

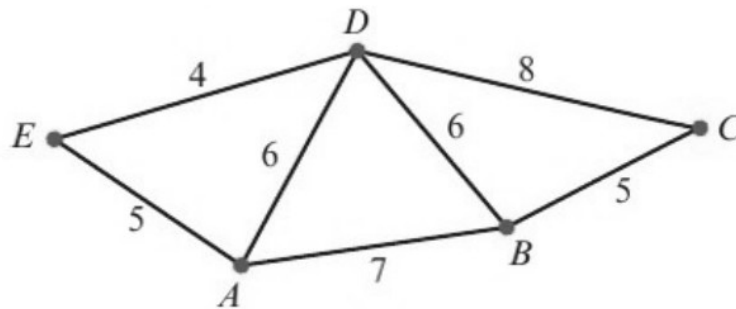
3A Kruskal's Algorithm

Minimum spanning tree (MST) / minimum connector:

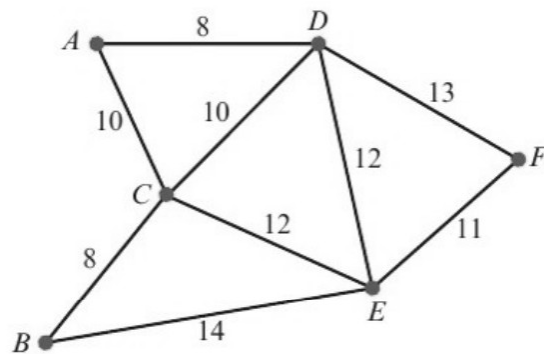
Kruskal's algorithm:

Number of arcs in a spanning tree of a network with n vertices:

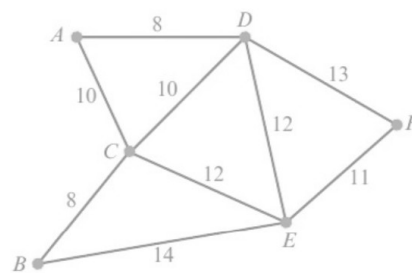
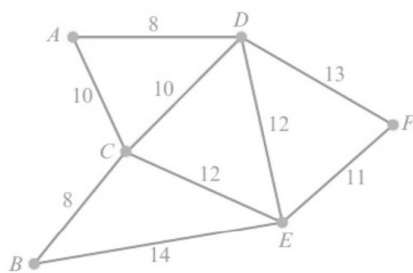
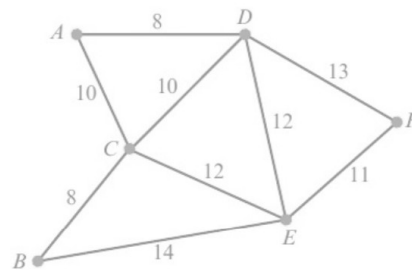
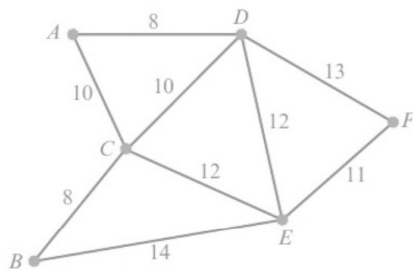
1. Use Kruskal's algorithm to find a minimum spanning tree for this network. List the arcs in the order that you consider them. State the weight of your tree.



2. Use Kruskal's algorithm and show that there are four minimum spanning trees for this network. State their weight.



Draw the four minimum spanning trees:

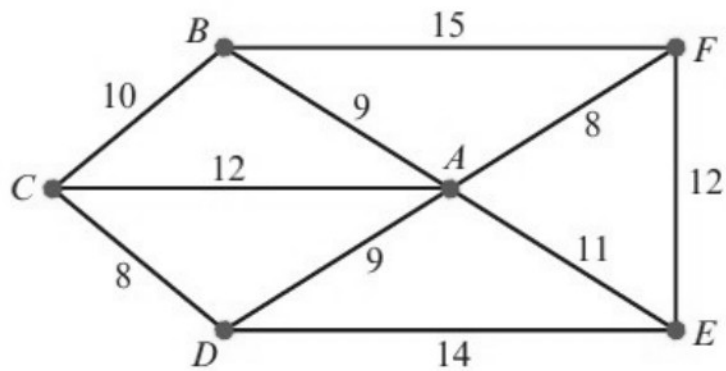


3B Prim's Algorithm

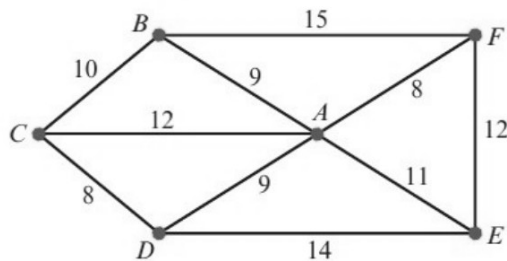
Prim's algorithm:

Differences between Prim's and Kruskal's algorithms:

1. Use Prim's algorithm to find a minimum spanning tree for the network below. List the arcs in the order in which you add them to your tree.



Draw your minimum spanning tree and state its weight:



3C Applying Prim's Algorithm to a Distance Matrix

Why use a distance matrix?

The distance matrix form of Prim's algorithm:

1. Apply Prim's algorithm to the distance matrix to find a minimum spanning tree. Start at A.

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>A</i>	–	27	12	23	74
<i>B</i>	27	–	47	15	71
<i>C</i>	12	47	–	28	87
<i>D</i>	23	15	28	–	75
<i>E</i>	74	71	87	75	–

Arcs in order of selection:

Draw the minimum spanning tree and state its weight:

2. Here is the distance matrix used in question 1.

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>A</i>	–	27	12	23	74
<i>B</i>	27	–	47	15	71
<i>C</i>	12	47	–	28	87
<i>D</i>	23	15	28	–	75
<i>E</i>	74	71	87	75	–

a) Work out the number of comparisons required to complete Prim's algorithm.

b) Work out the number of comparisons required for an $n \times n$ distance matrix.

c) State the order of this implementation of Prim's algorithm in terms of the number of vertices, n .

3D Using Dijkstra's Algorithm to Find the Shortest Path

Dijkstra's algorithm:

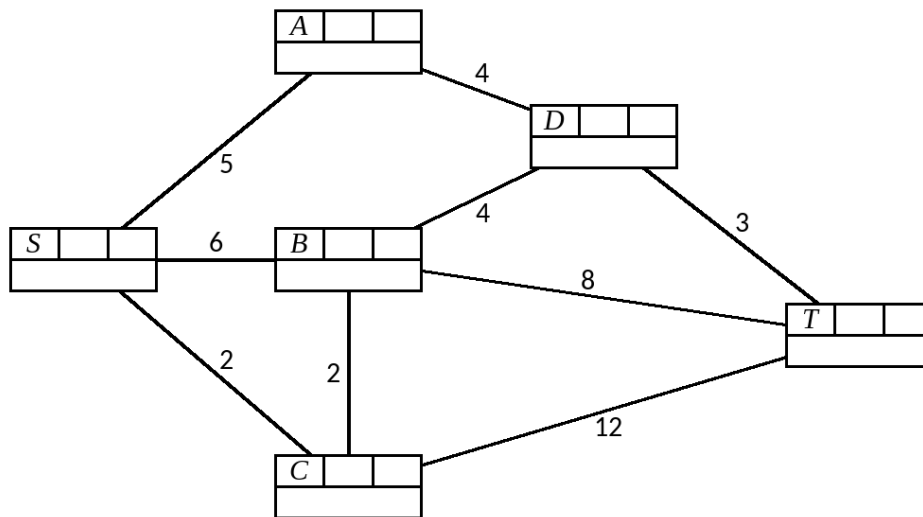
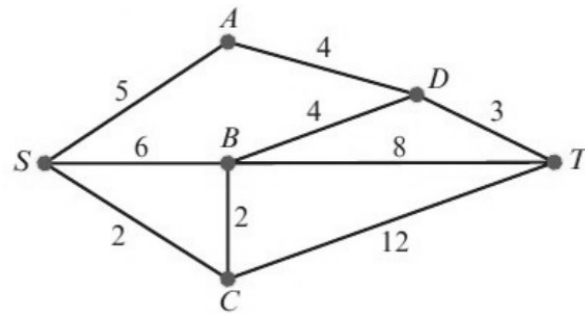
Final (permanent) label and working values (temporary labels):

To make the working clear, replace each vertex with a box like this:

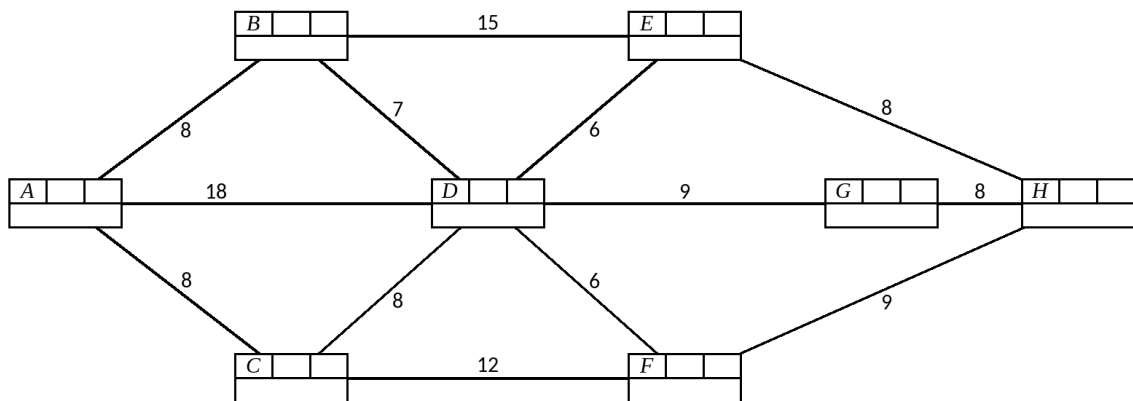
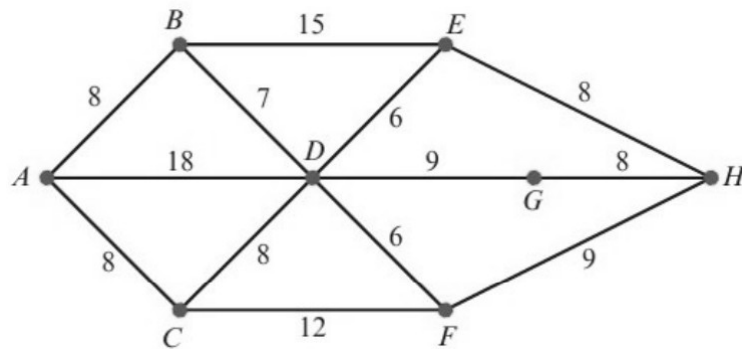
Vertex	Order of labelling	Final label
Working values		

Finding the shortest route (tracing back from T to S):

- Use Dijkstra's algorithm to find the shortest route from S to T in the network below.

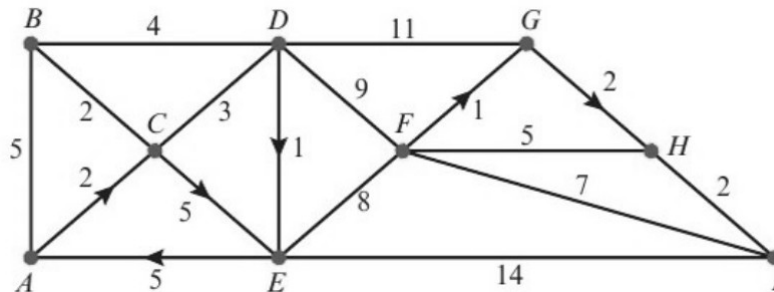


2. Use Dijkstra's algorithm in this network to find the length of the shortest route:
- from A to H
 - from A to G.
- List the routes you use.



Directed networks:

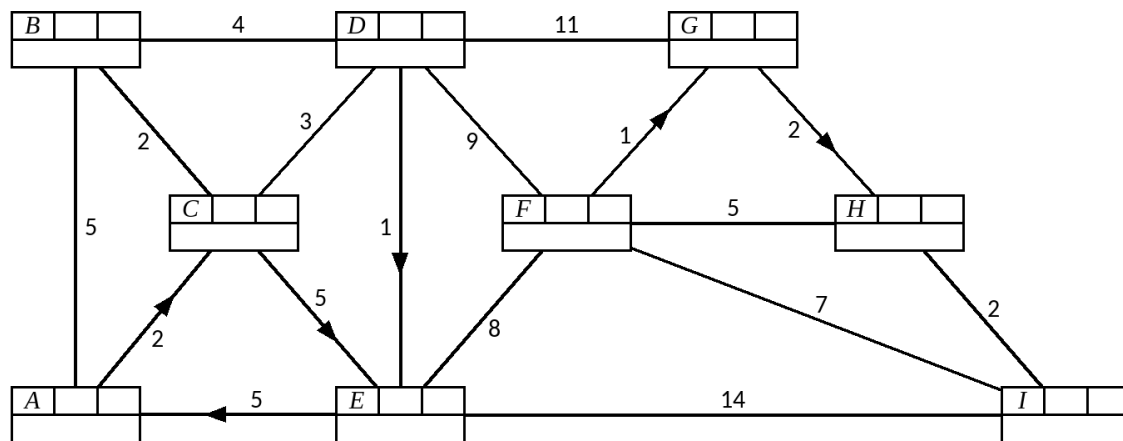
3. The network below represents part of a road system in a city. Some roads are one-way and these are indicated by directed arcs. The number on each arc represents the time, in minutes, to travel along that arc.



- a) Show that there are two quickest routes from A to I. Explain how you found your routes from your labelled diagram.

Road HI is closed due to roadworks.

- b) Find the quickest route from A to I, avoiding HI.



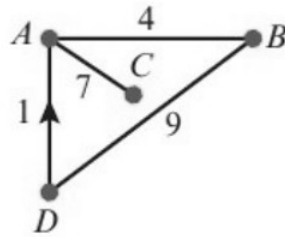
3E Floyd's Algorithm

A level only.

What does Floyd's algorithm find? (compare with Dijkstra's algorithm)

Floyd's algorithm:

1. This distance graph shows the direct distances, by road, between four towns A, B, C and D, in miles. The road from D to A is a one-way road as shown by the arrow.



- a) Use Floyd's algorithm to produce a table of shortest distances. You should give the distance table and route table for each iteration.

Initial tables:

Distance table

	A	B	C	D
A	-			
B		-		
C			-	
D				-

Route table

	A	B	C	D
A				
B				
C				
D				

First iteration:

Distance table

	A	B	C	D
A	-			
B		-		
C			-	
D				-

Route table

	A	B	C	D
A				
B				
C				
D				

Second iteration:

Distance table

	A	B	C	D
A	-			
B		-		
C			-	
D				-

Route table

	A	B	C	D
A				
B				
C				
D				

Third iteration:

Distance table

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>A</i>	-			
<i>B</i>		-		
<i>C</i>			-	
<i>D</i>				-

Route table

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>A</i>				
<i>B</i>				
<i>C</i>				
<i>D</i>				

Fourth iteration:

Distance table

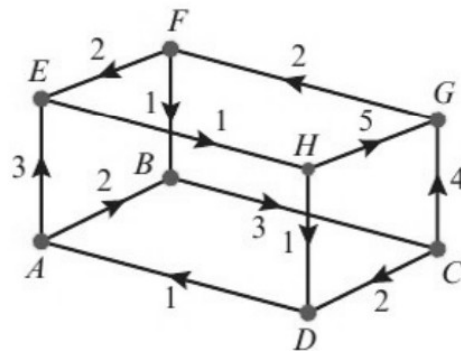
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>A</i>	-			
<i>B</i>		-		
<i>C</i>			-	
<i>D</i>				-

Route table

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>A</i>				
<i>B</i>				
<i>C</i>				
<i>D</i>				

- b) Find the route of minimum length from C to D.

2. 8 departure gates in an airport, which are linked by a system of travellers and escalators, are modelled using the directed network shown. The weight of each edge represents the travel time, in minutes, from one gate to another.



7 iterations of Floyd's algorithm are applied to the network, resulting in the following distance and route tables.

Distance table

	A	B	C	D	E	F	G	H
A	–	2	5	7	3	11	9	4
B	6	–	3	5	9	9	7	10
C	3	5	–	2	6	6	4	7
D	1	3	6	–	4	12	10	5
E	∞	∞	∞	∞	–	∞	∞	1
F	7	1	4	6	2	–	8	3
G	9	3	6	8	4	2	–	5
H	2	4	7	1	5	7	5	–

Route table

	A	B	C	D	E	F	G	H
A	A	B	B	C	E	G	C	E
B	D	B	C	C	D	G	C	E
C	D	D	C	D	D	G	G	E
D	A	A	B	D	A	G	C	E
E	A	B	C	D	E	F	G	H
F	D	B	B	C	E	F	C	E
G	F	F	F	F	F	F	G	F
H	D	D	D	D	D	G	G	H

- a) Apply the final iteration of Floyd's algorithm to give the final distance and route tables for the network.

Final iteration:

Distance table

	A	B	C	D	E	F	G	H
A	-							
B		-						
C			-					
D				-				
E					-			
F						-		
G							-	
H								-

Route table

	A	B	C	D	E	F	G	H
A								
B								
C								
D								
E								
F								
G								
H								

- b) Sarah needs to get from gate D to gate F. State the minimum time needed to make this journey, and determine the route she should take.