0</td><td>13</td><td>0</td></tr>
<tr><td>F</td><td>0</td><td>7</td><td>10</td><td>7</td><td>13</td><td>0</td><td>3</td></tr>
<tr><td>G</td><td>0</td><td>0</td><td>0</td><td>9</td><td>0</td><td>3</td><td>0</td></tr>
</table>
</td><td>Step 5: Repeat Step 3 and 4 until all row have been crossed out.
• circle 3 in column F and add edge FG
• circle 7 in column F and add edge FD
• circle 10 in column F and add edge FC
• circle 5 in column C and add edge CE</td></tr>
<tr><td>$\text{Total cost} = 4 + 7 + 3 + 7 + 10 + 5 = 36$. $\therefore$ the minimum cost to connect all the villages in the region to the electric grid is $36\,000$ euros.</td><td>State the minimum cost by summing th weights of all the edges in the tree.</td></tr>
</table>

Exam tip

The edges in the list for the minimum spanning tree should be listed in the order in which they were chosen by the algorithm.

3 section questions

3. Geometry and trigonometry / 3.16 Graph algorithms

Chinese postman problem

Student
view

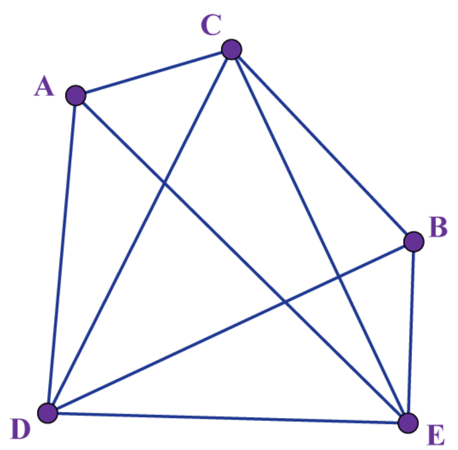
Overview (/study/app/math-ai-hl/sid-132-cid-761618/ov

Eulerian graphs

Making connections

Recall from section 3.14.0 (/study/app/math-ai-hl/sid-132-cid-761618/book/the-big-picture-id-28226/) that the problem involving the seven bridges of Königsberg prompted Leonhard Euler to begin developing graph theory. In the problem, each edge (bridge) was to be used only once.

Activity

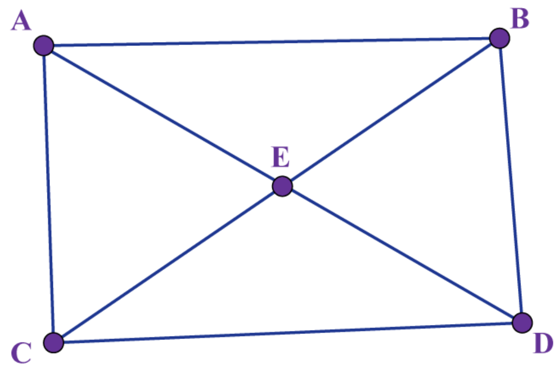


More information

The image is a diagram of a graph consisting of five points, labeled A, B, C, D, and E. These points are connected by a series of lines creating a network.\n\n- Point A is connected to points B, C, and D.\n- Point B is connected to points E and D.\n- Point C is connected to points A, D, and E.\n- Point D is connected to A, B, C, and E.\n- Point E is connected to B, C, and D.\n\nThis forms a complete graph where each point is directly linked to every other point by straight lines, showcasing multiple intersecting paths.

[Generated by AI]

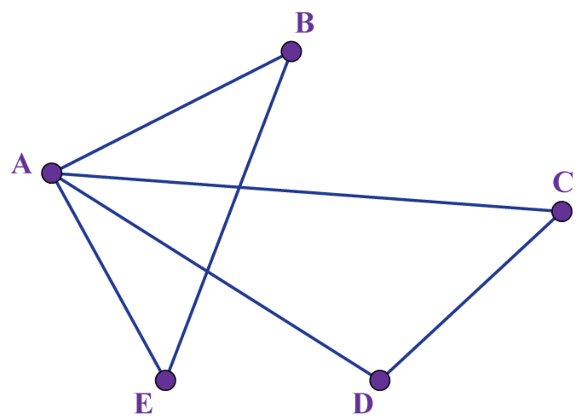
Student view



More information

The image is a geometric diagram of a quadrilateral formed by four labeled points: A, B, C, and D. These points are connected by straight lines to form a four-sided figure. Inside this quadrilateral, there is a central point labeled E. Straight lines are drawn from point E to each of the four corners, dividing the quadrilateral into four triangles: ABE, BCE, CDE, and DAE. The overall shape is trapezoidal with A and B forming the top edge, and C and D forming the bottom edge. The relationships among the points, as well as their connections, illustrate a geometric structure useful in studying properties of polygons.

[Generated by AI]



More information

The image depicts a simple graph diagram with five nodes labeled A, B, C, D, and E. Each node is represented by a purple circle, and they are connected by straight blue lines. The node A is central, connecting directly with nodes B, C, D, and E. Node B is additionally connected to node

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

C. Node C is connected to node D, and node D is connected to node E. This layout forms a network of connections resembling a pentagon with diagonals. The nodes are labeled with capital letters beside them, and the lines indicate relationships or connectivity between the points.

[Generated by AI]

1. Choose one of the graphs shown above and copy it onto a piece of paper.
2. Can you find a way to redraw the whole graph without lifting your pencil and without drawing an edge more than once?
3. Repeat Step 2 for the other two graphs. Can you find a relationship between the degrees of the vertices within a graph and the ability to redraw the graph following the rules in Step 2?
4. How many vertices of odd degree are there in each of these graphs? Is it possible to draw in this way a graph with three odd-degree vertices? Why or why not? ? (Hint: Recall from section 3.15.1 (/study/app/math-ai-hl/sid-132-cid-761618/book/adjacency-tables-id-28234/) that the sum of the degrees of all the vertices in a graph is equal to twice the number of edges, which means the sum is even.)

Important

A Eulerian trail is a trail that contains all the edges within a graph. A graph that contains a Eulerian trail is called a semi-Eulerian graph. A connected graph contains a Eulerian trail if and only if it contains exactly two vertices with an odd degree.

Exam tip

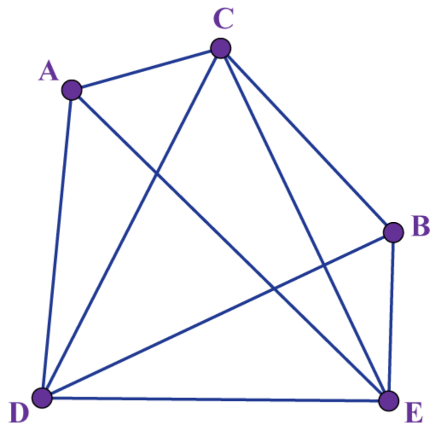
Recall that in a trail, no edges are repeated.

Example 1

★☆☆

Student view

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov



More information

The image is a graph consisting of five vertices labeled A, B, C, D, and E. Each vertex is connected by edges as follows:

- Vertex A is connected to vertices B, C, and D.
- Vertex B connects to vertices A, C, D, and E.
- Vertex C is linked to vertices A, B, and D.
- Vertex D connects to vertices A, B, C, and E.
- Vertex E connects to vertices B and D.

The task is to determine whether the graph contains a Eulerian trail. A Eulerian trail is a path that visits every edge exactly once. If the graph is semi-Eulerian, implying it contains a Eulerian trail starting and ending at different vertices, then one possible trail needs to be stated.

[Generated by AI]

Determine whether the graph above contains a Eulerian trail. If the graph is semi-Eulerian, state one possible Eulerian trail.

Steps	Explanation
$\deg(\mathrm{A}) = \deg(\mathrm{B}) = 3$ $\deg(\mathrm{C}) = \deg(\mathrm{D}) = \deg(\mathrm{E}) = 4$	Begin by determining the degree of each vertex.

Student view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

Steps	Explanation
One possible Eulerian trail is $\mathrm{AEDACBECDB}$.	As the graph contains exactly two vertices with a odd degree, the graph must contain a Eulerian trail.

Exam tip

A Eulerian trail will always begin at a vertex with an odd degree and end at the other vertex with an odd degree.

Important

A Eulerian circuit is a closed trail that contains all the edges within a graph. A graph that contains a Eulerian circuit is called a Eulerian graph. A connected graph contains a Eulerian circuit if and only if every vertex has an even degree.

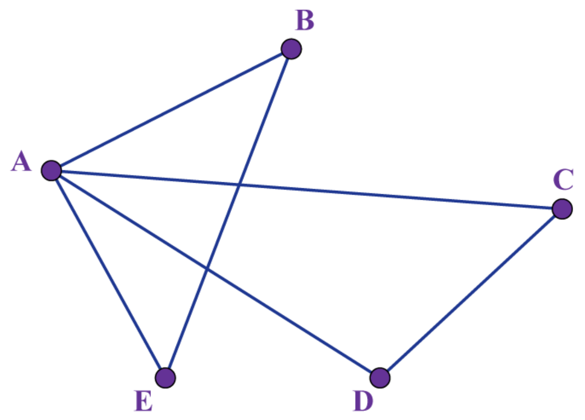
Exam tip

Recall that a circuit must be a trail that begins and ends at the same vertex.

Example 2

★☆☆

Student
view



More information

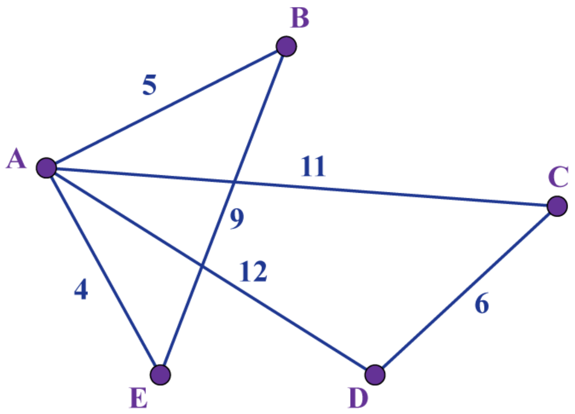
The image is a graph featuring five nodes labeled A, B, C, D, and E. The nodes are connected by various edges forming a network. Node A is connected to nodes B, C, and E. Node B is connected to nodes A and C. Node C is connected to nodes A, B, and D. Node D is connected to nodes C and E. Node E is connected to nodes A and D. The goal is to determine whether this graph contains an Eulerian circuit, which would mean visiting every edge once and returning to the starting node.

[Generated by AI]

Determine whether the graph below contains a Eulerian circuit. If the graph is Eulerian, state one possible Eulerian circuit.

Steps	Explanation
$\deg(\mathrm{A}) = 4$ $\deg(\mathrm{B}) = \deg(\mathrm{C}) = \deg(\mathrm{D}) = \deg(\mathrm{E}) = 2$	Begin by determining the degree of each vertex.
One possible Eulerian circuit is $\mathrm{ACDABEA}$.	As the degree of each vertex is even, the graph must contain a Eulerian circuit.

Chinese postman problem



More information

This is a graph illustrating the Chinese postman problem, where intersections in a small town are represented as vertices labeled A, B, C, D, and E. The edges between these vertices represent streets with assigned weights, indicating distances between the intersections:

- Edge from A to B: 5
- Edge from A to C: 11
- Edge from A to D: 9
- Edge from A to E: 4
- Edge from B to C: Not connected
- Edge from B to D: Not connected
- Edge from B to E: Not connected
- Edge from C to D: 6
- Edge from C to E: Not connected
- Edge from D to E: 12

The goal is for a postman to travel each road to deliver mail and return to the starting point. The problem is to determine the optimal path to achieve this cyclic route with minimal distance traveled.

[Generated by AI]

Consider again the graph shown above. Now, each edge has a weight. Imagine that the edges represent streets and the vertices intersections in a small town. A postman needs to travel on each road to deliver mail to all the houses along that road and wants to end up back at the depot. This is called the Chinese postman problem.

✓ **Important**

The length of a walk in a weighted graph is the sum of the weights of the edges in the walk.

Example 3

★☆☆

Determine the length of the shortest route the postman can take that allows him to travel along every road and return to where he started.

Steps	Explanation
$\text{Shortest route} = 5 + 4 + 9 + 11 + 12 + 6 = 47.$	In **Example 2**, you showed that this graph is Eulerian. Since a Eulerian circuit contains each edge of the graph exactly once, the lengths of al possible circuits will be the total weight, which is the sum of the weights of the edges.

If any vertex in a graph does not have an even degree, then no Eulerian circuit is possible. However, the postman needs to return to where they started, and do so for the route with the shortest length. They can do this by traversing some edges twice, that is, those edges that connect the vertices with an odd degree. In practice, we do this by duplicating some edges so that we can form a Eulerian circuit.

✓ **Important**

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

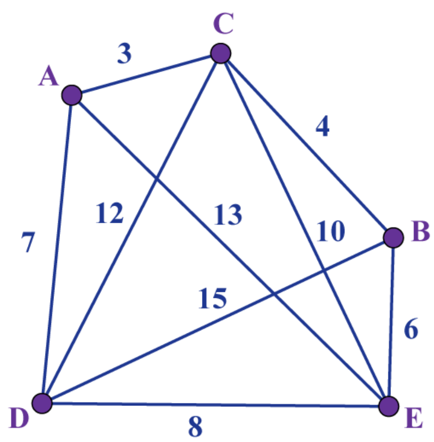
All connected graphs contain an even number of vertices with an odd degree.

Exam tip

For this course, you need to know how to solve the Chinese postman problem for graphs containing up to four vertices with an odd degree.

Example 4

★★☆



More information

The image is a graph representing a network of roads with five nodes labeled A, B, C, D, and E. Each edge between the nodes is labeled with a number, indicating the distance between the connected nodes. The connections are as follows:

- A to B: 13
- A to C: 3
- A to D: 7
- B to C: 4
- B to D: 15
- B to E: 6
- C to D: 12

Student
view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

- D to E: 8

The graph forms a network where multiple routes can be identified between nodes. The distances between the nodes indicate possible paths a postman can take to travel all roads and return to the starting point, forming a circuit.

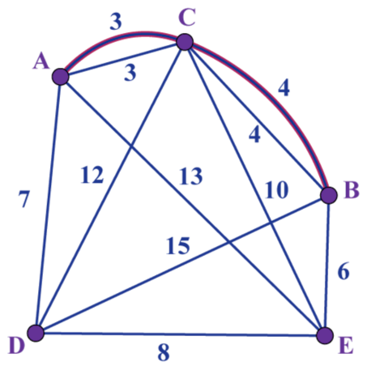
[Generated by AI]

The graph in above represents the road network of a town. Determine the length of the shortest route a postman can take that allows him to travel along every road and return to where he started.

Steps	Explanation
$\deg(\mathrm{A}) = \deg(\mathrm{B}) = 3$	In **Example 1**, you showed that this graph semi-Eulerian. It contai a Eulerian trail and has two vertices of an odd degree. The first step is to identify the two odd- degree vertices.
The shortest path between vertex A and vertex B is along edges AC and CB. It has a length of 7.	Next, find the shortest path between the two odd-degree vertices by looking at the graph.

Student
view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

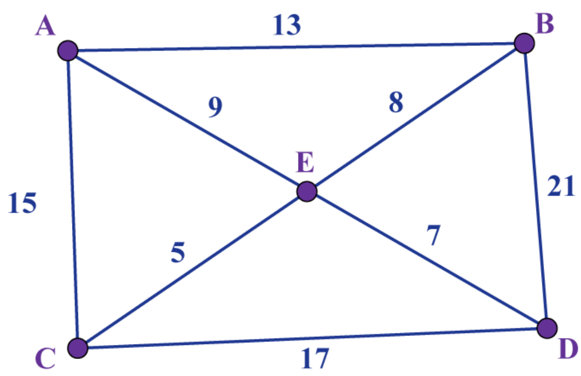
Steps	Explanation
	Add duplicate edges A and CB to the graph.
One possible circuit is AEDACBECDBCA. The length of the circuit is: $$7+8+13+3+4+15+12+10+6+4+3=85$$	All the vertices in the graph now have an eve degree, so it has Euleri circuits. You need to fir only one circuit, as all circuits will have the same length (as explained in **Example 3**

A similar process can be used when you have a graph with four vertices with odd degree. However, you must now connect two pairs of odd-degree vertices rather than just one as in **Example 4**. To do this, you will need to analyse the three possible ways of grouping the four odd-degree vertices into two pairs and then choose the combination of pairs that can be connected with the shortest paths.

Example 5

★★★

Student
view



More information

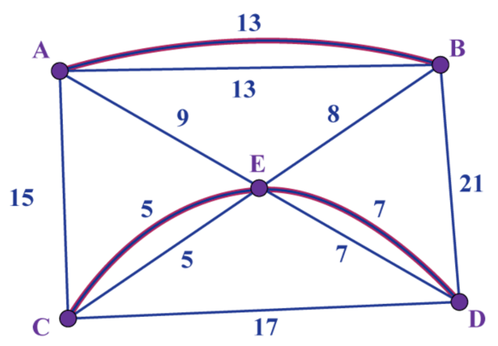
The graph represents a road network of a town with five points labeled as A, B, C, D, and E. The points are connected by roads with distances labeled in between them. Point A connects to B with a distance of 13, to E with a distance of 9, and to C with a distance of 15. Point C connects to E with a distance of 5, to D with a distance of 17, and directly to B with a distance of 21. Point B connects to E with a distance of 8. Point D connects to E with a distance of 7. The graph is labeled with distances: AB is 13, BC is 21, CD is 17, DA is 15, AE is 9, BE is 8, CE is 5, and DE is 7. The task is to determine the shortest route for a postman, Deni, to travel along every road and return to the starting point.

[Generated by AI]

The graph above represents the road network of a town in which Deni is a postman. Determine the length of the shortest route for Deni that allows him to travel along every road and return to where he started.

Steps	Explan
$\deg(\mathrm{A}) = \deg(\mathrm{B}) = \deg(\mathrm{C}) = \deg(\mathrm{D}) = 3$ $\deg(\mathrm{E}) = 4$	Begin b identifyi the deg of each vertex.

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

Steps	Explan
Option 1: AB and CD Option 2: AC and BD Option 3: AD and BC	As there four ver with an degree, the thre possible ways of sorting into two pairs.
Option 1: $\mathrm{AB} = 13$ (using edge AB) $\mathrm{CD} = 12$ (using edges CE and ED) $\text{Total} = 25$ Option 2: $\mathrm{AC} = 14$ (using edges AE and EC) $\mathrm{BD} = 15$ (using edges BE and ED) $\text{Total} = 29$ Option 3: $\mathrm{AD} = 16$ (using edges AE and ED) $\mathrm{BC} = 13$ (using edges BE and EC) $\text{Total} = 29$	For eacl option, the shor path betweer each pa
Option 1 is selected, so duplicate edges AB, CE, and ED are added to the graph. 	Choose option w the lowe total and the rele edges t graph.

Student
view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

Steps	Explan
$\text{Shortest circuit length} = 13+13+15+9+8+21+5+5+7+7+17$ $= 120$	The len the sho circuit f Deni is t sum of a the wei of the e in the n graph.

3 section questions

3. Geometry and trigonometry / 3.16 Graph algorithms

Travelling salesman problem

Hamiltonian graphs

Important

A Hamiltonian cycle is a cycle that includes every vertex within a graph with no vertex included more than once. A Hamiltonian graph is a graph that contains at least one Hamiltonian cycle.

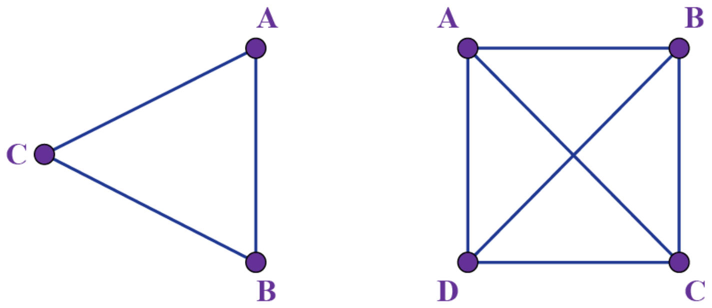
Making connections

Recall from section 3.14.2 (/study/app/math-ai-hl/sid-132-cid-761618/book/types-of-graphs-id-28228/) that a complete graph is a simple graph where each of the vertices is adjacent to every other vertex in the graph.

Student
view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

Activity



K_3 K_4

More information

The image displays two complete graphs:

1. **Graph K3**: This is a triangle formed by three nodes labeled as A, B, and C. Each node is connected to every other node by edges, creating a triangular shape.
2. **Graph K4**: This is a square formed by four nodes labeled as A, B, C, and D. The nodes, A and B, and C and D, are connected horizontally to form the top and bottom edges, respectively. The nodes A and D, and B and C, are connected vertically to form the left and right edges. Additionally, two diagonal lines connect node A to C and node B to D, creating an X shape within the square.

Both graphs are drawn in blue, and the labels are in purple. They visually demonstrate the concept of complete graphs with each node connecting to every other node within the graph.

[Generated by AI]

Section Student (0/0) Feedback Print (/study/app/math-ai-hl/sid-132-cid-761618/book/minimum-spanning-trees-id-28370/print/) Assign

For the graph K_3 above, there is only one possible Hamiltonian cycle. Although it can be written as ABCA, ACBA, BCAB, BACB, CABA, and CBAC, all these cycles contain the same edges and therefore are the same.

1. Find the number of different Hamiltonian cycles within K_4.
2. By drawing, or otherwise, find the number of different Hamiltonian cycles within the graphs K_5 and K_6.
3. By considering the emerging pattern, derive an expression for the number of different Hamiltonian cycles within K_n for $n \geqslant 3$.

Student
view

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

Important

The number of different Hamiltonian cycles within a complete graph can be found using the expression $\frac{1}{2}(n-1)!$

Be aware

While all complete graphs with at least three vertices have a Hamiltonian cycle, there is no definitive criterion to determine whether a general graph has a Hamiltonian cycle or not.

Exam tip

One rule that is sometimes helpful states that if all vertices in a graph of size n have a degree of at least $\frac{n}{2}$ then the graph will contain at least one Hamiltonian cycle.

Important

A Hamiltonian path is a path that contains all the vertices within a graph. A graph that contains a Hamiltonian path is called a semi-Hamiltonian graph .

Travelling salesman problem

Activity

In the applet below, five cities are represented as the vertices of a graph. The weight of each edge is the cost of travelling between the two cities incident with the edge. A salesman, who is based in city A, must visit each of the five cities and return home. The salesman has asked you to prepare the route for him with the cheapest possible cost.

Student view

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

Interactive 1. Travelling Salesman Problem.

Credit: GeoGebra (https://www.geogebra.org/m/yu9edvFu) GeoGebra Materials Team

More information for interactive 1

This interactive tool helps users explore and solve the classic Traveling Salesman Problem (TSP), where the objective is to determine the least expensive route that allows a salesman to visit a series of cities exactly once and return to the starting point. The visualization displays a complete weighted graph with five cities labeled A, B, C, D, and E, represented as vertices. The cities are interconnected by ten edges, each labeled with a travel cost in dollars. Users begin by selecting edges between cities using checkboxes next to each city pair, such as AB, AC, or DE. As edges are selected, they are highlighted in orange, and the total cost of the selected edges is dynamically displayed at the top of the screen. Below the graph, each edge has a horizontal slider allowing users to adjust its cost between $\$0$ and $\$300$, enabling them to simulate different pricing scenarios and test route optimizations. A reset button clears the current selections, letting users start over with a new route.

For example, a user could create the route $A \rightarrow C \rightarrow E \rightarrow D \rightarrow B \rightarrow A$ by selecting the edges AC, CE, DE, BD, and AB. Based on the default costs—$AC = \$119$, $CE = \$120$, $DE = \$199$, $BD = \$150$, and $AB = \$185$—the total route cost would be $\$773$, displayed at the top. The goal is to find the lowest-cost cycle that visits all cities once and returns to A. Through experimentation, users gain insight into optimization strategies and develop an intuitive understanding of graph theory and combinatorial problems.

This activity not only deepens mathematical problem-solving skills but also introduces users to the complexities of real-world logistics, delivery planning, and network routing.

Student view

Overview (/study/app/math-ai-hl/sid-132-cid-761618/ov

1. By selecting edges at the top left, create the cycle ABCDEA. What is the total cost of this route?
2. By experimenting with different selections, what is the cheapest route you can find for the salesman? How do you know when you have found the cheapest route?

The simple situation presented in the activity above is known as the travelling salesman problem. It has far reaching significance in many areas of study and has fascinated many mathematicians for nearly two centuries (and possibly longer). An interesting presentation discussing some of the current applications of the problem can be found here (https://www.youtube.com/watch?v=5VjphFYQKj8).

As the number of cities increases, finding the optimal route that includes all of them becomes very difficult to solve as the number of possible routes becomes very large, as you saw earlier in the discussion of Hamiltonian cycles. For this course, you need to find only an upper and lower bound in which the optimal route lies.

The nearest neighbour algorithm

This algorithm uses the nearest adjacent vertex, which is an adjacent vertex connected with an edge with the lowest weight. The following steps in the nearest neighbour algorithm find an upper bound for the shortest Hamiltonian cycle within a graph:

1. Pick a vertex of the graph as a starting point. A question will sometimes tell you which vertex to begin with. If not, choose any vertex.
2. Go from the current vertex to the nearest adjacent vertex that has not already been visited.
3. Repeat Step 2 until all vertices have been reached.
4. Complete the cycle by travelling back to the starting vertex.
5. Calculate the upper bound by summing the weights of all edges included in the cycle.

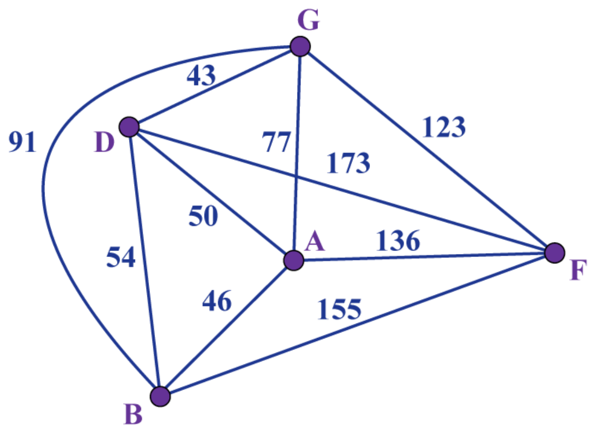
Example 1

★★☆

Student view

Enrique is planning a trip to Germany. He has friends in Arendsee, Beetzendorf, Dannenberg, Fehrbellin and Grabow. The graph in below represents the cities and the distances between them in kilometres.

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov



More information

The image is a graph representing several cities and the distances between them. The cities are labeled as A, B, D, F, and G. Distances are marked in kilometers on the lines connecting each pair of cities.

City connections and distances: - D to G is 43 km. - D to A is 50 km. - D to F is 123 km. - D to B is 54 km. - D to D is a loop of 91 km. - A to G is 77 km. - A to F is 136 km. - A to B is 46 km. - B to F is 155 km.

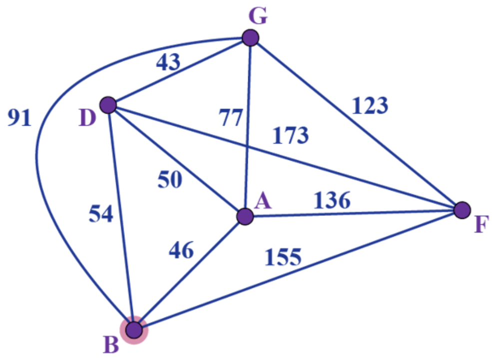
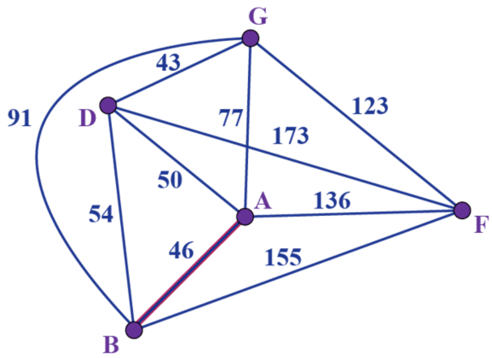
These connections form a network that Enrique could use to plan the shortest route visiting each city once, starting and ending in Beetzendorf.

[Generated by AI]

Enrique is planning to hire a rental car in Beetzendorf. Use the nearest neighbour algorithm to find an upper bound for the shortest trip that visits each of the towns only once and then returns to Beetzendorf.

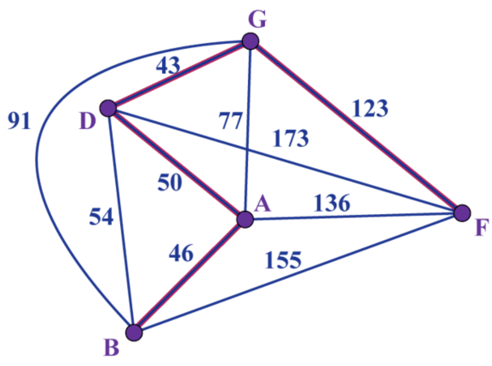
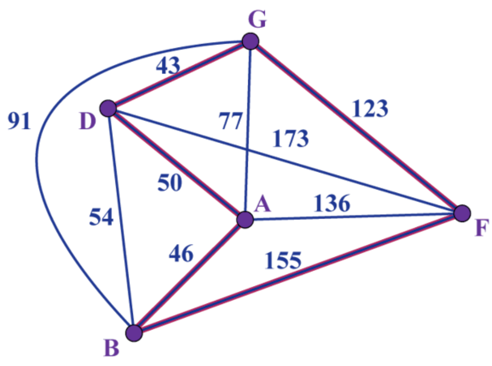
Student view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

Steps	Explanation
	Step 1: Choose a startir vertex. Since the problem states that Enrique will be renting car in Beetzendorf, the starting vertex is B.
	Step 2: From vertex B, begin the path by choosing the vertex adjacent to B with the lowest weight edge.

Student view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

Steps	Explanation
	Step 3: Continue choosing vertices by selecting the one with the lowest weight edge
	Step 4: Once all vertice have been reached, complete the cycle by returning to the startin vertex.
Upper bound: $46 + 50 + 43 + 123 + 155 = 417$	Find the upper bound f the shortest Hamiltonia cycle by adding the weights of the selected edges.

Student view

Important

Overview (/study/app/math-ai-hl/sid-132-cid-761618/ov

The upper bound found with the nearest neighbour algorithm will typically differ when using different starting vertices. When solving travelling salesman problems, we typically want the lowest possible upper bound. You may be required to calculate the upper bound multiple times using different starting vertices.

Deleted vertex algorithm

The following steps of the deleted vertex algorithm give a lower bound for the shortest Hamiltonian cycle within a graph:

1. Delete one vertex from the graph and all edges that are incident to the vertex. A problem may tell you which vertex to delete. If not, choose any of the vertices.
2. Calculate the length of the minimum spanning tree for the remaining vertices.
3. Create a subgraph by adding the two deleted edges with the lowest weight that connect the deleted vertex to the minimum spanning tree.
4. Find the lower bound by summing the weights of all the edges in the subgraph.

⚠ **Be aware**

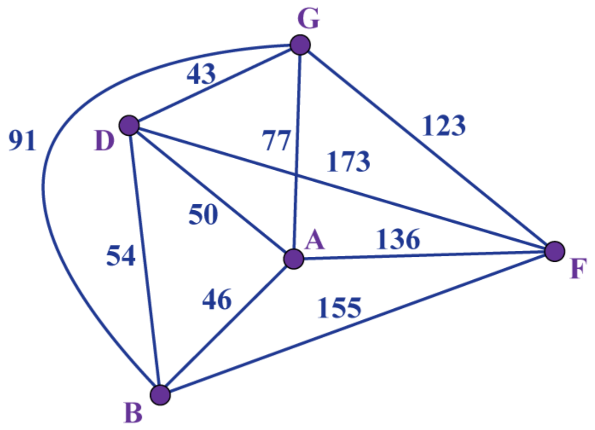
The deleted vertex algorithm will not always create a Hamiltonian cycle. Nevertheless, the sum of the weights of all edges in the resulting subgraph will be a lower bound for the minimal Hamiltonian cycle.

Example 2

★★☆

Student view

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov



More information

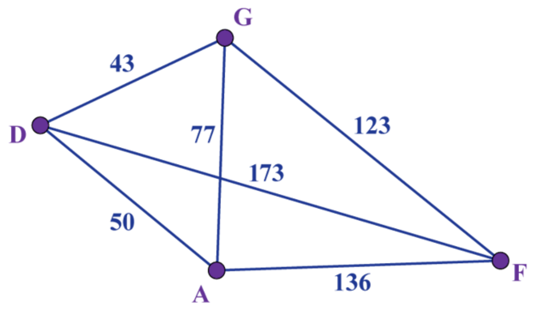
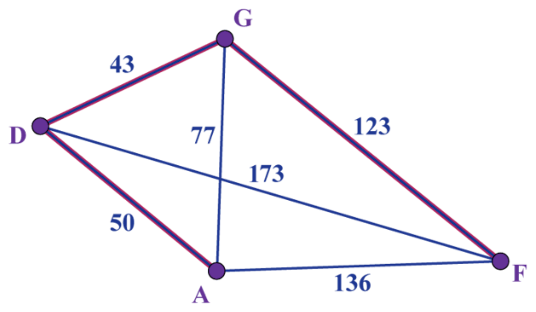
The diagram represents a network of five vertices labeled A, B, D, F, and G, connected by edges with numerical distances. The distances between the vertices are labeled on the connecting edges: between D and G is 43, D and B is 54, D and F is 123, B and F is 155, A and F is 136, D and A is 50, A and B is 46, and G and A is 173. There are also distances of 91 between D and B, and 77 between D and G. The diagram is used to identify paths and compute the shortest trip to visit each town once before returning to Beetzendorf, excluding one vertex, as per the deleted vertex algorithm.

[Generated by AI]

By deleting vertex B, use the deleted vertex algorithm to find a lower bound for the shortest trip that will allow Enrique to visit each of the towns once and then return to Beetzendorf.

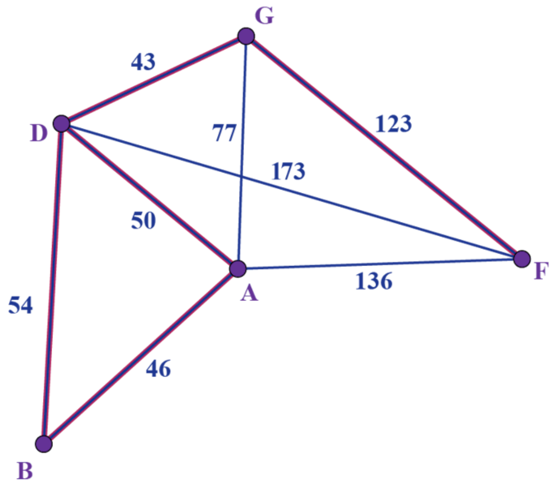
Student view

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

Steps	Explanation
	Step 1: Delete vertex B from the graph as instructed by the problem and all edges incident to vertex B.
	Step 2: Find the length of the minimum spanning tree for the remaining subgraph using Kruskal's algorith from section 3.16.2 (/study/app/math-ai-hl/sid-132-cid-761618/book/minimum-spanning-trees-id-28370/).

Student view

Overview
(/study/ap
ai-
hl/sid-
132-
cid-
761618/ov

Steps	Explanation
	Step 3: Add the two edges with the lowest weight that will connec vertex B to the minimu spanning tree.
Lower bound $: 46+54+50+43+123=316$	Find the lower bound fo the shortest Hamiltonia cycle by adding the weights of the selected edges.

From **Examples 1** and **2**, we can say that the shortest Hamiltonian cycle for Enrique's trip has a length greater than 316 km and less than 417 km.

Exam tip

There are two situations where you can state that you have found the exact length of the shortest Hamiltonian cycle for the graph:

- If the upper bound and lower bound are the same length.
- If you find a cycle with the same length as the lower bound.

3 section questions

Student
view

Checklist

What you should know

By the end of this subtopic you should be able to:

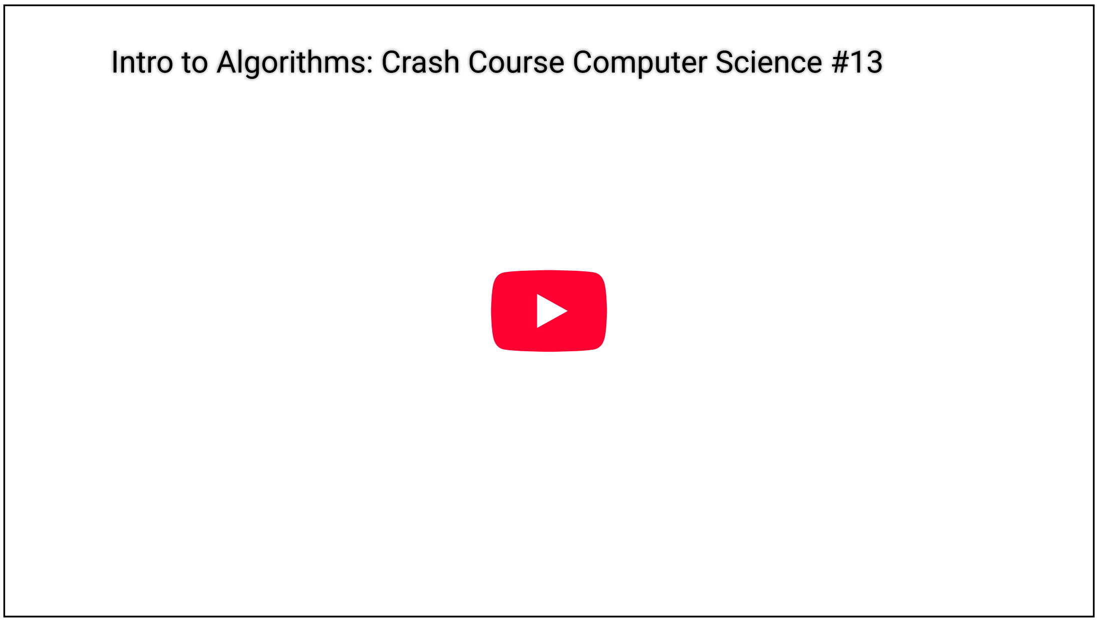
- define the following terms related to graph theory:
  - path
  - trail
  - cycle
  - circuit
  - connected graph
  - strongly connected graph
  - tree
  - Eulerian trails and circuits
  - Hamiltonian paths and cycles
- use Kruskal's algorithm to find the minimum spanning tree for a graph
- use Prim's algorithm to find the minimum spanning tree for a larger graph
- use Prim's algorithm on a weighted adjacency matrix without drawing a graph
- determine the shortest Chinese postman route around a weighted graph with up to four odd-degree vertices
- explain why the Chinese postman problem works and justify your choice of an algorithm based on the number of odd-degree vertices within the graph
- use the nearest neighbour algorithm to find an upper limit for the shortest Hamiltonian cycle within a graph
- use the deleted vertex algorithm to find a lower limit for the shortest Hamiltonian cycle within a graph.

Overview
(/study/app/math-ai-hl/sid-132-cid-761618/ov

Investigation

Section Student... (0/0) Feedback Print (/study/app/math-ai-hl/sid-132-cid-761618/book/investigation-id-28374/print/) Assign

In section 3.16.3 (/study/app/math-ai-hl/sid-132-cid-761618/book/chinese-postman-problem-id-28371/) and section 3.16.4 (/study/app/math-ai-hl/sid-132-cid-761618/book/travelling-salesman-problem-id-28372/) you learned about algorithms that allow you to travel around an entire graph with a minimum cost or distance. However, what if you just wanted to travel from one place to another? Consider this as you watch the following video.



After watching the video, visit the University of San Francisco's applet for Dijkstra's algorithm found here (https://www.cs.usfca.edu/~galles/visualization/Dijkstra.html). Experiment with various different graphs for each size until you fully understand how the algorithm works.

Now create your own graph with at least 10 vertices and explain how to find the shortest path from one vertex to another.

Student view