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

If not possible, return -1.

disconnected graph → dfs/bfs.
↳ D.S.U

Eg 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

① Graph should be connected

② min cost → minimum bridges/edges. (N-1)

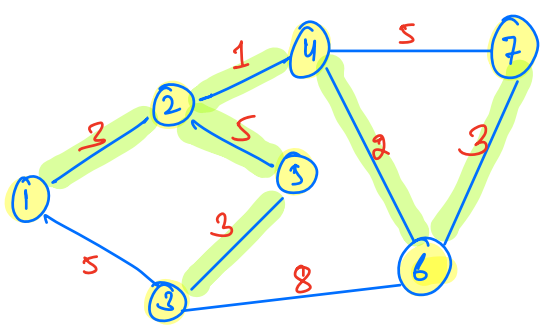
⇓

Tree.

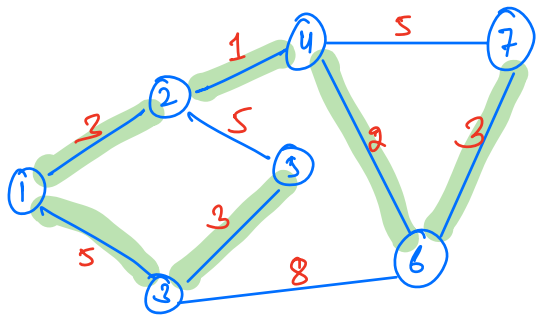
Cost of selected edges is minimum.

⇓

Minimum Spanning Tree (M.S.T)



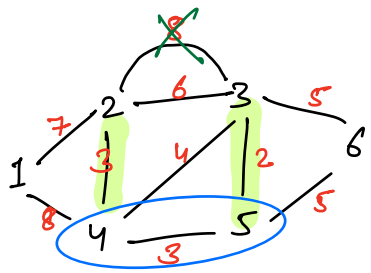
Cost = 3+1+5+3+2+3 = 17.



$\Sigma wt = 3+5+3+1+2+3 = \underline{17}$

Kruskal's Algorithm

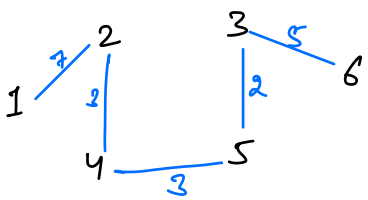
Select the edge with minimum weight, if it is not forming a cycle, till complete graph is connected.



4 and 5 are already visited but they are not forming cycle.

Step 1 → Sort the edges w.r.t edge weight in ascending order.

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	✗
2	8	3	✗



$\Sigma wt = 7+3+3+2+5 = \underline{20}$

Step 2. Select edges one by one if (u,v) doesn't belong to the same set $\Longrightarrow$ add it in M.S.T.

```
if( union(u,v) == true){
        ans += wt(u,v)
}
```

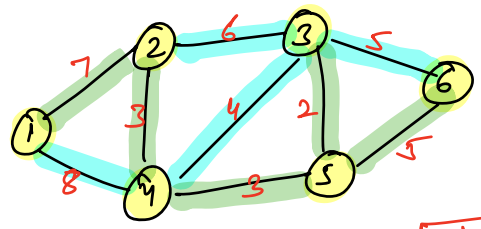
T.C $\rightarrow$ $O(E\log E + E) = O(E\log E)$

S.C $\rightarrow$ $O(N+E) \rightarrow O(E)$

least possible value of $E \rightarrow N-1$

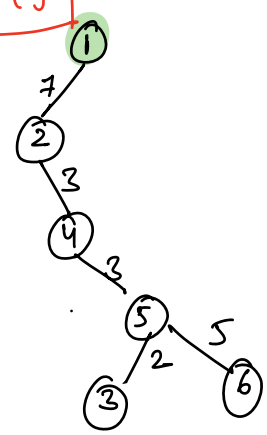
Prim's Algorithm

Start with any node as the root node of M.S.T. & keep on adding the other nodes with minimum weight.



visited[]

minHeap.



Steps:1.

① Select any node as root node & mark it visited.

② Insert all the connected edges of root in min Heap

① while heap.size() > 0 {

a. select the edge with min ed. wt from heap. u→v

b. if (visited[v] == true) continue

else.

visited[v] = true

ans += wt

for({w,x} : Adj[v]){

if (visited[x] == false){

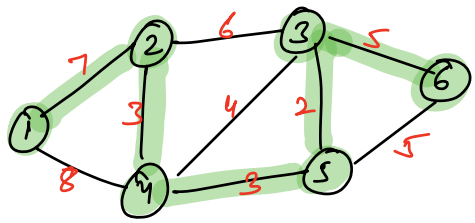
minHeap.insert(w,x)

}

}

}

[T.C → O(E log E) , S.C → O(N+E) = O(E)]



vis →

1	2	3	4	5	6
~~F~~ T	~~F~~ T	~~F~~ T	T	~~F~~ T	~~F~~ T

$u \xrightarrow{w} v$

(w, v)

~~(8,1)~~ ~~(3,2)~~ ~~(5,6)~~

~~(4,3)~~ ~~(3,5)~~

~~(1,1)~~, ~~(6,3)~~

~~(2,3)~~, ~~(5,6)~~

min-Heap.

ans : $3+3+2+5+7 = \underline{20}$

Q) Given a 2D matrix, each cell contains a 'X' or 'O'. Flip all 'O's that are surrounded by 'X' on all 4 sides.

or a group of O's.

X → rotten orange

O → fresh orange

X	X	X
X	X O	X
X	X	X
X	O	X

are

	0	1	2	3	4
0	X	X	X	X	X
1	X	X O	X O	X	O
2	X	X	X O	X	X
3	X	X	X	O	X
4	O	X	X	O	X

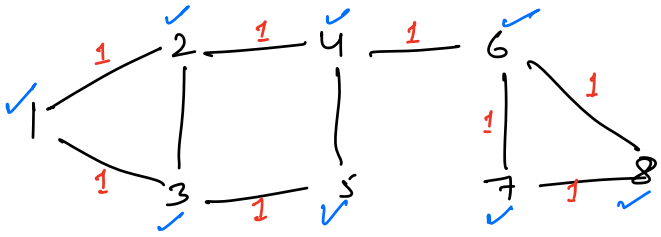
Solution →

① start traversal (BFS/DFS) from boundary of matrix with 'O' & mark connected O's as visited.

② ∀ unvisited O's, flip them to 'X'.

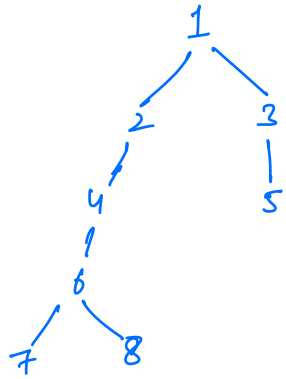
[T.C → O(N*m) , T.C → O(N*m)]

Q: Find the min no. of edges to reach v starting from u in undirected simple graph.



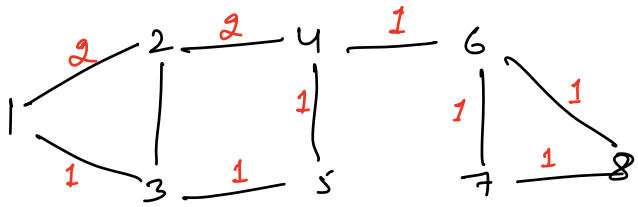
$u = 1, \quad v = 7$

ans = 4.

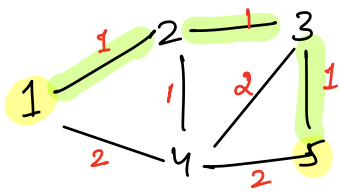


[Start B.F.S from u & find v. level of v = ans.]

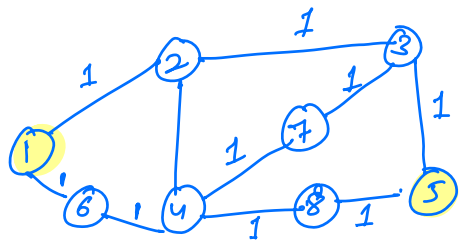
Q: Find the min cost to travel from u to v in given connected simple graph. [$1 \le$ cost of an edge ≤ 2]



[$u = 1, \quad v = 8$]

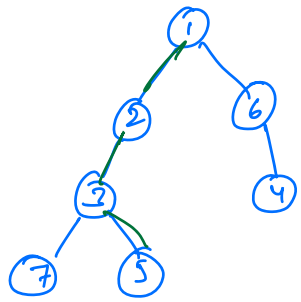


[$u = 1, \quad v = 5$]



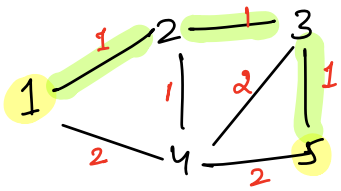
use dummy nodes

[Start B.F.S from u to find v.]
level of v = ans.



ans = 3.

How to solve for larger edge wts.? → to be continued..



[Start taking dummy nodes from 6.]

u	v	wt.
1	2	1
2	3	1
1	4	2
2	4	1
3	4	2
4	5	2
3	5	1

$\Rightarrow \{1 \quad 6 \quad 1\}$
$\{6 \quad 4 \quad 1\}$

$\Rightarrow \{3 \quad 7 \quad 1\}$
$\{7 \quad 4 \quad 1\}$

$\Rightarrow \{4 \quad 8 \quad 1\}$
$\{8 \quad 5 \quad 1\}$