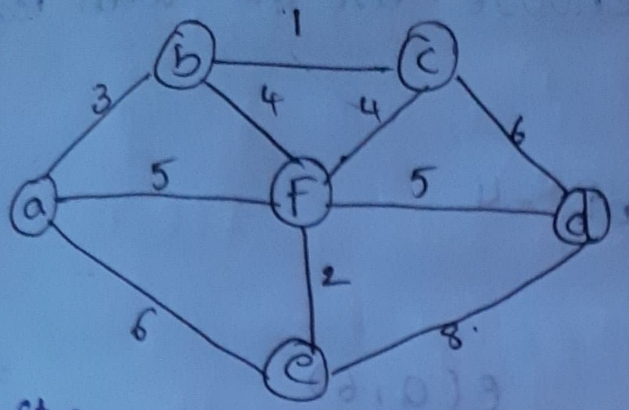
9/12/21

PRIM'S ALGORITHM



vertex (_ , _) → distance

↓

any neighbour in min spanning tree.

→ 1st vertex - a(Null, Null) = a(-, -)

since, it is the 1st vertex it has no adj vertex and no distance.

→ b (a, 3)   b has a, c, f as adj. But in min spanning tree only a is present)

c (-, ∞)

c does not have a neighbour, so it is considered at ∞.

d (-, ∞)   c, f, e are neighbours.

e (a, 6)   a, f, d are neighbours. only a is in min spanning tree.

f (a, 5)

Remaining vertices - Not there in min spanning tree.

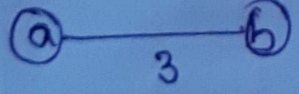
ab - 3   ae - 6   af - 5

b(a,3)   e(a,6)   f(a,5)

↓

Min.

So, b is added to min spanning tree.



→ b(a,3) | c(b,1)   b, f, d are neighbours present in min spanning tree

d(-, ∞)

e(a,6)   a, f, d are neighbours

f(   )   a, b, c, d, e are neighbours)

When more than 1 neighbours are present in min spanning tree, choose the one with least edge weight.

af - 5 bf - 4

f(b, 4)

c(b, 1) f(b, 4) e(a, 6)

Min spanning tree.

a(-, -)
b(a, 3)
c(b, 1)

d(c, 6)
e(a, 6)
f( ) a, b, c in min spanning tree.

af - 5 bf - 4 cf - 4

bf (or) cf can be choosen

f(b, 4).

Least weighted edge - f(b, 4).

Min spanning tree

a(-, -)
b(a, 3)
c(b, 1)
f(b, 4)

e(f, 2) ea - 6 ef - 2 e
d(f, 5) cd - 6 df - 5 d

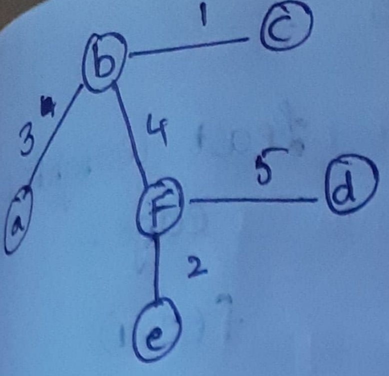
Least weighted edge - e(f, 2)

Min spanning tree

a(-, -) f(b, 4)
b(a, 3) e(F, 2)
c(b, 1)

d(f, 5) df - 5 de - 8 dc - 6

Add to min spanning tree.



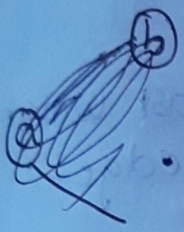
Min spanning tree

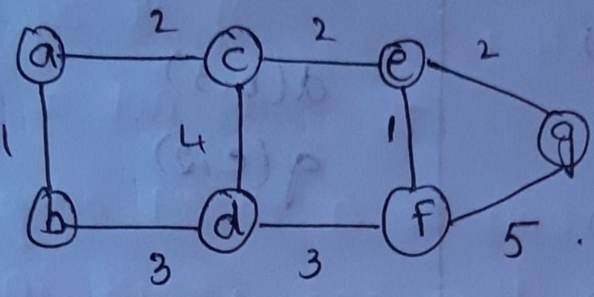
a(-, -) d(f, 5)

b(a, 3) e(f, 2)

c(b, 1) f(b, 4)

Total weights = 1 + 3 + 4 + 5 + 2 = 15





Min spanning tree using Prim's Algorithm.

1st vertex - a(-, -)

Min spanning tree - a(-, -)

b(a, 1)

c(a, 2)

Least weighted edge - b(a, 1).

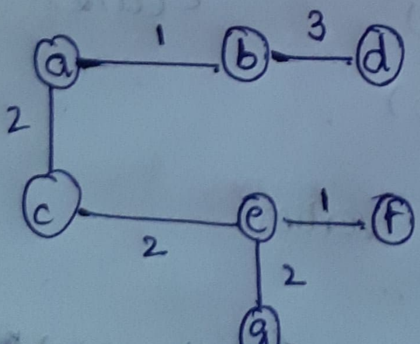
Min spanning tree		
a(-, -)	c(a, 2)	Least weighted edge - c(a, 2).
b(a, 1)	d(b, 3)	
	e(-, ∞)	
	f(-, ∞)	
	g(-, ∞)	

Min spanning tree	Remaining vertices	
a(-, -)	d(b, 3)	Least weighted edge - e(c, 2)
b(a, 1)	e(c, 2)	
c(a, 2)	f(-, ∞)	
	g(-, ∞)	

Min spanning tree	Remaining vertices	
a(-, -)	d(b, 3)	Least weighted edge -
b(a, 1)	f(e, 1)	f(e, 1)
c(a, 2)	g(e, 2)	
e(c, 2)		

Min spanning tree	Remaining vertices	
a(-, -)	d(b, 3)	Least weighted edge -
b(a, 1)	g(e, 2)	g(e, 2)
c(a, 2)		
e(c, 2)		
f(e, 1)		

Min spanning tree	Remaining vertices
a(-, -)	d(b, 3)
b(a, 1)	
c(a, 2)	
e(c, 2)	
f(e, 1)	
g(e, 2)	



Min spanning tree -

a(-, -), b(a, 1)

c(a, 2), d(b, 3)

e(c, 2), f(e, 1)

g(e, 2)

Sum of weights = 1+3+2+1+2+2 = 11

ALGORITHM Prim(G)

Input - A weighted connected graph $G = (V, E)$

V_T - vertices in min spanning tree

E_T - edges in min spanning tree.

Output - E_T, the set of edges composing a min spanning tree.

$V_T \leftarrow \{v_0\}$, the set of tree vertices can be initialized ~~by~~ with any vertex.

$E_T \leftarrow \phi$

for $i \leftarrow 1$ to $|V| - 1$ do

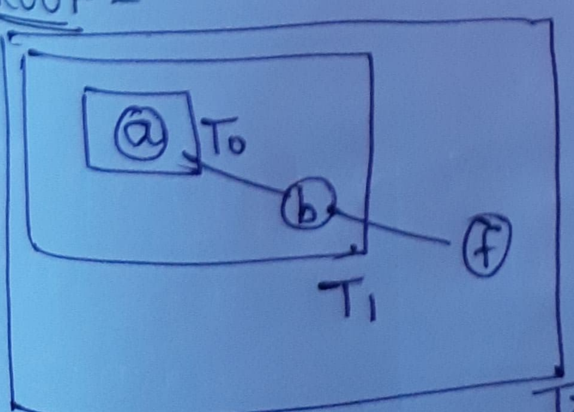
find a minimum-weight edge $e^* = (v^*, u^*)$ among all the edges (v, u). such that v is V_T and u is in $V - V_T$.

$V_T \leftarrow V_T \cup \{u^*\}$

$E_T \leftarrow E_T \cup \{e^*\}$

return E_T.

PROOF -



~~from~~ To T_{i-1} we add an elt to get T_i.

Prove that T_i gives min spanning tree.