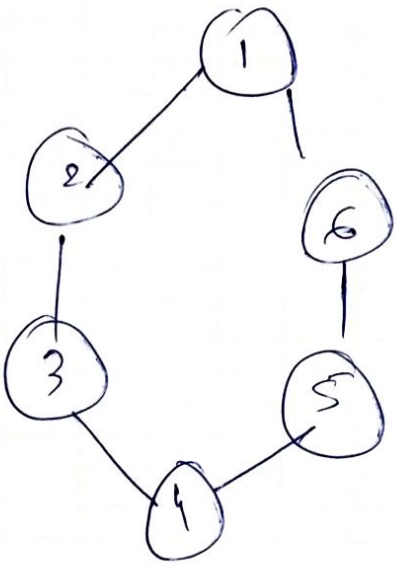
⇒ Spanning Tree.

⇒ Minimum Cost Spanning Tree

Assume example =

Graph G



G = (V, E)

V = { 1, 2, 3, 4, 5, 6 }

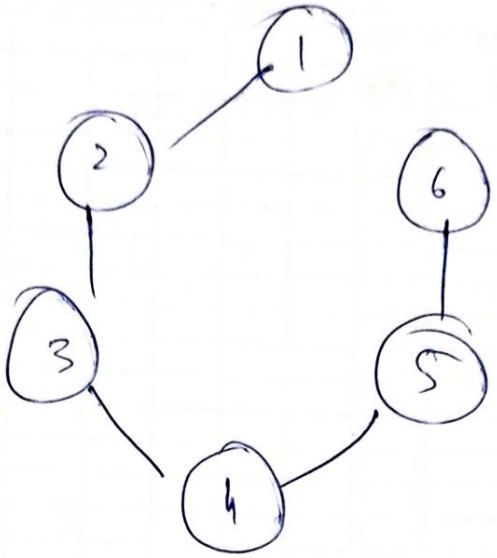
E = { (1,2), (2,3) ... }

So

Spanning tree is subset of graph. with some vertices & some edges.

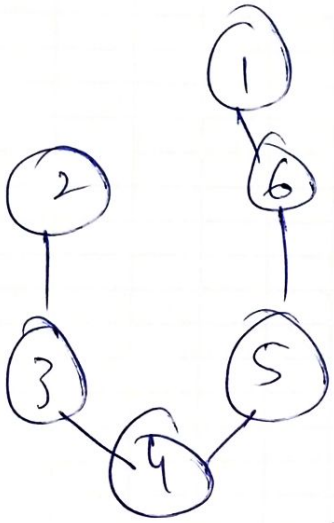
So $E = |V| - 1$

5a



=> Spanning tree

Another spanning tree.



=> spanning tree

Spanning tree has no cycle.

$S \subseteq G$

↓ subgraph

$S = (V', E')$

$V' = V$ & $E' = |V'| - 1$

⇒ How many ~~sub~~ spanning tree can be generated

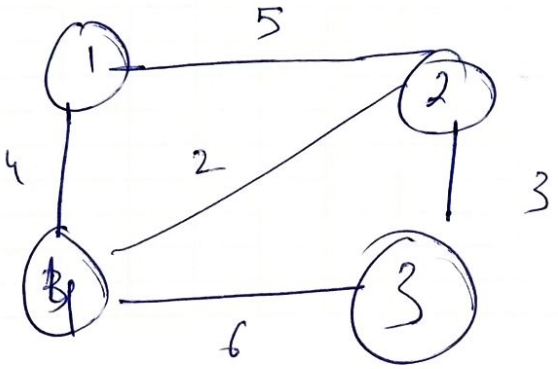
$$|S| = |E| C_{|V|-1} - \text{no. of cycles}$$

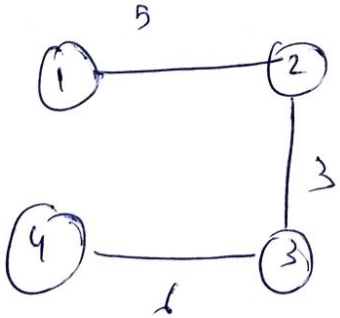
So $6C_5 - 0$ $\quad nC_r = \frac{n!}{r!(n-r)!}$

$=$

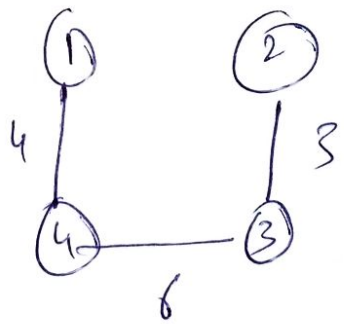
Cost

Minimum Spanning Tree

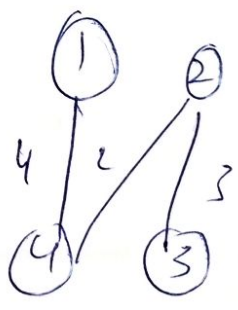




Cost = 14



13



9

⇒ All spanning tree have diff cost.

So can we find ~~Min~~ MST ?

Yes

① method, find all spanning tree & find the smallest,

2nd

Greedy Algos, which will help finding MST without finding all

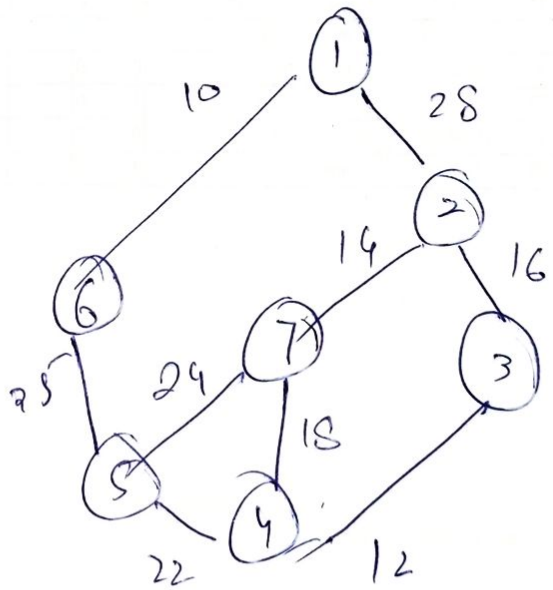
① → Prims Algo

② → Kruskals Algo.

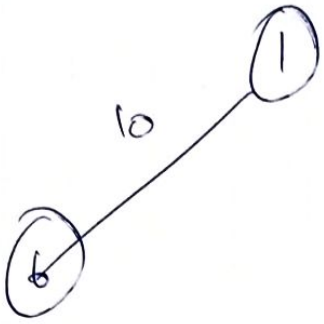
e.g.

Prims Algo.

How it works

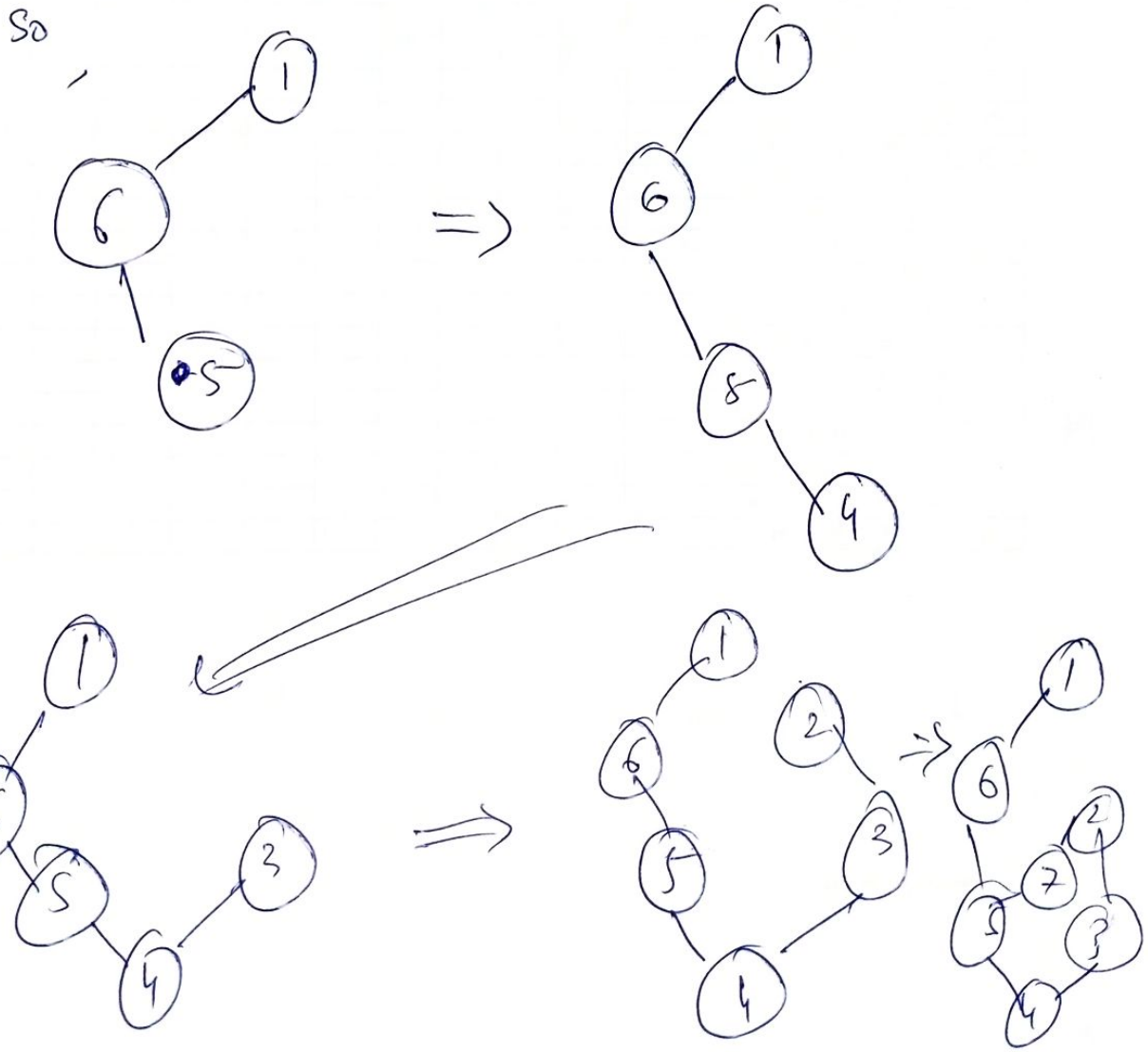


Identify minimum Cost edge



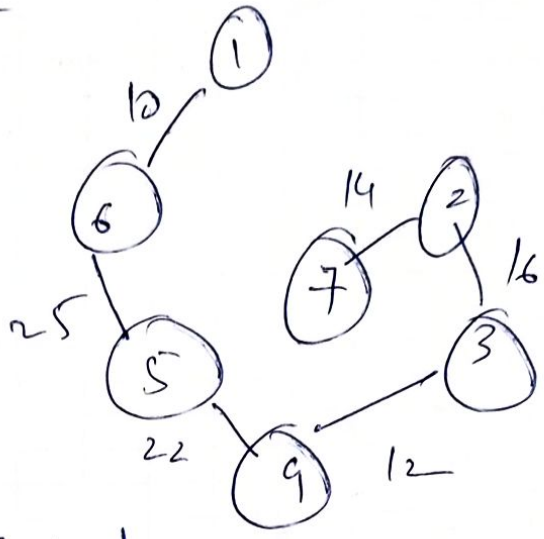
(2) Find minimum edge connected to already selected vertices.

So,



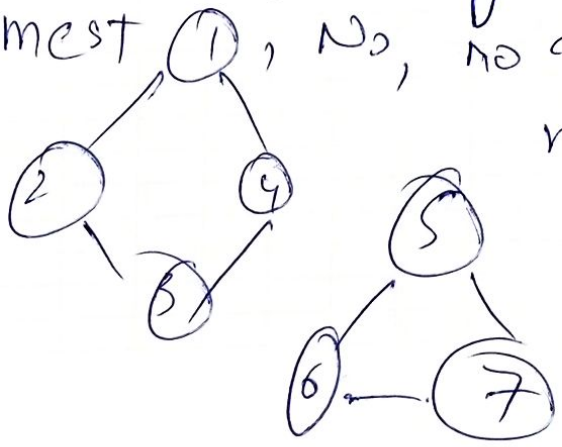
80

MST



Cost of Tree = 99.

If we have following graph can we find MST, No, no algo can because MST must be connected



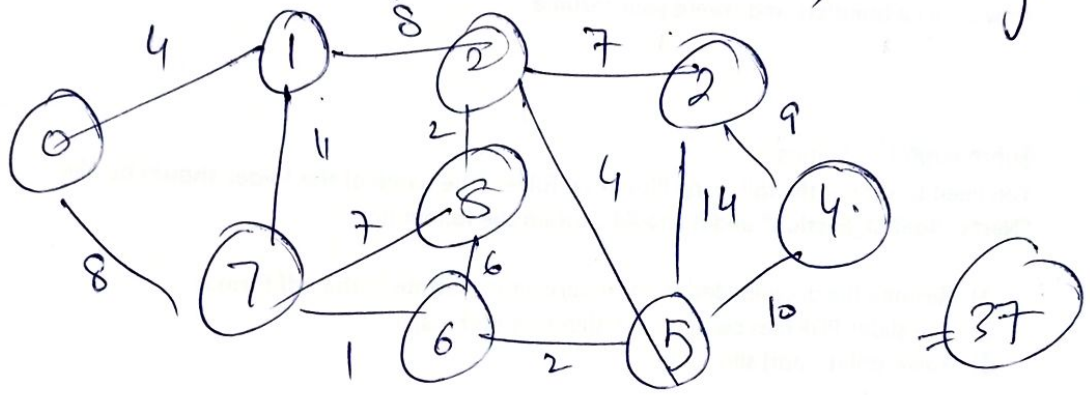
If we use prim, it will find 1 Component spanning tree.

Prim execute

(1) → Select a vSet (empty list) to contain MST vertices.

(2) → use infinite key for all & zero for starting

(3) → pick vertex
Include to vSet
update keys. if $w(u,v) <$ key.



vertex	0	1	2	3	4	5	6	7	8
key	0	∞	∞	∞	∞	∞	∞	∞	∞
	0	4	∞	∞	∞	∞	∞	8	0
	0	4	2	∞	∞	∞	∞	8	∞
(7)	0	4	8	—	∞	∞	1	8	7
	0	4	8	∞	∞	2	1	8	6
	0	4	4	14	10	2	1	8	6
	0	4	4	7	10	2	1	8	2
	0	4	4	7	9	2	1	8	2

6.8.8

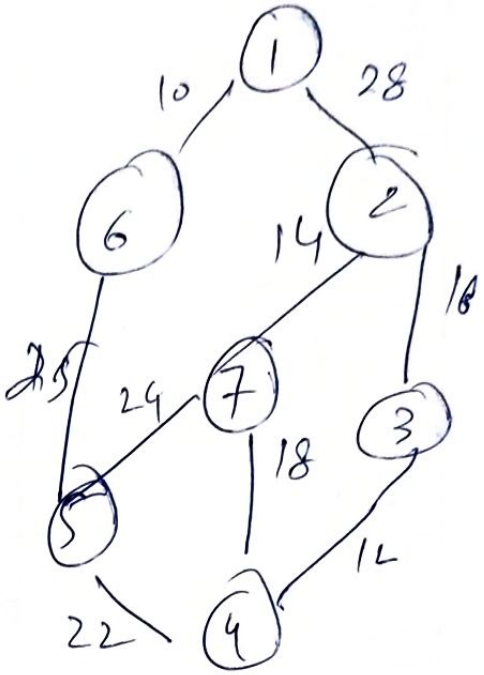
If adjacency matrix

$O = V^2$

If is the $E \log V$ with binary heap

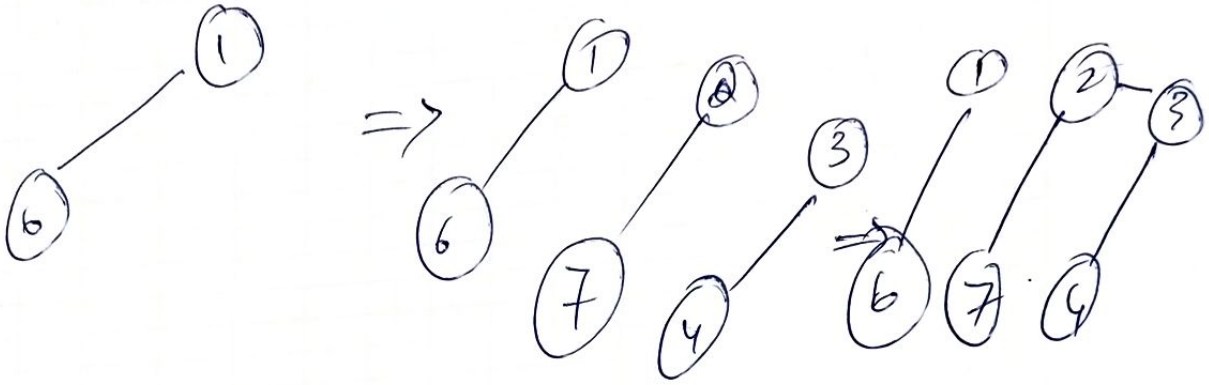
Krusked ns Algo

Also Greedy

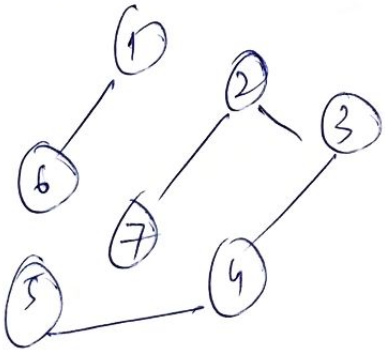


⇒ Always select minimum cost edge.

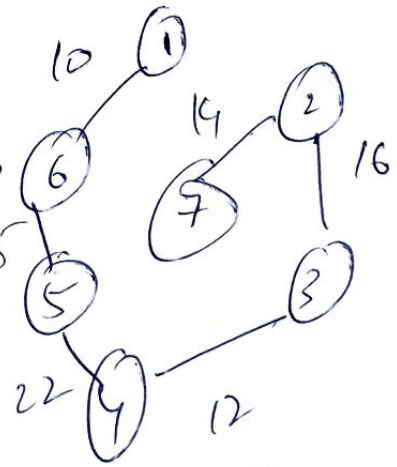
So



After this, the edge is 7 – 4 but it creates cycle, so avoid it, discard it.



⇒ 7-5 discard.



Cost = 99.

V. $E = V-1$

time by kruskal = $m \times n = N^2$.

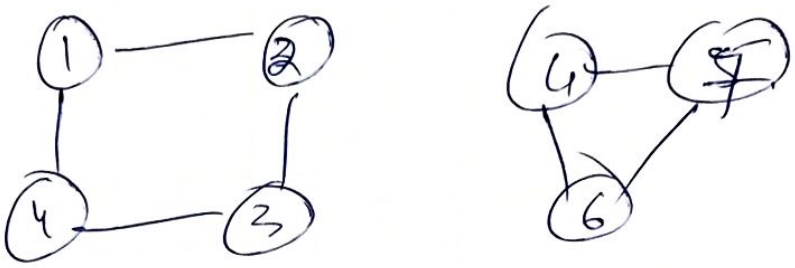
can be improved

How ?

with min heap. Always have sorted edges.

$n \log n$

If graph is not connected



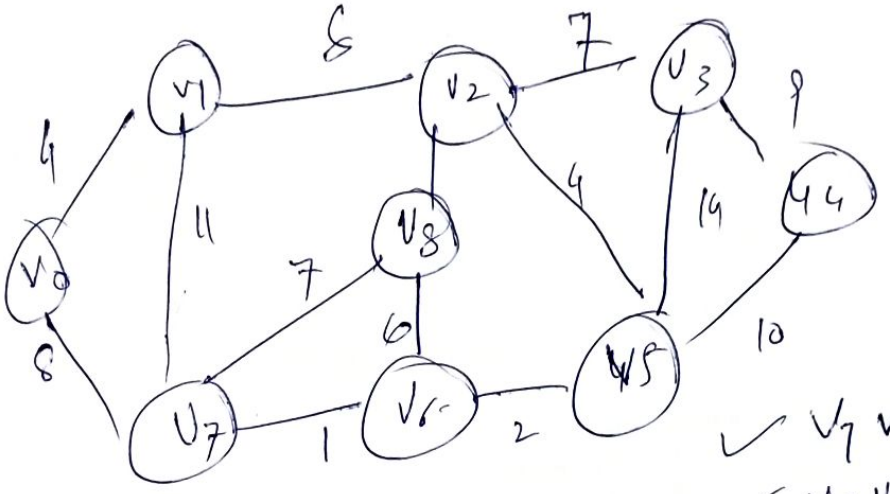
spanning tree is not possible but it find possible. ~~edges~~ need for components

Both ~~word~~ work with -ve weights

→ Sort all edges

→ Pick smallest edge, if cycle not created, add it.

→ Repeat until V-1 edge.



Cost 37

✓ $v_7 v_6$
✓ $v_8 v_2$
✓ $v_6 v_5$
✓ $v_0 v_1$
✓ $v_2 v_5$
✗ $v_8 v_1$
✓ $v_2 v_3$
✗ $v_7 v_8$
✓ $v_0 v_7$
✗ $v_1 v_2$
✓ $v_3 v_4$
$v_5 v_4$
$v_1 v_7$
$v_3 v_5$

Terminate