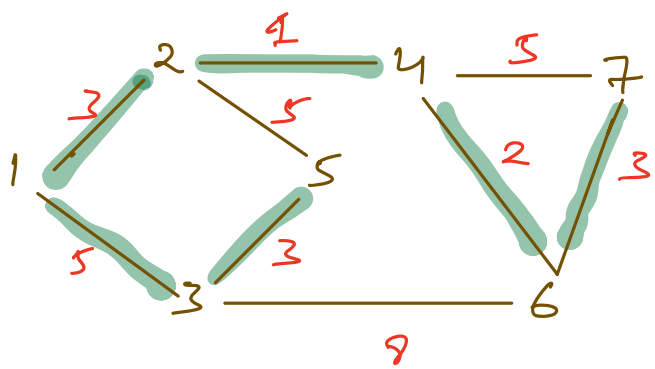
Welcome ☺

Agenda : Prim's algorithm
Minimum Spanning Tree (MST)
Dijkstra's algo.

Q N delivery centres of flikart.
Find min. cost of constructing roads b/w centres such that it is possible to travel from centre to another.

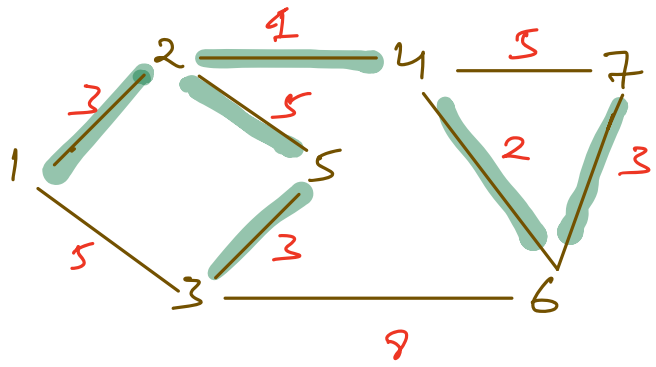
eg: N = 7

1 —3— 2
1 —5— 3
2 —1— 4
2 —5— 5
3 —3— 5
4 —2— 6
3 —4— 6
4 —5— 7
6 —3— 7



Q In a connected graph with N nodes, what is the # edges possible?

Ans = N-1 (Trees)



Minimum Spanning Tree ⇒ Tree generated from a connected weighted graph s.t all nodes are connected and sum of weight of all selected edges is minimum.

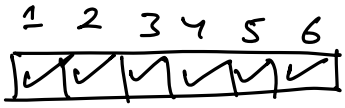
⇒ multiple MST possible for graph

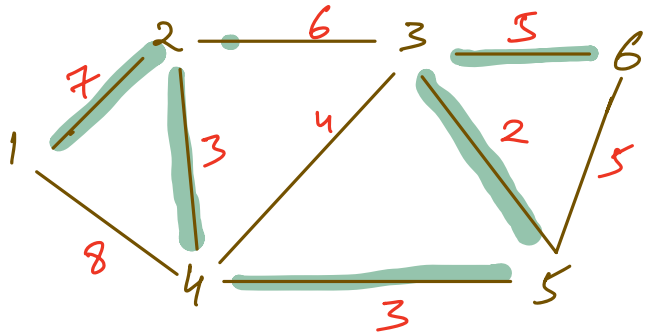
⇒ Graph with unique weights ⇒ unique MST

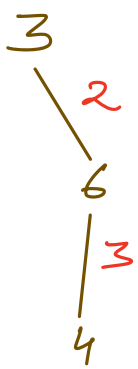
Algo. to find MST ⇒ 1) Prim's algo. ✓

2) Kruskal's algo. (Adv. 4.2)

Prim's algo.







sum = 20

1: Choose any node and insert all possible edges from that node in the min. heap.

Code

```
ans = 0
while ( ! heap.isEmpty() )          < weight, Node >
{
    p = heap.getMin()
    v = p.second
    if ( vis[v] == true )
        continue;

    vis[v] = true;
    ans += p.first

    for ( (u,w) in adj[v]
    {
        if ( ! vis[w] )
        {
            heap.add ( {w, u} )
        }
    }
}
```

10:23

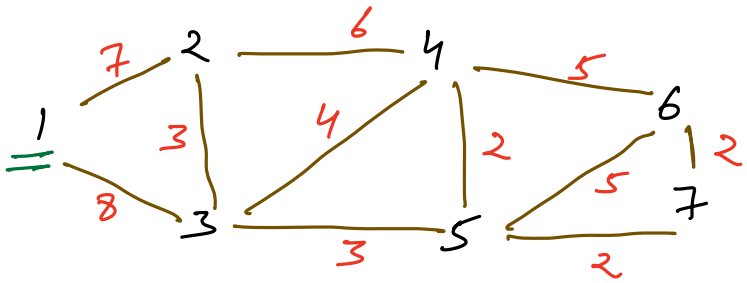
T.C ⇒ $O(E \log E)$

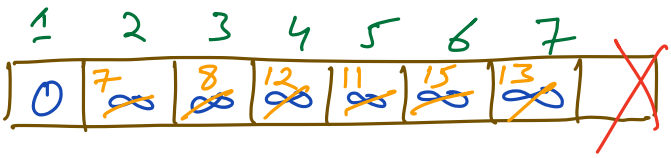
S.C ⇒ $O(N+E)$

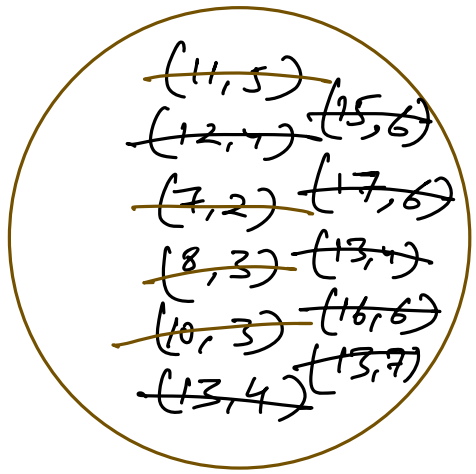
Dijsktra's Algorithm

⇒ Single source shortest path algo. for weighted graph with +ve weights

eg:







code

1. Insert neighbours of source initially in heap.

```
while ( ! heap.isEmpty() )
{
    p = heap.getMin()
    d = p.first
    u = p.second

    if ( d > dist[u] )
        continue ;

    for ( {u,y} in adj[u] )
    {
        newDist = d + u

        if ( newDist < dist[y] )
        {
            dist[y] = newDist
            heap.insert ( { newDist , y } )
        }
    }
}
```

T.C ⇒ $O(E+V)\log V$

S.C ⇒ $O(E+V)$