Tutorial 10

Paths and spanning trees

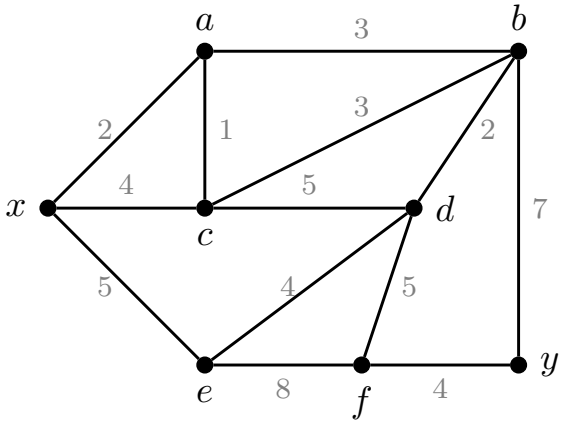
Question 1. Apply Dijkstra's Algorithm to each of the graphs below, using the specified starting and ending vertices.

1.

(a) start x, end y

(b) start a, end y

(c) start a, end e

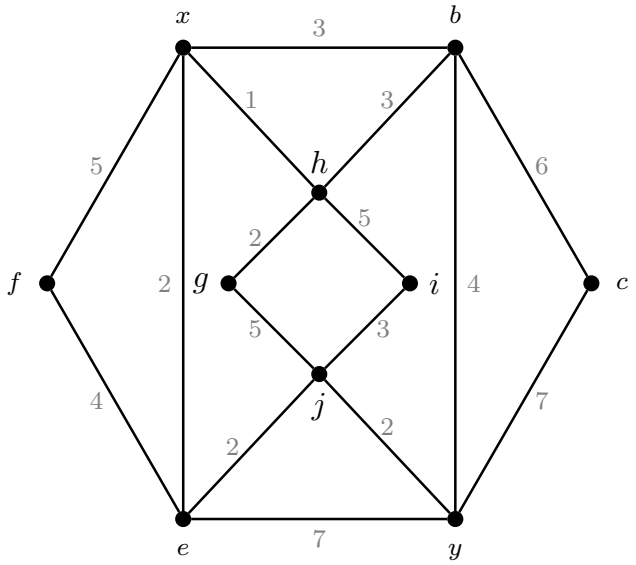


2.

(a) start x, end y

(b) start b, end y

(c) start f, end i

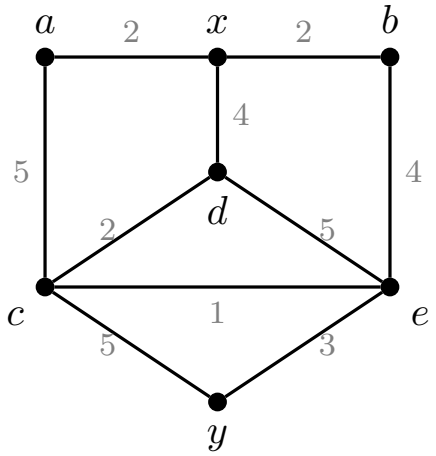


3.

(a) start x, end y

(b) start a, end e

(c) start b, end c

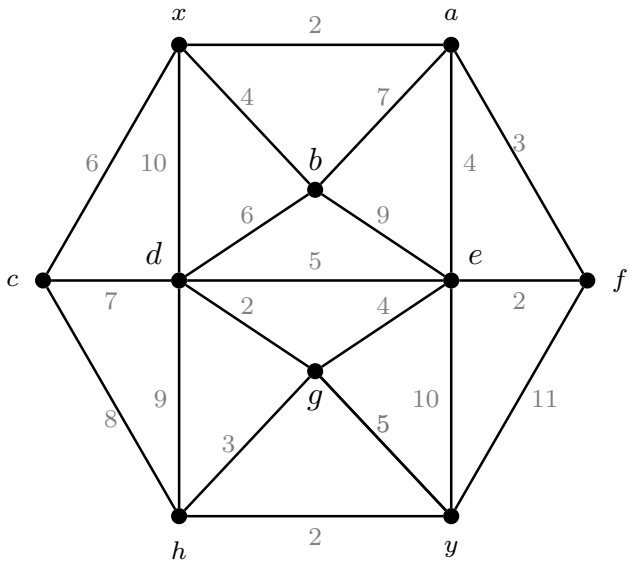


4.

(a) start x, end y

(b) start a, end h

(c) start b, end y

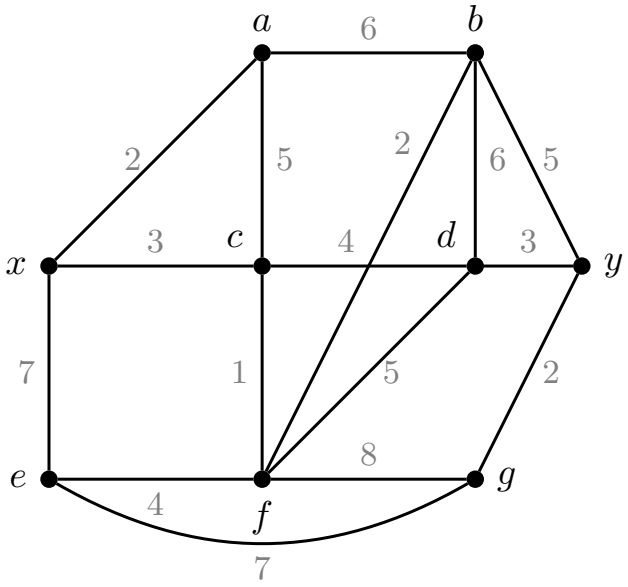


5.

(a) start x, end y

(b) start a, end g

(c) start b, end e

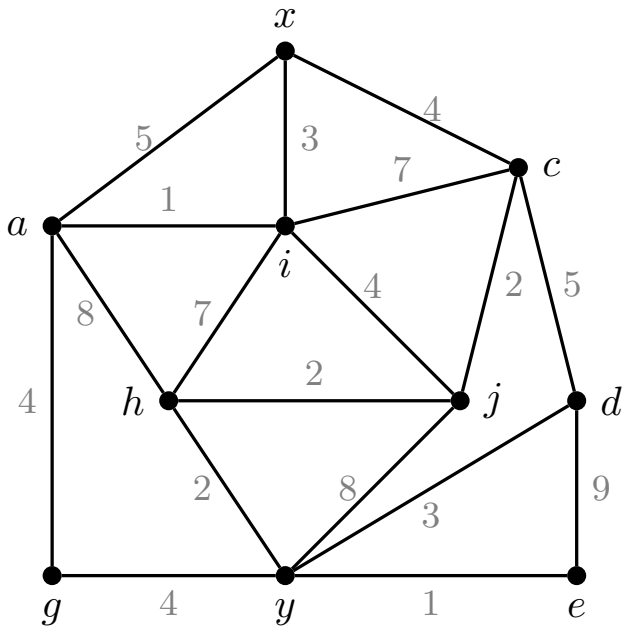


6.

(a) start x, end y

(b) start a, end d

(c) start g, end c

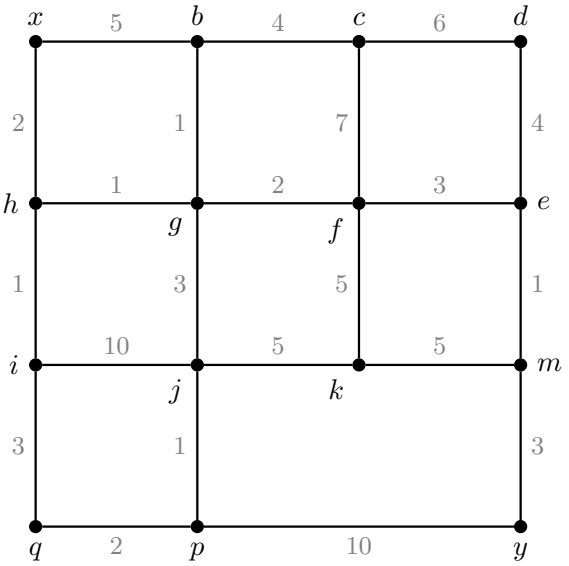


7.

(a) start x, end y

(b) start c, end p

(c) start d, end q

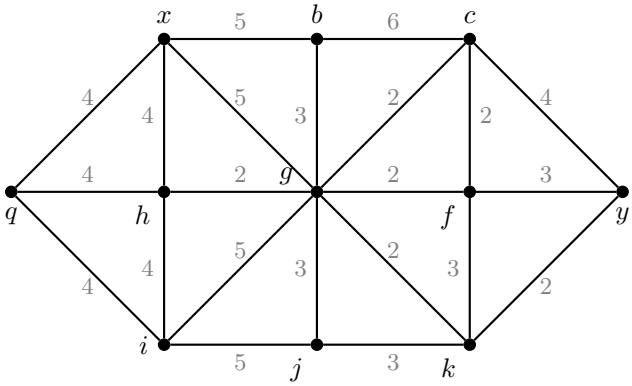


8.

(a) start x, end y

(b) start c, end q

(c) start i, end y

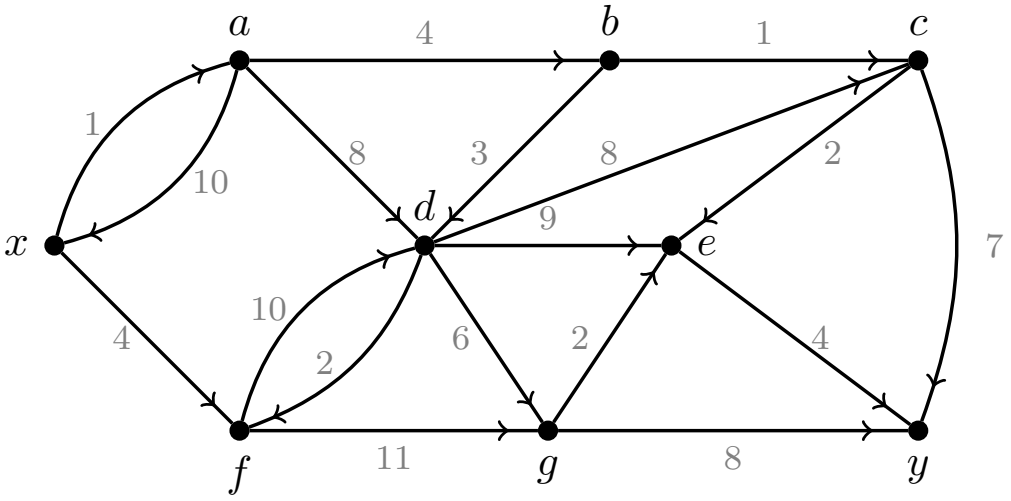


Question 2. Modify Dijkstra's Algorithm so that the output is the shortest path from the

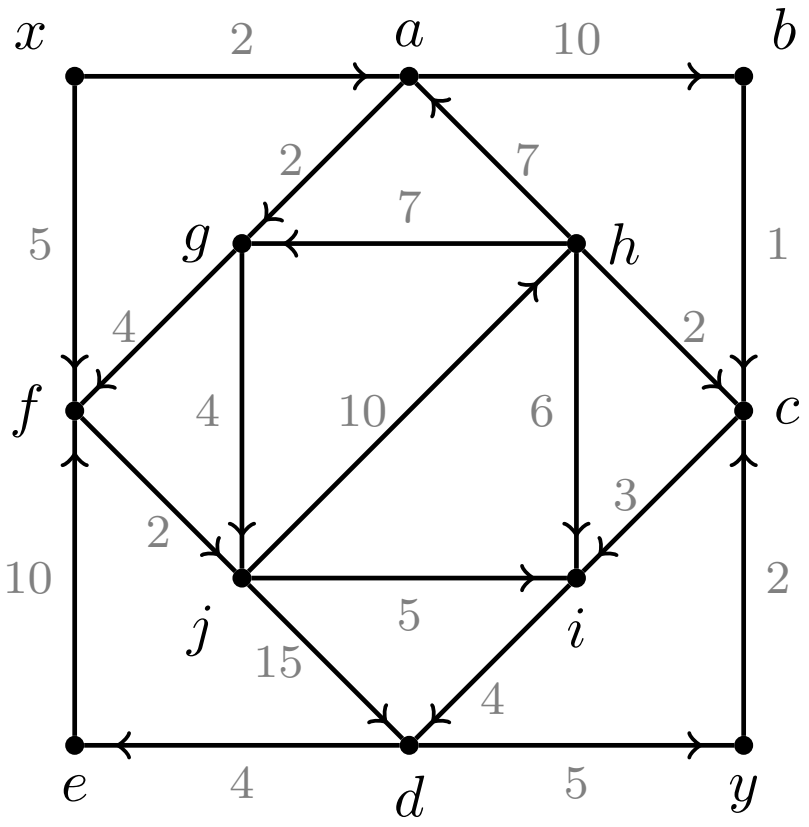
desired vertex to all other vertices in the graph.

Question 3. Apply Dijkstra's Algorithm to each of the directed graphs below, using the specified starting and ending vertices.

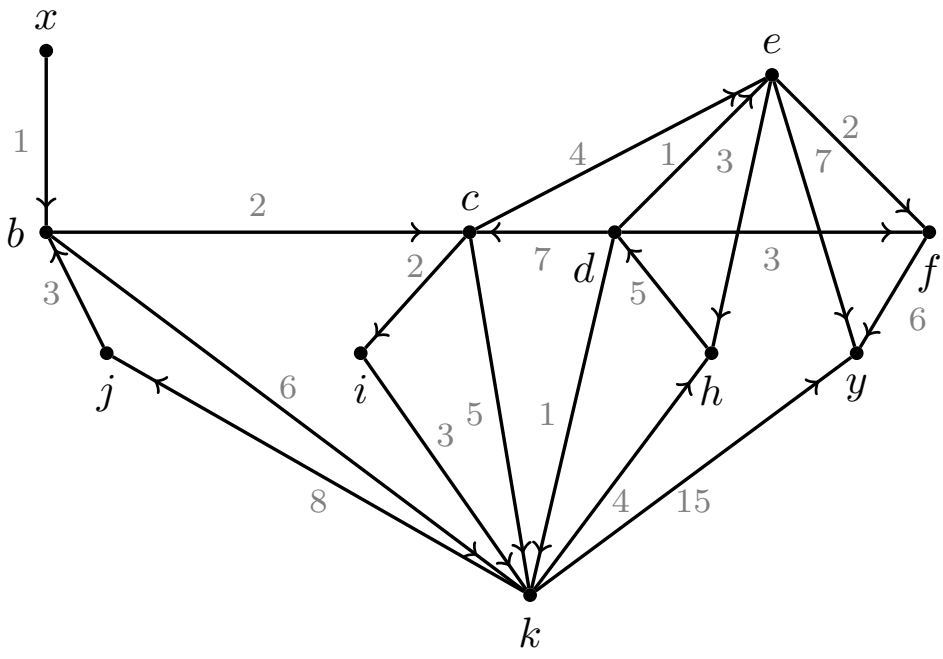
1. (a) start x, end y
   (b) start a, end y
   (c) start f, end c



2. (a) start x, end y
   (b) start a, end y
   (c) start e, end h
   (d) start y, end f

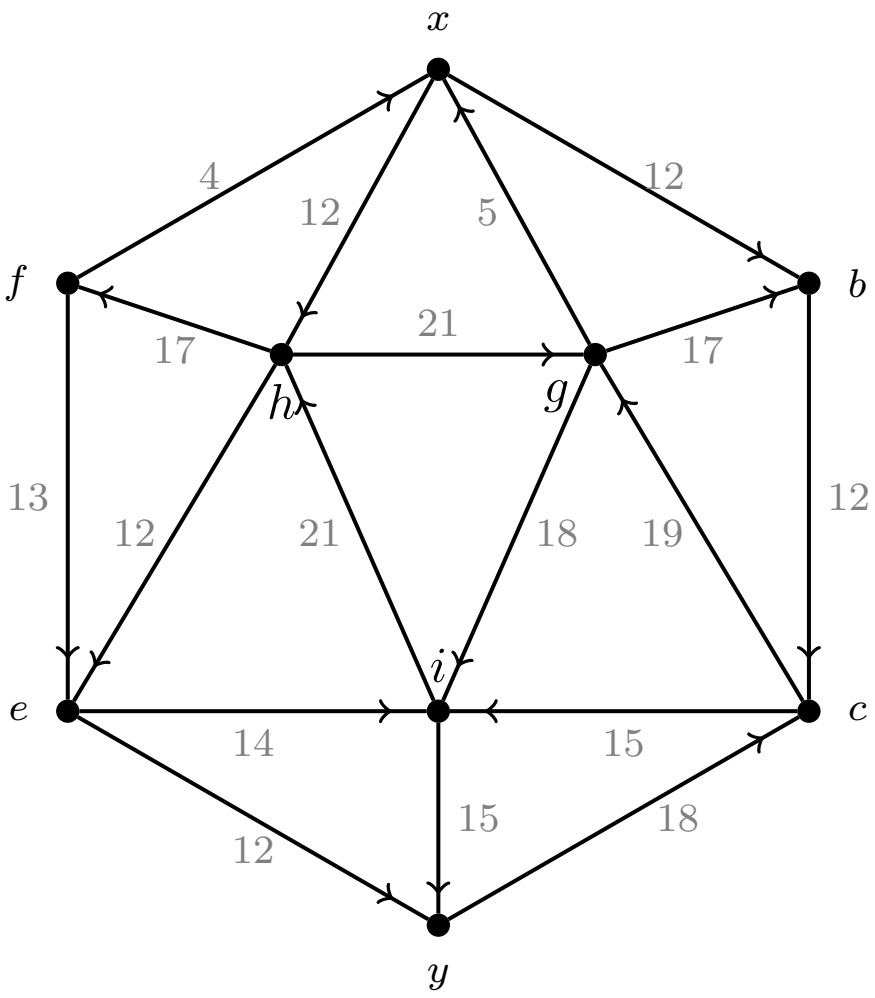


3. (a) start x, end y
   (b) start x, end h
   (c) start b, end f
   (d) start e, end i

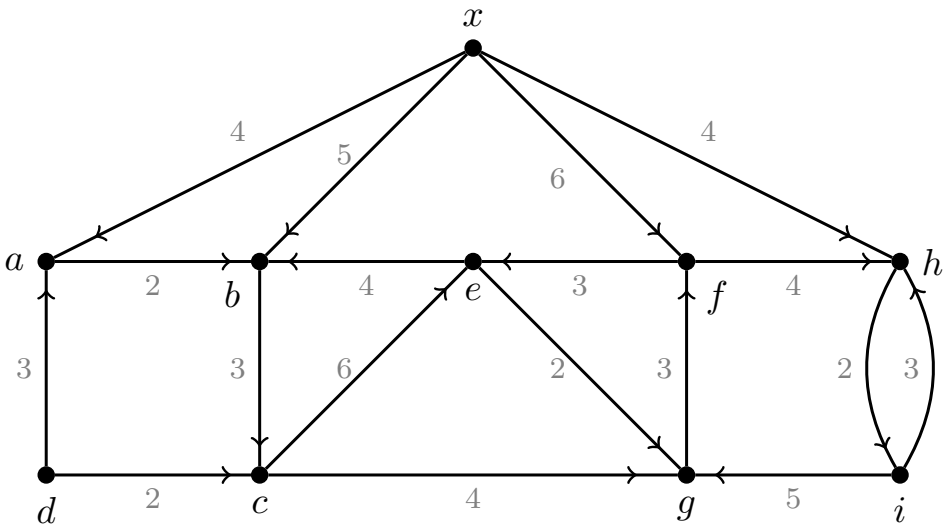


4. (a) start x, end y
   (b) start e, end b
   (c) start f, end c

(d) start f, end y



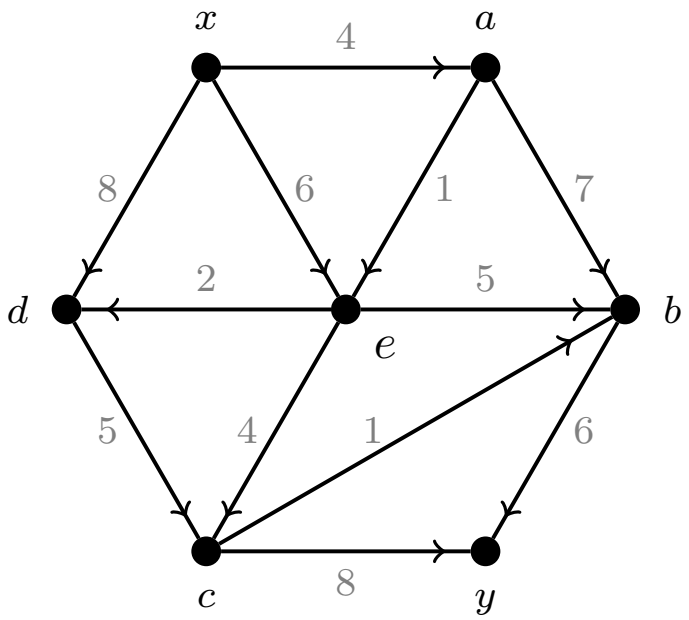
5. (a) start x, end i
   (b) start d, end h
   (c) start a, end g
   (d) start h, end c



6. (a) start x, end y

(b) start d, end y

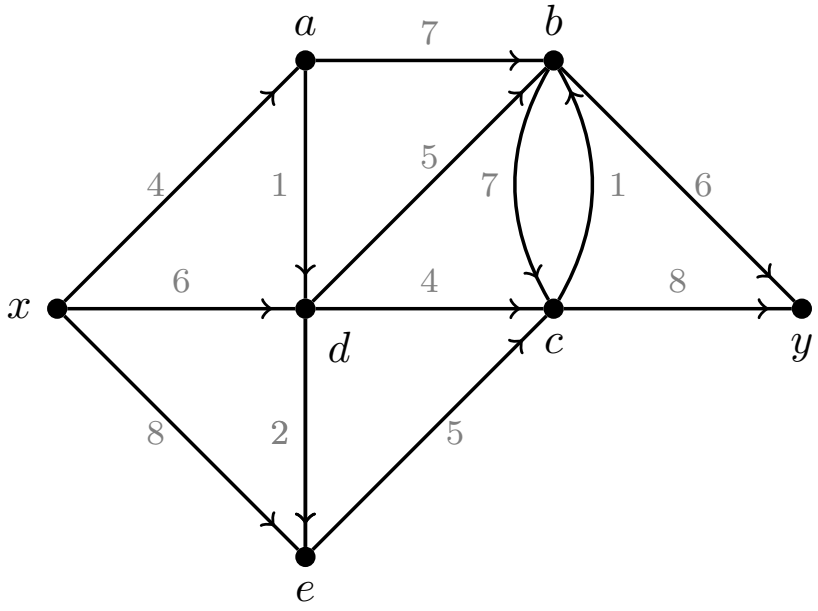
(c) start x, end b



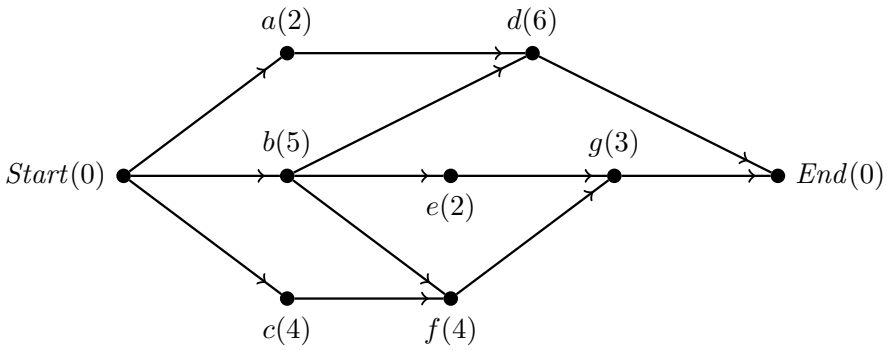
7. (a) start x, end y

(b) start x, end b

(c) start x, end c

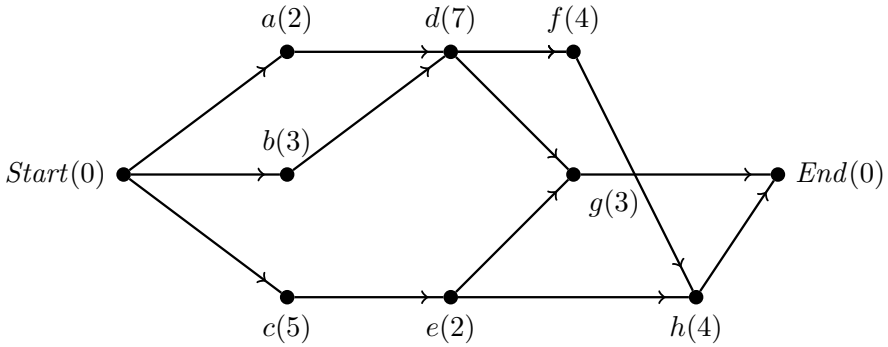


Question 4. Consider the project digraph shown below.



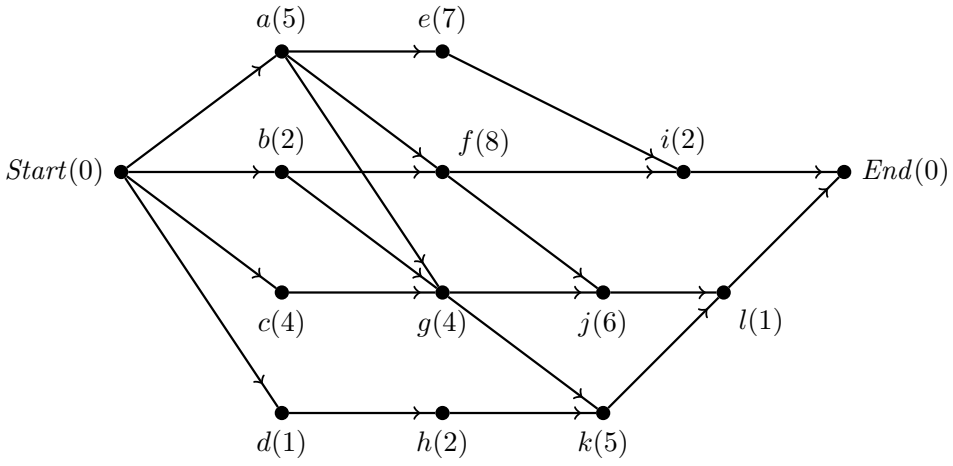
1. Use the Critical Path Algorithm to find a schedule with 2 processors.
2. Determine if the schedule is optimal. If not, find a better schedule using 2 processors.

Question 5. Consider the project digraph shown below.



1. Use the Critical Path Algorithm to find a schedule with 2 processors.
2. Determine if the schedule is optimal. If not, find a better schedule using 2 processors.

Question 6. Consider the project digraph shown below.



1. Use the Critical Path Algorithm to find a schedule with 2 processors.

2. Use the Critical Path Algorithm to find a schedule with 3 processors.
3. Determine if either schedule is optimal.

Question 7. The table below lists 9 tasks that comprise a project, as well as their processing times and precedence relationships.

Task	Processing Time	Precedence Relationships
a	2	
b	5	
c	1	a
d	2	a, b
e	4	b
f	6	c, e
g	7	d
h	6	e
i	2	f, g

1. Draw the project digraph.
2. Use the Critical Path Algorithm to find a schedule with 2 processors.
3. Use the Critical Path Algorithm to find a schedule with 3 processors.
4. Determine if either schedule is optimal.

Question 8. The table below lists 10 tasks that comprise a project, as well as their processing times and precedence relationships.

Task	Processing Time	Precedence Relationships
a	2	
b	4	
c	6	a
d	5	a
e	4	a, b
f	5	b
g	2	b
h	10	c
i	3	d, e, f
j	4	f, g

1. Draw the project digraph.
2. Use the Critical Path Algorithm to find a schedule with 2 processors.
3. Use the Critical Path Algorithm to find a schedule with 3 processors.
4. Determine if either schedule is optimal.

Question 9. The table below lists 13 tasks that comprise a project, as well as their processing times and precedence relationships.

Task	Processing Time	Precedence Relationships
a	2	
b	3	
c	1	
d	3	
e	4	a
f	6	b
g	4	c, d
h	4	e
i	3	e, f, g
j	2	e, f, g
k	11	d
l	2	h
m	1	i, j

1. Draw the project digraph.
2. Use the Critical Path Algorithm to find a schedule with 2 processors.
3. Use the Critical Path Algorithm to find a schedule with 3 processors.
4. Determine if either schedule is optimal.

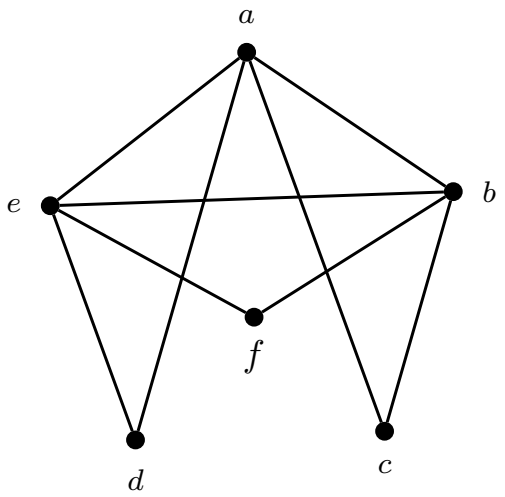
Question 10.

For each of the graphs described below, determine if G is (i) definitely a tree, (ii) definitely not a tree, or (iii) may or may not be a tree. Explain your answer or demonstrate with a proper graph.

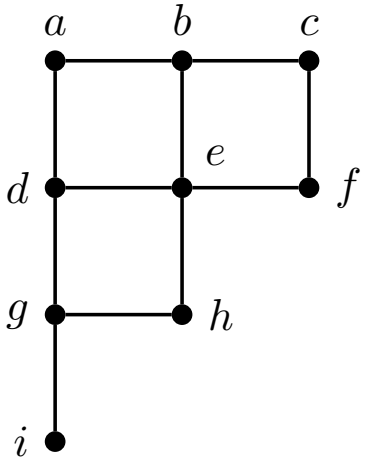
1. G has 10 vertices and 11 edges.
2. G has 10 vertices and 9 edges.
3. G is connected and every vertex has degree 1 or 2.
4. There is exactly one path between any two vertices of G.
5. G is connected with 15 vertices and 14 edges.
6. G is connected with 15 vertices and 20 edges.
7. G has two components, each with 9 vertices and 8 edges.

Question 11. Find a spanning tree for each of the graphs below.

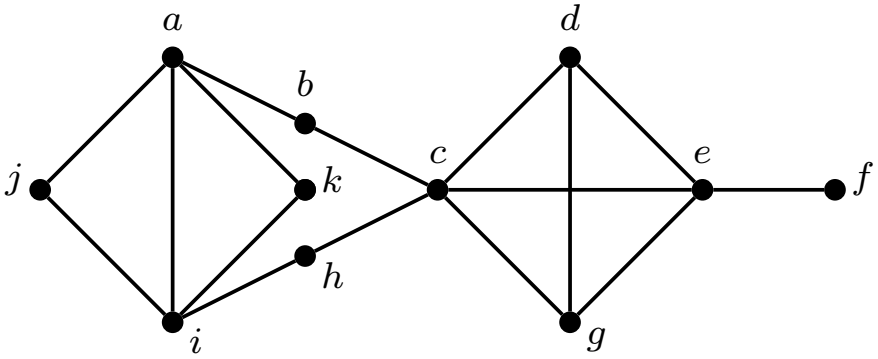
1.



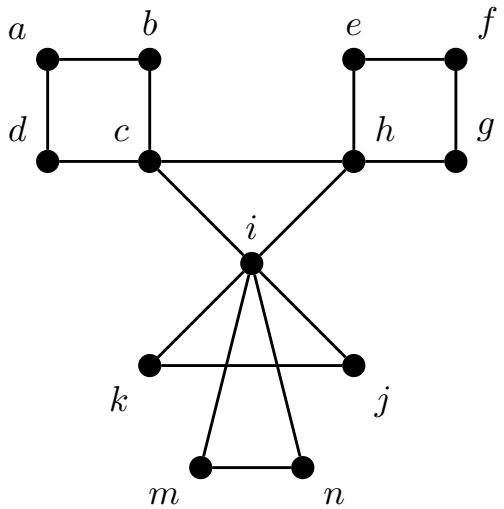
2.



3.



4.

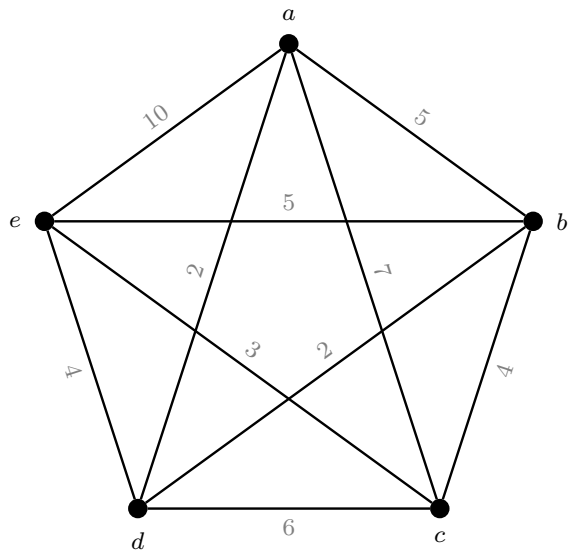


Question 12. Find a minimum spanning tree for each of the graphs below using

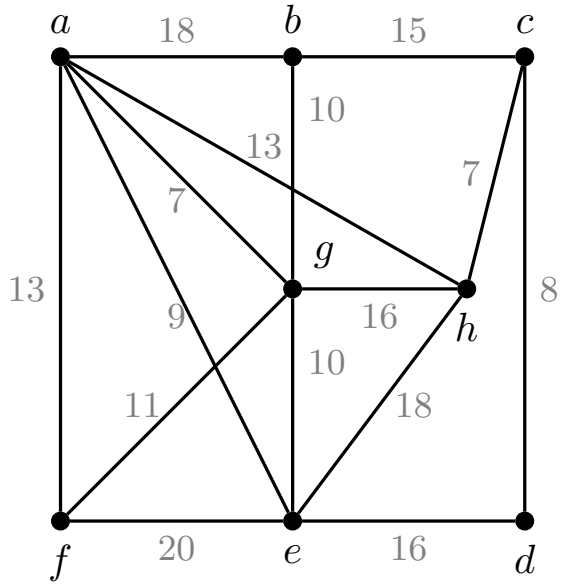
(i) Kruskal's Algorithm

(ii) Prim's Algorithm

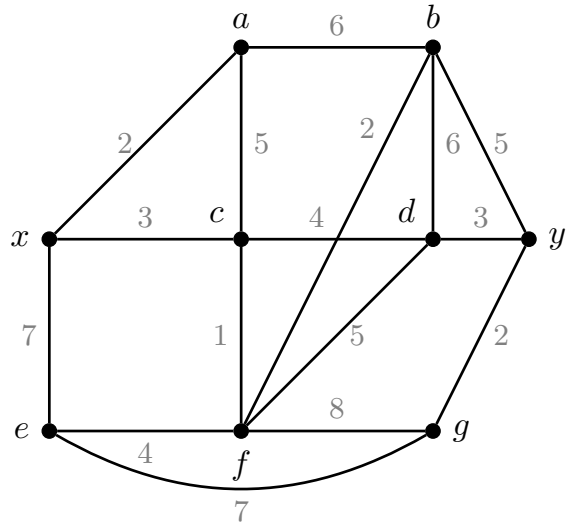
1.


a
b
c
d
e
10
5
5
2
1
4
3
2
4
6

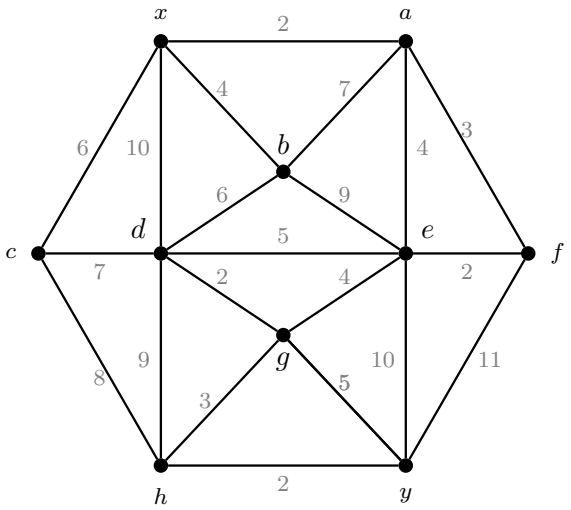
2.


a
18
b
15
c
10
13
7
7
g
13
8
9
16
h
10
11
18
f
20
e
16
d

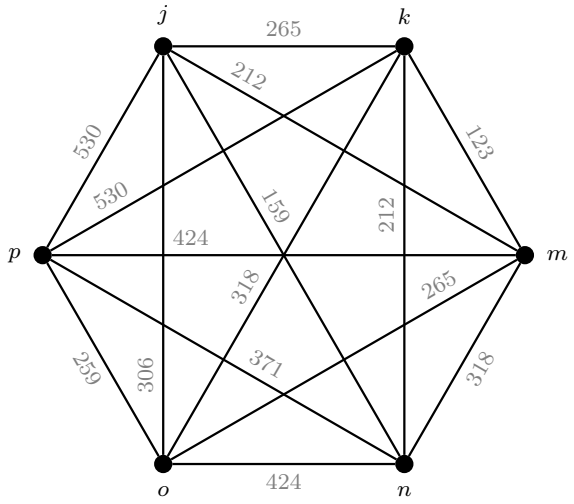
3.


a
6
b
2
5
2
6
5
x
3
c
4
d
3
y
7
1
5
2
8
e
4
f
g
7

4.


x
2
a
4
7
6
10
b
4
3
6
9
d
5
e
c
f
7
2
4
2
9
g
10
11
8
3
5
h
2
y

5.


j
265
k
212
530
123
530
159
212
p
424
m
318
265
259
306
371
318
o
424
n

Question 13. Find a minimum spanning tree for the graph represented by the table below.

	a	b	c	d	e	f	g
a	·	5	7	8	10	3	11
b	5	·	2	4	1	12	7
c	7	2	·	6	7	5	4
d	8	4	6	·	2	10	12
e	10	1	7	2	·	6	9
f	3	12	5	10	6	·	15
g	11	7	4	12	9	15	·

Question 14. Kruskal's Algorithm and Prim's Algorithm are both written with a connected graph as an input. Determine how each of these would perform if the input was a disconnected graph.

Question 15. How would you modify Kruskal's and Prim's algorithm if a specific edge must be included in the spanning tree? Would the resulting tree be a minimum spanning tree? Explain your answer.

Question 16. The Optos Cable Company is in the process of expanding its fiber optic network over the coming years. While this expansion will require the installation of new cable lines, the company aims to minimize overall costs. Given that the cost per meter of cable is fixed, it is essential to construct the network using the shortest total length of cable possible.

Construction has already begun: the first two phases of the expansion are complete, with cables laid between Quechee and Vinton, and between Vinton and Tempe. However, due to changes in other infrastructure projects, the required cable lengths between certain cities have been updated.

Using the already-installed connections and the revised cable lengths provided below, determine the most efficient way to complete the network expansion. Your goal is to minimize the total cable length required to connect all cities. Calculate the total length of cable that will be needed to complete the network.

	Mesa	Natick	Quechee	Rutland	Tempe	Vinton
Mesa	·	29	32	25	34	45
Natick	29	·	50	46	45	51
Quechee	32	50	·	41	28	19
Rutland	25	46	41	·	40	35
Tempe	34	45	28	40	·	15
Vinton	45	51	19	35	15	·

Question 17. The Reverse Delete Algorithm finds a minimum spanning tree by deleting the largest weighted edges as long as you do not disconnect the graph. In essence, it is Kruskal's Algorithm in reverse. Apply the Reverse Delete Algorithm to the graphs in Question 12.

Question 18. Under what circumstances would Reverse Delete be a better choice than Kruskal's Algorithm? Under what circumstance would Kruskal's be a better choice? Explain your answer.

Question 19. We studied two different algorithms for finding a minimum spanning tree. Both Kruskal and Prim cited the work of the Czech mathematician Otakar Borůvka, who is credited with the first minimum spanning tree algorithm from 1926. Below is a description of his algorithm.

Borůvka's Algorithm

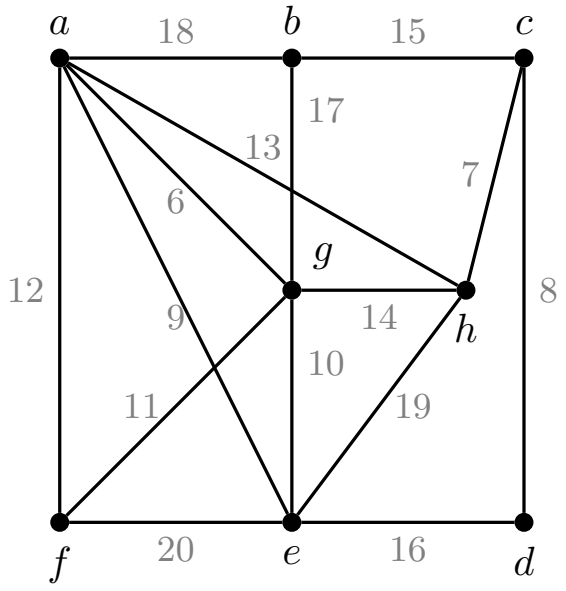
Input: Weighted connected graph $G = (V, E)$ where all the weights are distinct.
Steps:

1. Let T be the forest where each component consists of a single vertex.
2. For each vertex v of G, add the edge of least weight incident to v to T.
3. If T is connected, then it is a minimum spanning tree for G. Otherwise, for each component C of T, find the edge of least weight from a vertex in C to a vertex not in C. Add the edge to T.
4. Repeat Step (3) until T has only one component, making T a tree.

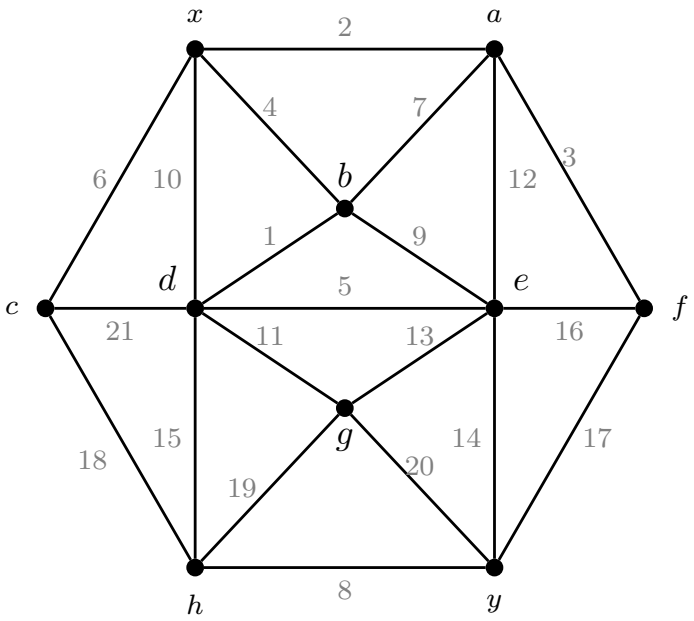
Output: A minimum spanning tree for G.

Apply Borůvka's Algorithm to the following two graphs. Use either Kruskal's Algorithm or Prim's Algorithm to verify that Borůvka's Algorithm found a minimum spanning tree.

1.



2.



Enrichment Questions

Question 20. You have decided to start a small business that manufactures a product. You are trying to determine the number of employees you will need, as well as how long it will take to produce one item. Write a report where Critical Paths are used to answer these questions. To complete the report, you will need to:

1. Name your business and the product.
2. Determine the different stages of production and their precedence relationships.
3. Draw the project digraph.
4. Include at least 10 different tasks.
5. Ensure at least 2 tasks must rely on multiple tasks.
6. Form schedules using the Critical Path Algorithm using 2, 3, and 4 processors.

Question 21. You are starting a new business that will serve customers in 7 different cities. You need to determine the distance from your hub location to each of your customers.

1. Choose 7 cities from the table on the next page and choose one to be your hub location.
2. Use the modification of Dijkstra's Algorithm from Problem 2 to minimize the mileage from your hub location to each of the customer locations.

City (Acronym)	NYC	LAX	CHI	HOU	PHI	PHX	SAT	SND	DAL	SJC	AUS	JAX	SFO
New York (NYC)	*	2448	712	1419	81	2142	1583	2431	1372	2552	1513	836	2569
Los Angeles (LAX)	2448	*	1744	1372	2391	357	1203	112	1239	306	1226	2146	348
Chicago (CHI)	712	1744	*	943	664	1453	1054	1733	806	1840	981	865	1857
Houston (HOU)	1419	1372	943	*	1341	1015	189	1303	225	1610	146	822	1644
Philadelphia (PHI)	81	2391	664	1341	*	2080	1507	2370	1299	2502	1437	759	2520
Phoenix (PHX)	2142	357	1453	1015	2080	*	848	299	886	615	869	1793	654
San Antonio (SAT)	1583	1203	1054	189	1507	848	*	1128	253	1453	74	1011	1489
San Diego (SND)	2431	112	1733	1303	2370	299	1128	*	1183	417	1156	2090	459
Dallas (DAL)	1372	1239	806	225	1299	886	253	1183	*	1451	182	908	1483
San Jose (SJC)	2552	306	1840	1610	2502	615	1453	417	1451	*	1466	2344	42
Austin (AUS)	1513	1226	981	146	1437	869	74	1156	182	1466	*	960	1502
Jacksonville (JAX)	836	2146	865	822	759	1793	1011	2090	908	2344	960	*	2373
San Francisco (SFO)	2569	348	1857	1644	2520	654	1489	459	1483	42	1502	2373	*

Table 1: Distance matrix between major U.S. cities