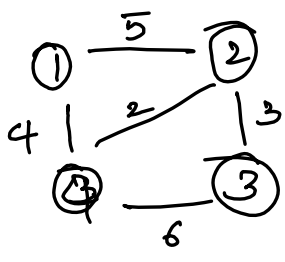
Minimum Spanning trees :-

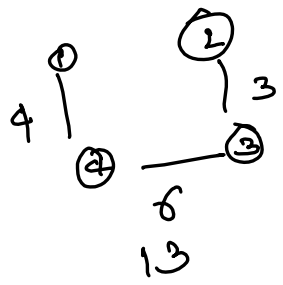
↳ weight is minimised.

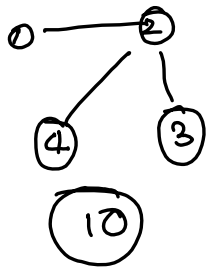
→ Shortest path b/w 2 points in graph

→ Remove all the redundant edges.

→ Edges = n-1

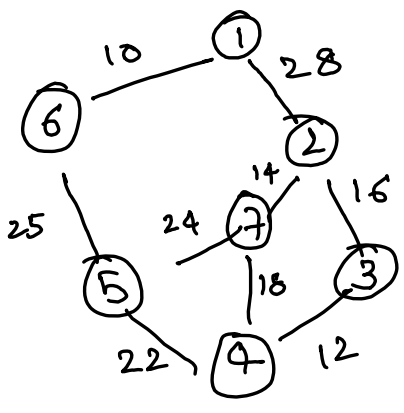


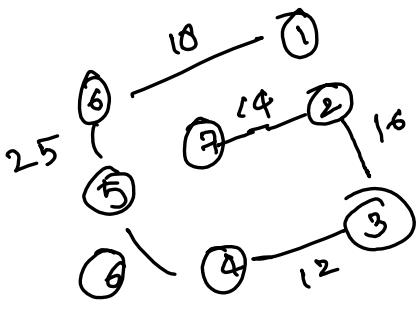




for lengthy trees.

(1) Kruskal's Algorithm :-





→ shouldn't form a cycle.

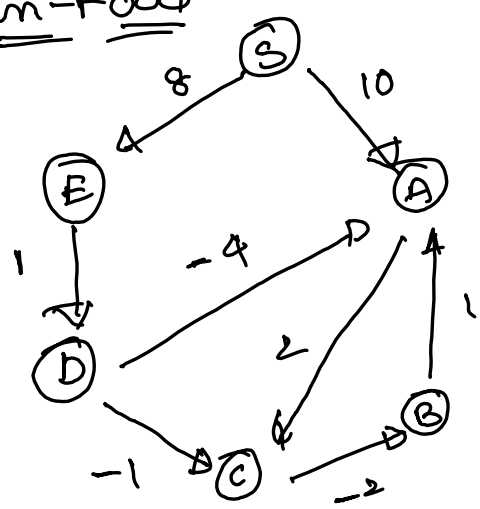
→ take min weight edges first

* Prim's :-

→ Start with a vertex and go with min weight adjacent edge.

* Single Source shortest path.

Bellman-Ford



	10	10	12	9	8
0	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~
S	A	B	C	D	E

1st iter :- Start with S

next A: 10+2 =12

next B: SKIP

next C: ~~B~~ 10

next D: SKIP

next E: for D 8+1 = 9

2nd iter :

0	10	10	~~12~~ 8	9	8
S	A	B	C	D	E

A, B, C → not changing anything

D: C 9 - 1 = 8 i.e., C

~~E~~: A 9 - 4 = 5 i.e., A

Repeat