Q.6 Prim's Algorithm For MST.

⇒ i) A spanning tree of a connected g undirected graph G, is a sub-graph of G which is a tree that connects all vertices together.

ii) A MST is defined as a spanning tree with weight less than or equal to the weight of every other spanning tree.

iii) Prim's algorithm is a greedy algorithm that used to form MST for a connected weighted undirected graph.

iv) In other words, the algorithm builds a tree that includes every vertex and a subset of the edges in such a way that the total weight of all the edges in the tree is minimized.

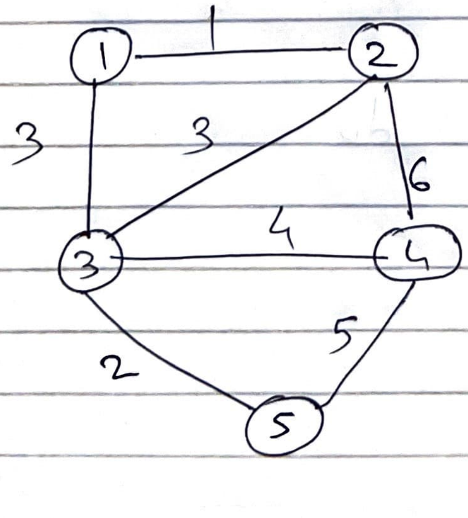
• Algorithm :

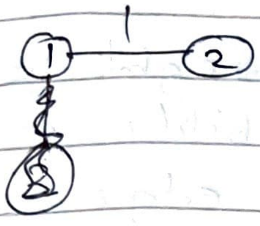
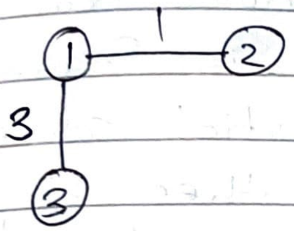
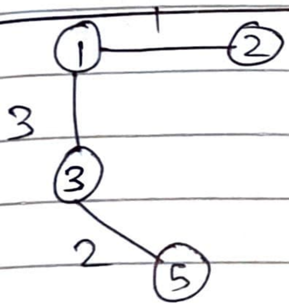
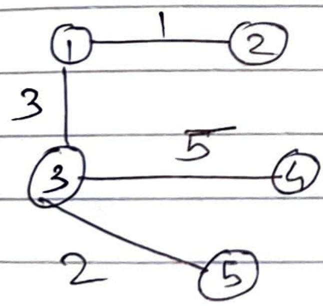
```
PRIM (G,n)
{

cost ← 0

for i ← 1 to n-1 do
    MST[i] ← 0

MST[0] ← 1
```

```
for k ← 1 to n do
    min_dist ← ∞
    for i ← 1 to n-1 do
    for j ← 1 to n-1 do

    if a[i, j] AND((MST[i] AND MST[j])
        OR (MST[i] AND MST[j])) then

        if a[i, j] < min_dist then
            min_dist ← a[i, j]
            u ← i
            v ← j

print (u, v, min_dist)
MST[u] ← MST[v] ← 1
cost ← cost + min_dist

print ("Total cost = ", cost)
}
```

- The time complexity is $O(n^2)$.

- For e.g.,



Partial solution	Set of neighbour edges	Cost	Updated U_V
1) $U_V = \{1, 2, 3, 4, 5\}$ Initial partial solution 1	<1, 2> <1, 3>	1 3	$\{2, 3, 4, 5\}$
2) 	<1, 3> <2, 3> <2, 4>	3 3 6	$\{3, 4, 5\}$
3) 	<2, 3> <2, 4> <3, 4> <3, 5>	3 6 4 2	$\{4, 5\}$
4) 	<2, 4> <3, 4> <5, 4>	6 4 5	$\{4\}$
5) 			

$\therefore$ Cost of solution

$= W(1,2) + W(1,3) + W(3,4) + W(3,5)$

$= 1 + 3 + 5 + 2$

$= 10$

Q.7 Kruskal's Algorithm

⇒ i) Kruskal's algorithm is used to find the MST for a connected weighted graph.

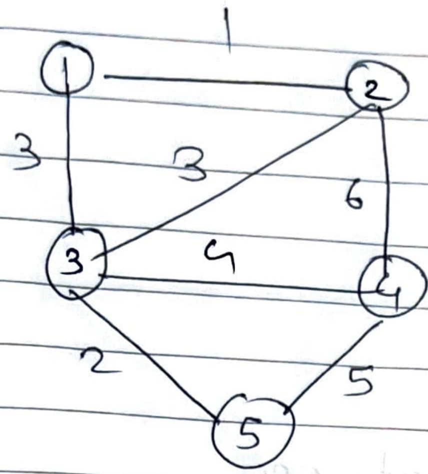
ii) This algorithm first sorts all the edges in non-decreasing order of their weight.

iii) Edge within minimum weight is selected and its feasibility is tested.

iv) If inclusion of the edge to a partial solution does not form the cycle, then the edge is feasible and added to the partial solution.

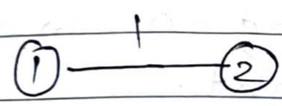
v) If it is not feasible then skip it and check for the next edge.

For e.g.,

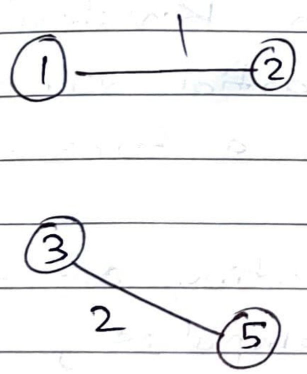


Edge	<1,2>	<3,5>	<1,3>	<2,3>	<3,4>	<4,5>	<2,4>
cost	1	2	3	3	4	5	6

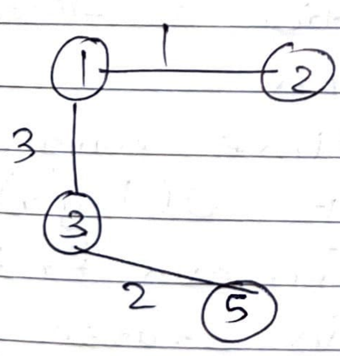
1)



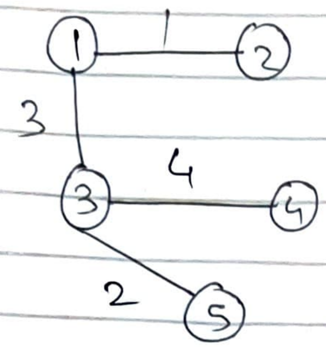
2)



3)



<2,3> edges creates cycle so remove it



Cost = 10 //

Difference between

Prim's Algorithm	Kruskal's Algorithm
i) Prim's algorithm always chooses the next edge which is a neighbour of vertices in partially generated solution.	i) The Kruskal algorithm chooses the edge having a minimum weight.
ii) Prim's algorithm ensures that partial solution is always a tree.	ii) In Kruskal's algorithm a partial solution can be a forest.
iii) Sorting of edges is not required.	iii) Sorting of edges is compulsory.
iv) In prim's algorithm graph must be a connected graph.	iv) Kruskal's can to function on disconnected graph.
v) Time complexity of $O(n^2)$	v) Time complexity of $O(\log V)$.
vi) Can run faster in dense graph	vi) Can run faster in sparse graph.