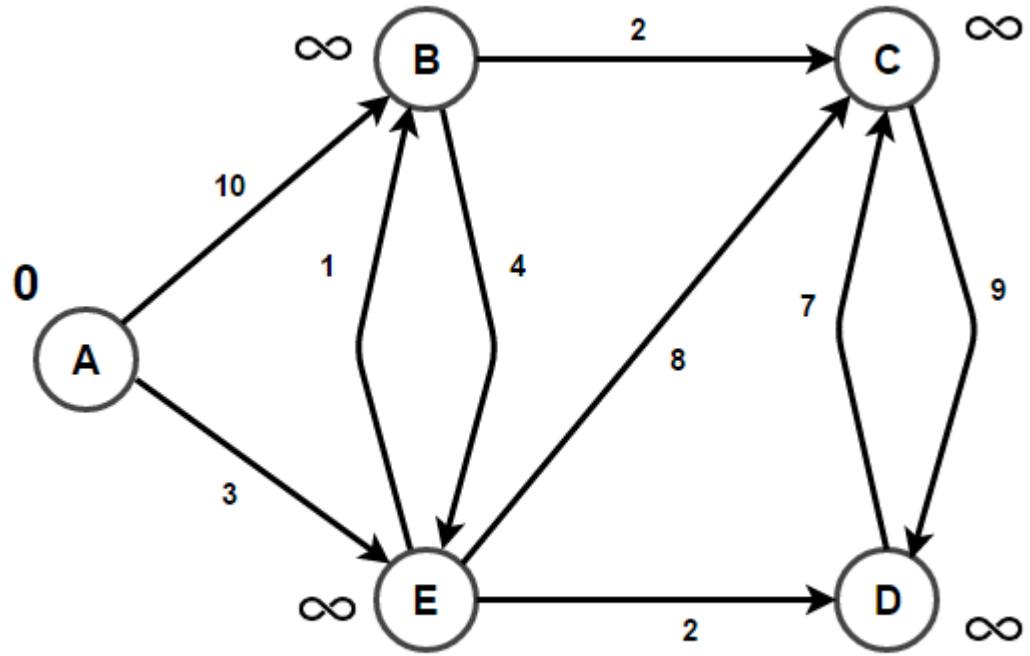
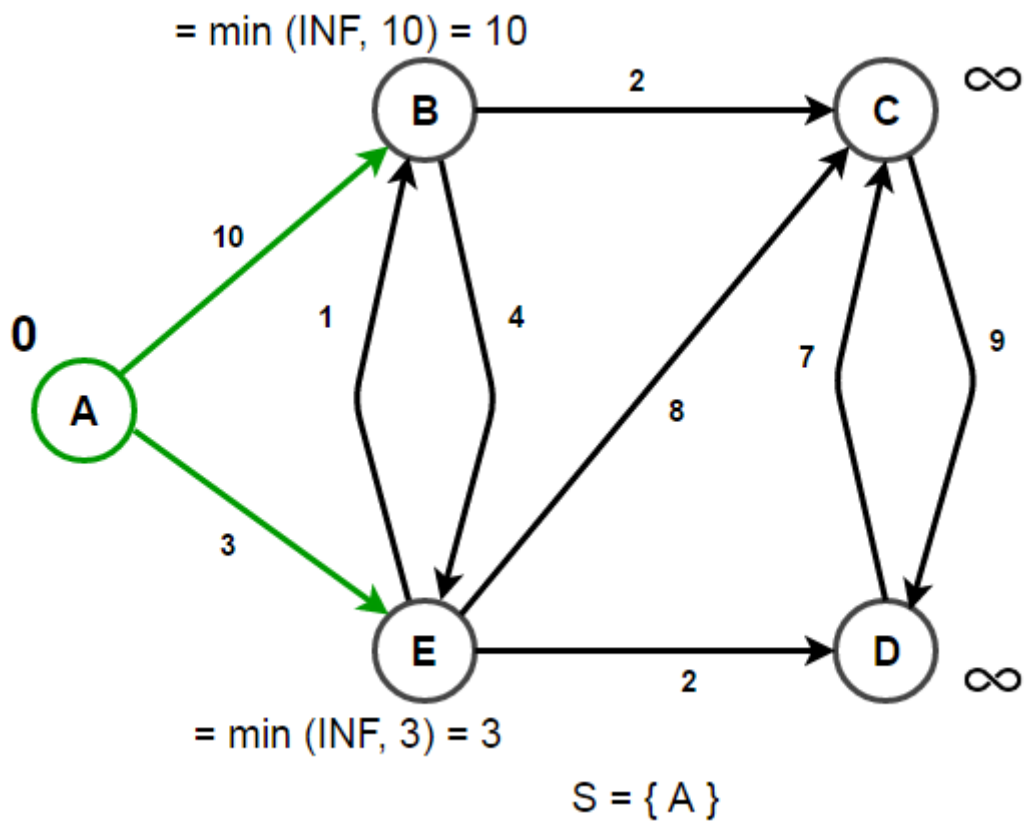
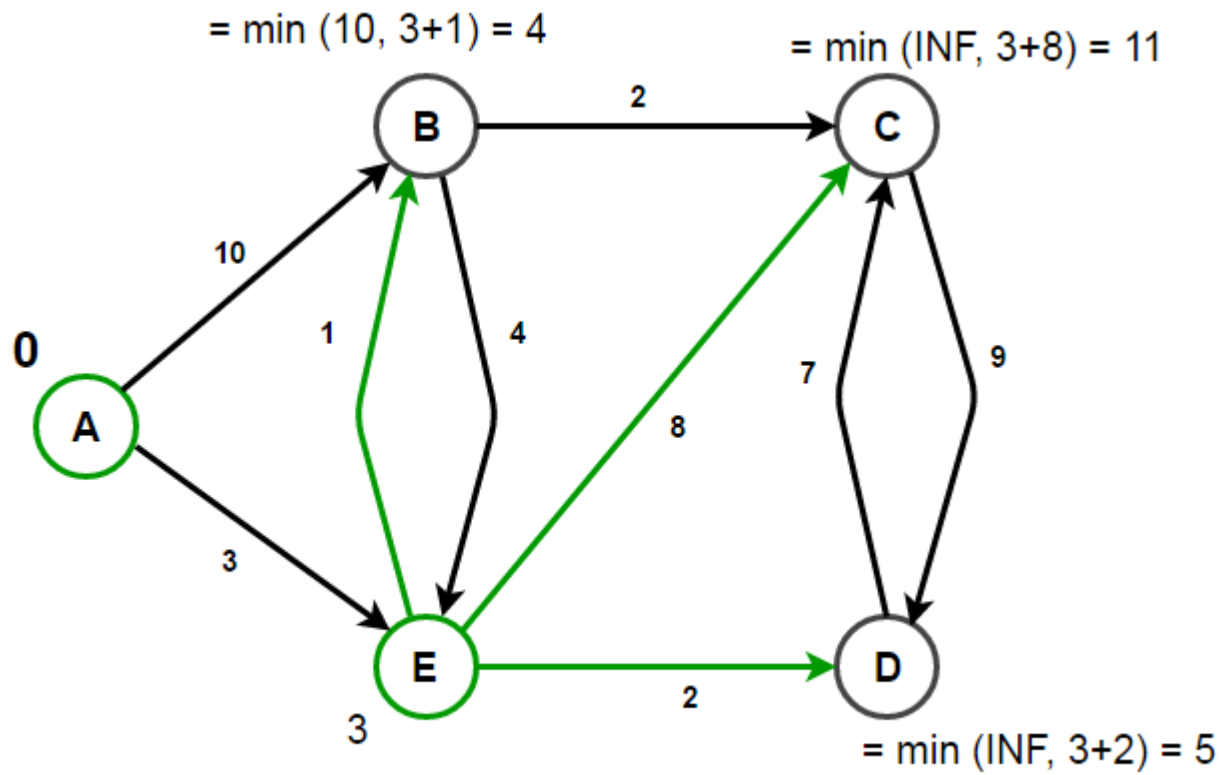
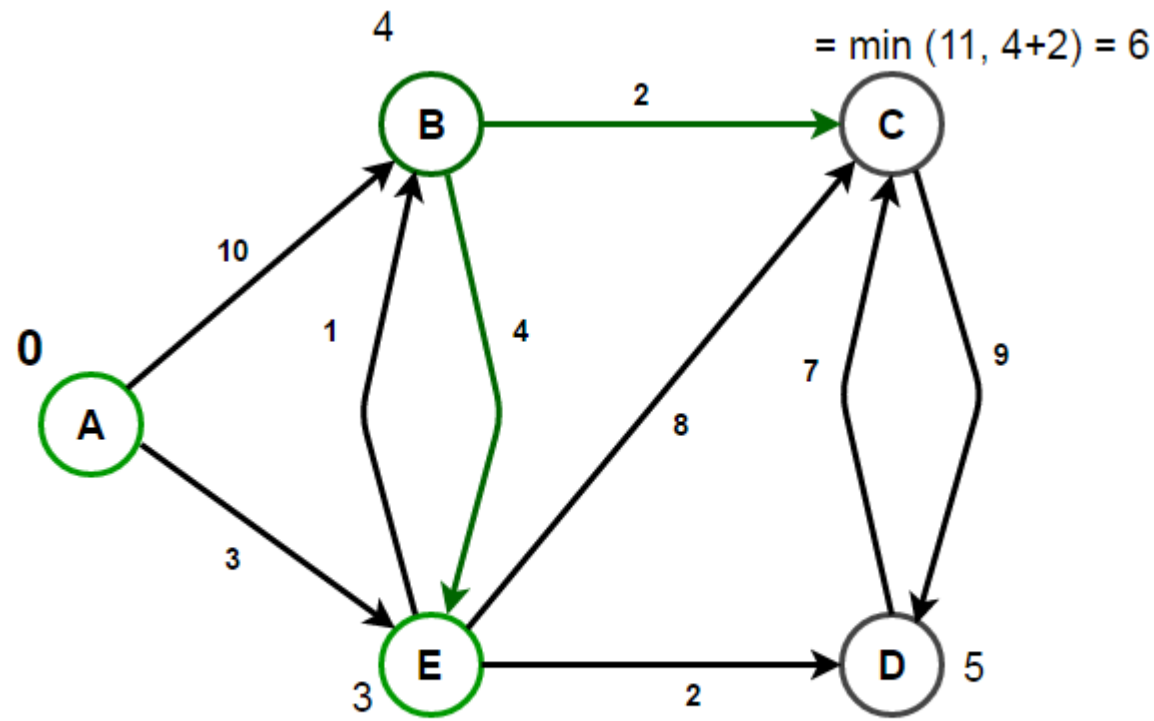
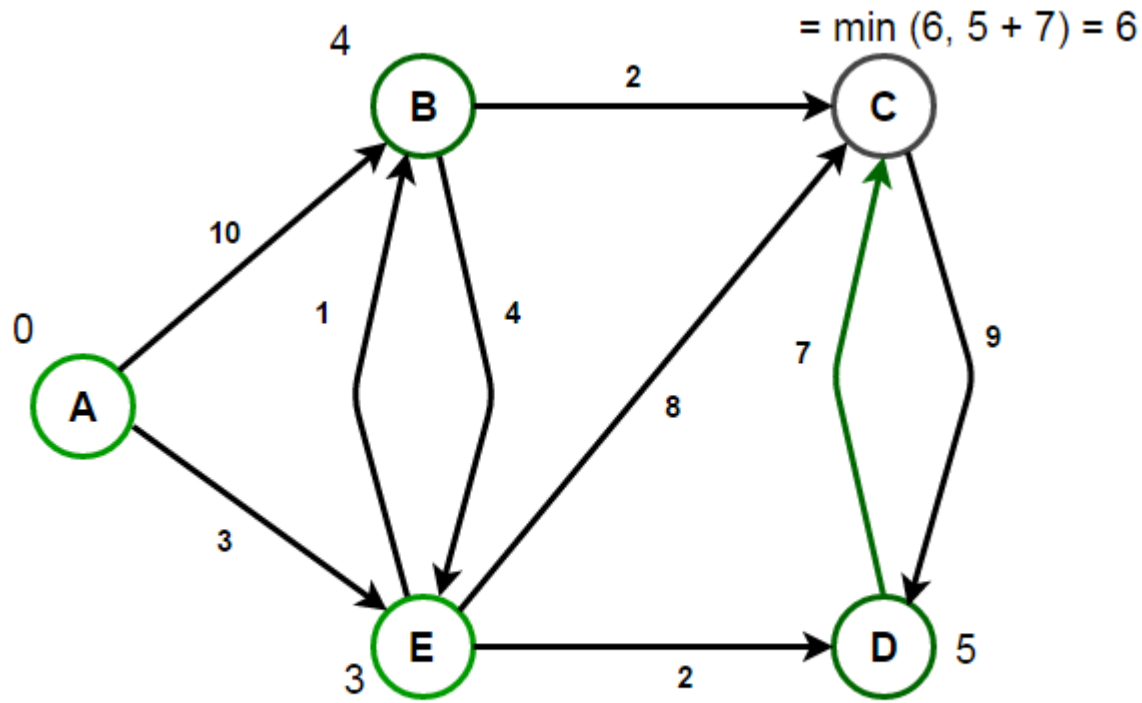
Consider the following graph. What is the shortest path from node A to node F?



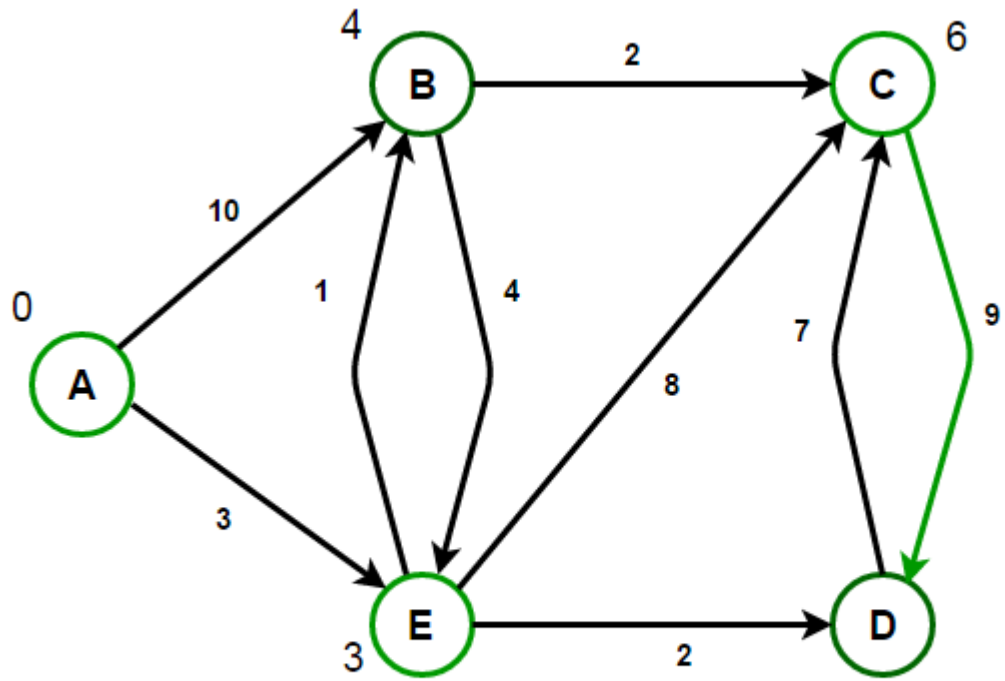

= min (INF, 10) = 10
B
2
C
∞
10
0
1
4
9
7
A
8
3
E
2
D
∞
= min (INF, 3) = 3
S = { A }


= min (10, 3+1) = 4
= min (INF, 3+8) = 11
B
2
C
10
0
1
4
7
9
A
8
3
E
D
2
3
= min (INF, 3+2) = 5
S = { A, E }


4
= min (11, 4+2) = 6
B
2
C
10
1
4
0
A
7
9
8
3
E
D
5
3
2
= min (3, 4 + 4) = 3
S = { A, E, B }

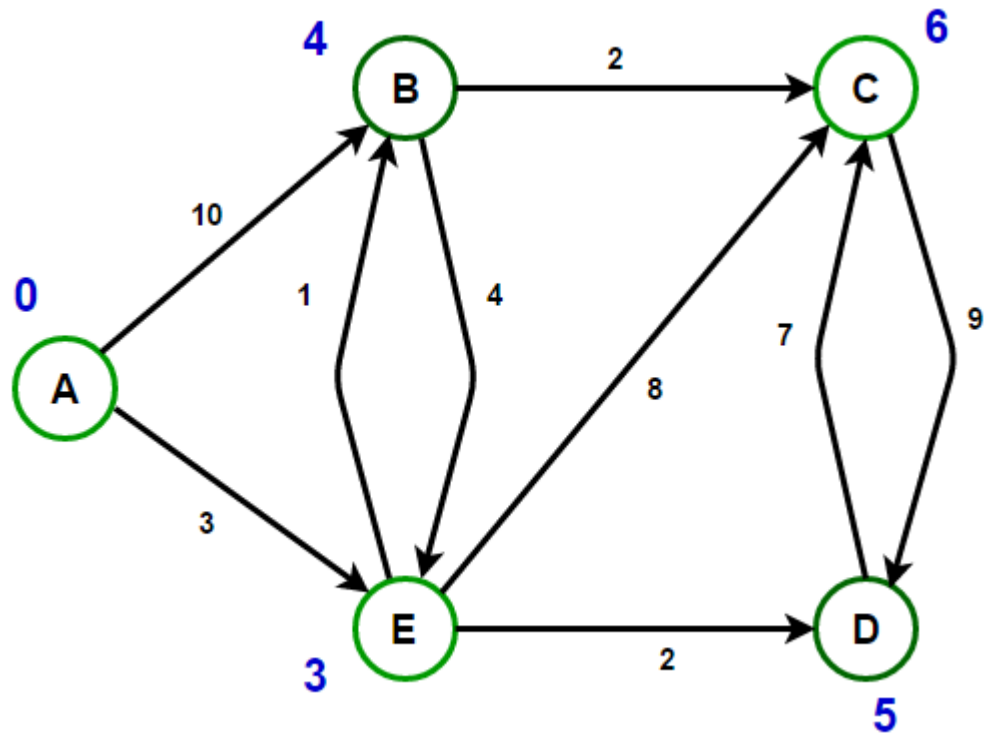


S = { A, E, B, D }



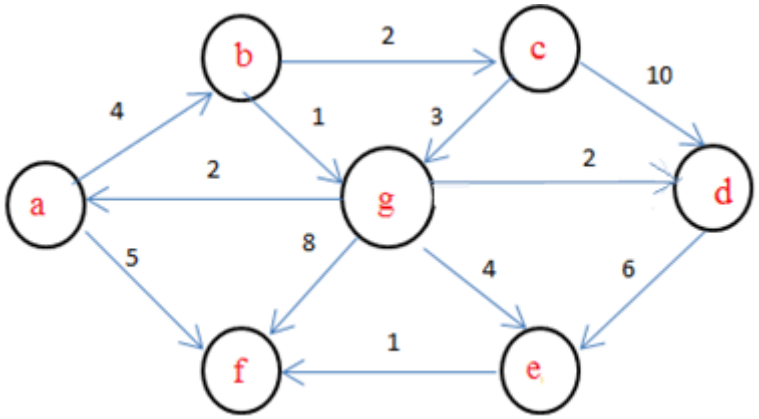
= min (5, 6 + 9) = 5

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


4
6
B
2
C
10
0
1
4
7
9
A
8
3
E
D
2
3
5

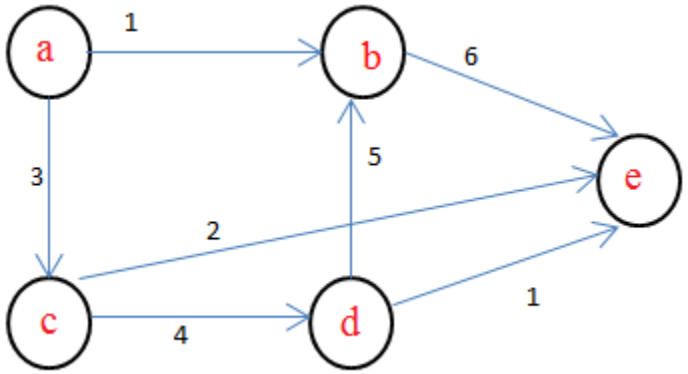
Vertex	Minimum Cost	Route
A —> B	4	A —> E —> B
A —> C	6	A —> E —> B —> C
A —> D	5	A —> E —> D
A —> E	3	A —> E

Consider the following graph If b is the source vertex, what is the minimum cost to reach f vertex?



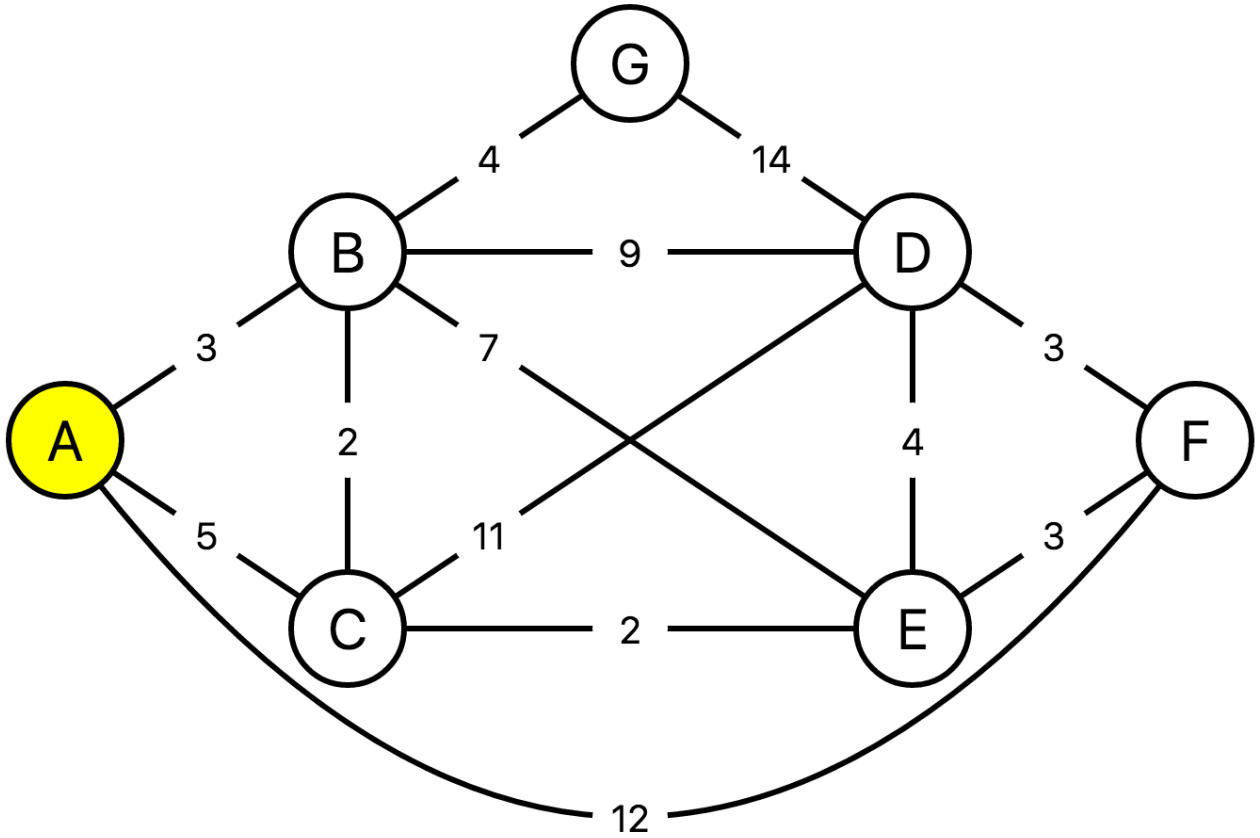
Explanation: The minimum cost to reach f vertex from b vertex is 6 by having vertices g and e as intermediates.
b to g, cost is 1
g to e, cost is 4
e to f, cost is 1
hence total cost 1+4+1=6.

In the given graph, identify the shortest path having minimum cost to reach vertex E if A is the source vertex.



Explanation: The minimum cost required to travel from vertex A to E is via vertex C
A to C, cost= 3
C to E, cost= 2
Hence the total cost is 5.

What is the shortest path from node A to node F?



Answer : **A** -5-> **C** -2-> **E** -3-> **F** = **10**

In a weighted, undirected graph if we apply Dijkstra's algorithm to find the shortest path between two nodes. If we add 1 to all the edge weights, does the shortest path remain the same?

A - Yes

B - No

Answer : If we use the graph on question 2 and increase all edge weights by 1, the shortest path from node A to node F is no longer A -> C -> E -> F, it becomes A -> F.

To implement Dijkstra’s shortest path algorithm on unweighted graphs so that it runs in linear time, the data structure to be used is:

A. Stack
B. Heap
C. Queue
D. Binary Tree

Answer : C

If we use Queue (FIFO) instead of Priority Queue (Min Heap), we get the shortest path in linear time O(|V| + |E|).

If all edges have the same weight in an undirected graph, which algorithm will find the shortest path between two nodes more efficiently?

A - Dijkstra
B - Bellman-Ford
C - Depth-First Search
D - Breadth-First Search

Answer : D

Breadth-First Search has time complexity O(|V| + |E|).

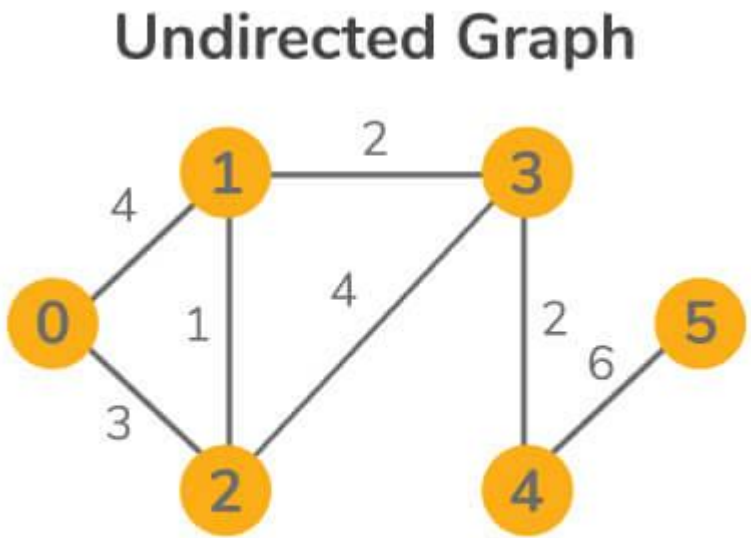
Dijkstra's algorithm is based on which paradigm?

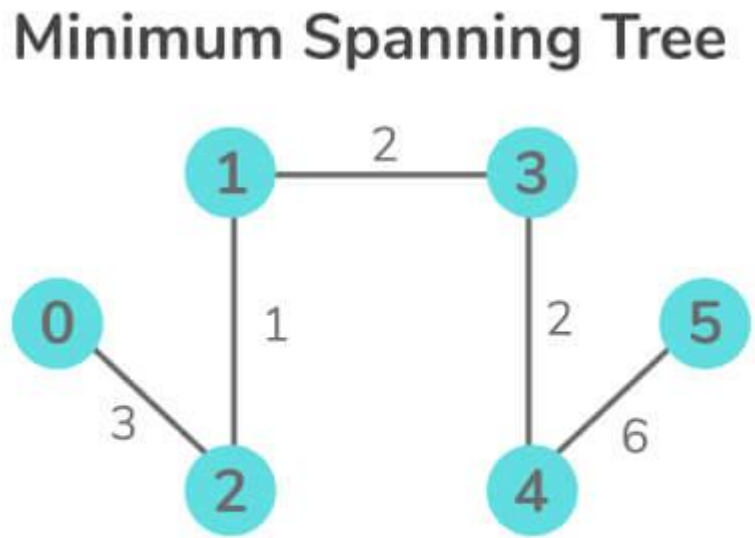
A - Greedy paradigm
B - Backtracking paradigm
C - Dynamic Programming paradigm
D - Divide and Conquer paradigm

Answer : A

Dijkstra relates to the greedy approach since we select the node with the shortest distance from the set of unvisited nodes.

Question:
Find out the MST of the given graph.

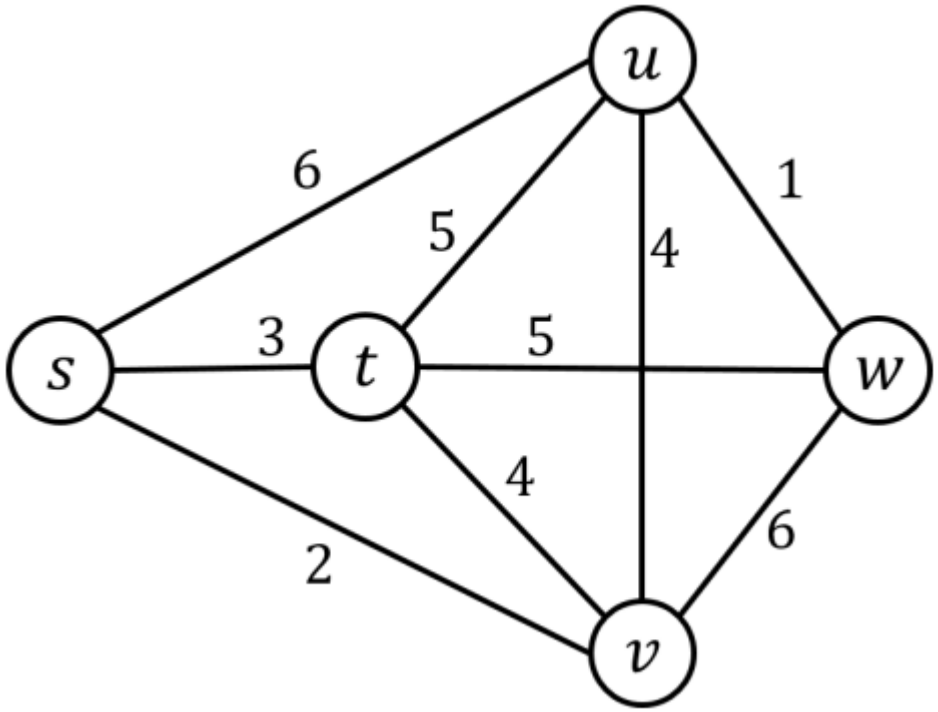




InterviewBit

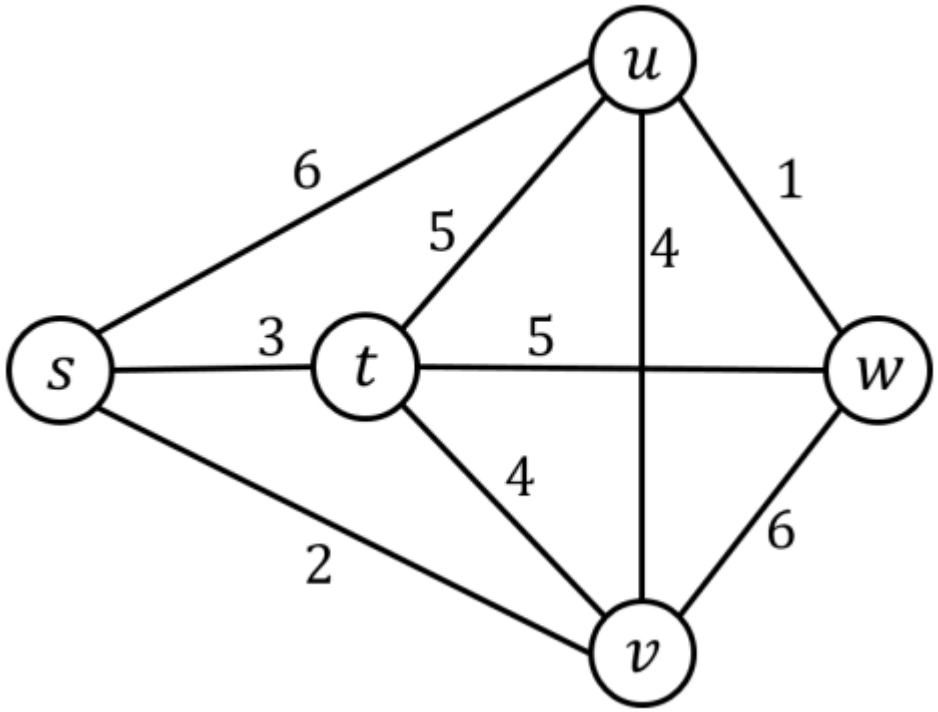
Question:

Draw the minimum spanning tree that would result from running Prim's algorithm on this graph, starting at vertex s. List the order in which edges are added to the tree.

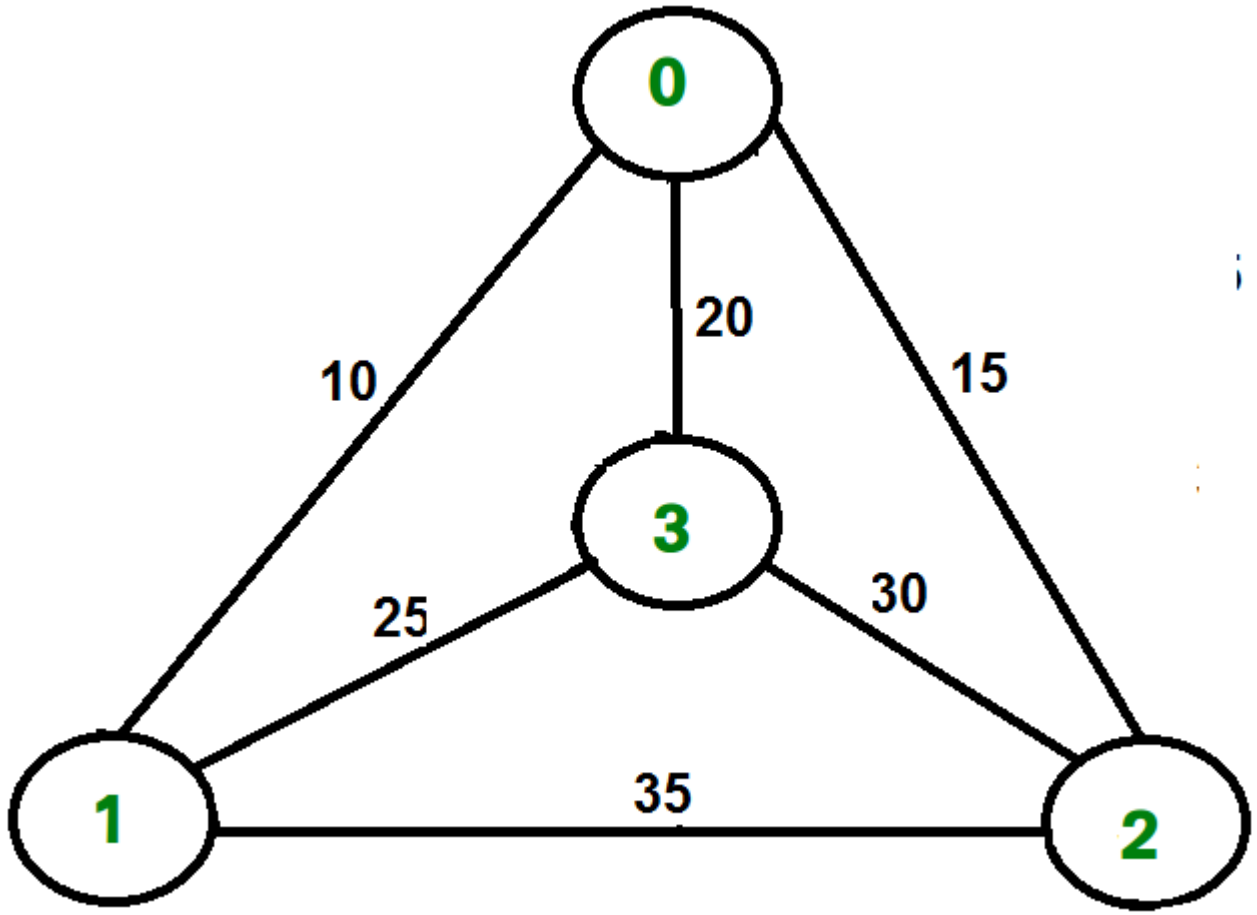


Question:

Draw the minimum spanning tree that would result from running Kruskal's algorithm on this graph. List the order in which edges are added to the tree.



Question:
Given a set of cities and distance between every pair of cities, the problem is to find the shortest possible tour that visits every city exactly once and returns to the starting point.



Obtain reduced cost matrix for travelling sales person problem. Consider the instance define by the cost matrix:

$$\begin{bmatrix} \infty & 5 & 1 & 10 & 6 \\ 1 & \infty & 4 & 12 & 7 \\ 3 & 6 & \infty & 4 & 16 \\ 7 & 1 & 3 & \infty & 9 \\ 16 & 12 & 7 & 6 & \infty \end{bmatrix}$$

Find optimal solution to the knapsack problem instance n=6, m=15, (p1…p6) = (10,5,15,7,6,18), (w1…w6) = (2,3,5,7,1,4)