Shri Vile Parle Kelavani Mandal’s

Institute of Technology Dhule

Department of Information Technology

Design and Analysis of Algorithms lab

Name:-Mrunali Dipak Patil
Class:-SYIT
Roll no:-68
PRN:-23054491246503

Problem Statement :- Find minimum spanning tree of given connected undirected graph using the Prim’s Algorithm.

Theory :

Prim's algorithm is a greedy algorithm that finds a minimum spanning tree for a weighted undirected graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized.

The method of making minimum spanning tree from Prim's algorithm is like Dijkstra's algorithm for shortest paths. Each vertex is given a status, which can be permanent or tempõrary.

Initially all the vertices are temporary and at each step of the algorithm, a temporary vertex is made permanent. The process stops when all the vertices are made permanent. Making a vertex permanent means that it has been included in the tree. The temporary vertices are those vertices which have not been added to the tree.

We label each vertex with length and predecessor, The label length represents the weight of the shortest edges connecting the vertex to a permanent vertex and predecessor represents that permanent vertex. Once a vertex is made permanent,it is not relabeled. Only temporary vertices will be relabeled if required.

Applying the greedy approach, the temporary vertex that has the minimum value of length is made permanent. In other words we can say that the temporary vertex which is adjacent to permanent Vertex by an edge of least weight is added to the tree.

The Step for making a minimum spanning tree by Prim's algorithm are as :

A) Initialize the length of all vertices to infinity and predecessors of all vertices to NIL. Make the status of all vertices temporary.

B) Select any arbitrary vertex the root vertex and make its length label equal to zero.

C)From all the temporary vertices in the graph,find out the vertex that has smallest value of length, make it permanent and now that is our current vertex(If there many with the same value of length then anyone can be selected)

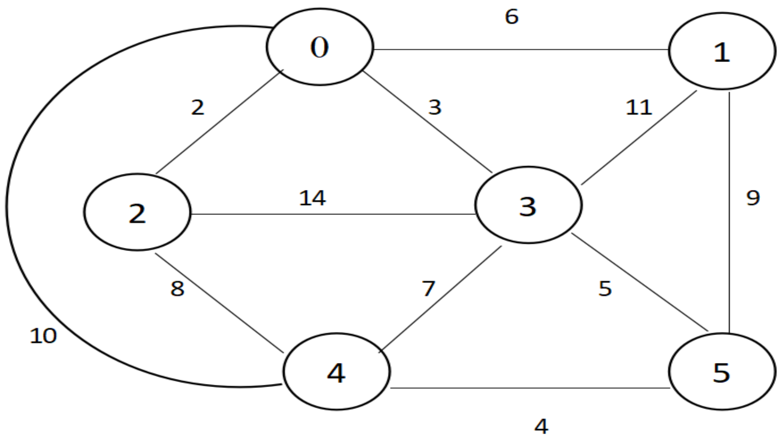
D)Examine all the temporary vertices adjacentto the current vertex. Suppose Current is the vertex and v is a temporary vertex adjacent to current.

```
(i) If weight (current ,v)<length(v)
        Relabel the vertex v
    Now length(v) = weigth (current,v)
        Preddecessor(v) =current
(ii) If weigth(current ,v)>=length(v)
        Vertex v is not relabeled
```

E) Repeat step (C) and (D) till there are no temporary vertices left,or all the temporary length equal to infinity.If the graph is connected then thevprocedure will stop when all n vectors have been made and n-1 edgesnare added to the spanning tree.If the graph is not connected in the then those vertices that are not reachable for the roots vertex will be remain temporary with the length infinity.In this case with no spanning tree it's possible

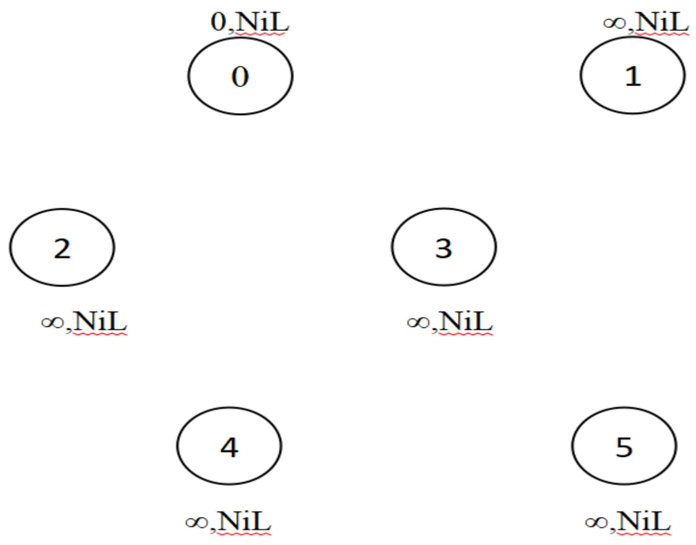
Solution :

Let us take an undirected connected graph and construct the minimum spanning tree.



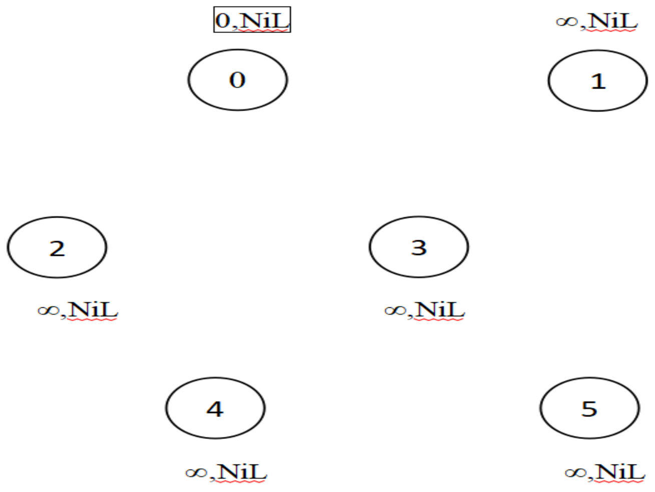
Initially length value for all the vertices are set to very large number suppose infinity is such a number.We have take predecessor of all vertices NIL in beginning. Initially all vertex are temporary. We select the first vertex 0.

Step 1:-



Vertex	Length	Predecessor	Status
0	0	NIL	Temp
1	∞	NIL	Temp
2	∞	NIL	Temp
3	∞	NIL	Temp
4	∞	NIL	Temp
5	∞	NIL	Temp

From all the temporary vertices, vertex 0 has the smallest length, so it will be made permanent. This is the first vertex to be included in the tree. Its predecessor will remain NIL. Now vertex 0 is the current vertex.



vertex	Length	Predecessor	Status
0	0	NIL	Perm
1	∞	NIL	Temp
2	∞	NIL	Temp
3	∞	NIL	Temp
4	∞	NIL	Temp
5	∞	NIL	Temp

Vertices 1, 2, 3, 4 are temporary vertices adjacent to 0

weight(0,1) < length(1) 6 < ∞ Relabel 1

predecessor[1]=0, length[1]=6

weight(0,2) < length(2) 2 < ∞ Relabel 2

predecessor [2] = 0 , length[2]=2

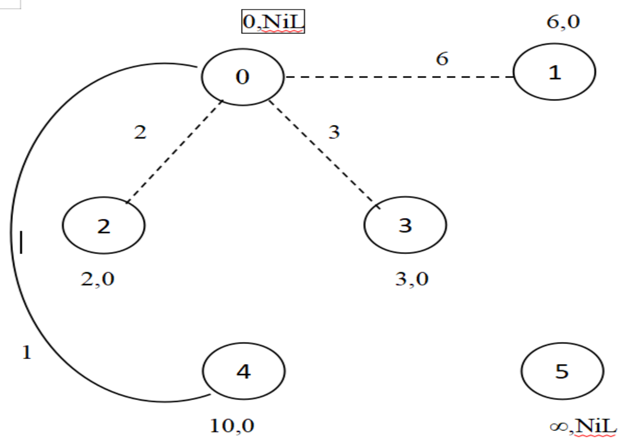
weight(0,3) < length(3) 3 < ∞ Relabel 3

predecessor [3] = 0 , length[3]=3

weight t(0, 4) < length(4) 10 < ∞ Relabel 4

predecessor r[4] = 0 length[4]=1

Step: 2



Vertex	Length	Predece -ssor	Status
0	0	NIL	Perm
1	6	NIL	Temp
2	2	NIL	Temp
3	3	NIL	Temp
4	10	NIL	Temp
5	∞	NIL	Temp

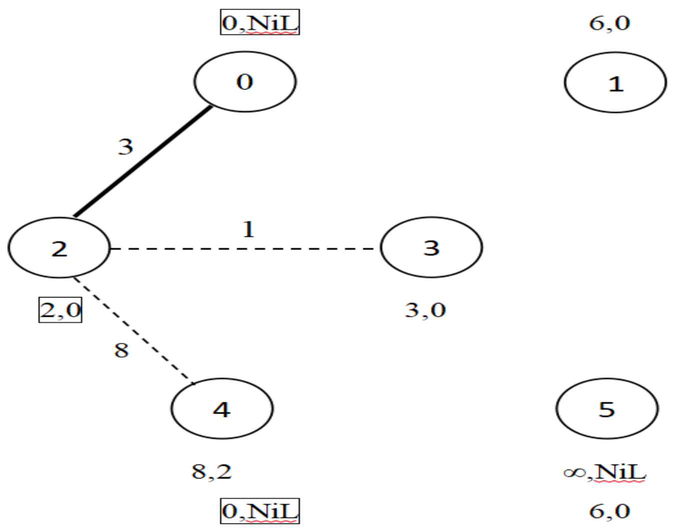
From all the temporary vertices, vertex 2 has the smallest length so make it permanent i.e. include it in the tree. Its predecessor is 0, so the edge that is added to the tree is (0,2). Now vertex 2 is the current vertex. Vertices 3, 4 are temporary vertices adjacent to vertex 2.

weight(2,3) > length(3) 14 > 3 Don't Relabel 3

weight(2,4) < length(4) 8 <10 Relabel 4

predecessor[4]=2, length[4]=8

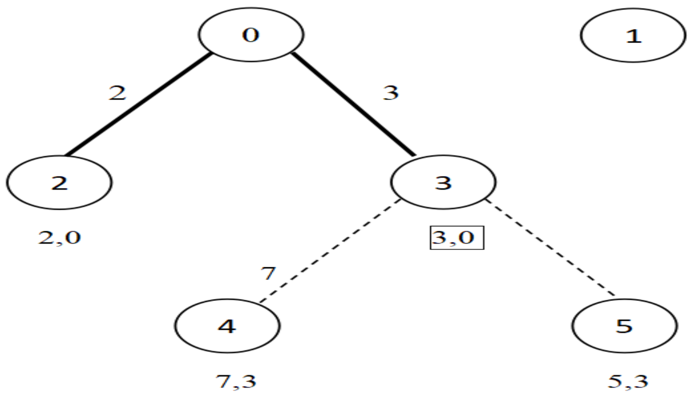
Step:-3



Vertex	Length	Predecessor	Status
0	0	NIL	Perm
1	6	0	Temp
2	2	0	Perm
3	3	0	Temp
4	8	2	Temp
5	∞	NIL	Temp

From all temporary vertices, vertex 3 has the smallest value of length so make it permanent. Its predecessor is 0, so the edge (0,3) is included in the tree. Now vertex 3 is the current working vertex. Vertices 1, 4, 5 are temporary vertices adjacent to vertex 3.
weight(3,1) > length(1) 11>6 Don't relabel I
weight(3,4) < length(4) 7 <8 Relabel 4
predecessor[4]=3, length[4]=7
weight(3,5) < length(5) 5 <∞ Relabel 5
predecessor[5]=3, length[5]=5

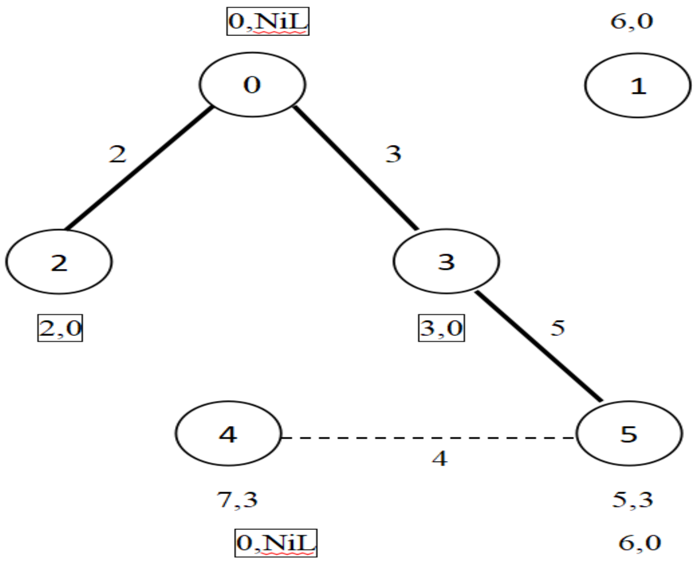
Step:4



Vertex	Length	Predece-ssor	Status
0	0	NIL	Perm
1	6	0	Temp
2	2	0	Perm
3	3	0	Perm
4	7	3	Temp
5	5	3	Temp

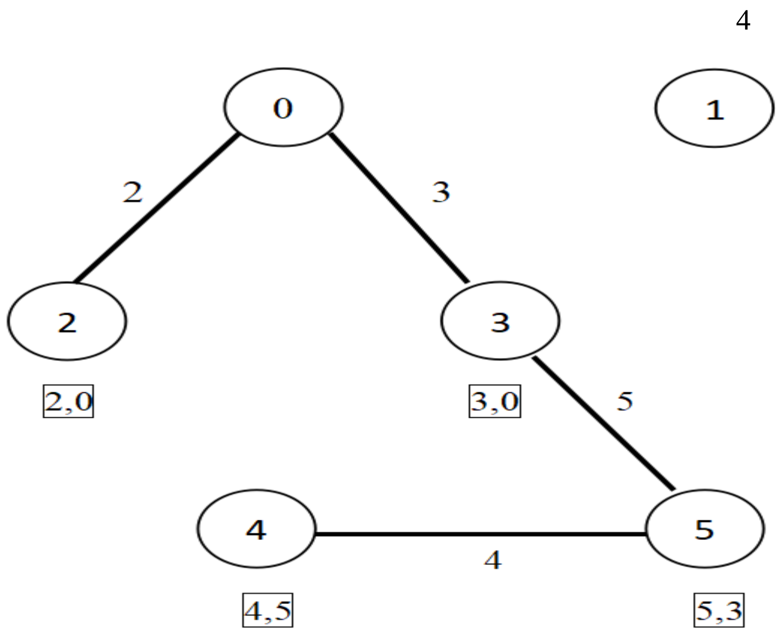
From all temporary vertices, vertex 5 has the smallest length so make it permanent. Its predecessor is 3 so include edge (3,5) in the tree. Now vertex 5 is the current vertex. Vertices 1, 4 are temporary vertices adjacent to vertex 5 weight(5,1) > length(1) 9>6 Don't relabel 1
weight(5,4) < length(4) 4 <7 predecessor[4]=5, length[4]=4 Relabel 4

Step :5



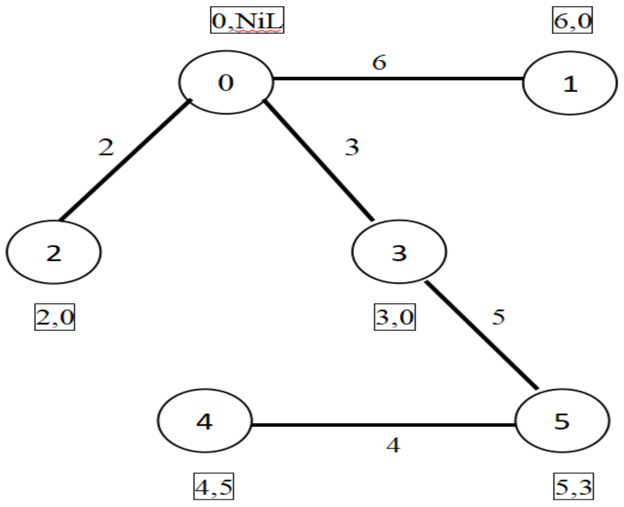
Vertex	Length	Predecessor	Status
0	0	NIL	Perm
1	6	0	Temp
2	2	0	Perm
3	3	0	Perm
4	4	5	Term
5	5	3	Perm

From all temporary vertices, vertex 4 has the smallest length so make it permanent. Its predecessor is 5, so include edge (5,4) in the tree. Now vertex 4 is the current vertex. There are no temporary vertices adjacent to 4



Vertex	Length	Predecessor	Status
0	0	NIL	Perm
1	6	0	Perm
2	2	0	Perm
3	3	0	Perm
4	4	5	Perm
5	5	3	Perm

Vertex 1 is the only temporary vertex left and its length is 6, so make it permanent. Its predecessor is 0, so edge (0,1) is included in the tree.



Now all the vertices are permenant so we stop our procedure.

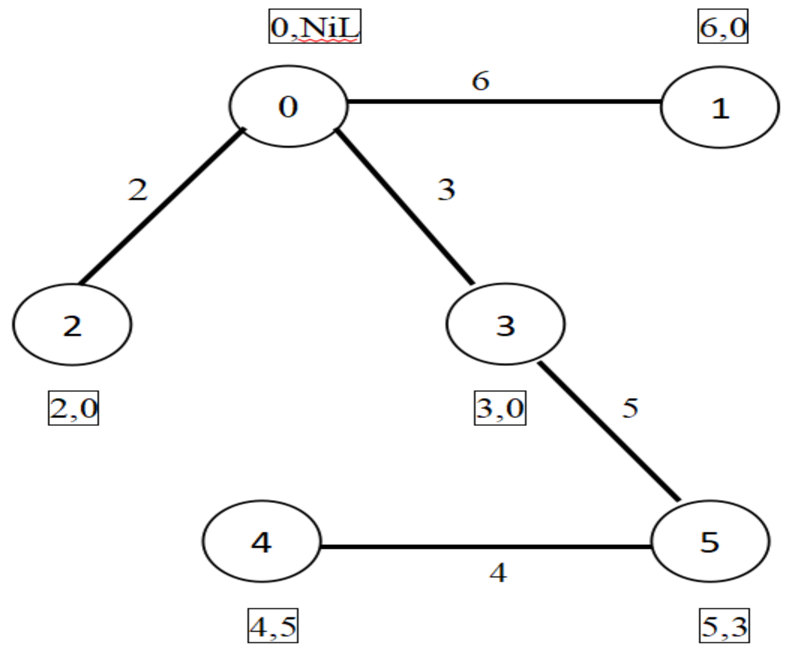
Vertex	Length	Predecessor	Status
0	0	NIL	Perm
1	6	0	Perm
2	2	0	Perm
3	3	0	Perm
4	4	5	Perm
5	5	3	Perm

Now we have a complete minimum spanning tree. The edges that belong to minimum spanning tree are- (0,1), (0,2), (0,3), (5,4), (3.5)

Weight of minimu... spanning tree will be- 6+2+3+4+5=20
Now let us take a graph that is not connected.

After making vertices 0,1,2,3,1 permanent,the situation would be-

Vertex	Length	Predecessor	Status
0	0	NIL	Perm
1	6	0	Perm
2	3	0	Perm
3	4	4	Perm
4	5	2	Perm
5	∞	NIL	Temp
6	∞	NIL	Temp
7	∞	NIL	Temp


0,NiL
6,0
6
0
1
2
3
2
3
2,0
3,0
5
4
5
4
4,5
5,3