20/9/2023

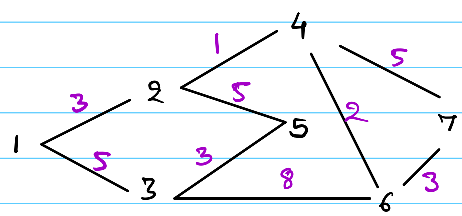
Graphs - 3

Q Given N islands & cost of construction of a bridge b/w multiple pair of islands. Find mini cost of construction required such that it is possible to travel from one island to any other island via bridges (cost > 0)

If not possible, return -1.

N = 7

1 —3— 2
1 —5— 3
2 —1— 4
2 —5— 5
3 —3— 5
4 —2— 6
3 —8— 6
4 —5— 7
6 —3— 7

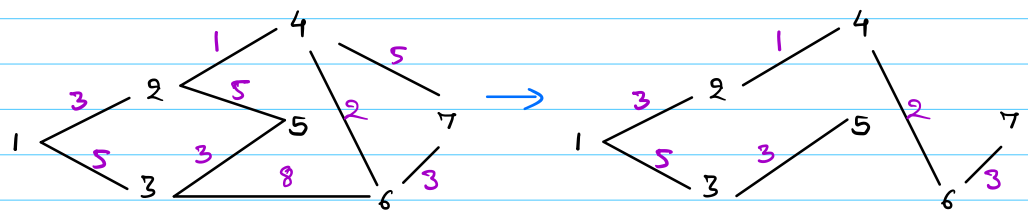


In a connected graph with N nodes, mini # edges possible = N-1 (tree)

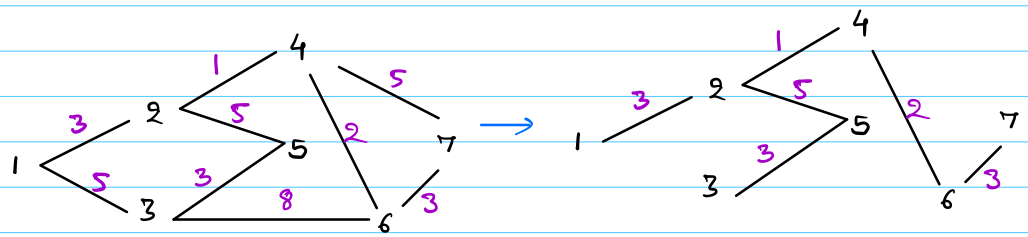
Ans = -1, if graph is disconnected.

(check using DSU)

Minimum Spanning tree : Tree generated from a connected weighted graph s.t all nodes are connected & sum of weights of all selected edges is minimum



Sum of weight = 17



Sum of weight = 17

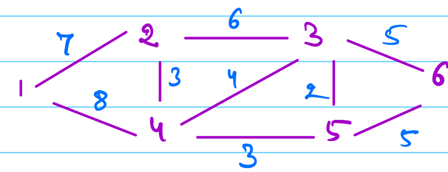
Multiple MST possible for any graph.

Graph with unique weights → unique MST

Algo to find MST → 1> Krushkal's Algo
2> Prim's Algo } greedy.

Krushkal Algo

Select edge with minimum weight if it is not forming a cycle, till complete graph is connected.

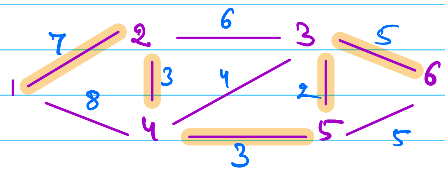


Steps →

1> Sort the edges w.r.t weight. → TC: $O(E \log E)$

2> Consider each node as a set, i.e parent[i] = i. (DSU)

3> Travel all the edges (u, v) & take union if disjoint
⇒ add in answer.



3	2	5	✓
2	3	4	✓
4	3	5	✓
3	4	4	✗
3	5	6	✓
5	5	6	✗
2	6	3	✗
1	7	2	✓
1	8	4	✗

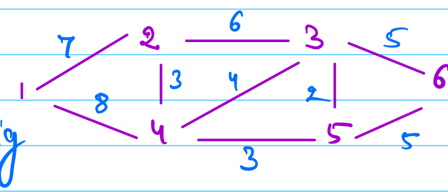
Ans ⇒ 2+3+3+5+7 ⇒ 20

TC: $O(E \log E + E * 1)$

⇒ $O(E \log E)$

SC: $O(N)$ → parent[].

Prims Algo



1) Start with any node as root of MST & keep adding the other nodes as its children.

2) Give priority to nodes that have less edge weight.

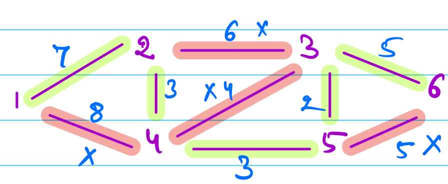
Steps

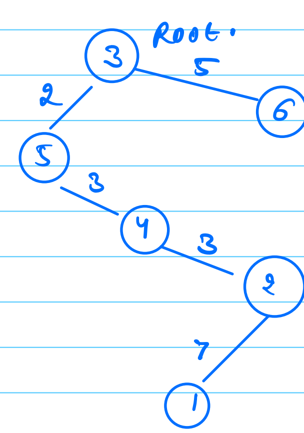
1) Start with root, insert its edges in a min heap (w.r.t edge weight)

2) Pick min wt edge from heap ⇒ if it forms a cycle, i.e connecting both visited nodes ⇒ repeat step 2

   else add the other node as part of tree & insert all its connected edges in min heap.

3) Continue step 2, till complete tree is formed.





Ans = 20

$TC: O(E \log E + E)$

$\approx O(E \log E)$

$SC: O(E+N)$

heap → visited array.

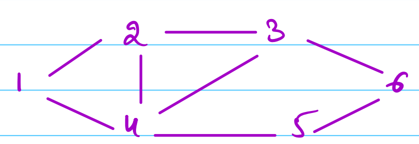
Code

$u \rightarrow \{v1, wt1\}, \{v2, wt2\}$

```
insert (u){
    for ({v,wt} : adj(u)){
        pq.add(v,wt);
    }
}
```

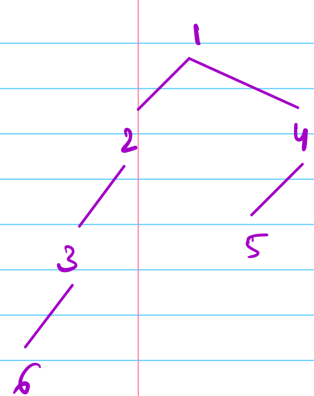
Meet at 8:26 am IST

Q Find min # edges to travel from u to v in undirected simple graph.



u = 1
v = 6

Ans = 3

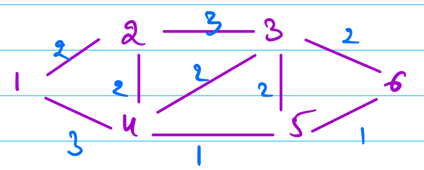


TC: O(N+E)
SC: O(N)

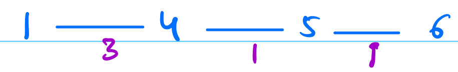
~~1~~ ~~2~~ ~~4~~ ~~3~~ ~~5~~ ~~6~~

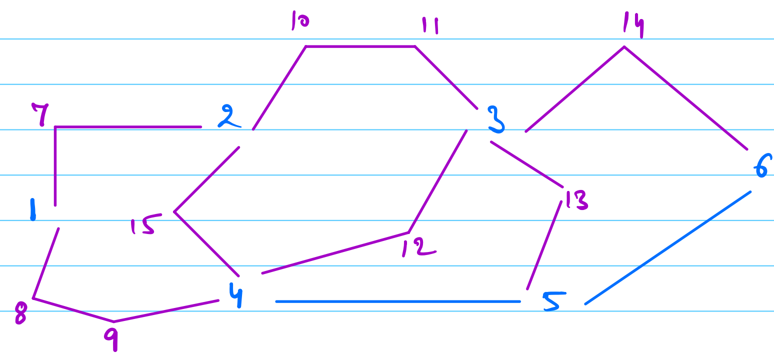
Ans → start BFS from u & level of v will be the answer.

Q. Find min distance to travel from u to v in undirected simple graph. $(1 <= wt <= 3)$



$u = 1$, $v = 6$, Ans $= 5$





Insert dummy nodes, s.t $wt = 1$ $\forall$ edges & apply BFS.

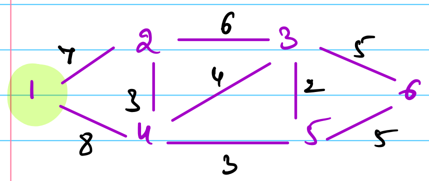
TC: $O(N+E)$

Will it work for large weight? No

Dijkstra's algo → Single source shortest path algorithm for weighted graph with +ve weights

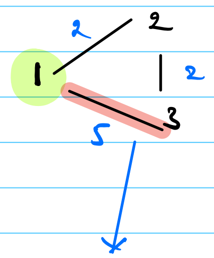
Q. There are N cities in a country. You are living in city 1. Find mini distance to reach every city from city 1.

$C_1 \xrightarrow{dist} C_2$



Ans ⇒ $d[i], \forall i$

$d[1] = 0$ (source)



$d[1] = 0$
$d[2] = 2$
$d[3] = 4 \; (2+2)$

Relaxing an edge

intermediate node

if ($d[u\_w] > d[u\_v] + d[v\_w]$)
$d[u\_w] = d[u\_v] + d[v\_w]$;

To be continued....