26.03.2020 **Prim's Algorithm**

(as Pseudocode

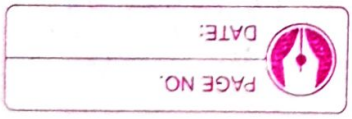
procedure prim (G, w) $\quad G = (V, E)$

Input: A connected undirected graph G with edge weights w.

Output: A minimum spanning tree defined by the array prev.

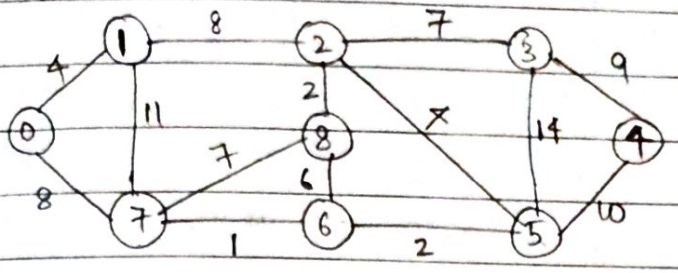
```
for all u ∈ V:
    cost(u) = ∞
    prev(u) = nil
Pick any initial node u0.
Cost(u0) = 0.

H = makequeue(V)    (priority queue, using cost-values as keys).
while H is not empty:
    v = deletemin(H)
    for each {v,z} ∈ E:
        if cost(z) > w(v,z):
            cost(z) = w(v,z)
            prev(z) = v
            decreasekey(H,z)
```



(8) Using the algorithm, find the min. cost spanning tree for the graphs given below:

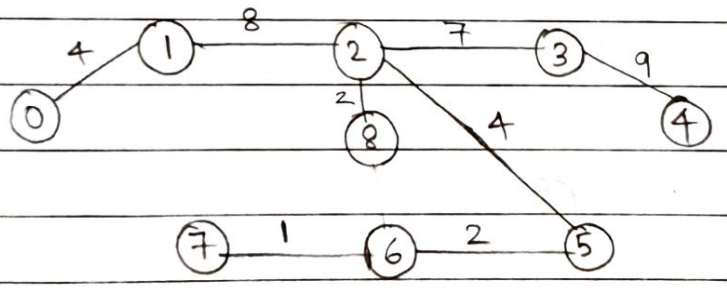
(i)



start from vertex 0.
table has cost/prev values.

Set S	0	1	2	3	4	5	6	7	8
{}	0/nil	∞/nil	∞/nil	∞/nil	∞/nil	∞/nil	∞/nil	∞/nil	∞/nil
0		4/0	∞/nil	∞/nil	∞/nil	∞/nil	∞/nil	8/0	∞/nil
0,1			8/1	∞/nil	∞/nil	∞/nil	∞/nil	11/1	∞/nil
0,1,2				7/2	∞/nil	4/2	∞/nil	∞/nil	2/2
0,1,2,8				7/2	∞/nil	4/2	6/8	7/8	-
0,1,2,8,5				7/2	10/5	-	2/5	7/8	-
0,1,2,8,5,6				7/2	10/5		-	1/6	
0,1,2,8,5,6,7				7/2	10/5			-	
0,1,2,8,5,6,7,3				-	9/3				
0,1,2,8,5,6,7,3,4									

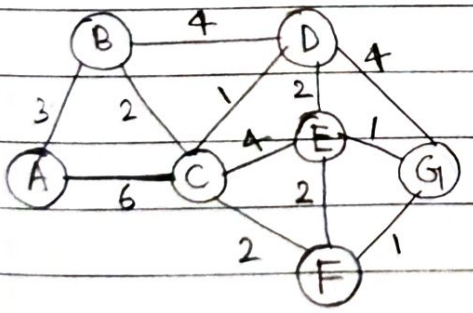
MST:



min. cost = 4+8+2+7+4+9+2+1

= 32

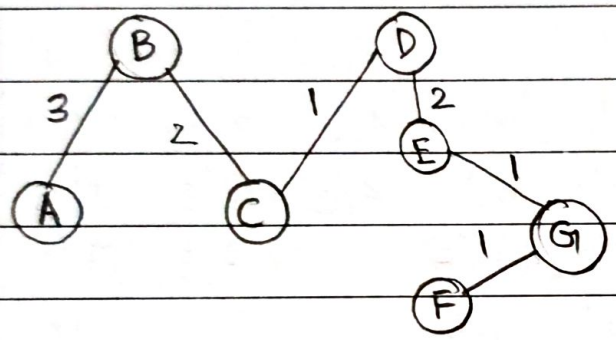
(ii)



start from vertex A.

Set S	A	B	C	D	E	F	G
{}	0/nil	∞/nil	∞/nil	∞/nil	∞/nil	∞/nil	∞/nil
A		3/A	6/A	∞/nil	∞/nil	∞/nil	∞/nil
A, B			2/B	4/D	∞/nil	∞/nil	∞/nil
A, B, C				1/C	4/C	2/C	∞/nil
A, B, C, D					2/D	2/C	4/D
A, B, C, D, E						2/C	1/E
A, B, C, D, E, G						1/G	
A, B, C, D, E, G, F							

MST:



Min. cost: 3+2+1+2+1+1

= 10