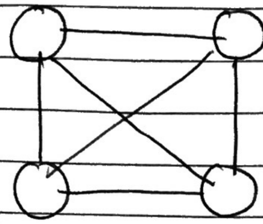
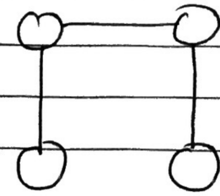
e.g. edges → communication links
minimum → links

Minimum Cost Spanning Tree



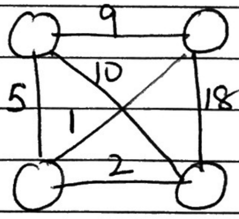
Undirected
Connected
graph G



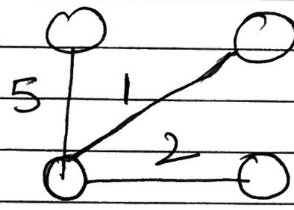
A spanning tree T of G
(no cycle)

$n = 4$

edges in T $= n-1 = 3$



Graph G

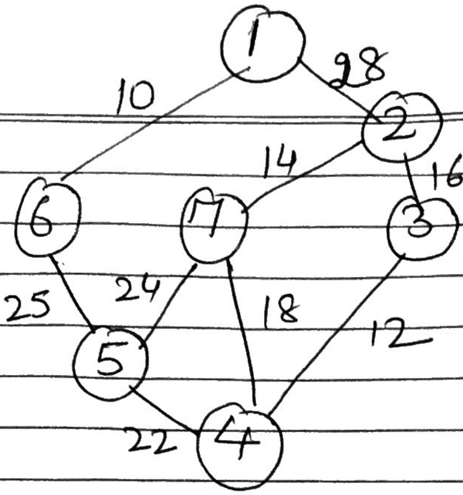


Minimum Cost
Spanning Tree of G
(no cycle)

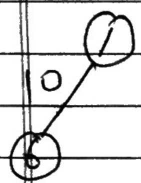
Greedy Algorithms
to
Solve Minimum Cost Spanning
Tree

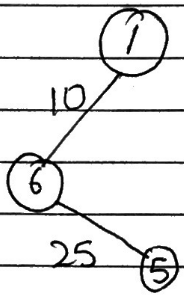
Prim's Algorithm

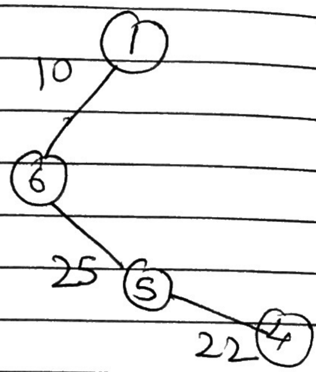
Kruskal's Algorithm

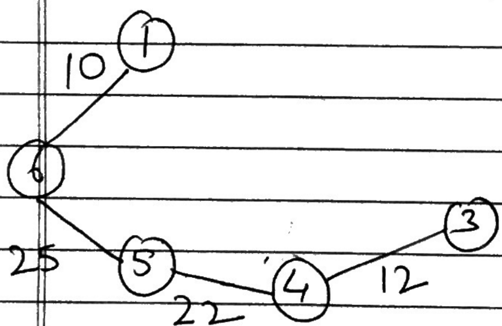


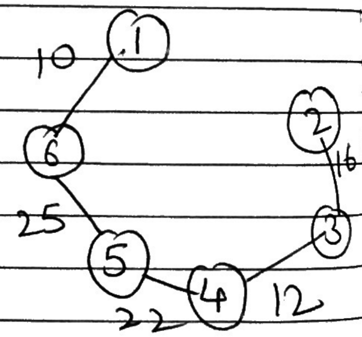
Prim's Algorithm :-

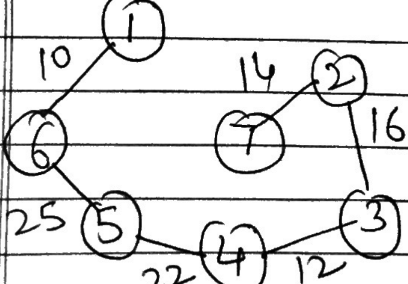












Minimum Cost Spanning Tree (cost = 99)

Time complexity :-

If heap (min) is used $\Rightarrow O(E \log V)$

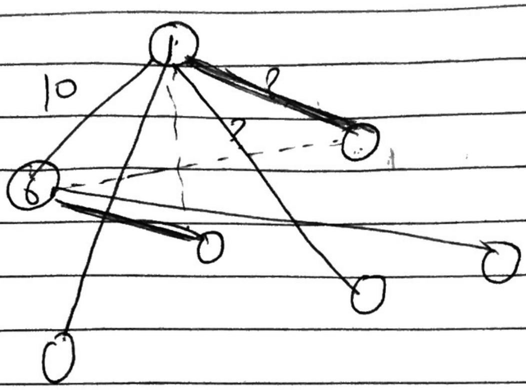
$E \rightarrow$ edges

$V \rightarrow$ vertices

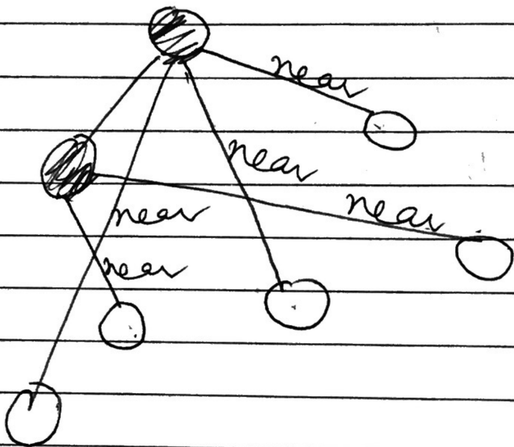
(using binary heap)

Analysis of Prim's Algorithm

min



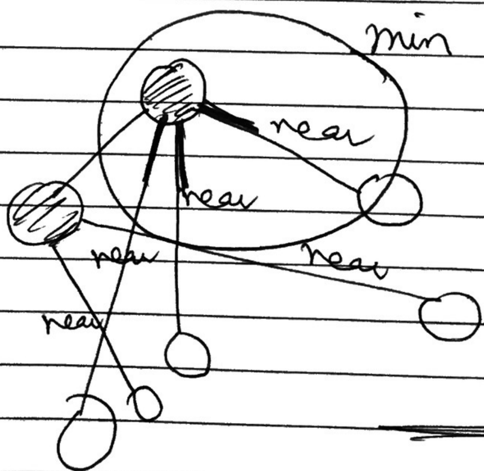
near



Binary Heap

$O(\log V)$

min ? cost?



$O(E)$

$O(E \log V)$

E → edges

V → vertices