Greedy method

By

Prof. Shaik Naseera
Department of CSE
JNTUA College of Engg., Kalikiri

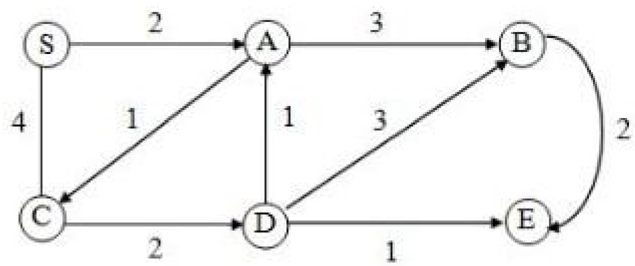
Objectives

- Activity selection problem
- Minimum Cost Spanning Tree
- Previous Gate Questions

Q. No. 4

If we run disjkstra algorithm on source vertex (S) to the following graph then which of the following is possible order of visiting nodes?

I. S, A, C, D, E, B II. S, A, D, C, E, B III. S, A, D, E, C, B



(A) only I, III (B) only II, III

(C) only I, II (D) I, II, III

{S, A, C, D, B, E}

S	A	B	C	D	E
$\boxed{0}$	∞	∞	∞	∞	∞
	$\boxed{2}$	∞	4	∞	∞
		5	$\boxed{3}$	∞	∞
		5		$\boxed{5}$	∞
		$\boxed{5}$			6
					$\boxed{6}$

{S, A, C, B, D, E}.

Answer : None of the above

Greedy Method (Recap)

- Greedy method suggests that one can devise an algorithm that works in stages, considering one input at a time.
- At each stage, a decision is made regarding whether a particular input is in an optimal solution.
- If the inclusion of the next input into the partially constructed optimal solution will result in an infeasible solution, then this input is not added otherwise it is added.
- The selection procedure is based on some optimization measure (objective function)

Activity Selection Problem

- *You are given n activities with their start and finish times. Select the maximum number of activities that can be performed by a single person, assuming that a person can only work on a single activity at a time.*
- Example

  Consider the following 6 activities.

  start[] = {1, 3, 0, 5, 8, 5};

  finish[] = {2, 4, 6, 7, 9, 9};

Solution:

The maximum set of activities that can be executed by a single person is {0, 1, 3, 4}

Activity-Selection Problem

For a set of proposed activities that wish to use a lecture hall, select a maximum-size subset of “compatible activities”.

- Set of activities: $S=\{a_1,a_2,\ldots a_n\}$
- Duration of activity a_i: $[start\_time_i, finish\_time_i)$
- Activities ***sorted in increasing order of finish time***:

i	1	2	3	4	5	6	7	8	9	10	11
$start\_time_i$	1	3	0	5	3	5	6	8	8	2	12
$finish\_time_i$	4	5	6	7	8	9	10	11	12	13	14

The solution is x1,x4,x8,x11

- The greedy choice is to always pick the next activity whose finish time is least among the remaining activities and the start time is more than the finish time of previously selected activity.
- We can sort the activities according to their finishing time so that we always consider the next activity as minimum finishing time activity.

procedure

1. Sort the activities according to their finishing time
2. Select the first activity from the sorted array and print it.
3. Do the following for remaining activities in the sorted array.
   a) If the start time of this activity is greater than the finish time of previously selected activity then select this activity and print it.

Implementation

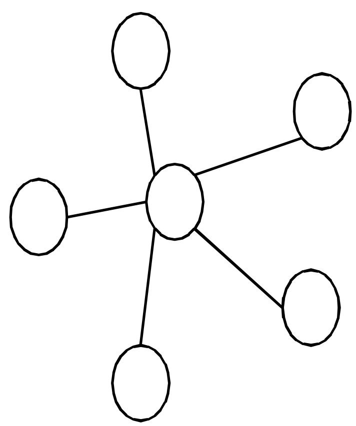
- activity selection problem.docx

Minimum Cost Spanning Tree

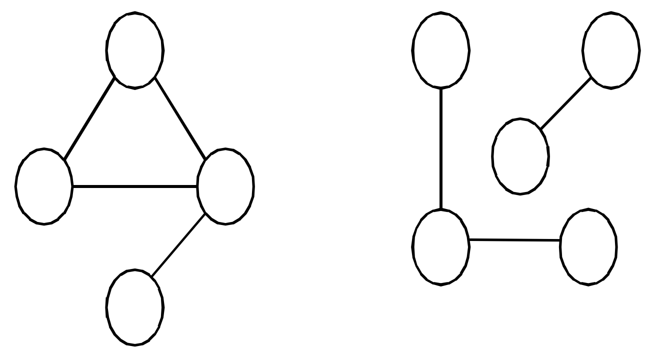
Tree

A tree is a graph with the following properties:

- The graph is connected (Each vertex is reachable)
- There are no cycles(acyclic)



Tree



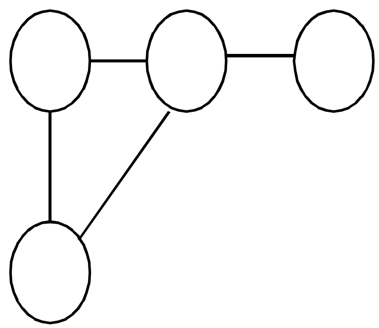
Graphs that are not trees

Minimum Spanning Tree (MST)

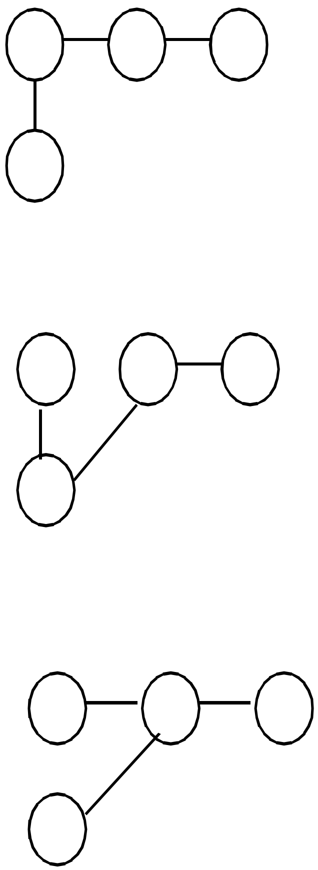
- Let G=(V,E) be an undirected connected graph.
- A sub graph T=(V, E) of G is a spanning tree of G if
  - T is a tree (i.e., it is acyclic)
  - T covers all the vertices ***V***
    - contains **|*V*| - *1*** edges
  - T has minimum total weight
  - A single graph can have many different spanning trees.

A Graph with different Spanning Trees

Connected undirected graph

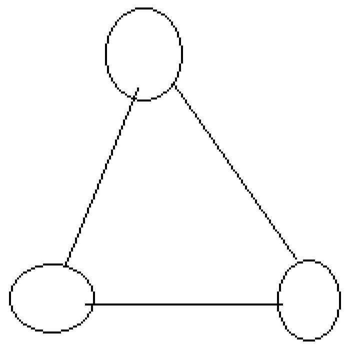


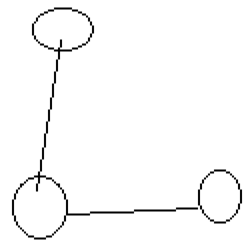
Spanning Tree

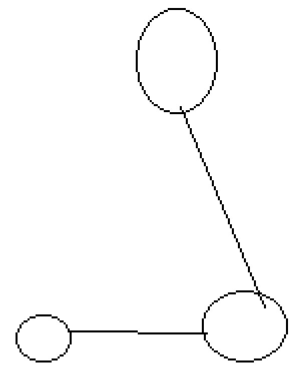


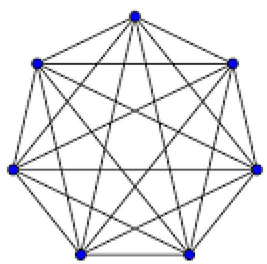
Complete Graph

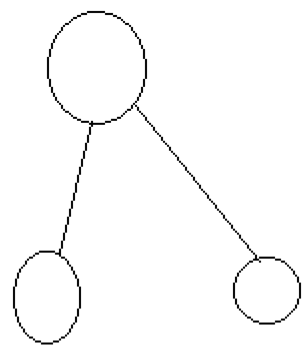
Total **number** of **Spanning Trees in a Graph**. If a **graph** is a complete **graph** with n vertices, then total **number** of **spanning trees** is $n^{(n-2)}$ where n is the **number** of nodes in the **graph**.



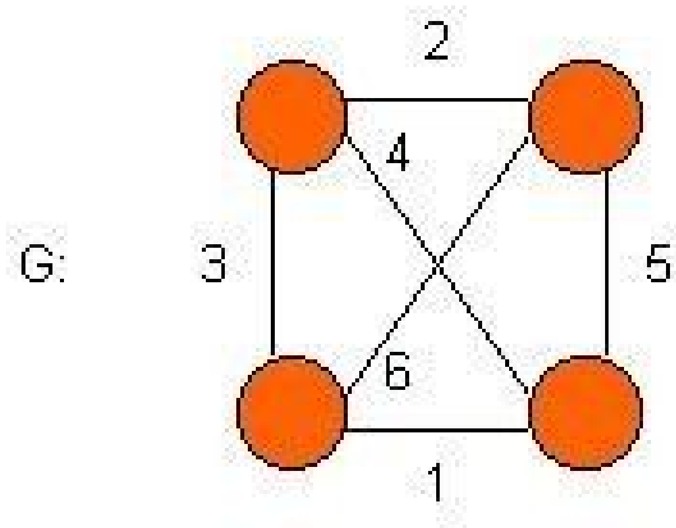




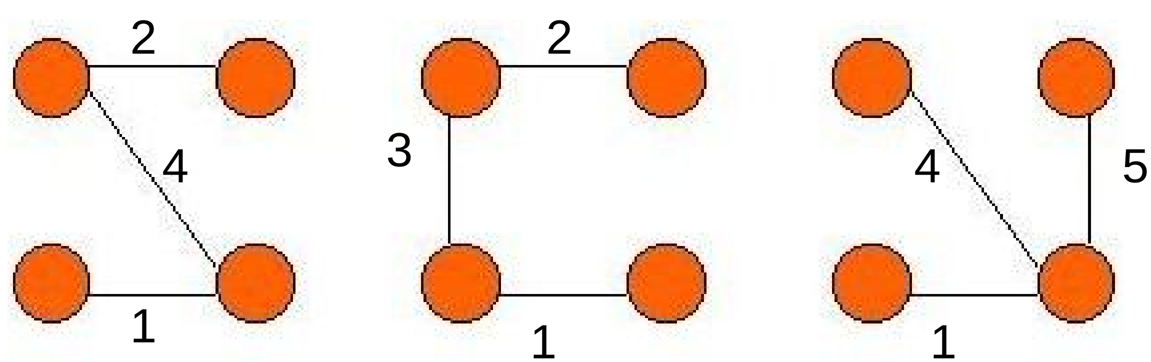




Weighted Graph graph G



Spanning Tree from Graph G



Minimum Cost Spanning Tree

1. Kruskal's Algorithm
2. Prim's Algorithm

Kruskal's Algorithm

- kruskals algorithm.pptx

Prim’s Algorithm

- The algorithm was discovered in 1930 by mathematician Vojtech Jarnik and later independently by computer scientist Robert C. Prim in 1957.
- The algorithm continuously increases the size of a tree starting with a single vertex until it spans all the vertices.
- Prim's algorithm is faster on dense graphs.

Introduction

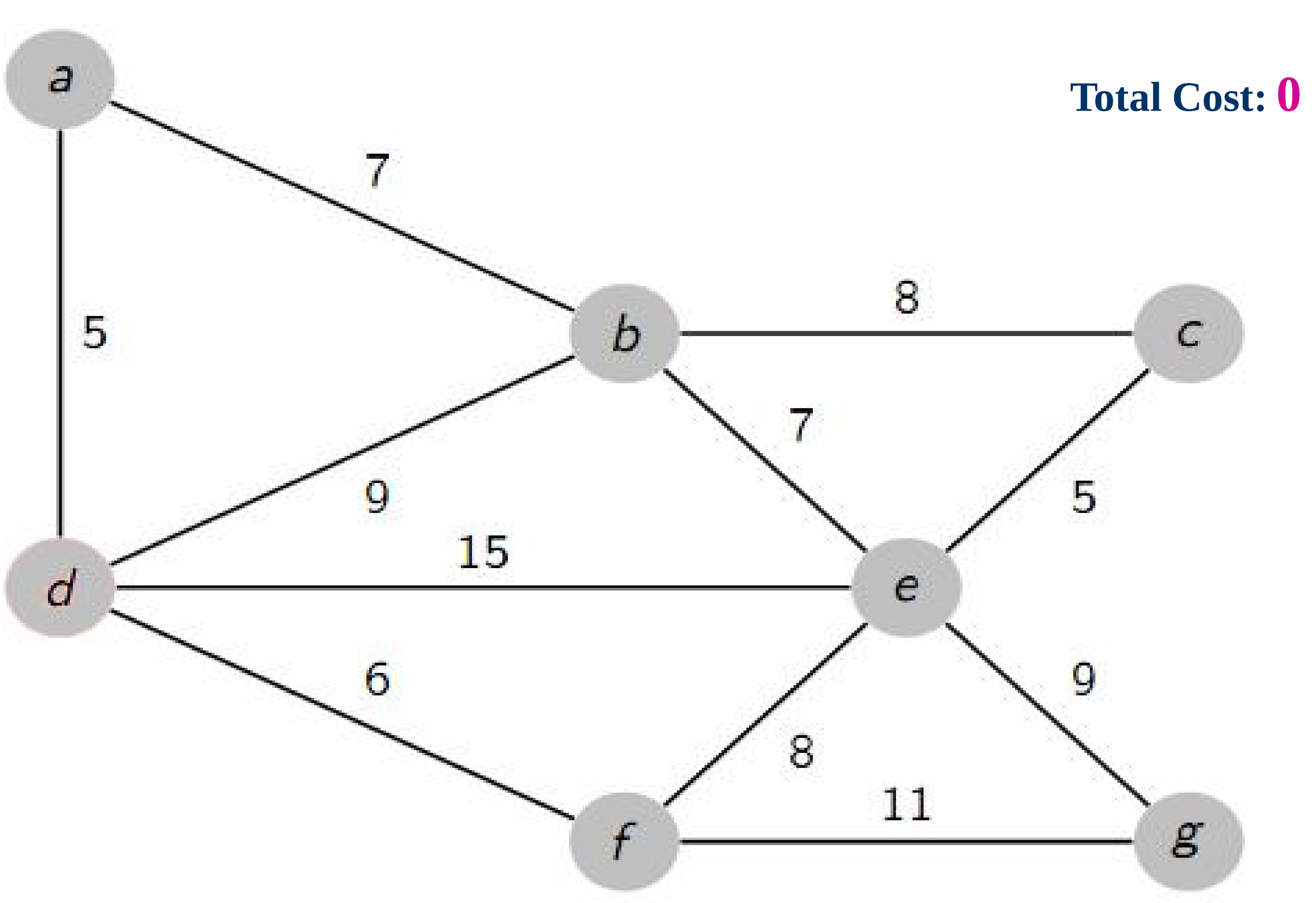
- Prim's algorithm for finding a minimal spanning tree parallels closely the depth- and breadth-first traversal algorithms.
- Just as these algorithms maintained a closed list of nodes and the paths leading to them, Prim's algorithm maintains a closed list of nodes and the edges that link them into the minimal spanning tree.
- Whereas the depth-first algorithm used a stack as its data structure to maintain the list of open nodes and the breadth-first traversal used a queue, Prim's uses a priority queue.

Procedure: Prim’s Algorithm

1. Randomly choose any vertex. The vertex connecting to the edge having least weight is usually selected and is added to the partially constructed spanning tree.
2. Find all the edges that connect the tree to new vertices.
3. Find the least weight edge among those edges and include it in to the added to the partially constructed spanning tree.
4. If including that edge creates a cycle, then reject that edge and look for the next least cost edge.
5. Keep repeating step-02 until all the vertices are included and Minimum Spanning Tree (MST) is obtained.

Implementation

- At first we declare an array named: closed list.
- And consider the **open list** as a priority queue with min-heap.
- Adding a node and its edge to the closed list (partially constructed spanning list) indicates that we have found an edge that links the node into the minimal spanning tree.
- As a node is added to the closed list, its successors (immediately adjacent nodes) are examined and added to a priority queue of open nodes.

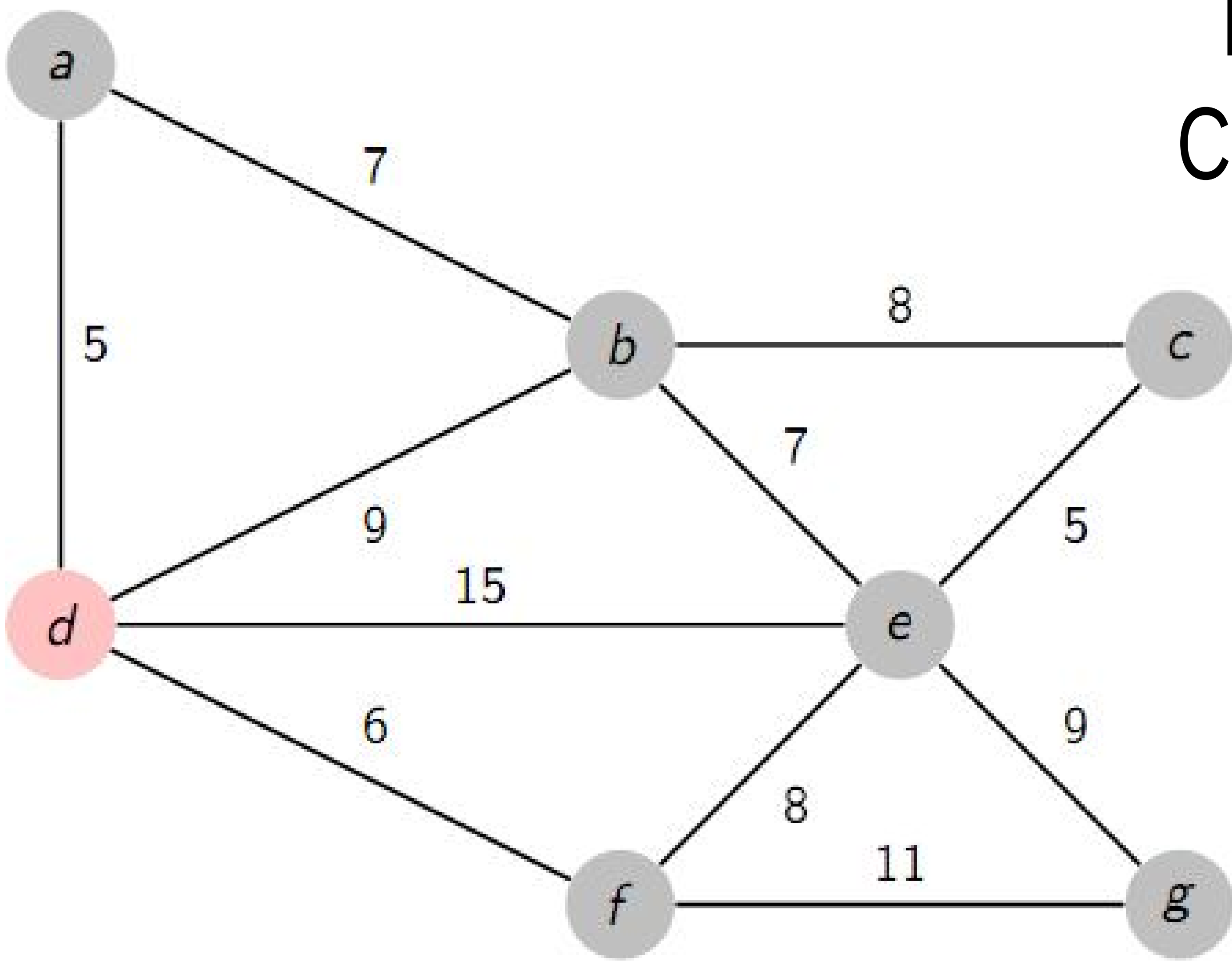


Total Cost: 0

Open List: **d**
Close List:

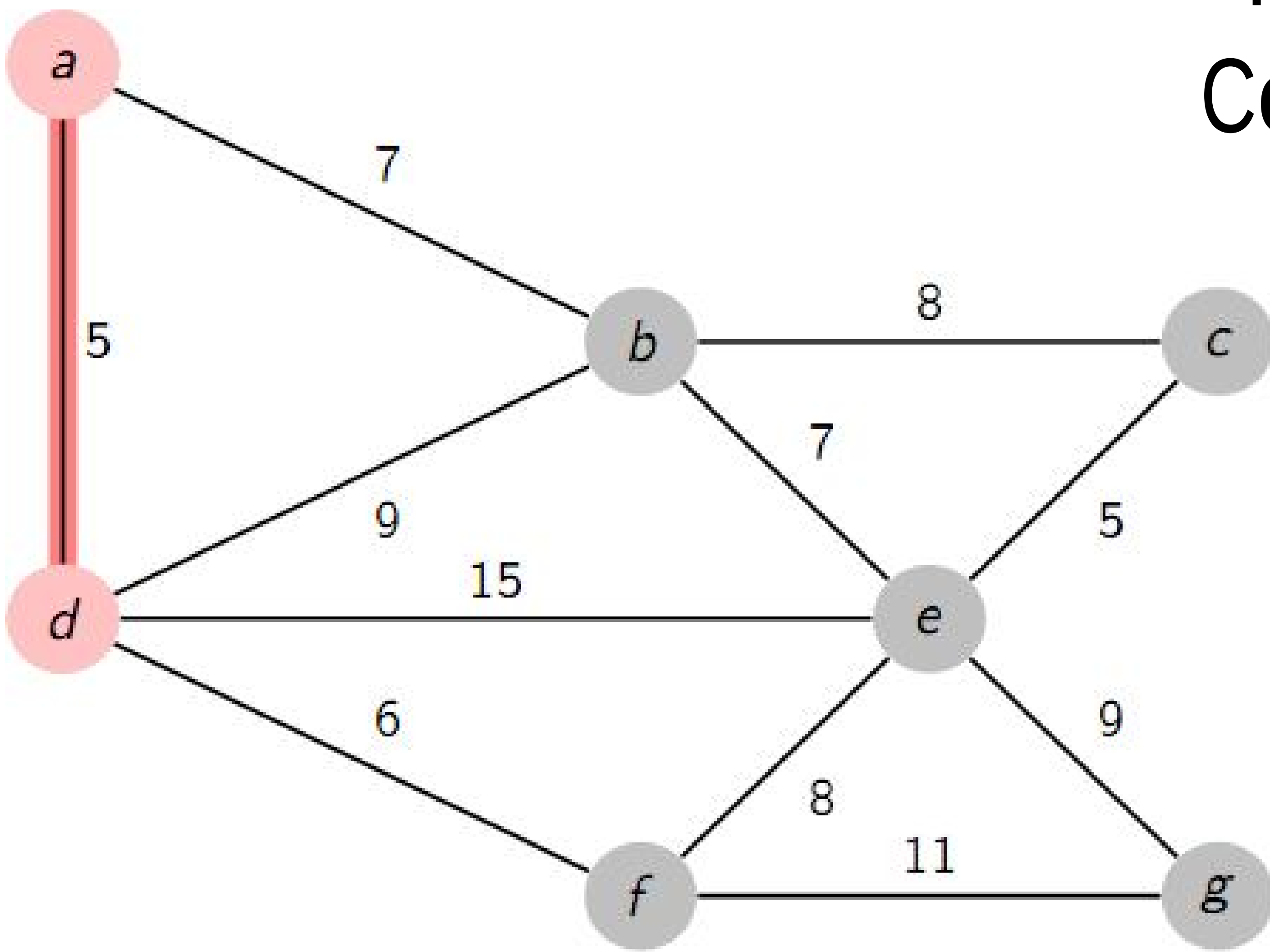
Spanning Tree

Total
Cost: 0



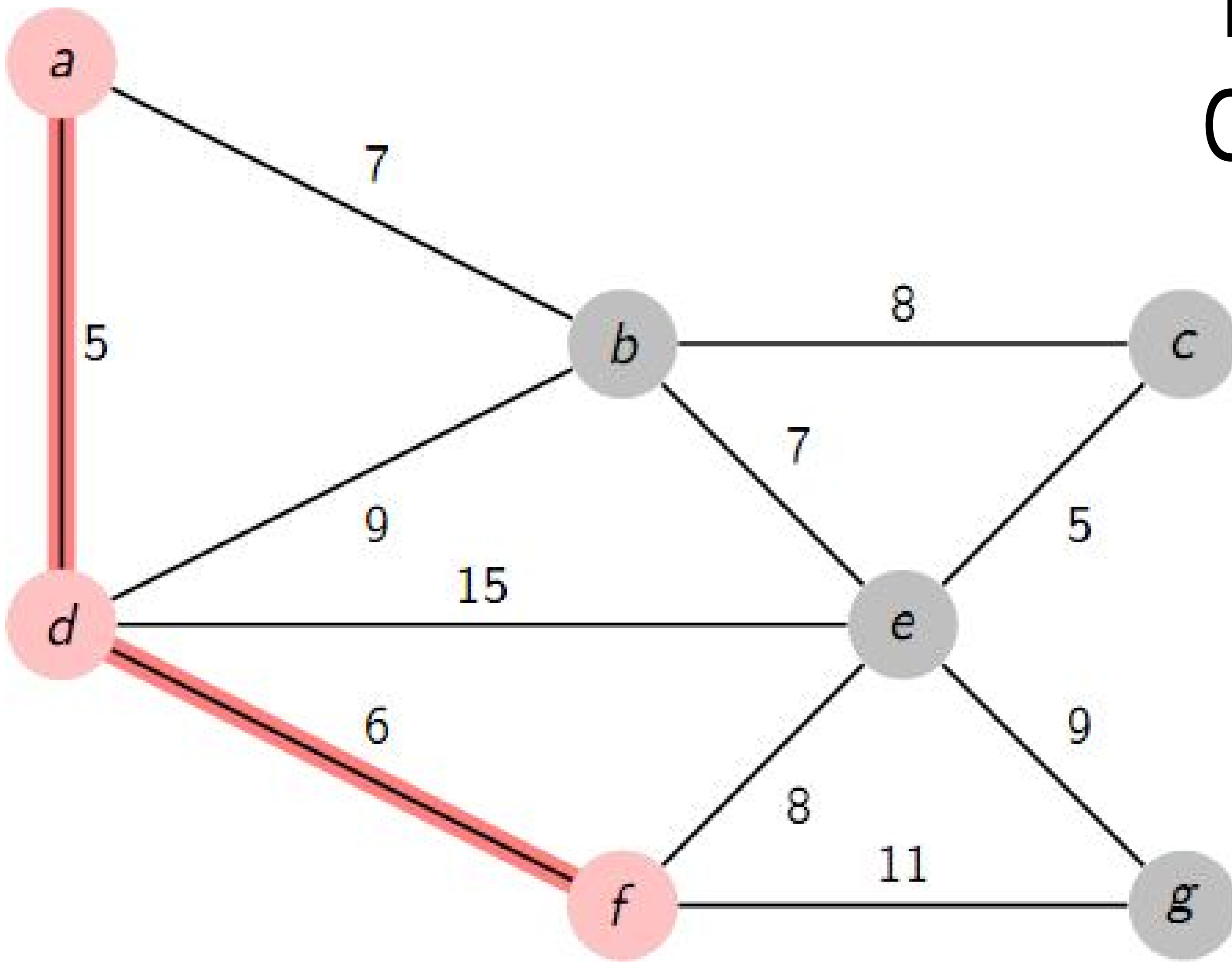
Open List: **a, f, e, b**
Close List: **d**

Total Cost: 5



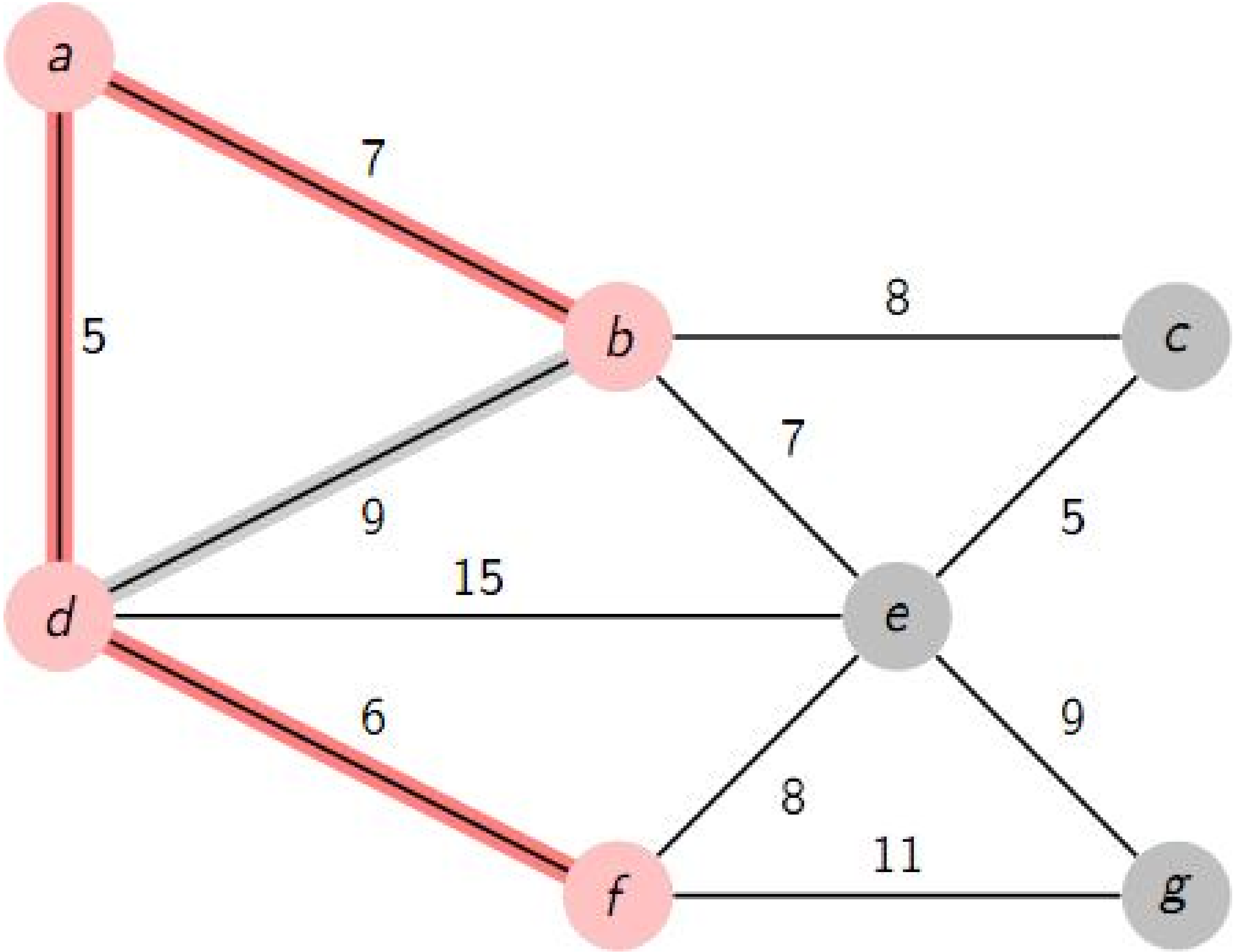
Open List: **f, e, b**

Close List: **d, a**



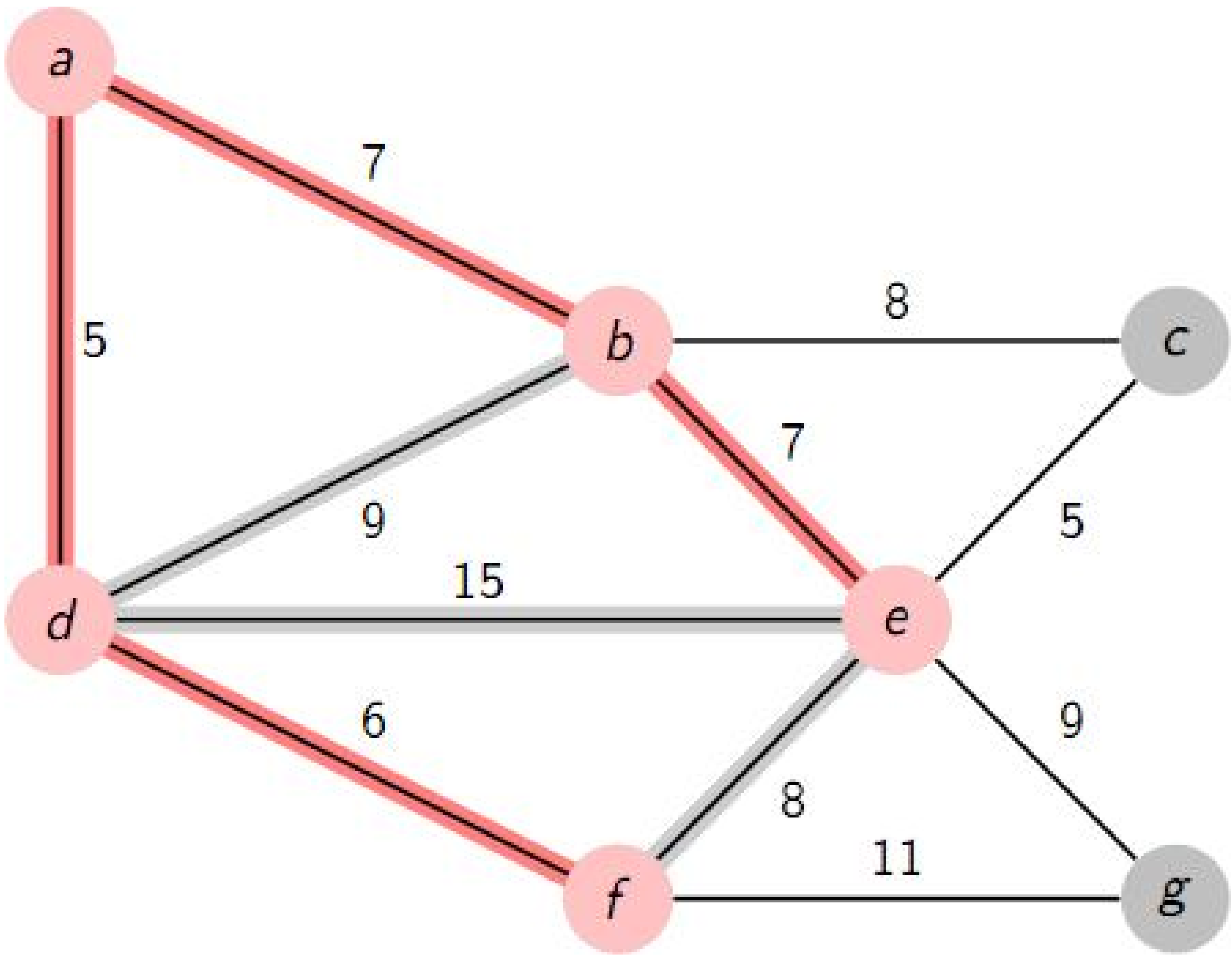
Total Cost:
11

Open List: **b, e, g**
Close List: **d, a, f**



Total Cost:
18

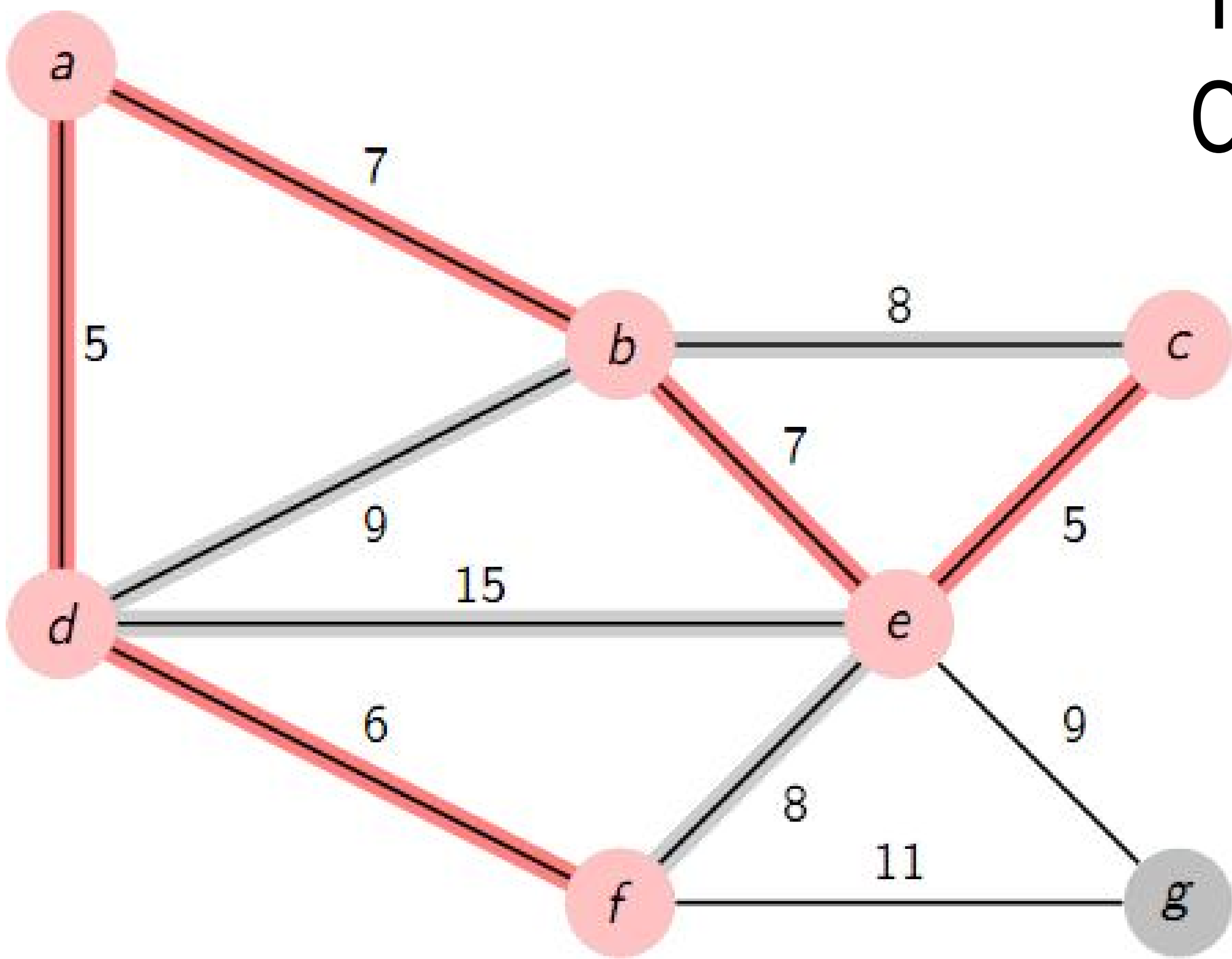
Open List: **e, g, c**
Close List: **d, a, f, b**



Total Cost:
25

Open List: **c, g**

Close List: **d, a, f, b, e**



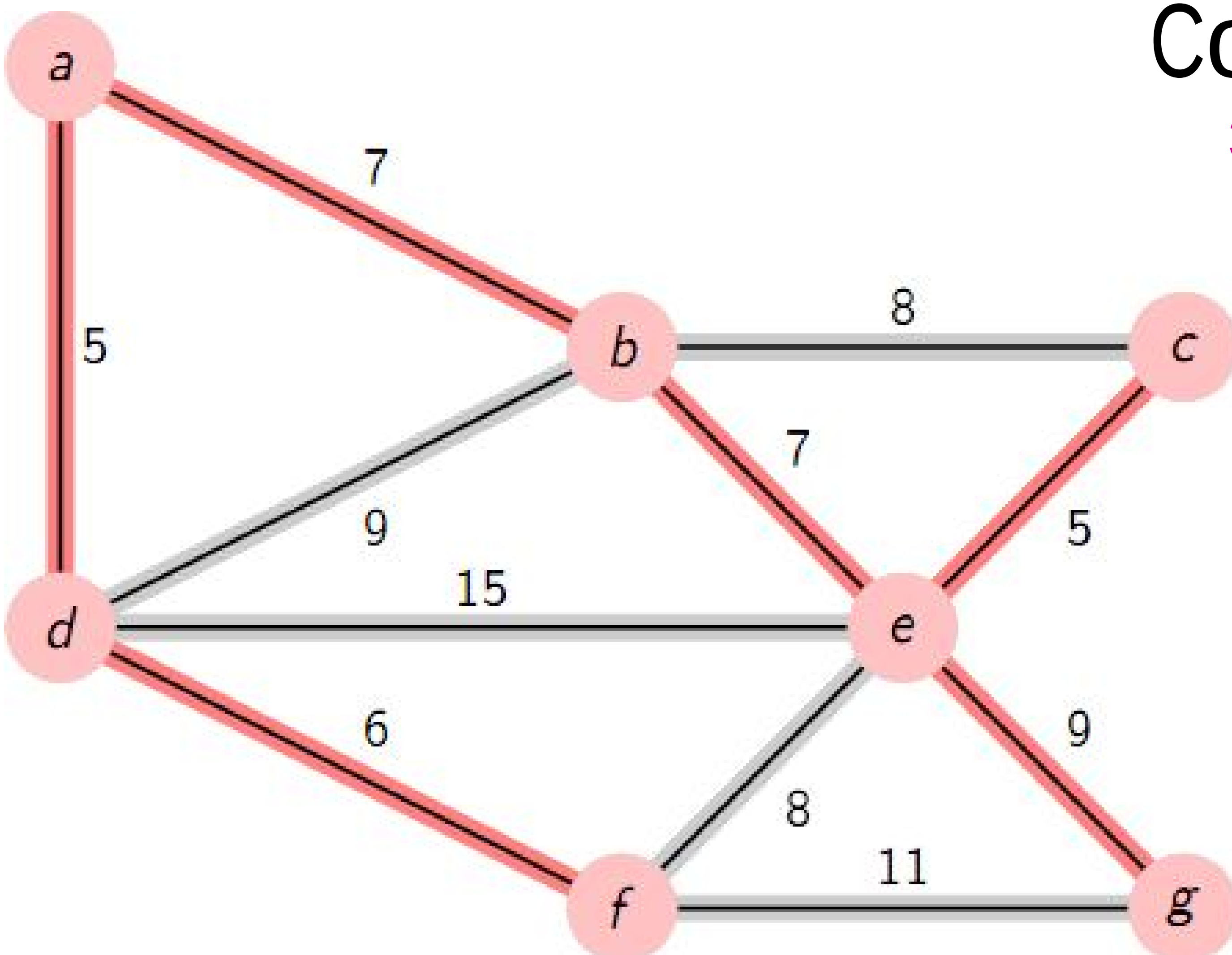
Total Cost:
30

Open List: **g**

Close List: **d, a, f, b, e, c**

Total Cost:

39



Open List:

Close List: **d, a, f, b, e, c**

- **Algorithm for Prim's minimum spanning tree**

```
//Let T be the set of selected edges. initialize T=ø.
//Let TV be the set of vertices already in the tree. set TV={u}.
//Let E be the set of network edges.
while ( (E ≠ ø) and (T ≠ n-1))                                  // E+V
{
   let (u,v) be a least cost edge such that u ϵ TV and v≠TV.       //log V using heap
    if( there is no such edge)
        break;
  E=E-{(u,v)}
  add edge (u,v) to T.
   add vertex v to TV.
}
if (|T| == n-1)
      T is a minimum cost spanning tree.
else
    The network is not connected and has no spanning tree.
```

Complexity Analysis

Minimum edge weight data structure	Time complexity (total)
adjacency matrix, searching	**$O(V*V)$ //to search for a min edge**

binary heap and adjacency list	**$O((V + E) \log(V)) = O(E\log(V))$**
	//V+E for search for min edge using BFS
	// log V to search for vertex in min heap

Fibonacci heap and adjacency list	**$O(E + V \log(V))$**

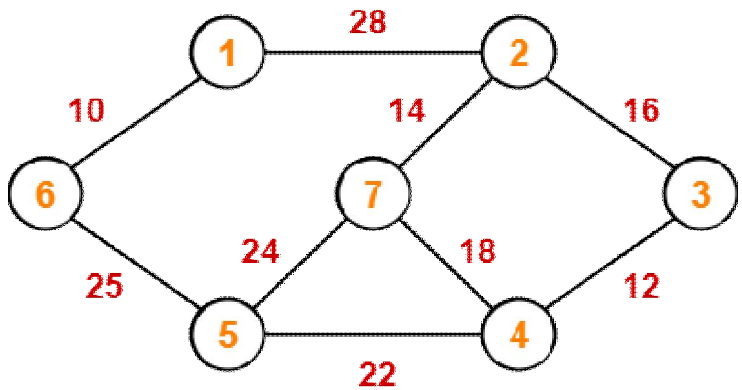
Since, $|E| \leq |V|^2 \Rightarrow \mathbf{\log|E|=(\log V^2) = (2\log V)=O(\log V)}$.

Application

- One practical application of a MST would be in the design of a network. For instance, a group of individuals, who are separated by varying distances, wish to be connected together in a telephone network. Because the cost between two terminal is different, if we want to reduce our expenses, Prim's Algorithm is a way to solve it
- Connect all computers in a computer science building using least amount of cable.
- A less obvious application is that the minimum spanning tree can be used to approximately solve the traveling salesman problem. A convenient formal way of defining this problem is to find the shortest path that visits each point at least once.
- Another useful application of MST would be finding airline routes. The vertices of the graph would represent cities, and the edges would represent routes between the cities. Obviously, the further one has to travel, the more it will cost, so MST can be applied to optimize airline routes by finding the least costly paths with no cycles.

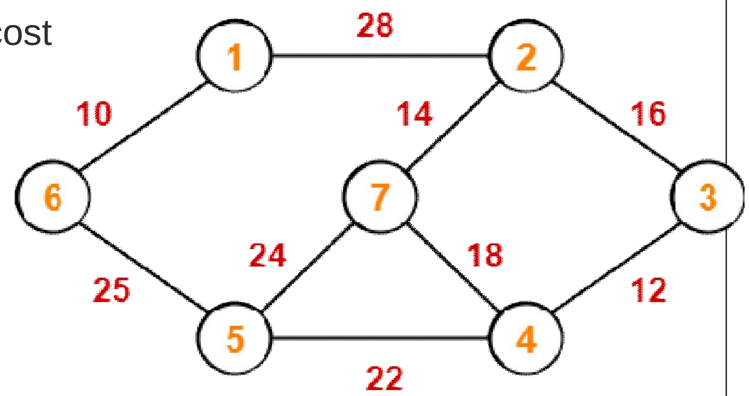
Practice Problems

Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-

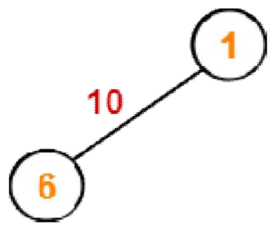


Prim's Algorithm-

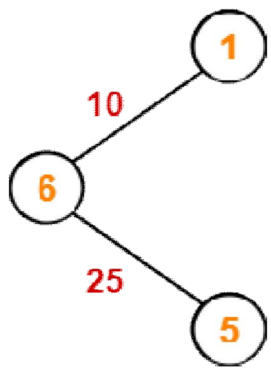
The above discussed steps are followed to find the minimum cost spanning tree using Prim's Algorithm-



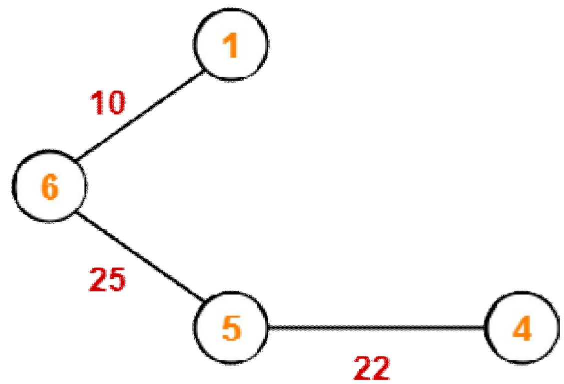
Step-01:



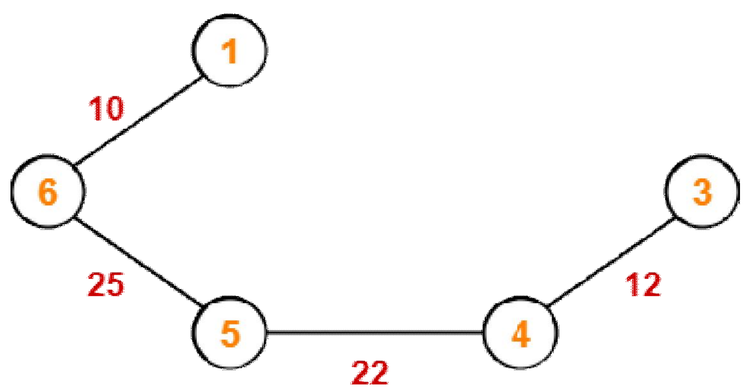
Step-02:



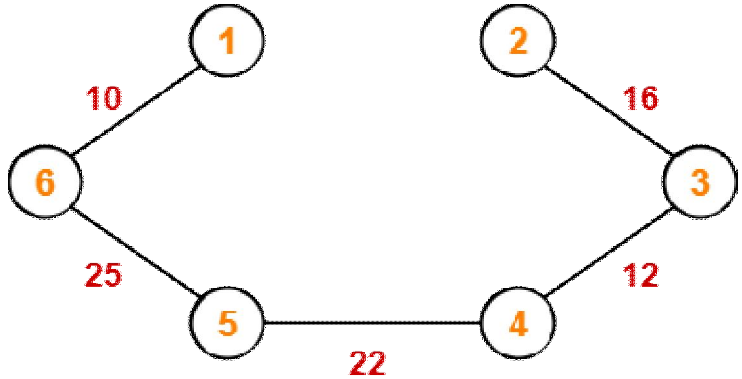
Step-03:



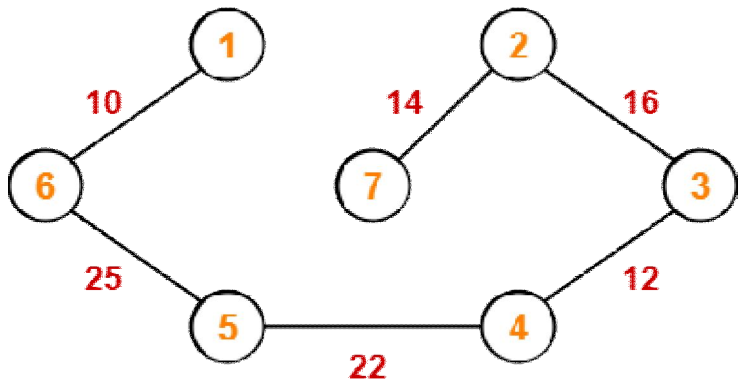
Step-04:



Step-05:



Step-06:



Since all the vertices have been included in the MST, so we stop.

Now, Cost of Minimum Spanning Tree

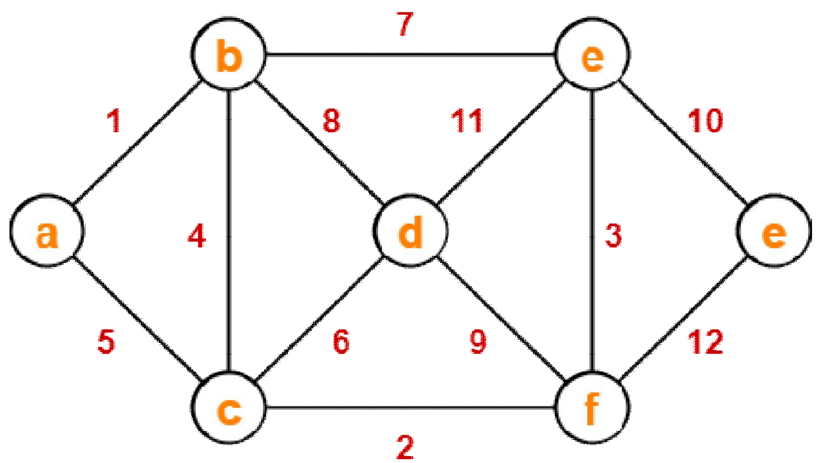
= Sum of all edge weights

= 10 + 25 + 22 + 12 + 16 + 14

= 99 units

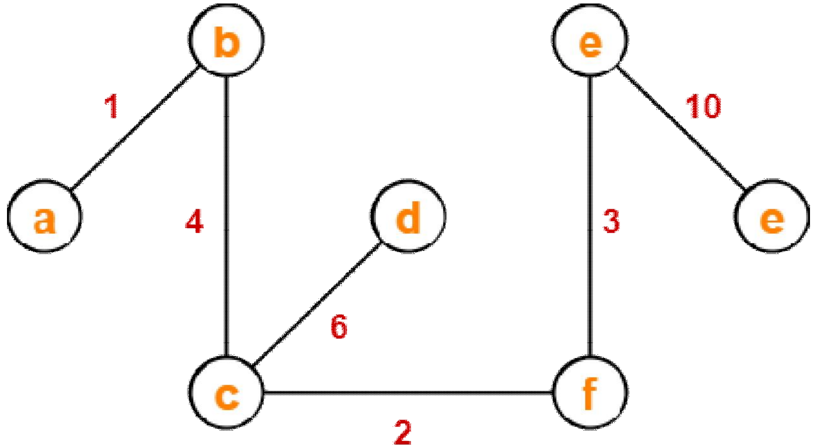
Problem-02:

Using Prim’s Algorithm, find the cost of minimum spanning tree (MST) of the given graph-



Solution-

The minimum spanning tree obtained by the application of Prim’s Algorithm on the given graph is as shown below-



Now, Cost of Minimum Spanning Tree

= Sum of all edge weights

= 1 + 4 + 2 + 6 + 3 + 10

= 26 units

Previous Year Gate Questions

Q. No. 1

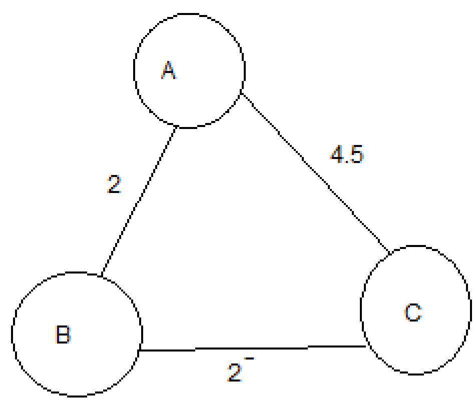
GATE CSE 2016 Set 1

Let G be a weighted connected undirected graph with distinct positive edge weights. If every edge weight is increased by the same value, then which of the following statements is/are **TRUE**?

P : Minimum spanning tree of G does not change

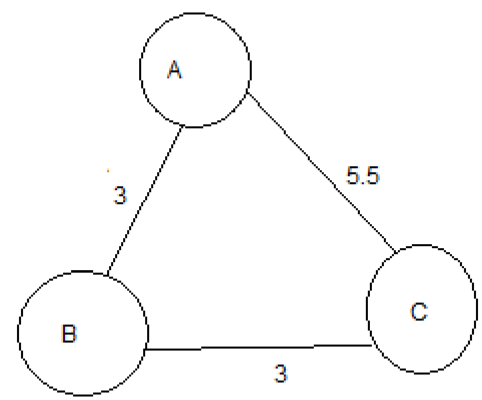
Q : Shortest path between any pair of vertices does not change

- (A) P only
- (B) Q only
- (C) Neither P nor Q
- (D) Both P and Q



Shortest path is A-B-C

==>



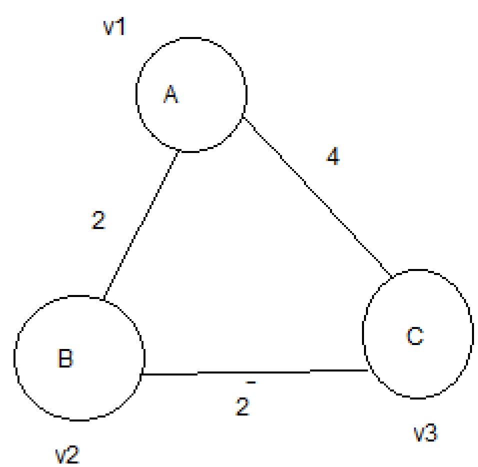
Shortest path is A-C

Q. No. 2

GATE CSE 2006

Consider a weighted complete graph G on the vertex set {v1, v2, ..vn} such that the weight of the edge (vi, vj) is $2|i-j|$. The weight of a minimum spanning tree of G is:

- **A** n — 1
- **B** 2n — 2
- **C** $\frac{n}{2}$
- **D** n^2



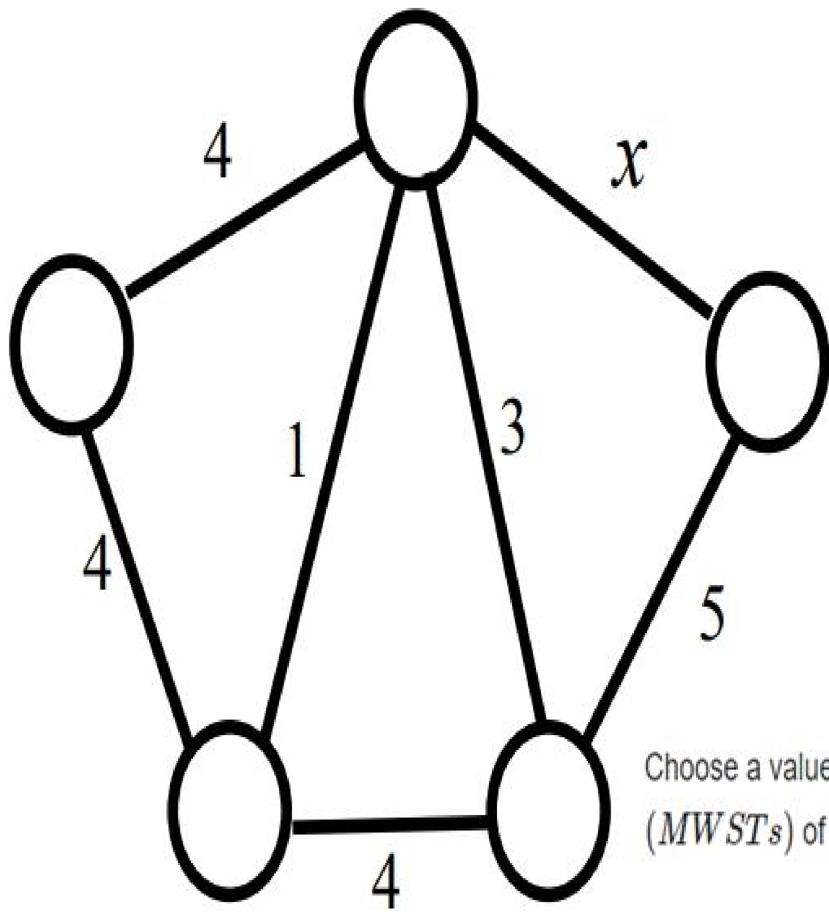
2+2 = 4

2n-2=6-2=4

Q. No. 3

GATE CSE 2018

Consider the following undirected graph G :



Choose a value for x that will maximize the number of minimum weight spanning trees $(MWSTs)$ of G. The number of $MWSTs$ of G for this value of x is _____.

A) 4
B) 5
C) 2
D) 3

Answer is B

Edge weights
1
3
4
4
4
5
1,3,4(1),5,x
1,3,4(1),x,5
1,3,4(2),5,x
1,3,4(2),x,5

Q. No. 4

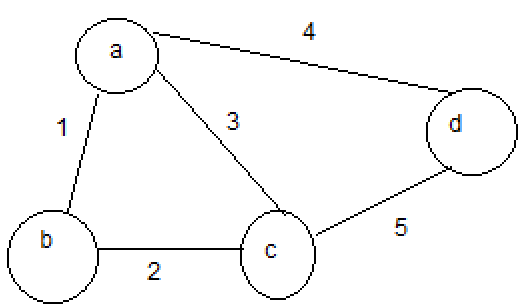
GATE CSE 2016 Set 1

$G = (V, E)$ is an undirected simple graph in which each edge has a distinct weight, and e is a particular edge of G. Which of the following statements about the minimum spanning trees $(MSTs)$ of G is/are **TRUE**?

I. If e is the lightest edge of <u>some</u> cycle in G, then every MST of G <u>includes</u> e

II. If e is the heaviest edge of <u>some</u> cycle in G, then every MST of G <u>excludes</u> e

- (A) I only
- (B) II only
- (C) both I and II
- (D) neither I nor II

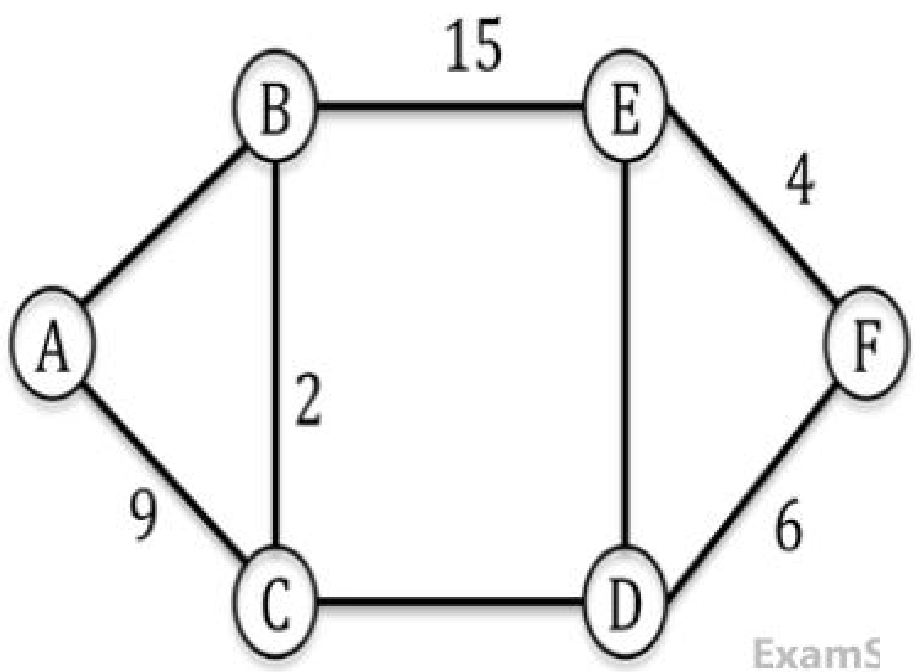


1,2,4

Q. No. 5

GATE CSE 2015 Set 1

The graph shown below has 8 edges with distinct integer edge weights. The minimum spanning tree (MST) is of weight 36 and contains the edges: {(A, C), (B, C), (B, E), (E, F), (D, F)}. The edge weights of only those edges which are in the MST are given in the figure shown below. The minimum possible sum of weights of all 8 edges of this graph is ___________.



2
4
6
9
15
Total : 36

ED 7,
AB 10
CD 16
Total : 33
69

Answer

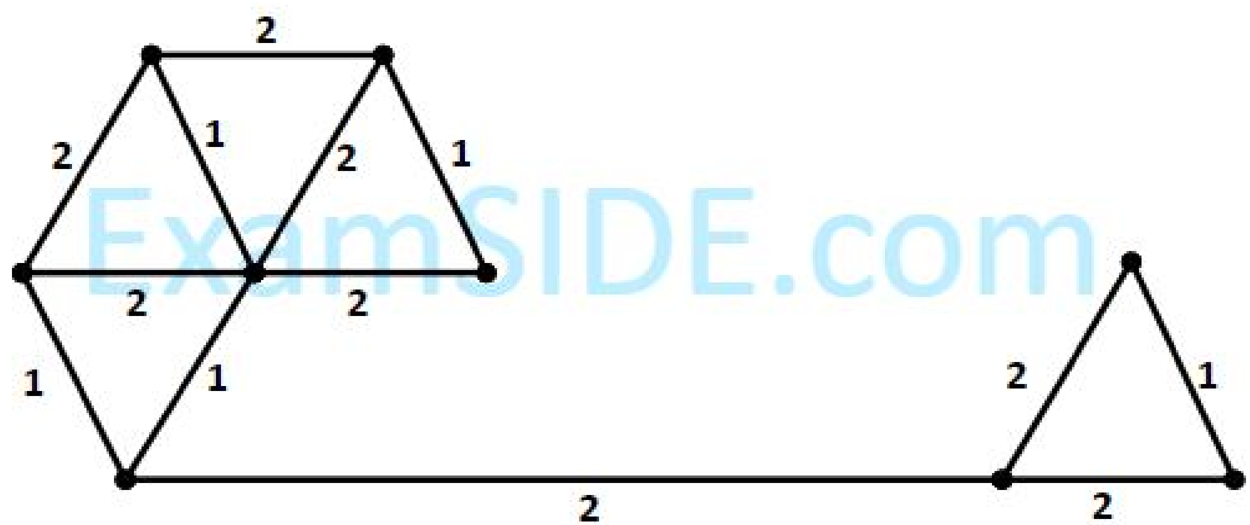
Correct Answer is **69**

Q. No. 6

GATE CSE 2014 Set 2

The number of distinct minimum spanning trees for the weighted graph below is

____



Answer

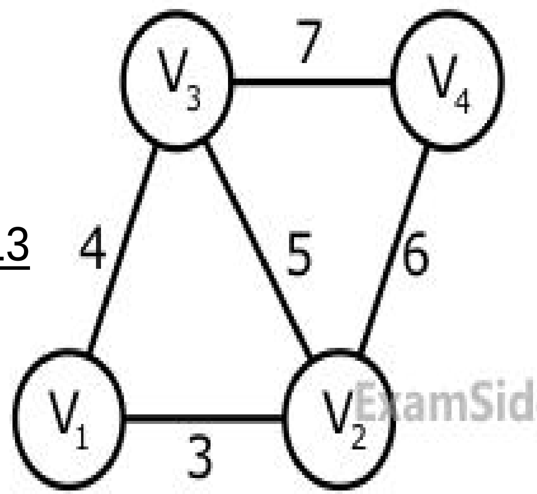
Correct Answer is **6**

GATE CSE 2011

Q. No. 7

An undirected graph G(V, E) contains n (n > 2) nodes named v_1 , v_2 ,....v_n. Two nodes v_i , v_j are connected if and only if $0 < |i - j| <= 2$. Each edge (v_i, v_j) is assigned a weight i + j. A sample graph with n = 4 is shown below.

MST cost is 3+4+6=13
Let n=4 in option (B) :16-4+1=13



A $\frac{1}{12}(11n^2 - 5n)$

B $n^2 - n + 1$

C 6n – 11

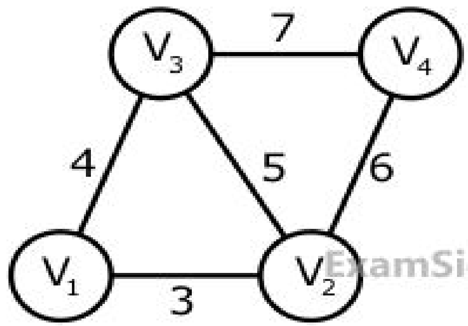
D 2n + 1

What will be the cost of the minimum spanning tree (MST) of such a graph with n nodes?

Q. No. 8

GATE CSE 2011

An undirected graph G(V, E) contains n ($n > 2$) nodes named v_1 , v_2 ,….v_n. Two nodes v_i , v_j are connected if and only if $0 < |i - j| <= 2$. Each edge (v_i, v_j) is assigned a weight $i + j$. A sample graph with $n = 4$ is shown below.



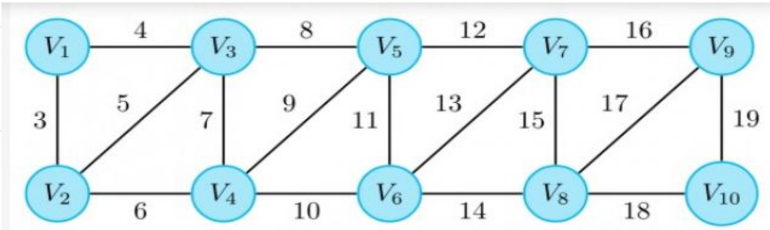
The length of the path from v5 to v6 in the MST of previous question with n = 10 is

A 11

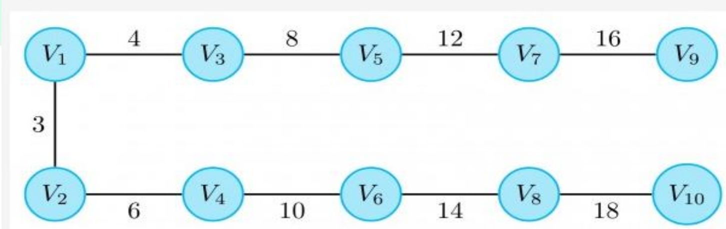
B 25

C 31

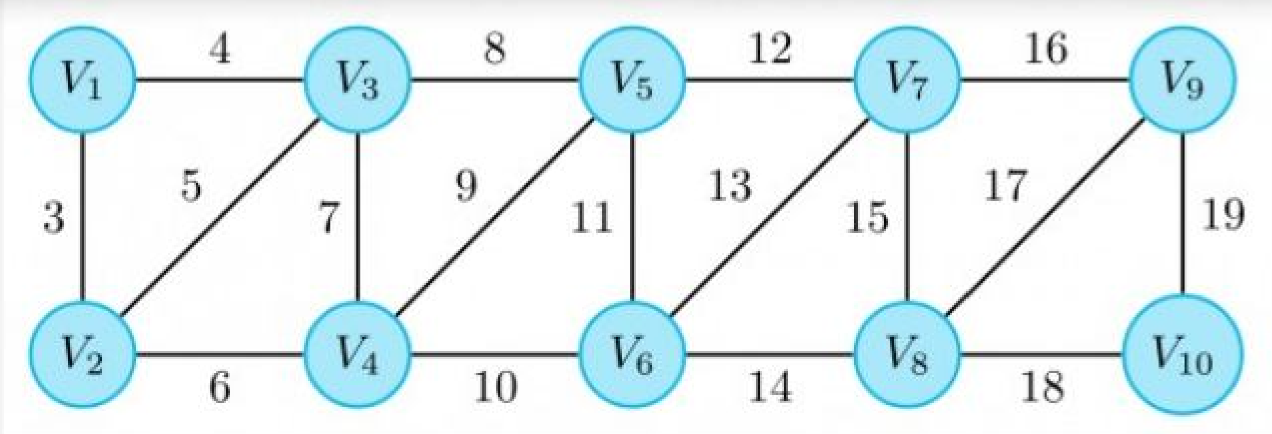
D 41



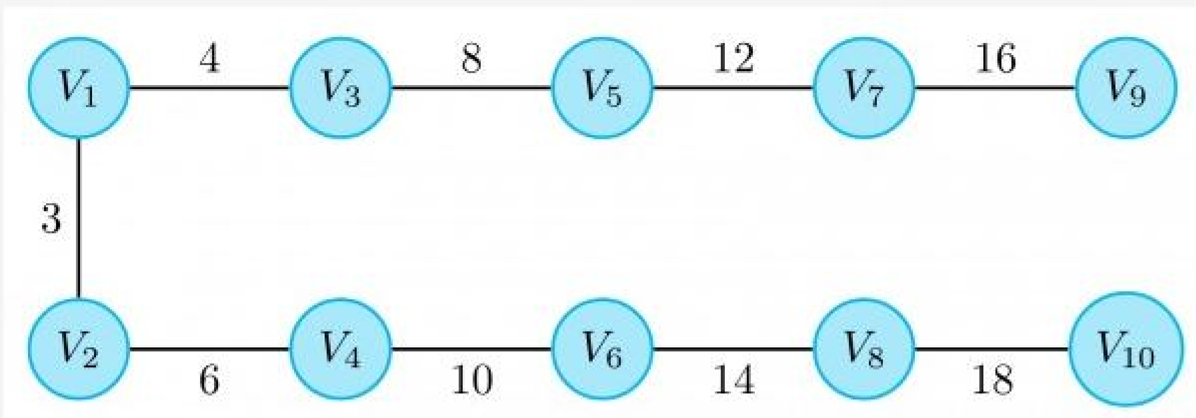
Above is the graph.
Below is the MST.



Length of the path from v_5 to $v_6 = 8 + 4 + 3 + 6 + 10 = 31$ **(Answer)**



Above is the graph.
Below is the MST.



Length of the path from v_5 to $v_6 = 8 + 4 + 3 + 6 + 10 = 31$ **(Answer)**

Q. No. 9

GATE CSE 2010

Consider a complete undirected graph with vertex set {0, 1, 2, 3, 4}. Entry W(ij) in the matrix W below is the weight of the edge {i, j}.

$$w = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

What is the minimum possible weight of a spanning tree T in this graph such that vertex 0 is a leaf node in the tree T?

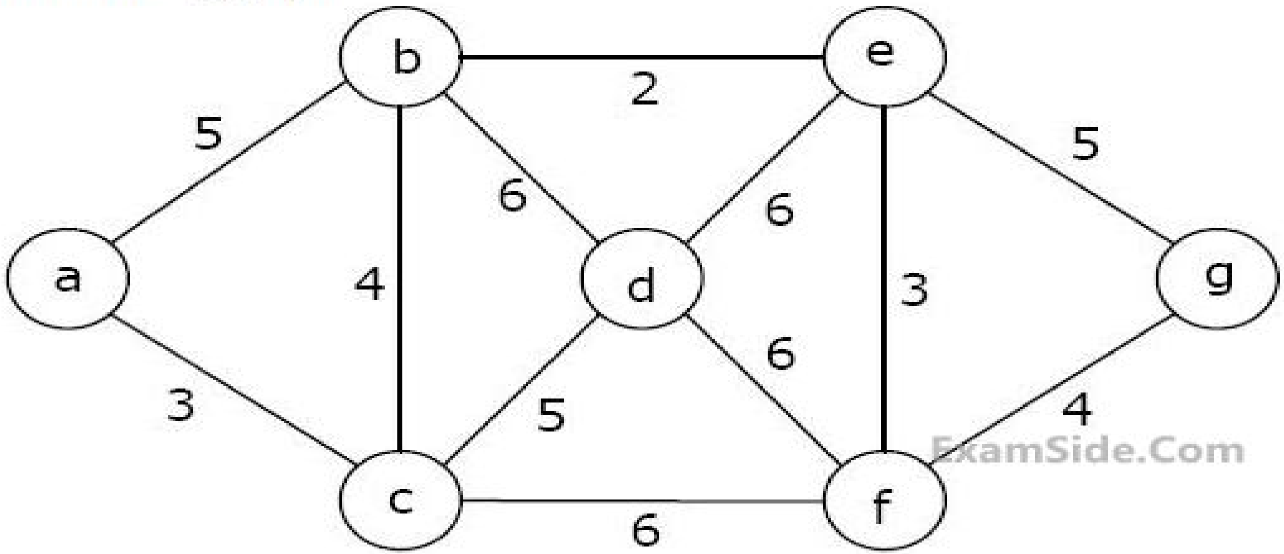
- A 7
- B 8
- C 9
- D 10

To get the minimum spanning tree with vertex 0 as leaf, first remove 0th row and 0th column and then get the minimum spanning tree (MST) of the remaining graph. Once we have MST of the remaining graph, connect the MST to vertex 0 with the edge with minimum weight (we have two options as there are two 1s in 0th row).

Q. No. 10

GATE CSE 2009

Consider the following graph:



Which one of the following is NOT the sequence of edges added to the minimum spanning tree using Kruskal's algorithm?

- A (b,e)(e,f)(a,c)(b,c)(f,g)(c,d)
- B (b,e)(e,f)(a,c)(f,g)(b,c)(c,d)
- C (b,e)(a,c)(e,f)(b,c)(f,g)(c,d)
- D (b,e)(e,f)(b,c)(a,c)(f,g)(c,d)

Q. No. 11

GATE CSE 2007

Let w be the minimum weight among all edge weights in an undirected connected graph. Let e be a specific edge of weight w . Which of the following is FALSE?

A. There is a minimum spanning tree containing e.

B. If e is not in a minimum spanning tree T, then in the cycle formed by adding e to T, all edges have the same weight.

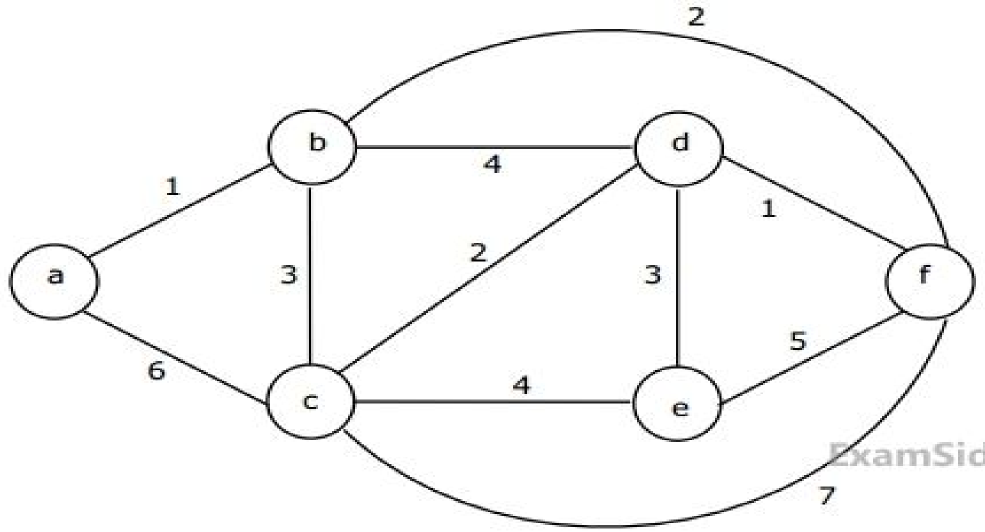
C. Every minimum spanning tree has an edge of weight w .

D. e is present in every minimum spanning tree.

GATE CSE 2006

Q. No. 12

Consider the following graph:



a-b	1
df	1
cd	2
bf	2
bc	3
de	3
bd	4
ce	4
ef	5
ac	6
cf	7

Which one of the following cannot be the sequence of edges added, in that order, to a minimum spanning tree using Kruskal's algorithm?

A $(a-b), (d-f), (b-f), (d-c), (d-e)$

B $(a-b), (d-f), (d-c), (b-f), (d-e)$

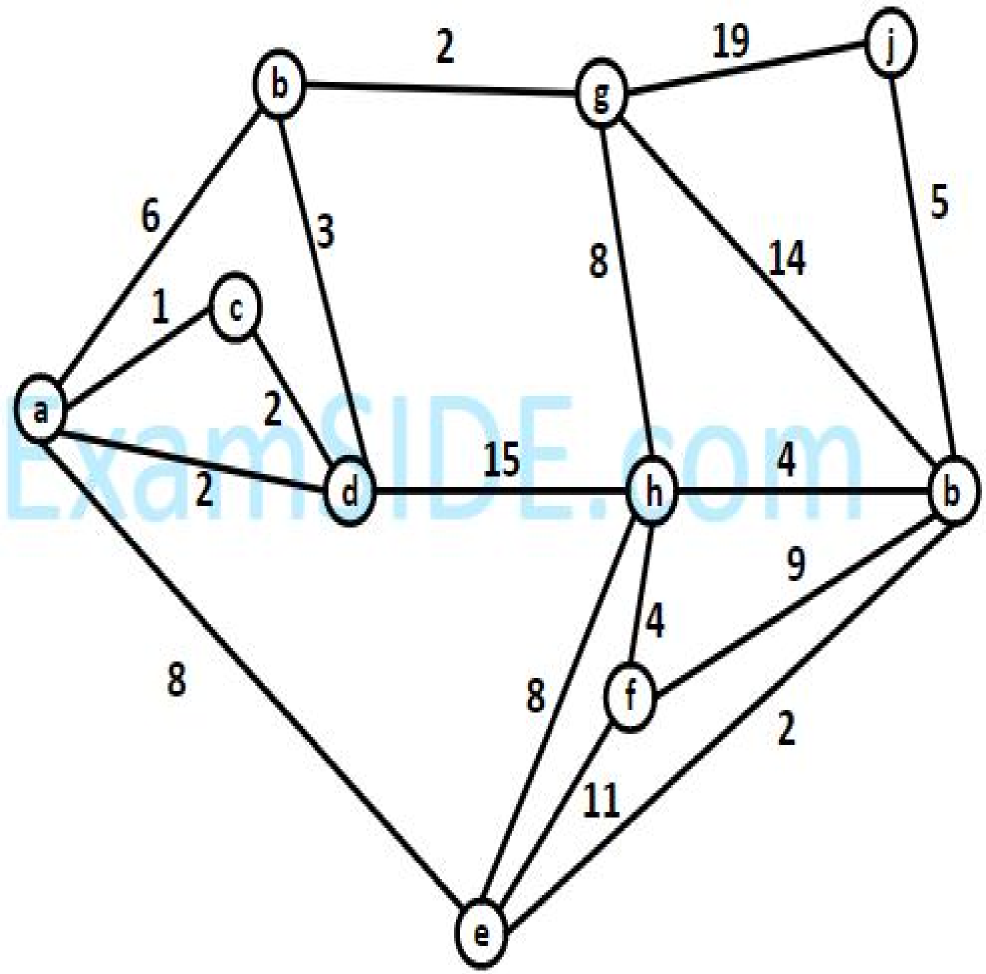
C $(d-f), (a-b), (d-c), (b-f), (d-e)$

D $(d-f), (a-b), (b-f), (d-e), (d-c)$

Q. No. 13

GATE CSE 2003

What is the weight of a minimum spanning tree of the following graph?



A 29

B 31

C 38

D 41

Q. No. 14

GATE CSE 2000

Let G be an undirected connected graph with distinct edge weight. Let emax be the edge with maximum weight and emin the edge with minimum weight. Which of the following statements is false?

A Every minimum spanning tree of G must contain emin

B If emax is in a minimum spanning tree, then its removal must disconnect G

C No minimum spanning tree contains emax

D G has a unique minimum spanning tree

Q. No. 15

GATE CSE 2010

Consider a complete undirected graph with vertex set {0, 1, 2, 3, 4}. Entry W(ij) in the matrix W below is the weight of the edge {i, j}.

$$w = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

What is the minimum possible weight of a path P from vertex 1 to vertex 2 in this graph such that P contains at most 3 edges?

- A 7
- B 8
- C 9
- D 10

Thank You