1) DSU ⇒ Optimisations

2) Prim's (MST)

3) Outdegree of Topological Sorted Order

4) Multi Source BFS & Heap

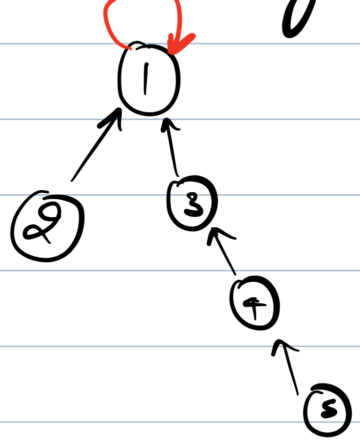
Monday ⇒ Interview Problems.

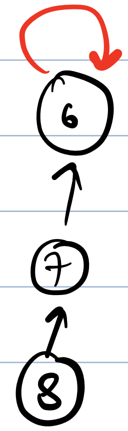
Reduce T.C.

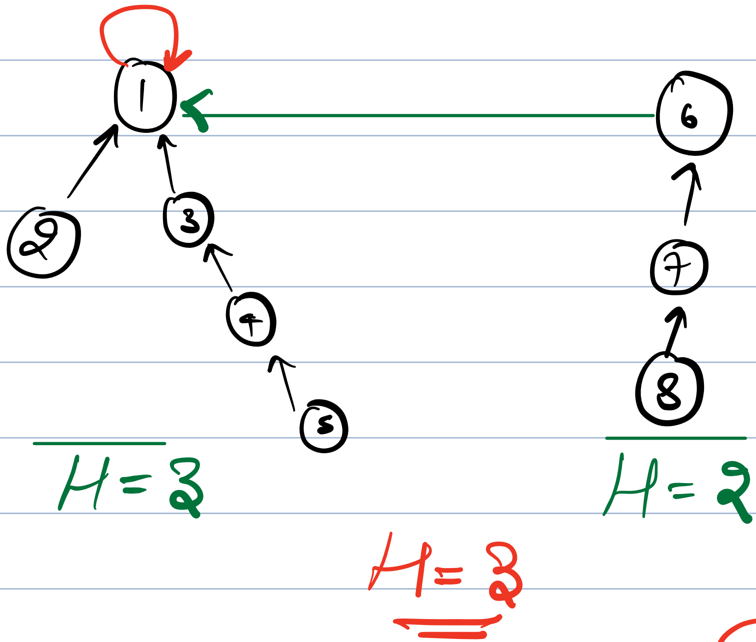
1) Union By Rank.

2) Path Compression

1) Union By Rank → Height







Union (5, 8) {

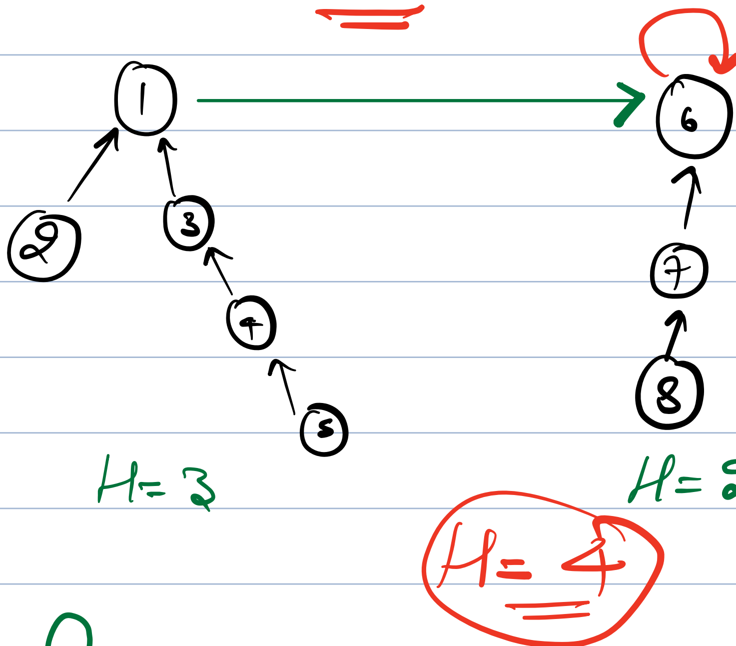
root(5) = 1 →

root(8) = 6 →

parent[1] = 6

parent[6] = 1;

}



One Union ⇒ $O(H)$

↓

Goal: Minimize resulting Height of Tree (set) after every union.

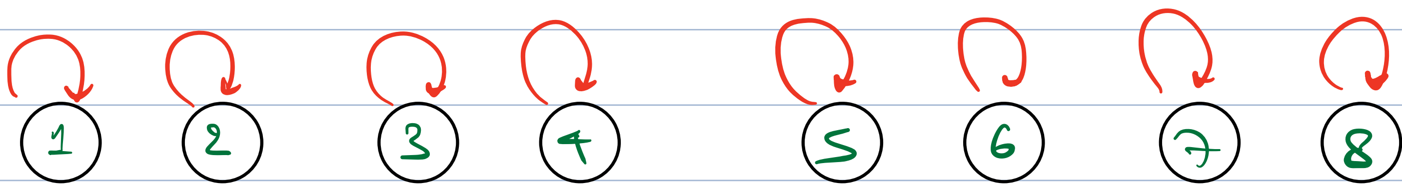
Initially all Nodes are present in different set

$$
\begin{array}{rccccccccc}
 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\text{parent} = & [\cancel{} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8]
\end{array}
$$

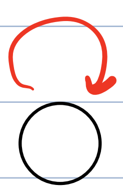
$$\text{Height} = [\cancel{}\ 0, 0, 0, 0, 0, 0, 0, 0]$$

$\langle 1, 2 \rangle$ $\langle 2, 3 \rangle$

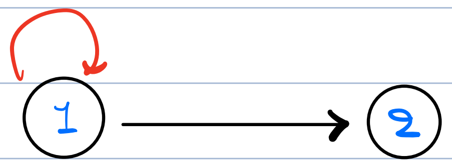
$\langle 4, 5 \rangle$ $\langle 3, 5 \rangle$

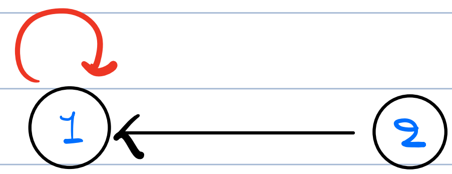


for a graph with N nodes, what can be the max Height of the DSU Tree formed.

$N = 1$  $\Rightarrow$ $\underline{H = 0}$.

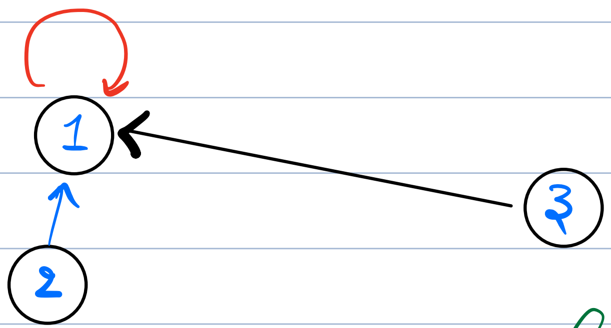
$N = 2$ $\qquad 1 \quad + \quad 1$





$H = 1$

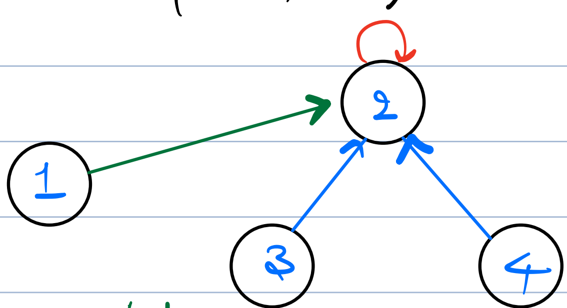
$\underline{N = 3}$ $\qquad$ Union (2, 1)

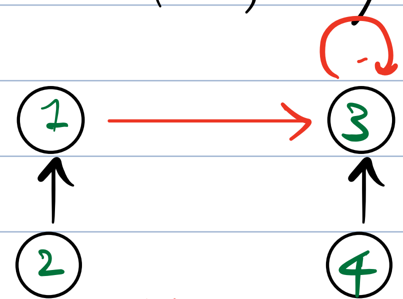


Height = 1

$N = 4$

(1, 3) $\quad , \quad$ (2, 2)



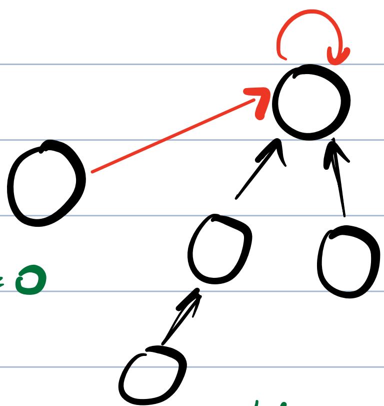


$\underline{H = 1}$ $\qquad \underline{H = 2}$

$H = 2$

$N = 5$

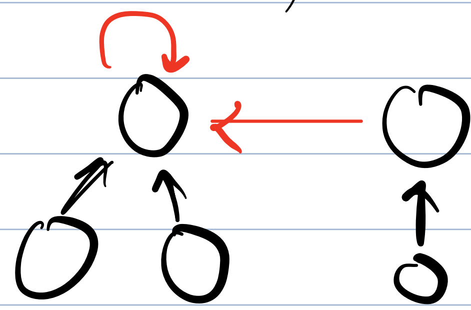
1, 4



$H=0$

$H=2$

3, 2



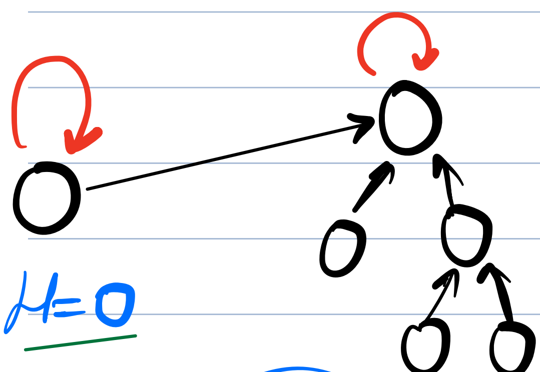
$H=1$ $H=1$

$H=2$

$H=2$

$N = 6$

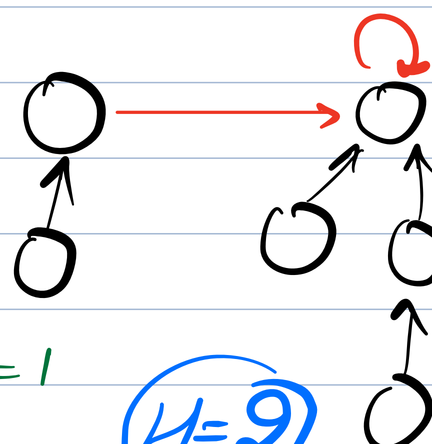
(1, 5)



$H=0$

$H=2$

(2, 4)



$H=1$

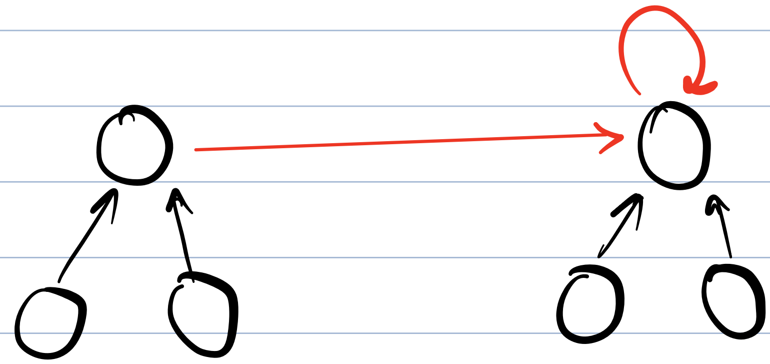
$H=2$

$H=2$

H = 2

H=2

(3, 3)



H=1 H=1

H=2

N=7

(1, 6) 0, 2 H=2

(2, 5) 1 2 H=2

(3, 4) 1 2 H=2

N=8

0, 2 (1, 7) 2

1 2 (2, 6) 2

1 2 (3, 3) 2

2 2 (4, 4) H=3

1 2 3 4 5 6 7 8 9 10 .... 15 16...

0 1 1 2 2 2 2 3 3 3 ... 3 4 ...

$N = \quad \log_2 N$

$H[N+1] = \{0\}$

findroot()

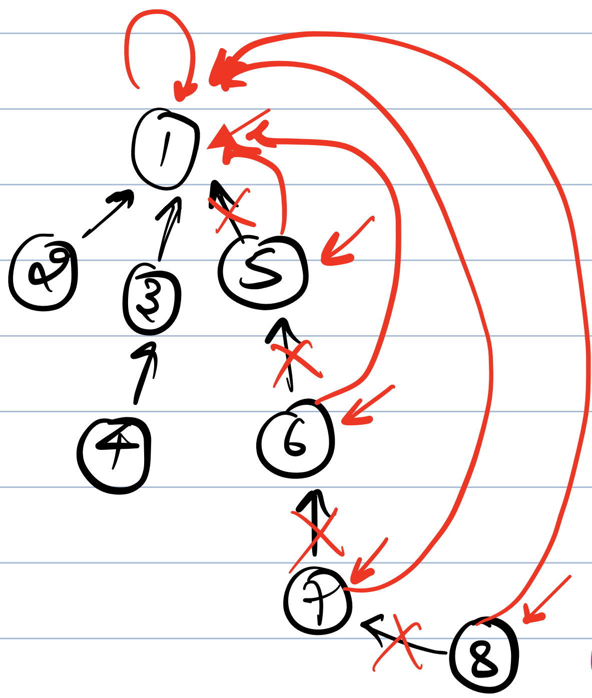
```
bool union(u, v) {
    x = findroot(u);
    y = findroot(v);

    if (x == y) {
        return false;
    }

    if (H[x] < H[y]) {
        parent[x] = y;
    } else if (H[x] > H[y]) {
        parent[y] = x;
    } else {
        parent[y] = x;
        H[x]++;
```

}

return True;

}

Path Compression



findroot (8) ⇒ 1

⇓

T.C. = O (Height)

⇓

$\log_2 N$

$O(\log_2 N)$

```
int findroot (x) {
    if (parent[x] == x) {
            return x;
    }

parent[x] = findroot (parent[x]);
```

H.W.

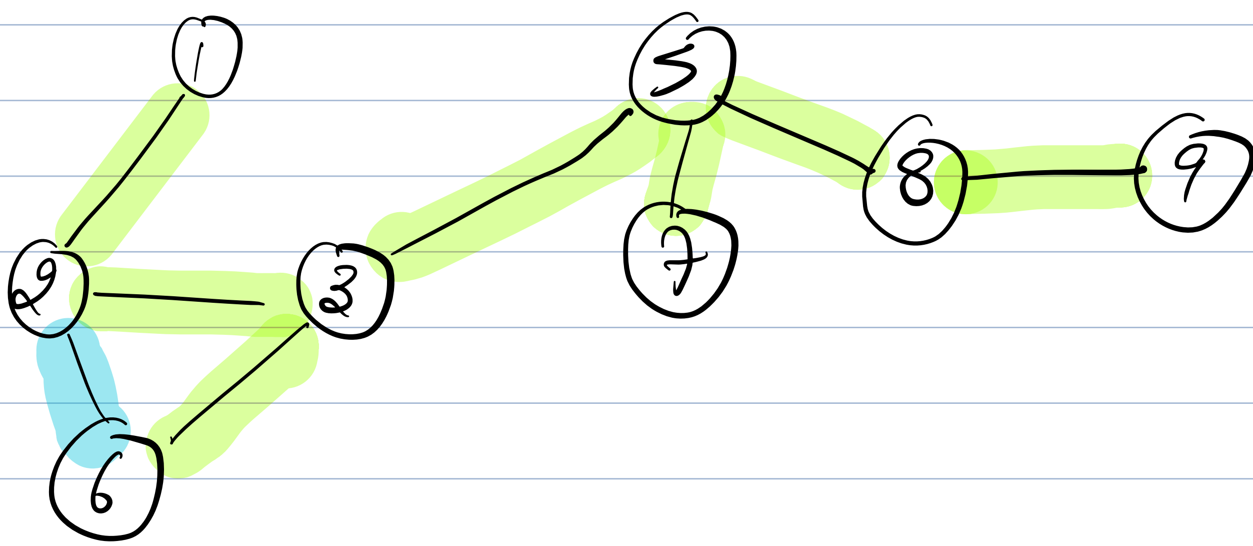
return parent[x];
}

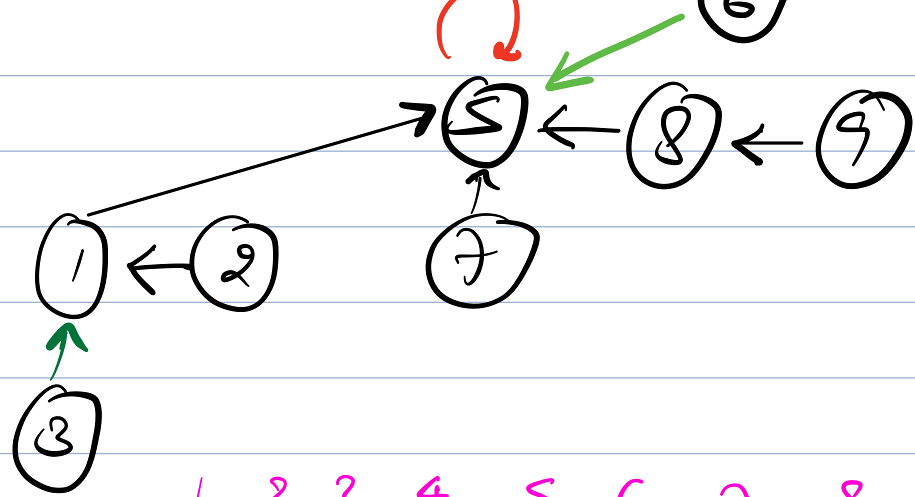
Amortized

T.C. = $O(1)$

UseCase of DSU

1) Cycle Detection





1	2	3	4	5	6	7	8	9
5	1	1	4	5	5	5	5	8

$O(N)$

Union (1, 2) ✓
Union (8, 9) ✓
Union (5, 7) ✓
Union (5, 8) ✓
Union (2, 3) ✓
Union (3, 5) ✓
Union (3, 6) ✓

$O(E)$

$O(1)$

Union (2, 6)

5 5 ✗ ⇒ Cycle is Present

$$T.C. = O(\underbrace{N+E}_{})$$

V+E

2) Check if graph is connected

After performing all Unions

⇓

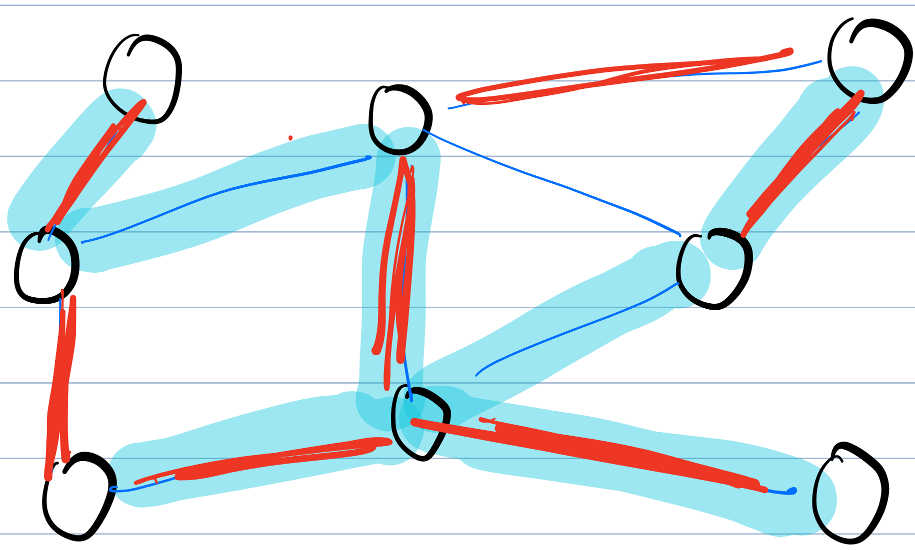
$\forall$ pairs $(i, j) \Rightarrow \underline{\text{union}}(i, j)$

↓

False

MST

Eg



Goal is to construct roads to connect everything

Q With a connected graph of N nodes & E edges.

What is the min no. of edges required to keep the graph connected.

Ans = N-1

(N nodes ⇒ N-1 edges) ⇒ Tree

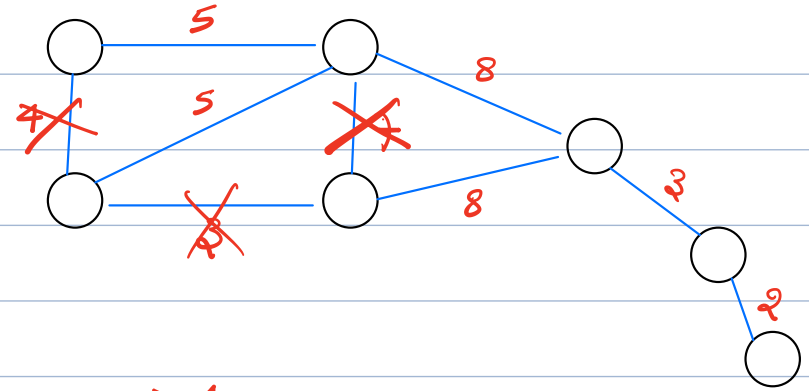
Connected Graph N nodes $\xrightarrow{\text{Remove extra edges}}$ Spanning Tree

s.t. graph is connected. & no. of edges = (N-1)

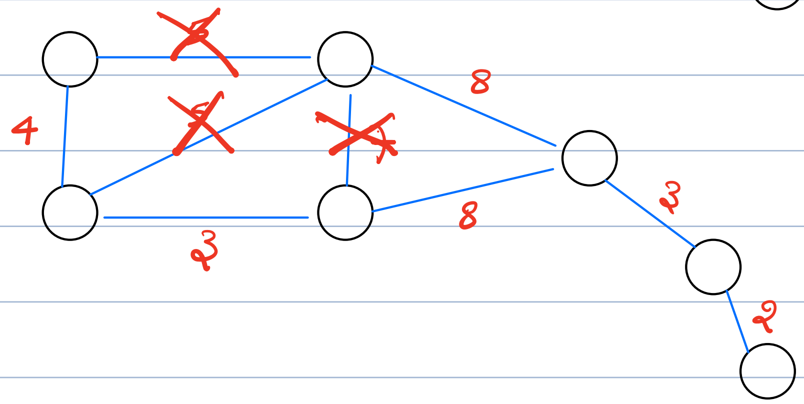
A graph can have multiple S.T.

$N = 7$

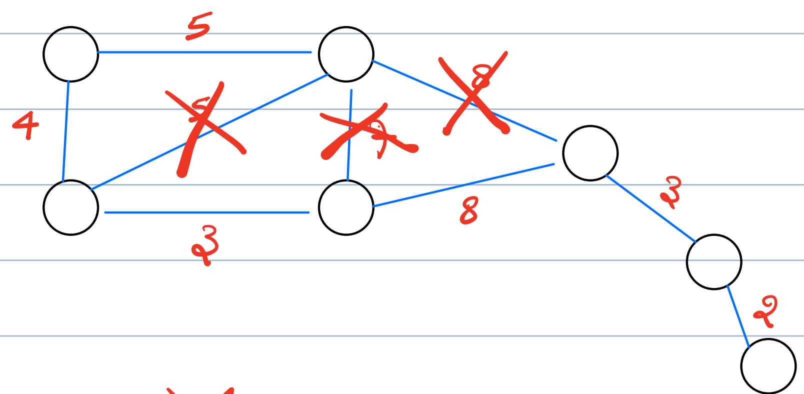
$E = 9$ (-3)



Σ Weights = 31

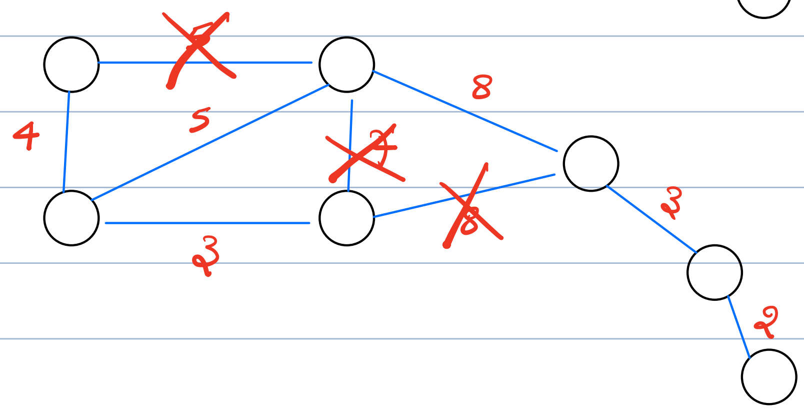


Σ Weights = 28



Σ Weights = 25

min

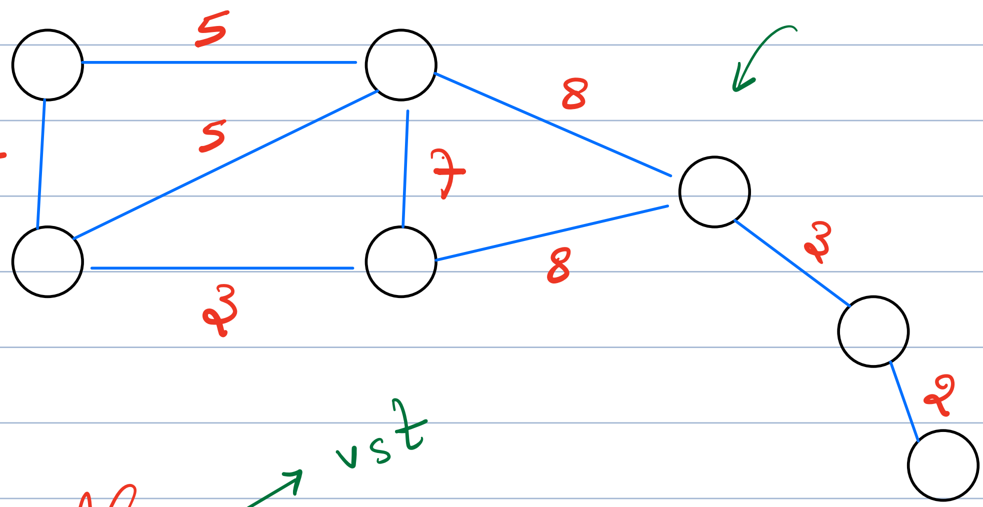


Σ Weights = 25

Minimum Spanning Tree.

There can be multiple MSTs of a graph.

How to find MST



1) Prim's Algo → vst
2) Kruskal's Algo. → DSU

Goal ⇒ Select the edges with min possible weights.

⇓

1) Sort the edges based on weights
2) Use Min Heap

Heap(<Pair<int, Pair<int, int>>>)

Edge: Pair<int, int> ; Weight: int ; u, v

$\langle 3, \langle 5, 6\rangle\rangle$

$\langle 3, \langle 2, 4\rangle\rangle$

$\langle 4, \langle 1, 2\rangle\rangle$

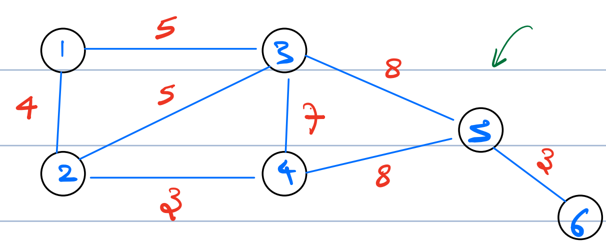
$\langle 5, \langle 1, 3\rangle\rangle$

$\langle 5, \langle 2, 3\rangle\rangle$

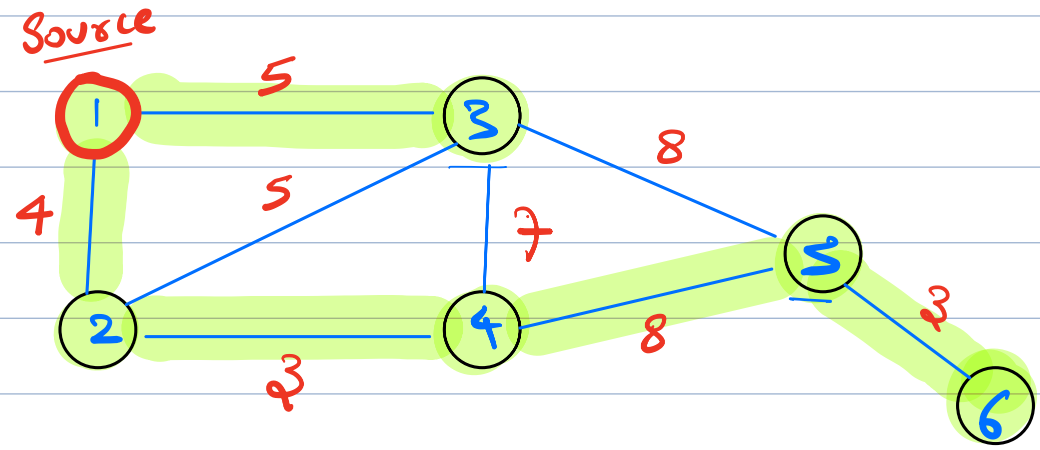
$\langle 7, \langle 3, 4\rangle\rangle$

$\langle 8, \langle 3, 5\rangle\rangle$

$\langle 8, \langle 4, 5\rangle\rangle$



1) Prim's Algo. (BFS)



~~$\langle 4, \langle 1, 2 \rangle \rangle$~~

~~$\langle 5, \langle 1, 3 \rangle \rangle$~~

~~$\langle 3, \langle 2, 4 \rangle \rangle$~~

~~$\langle 5, \langle 2, 3 \rangle \rangle$~~

~~$\langle 7, \langle 4, 3 \rangle \rangle$~~

~~$\langle 8, \langle 4, 5 \rangle \rangle$~~

$\langle 8, \langle 3, 5 \rangle \rangle$

~~$\langle 3, \langle 5, 6 \rangle \rangle$~~

w u v

$\langle 4, \langle \boxed{1}, \cancel{2} \rangle \rangle$

$\langle 3, \langle \boxed{2}, 4 \rangle \rangle$

$\langle 5, \langle \boxed{1}, \cancel{3} \rangle \rangle$

$\langle 5, \langle \boxed{2}, \boxed{3} \rangle \rangle$ ✗

$\langle 7, \langle \boxed{4}, \boxed{3} \rangle \rangle$ ✗

$\langle 8, \langle \boxed{4}, \cancel{5} \rangle \rangle$

$\langle 3, \langle \boxed{5}, \cancel{6} \rangle \rangle$

$\langle 4, \langle 1, 2 \rangle \rangle$

$\langle 3, \langle 2, 4 \rangle \rangle$

$\langle 5, \langle 1, 3 \rangle \rangle$

$\langle 8, \langle 4, 5 \rangle \rangle$

$\langle 3, \langle 5, 6 \rangle \rangle$

Ans

Code

Node Weight

```
1 : [ <2, 5>, <4, 7> ]
```

```
MinHeap <Pair <int, Pair <int, int>>> mh;
vst [N+1] = {false};
```

```
Pair < first, second >
```

u ⇒ Source

```
List<<Pair<int, Pair <int, int>>> createMST (u) {

    List < Pair<int, Pair <int, int>>> mst;

    vst [u] = true;
    for (all nodes v connected to u) {
            mh.insert <w, <u, v>>;
    }

    while ( mst.size () <(N-1)) {

        x = mh.getMin ();
        Weight = x.first ();
        Src = x.second(). first ();
        dest = x.second(). second();

        if (vst [dest] == false) {
```

```
            vst[dest] = True;
            mst.add(x);
            for (all z connected to dest){
                if (vst[z] == false){
                    mh.add(<weight, <dest, z>>);
                }
            }
        }
    }
    return mst;
}
```

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

Next Weekend → Hol.

Weekend

DSA1, 2, 3, 4, SQL

Modulewise