'Assignment 03'

Amirun Nahin

ID: 23201416

Sec: 06

Course: CSE221

Date: 16/05/2025

Ans. to the que no-01

(a)

parent =

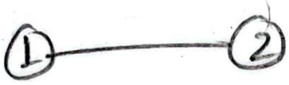
	0	1	2	3	4	5	6	7	8	9	10	11	12
parent	~~0~~ 3	~~1~~ 3	~~2~~ 1	3	~~4~~ 3	~~5~~ 3	6	~~7~~ 1	~~8~~ 0	~~9~~ 0	~~10~~ 8	11	~~12~~ 3

size =

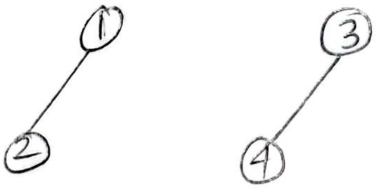
	0	1	2	3	4	5	6	7	8	9	10	11	12
size	~~1~~ ~~2~~ 4	~~1~~ ~~2~~ 3	1	~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~7~~ 11	1	1	1	1	~~1~~ 2	1	1	1	1

Now, let's do union by size (Without Path Compression)

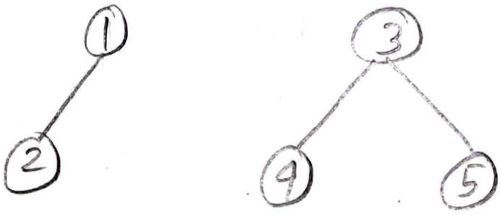
(i) Union (1,2)



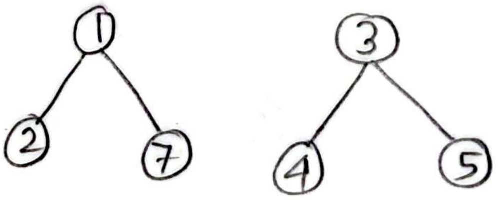
(ii) Union (3,4)



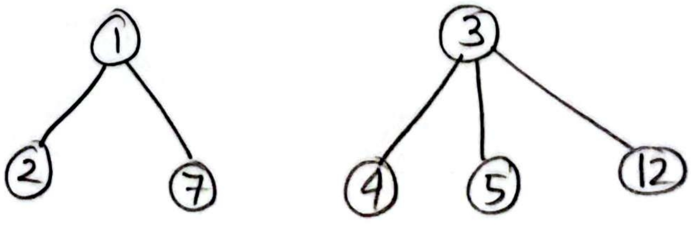
(iii) Union(3,5)



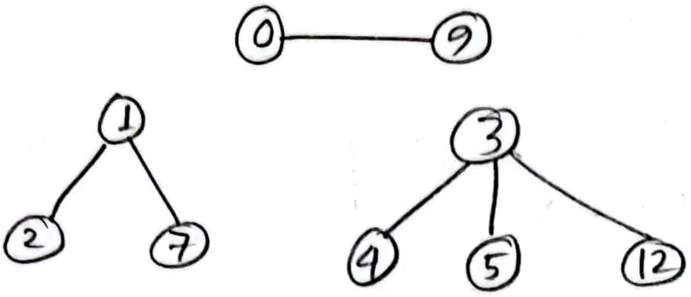
(iv) Union(1,7)



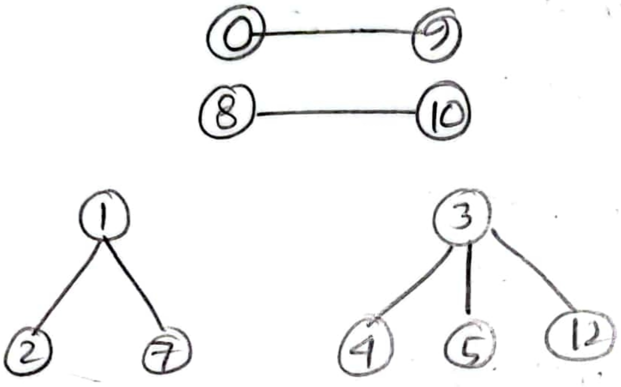
(v) Union(3,12)



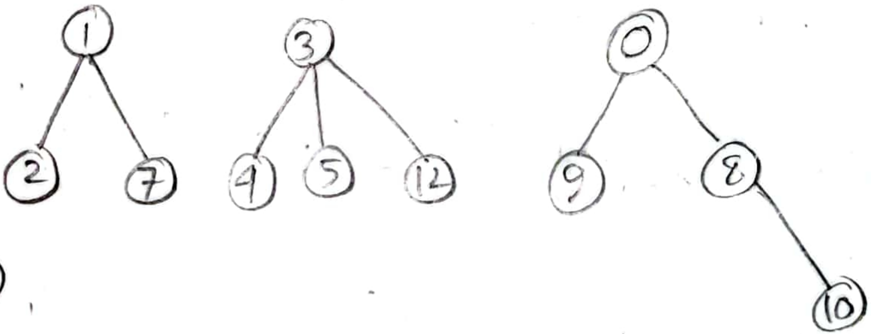
(vi) Union (0,9)



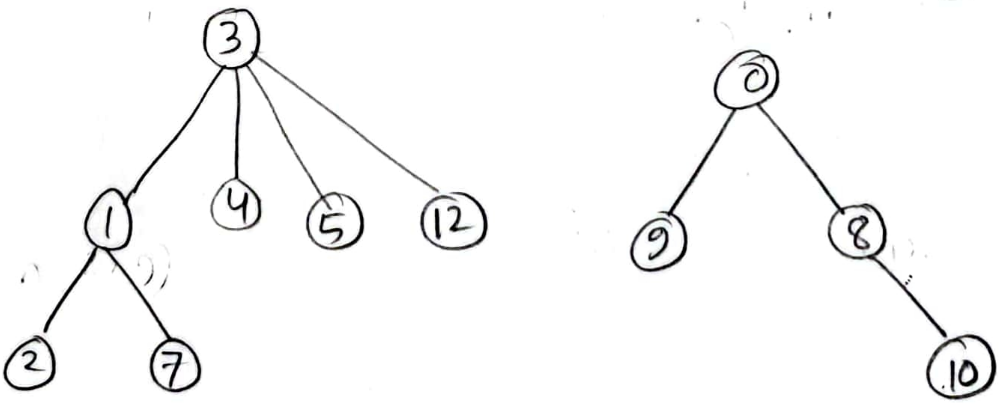
(vii) Union (8,10)



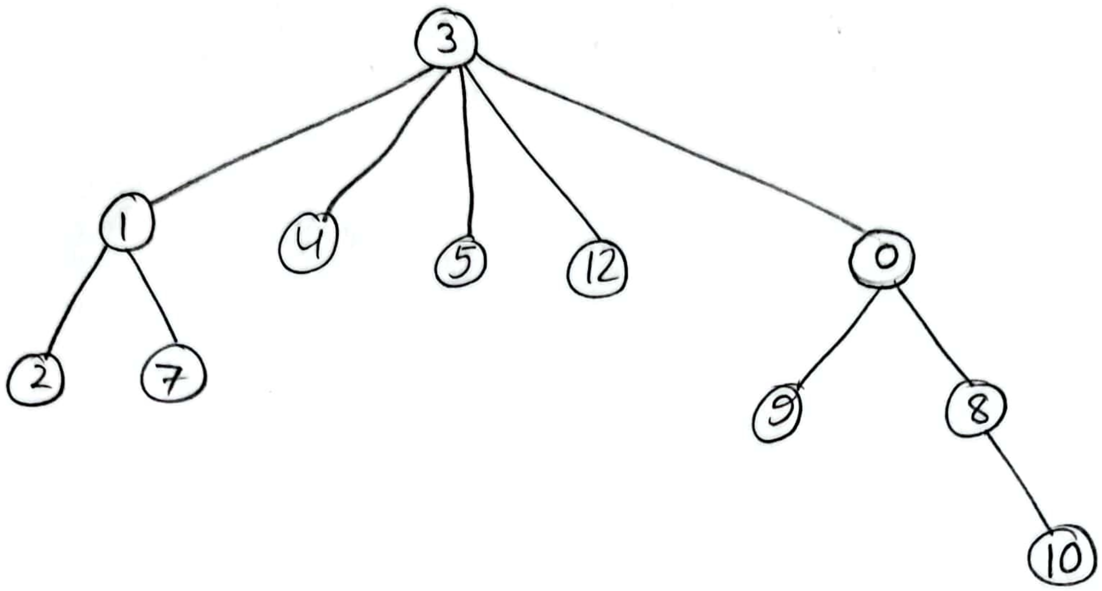
(viii) Union (8,9)



(ix) Union(7,4)



⊗ Union (2,9)



Finally, the parent and size arrays. are,

	0	1	2	3	4	5	6	7	8	9	10	11	12
parent =	3	3	1	3	3	3	6	1	0	0	8	11	3

	0	1	2	3	4	5	6	7	8	9	10	11	12
size =	4	3	1	11	1	1	1	1	2	1	1	1	1

(b)

parent =

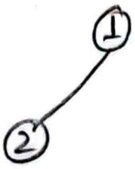
	0	1	2	3	4	5	6	7	8	9	10	11	12
	8		1	1	3	3		1	1	0	8		3
parent	~~0~~	1	~~2~~	~~3~~	~~4~~	~~5~~	6	~~7~~	~~8~~	~~9~~	~~10~~	11	~~12~~

height =

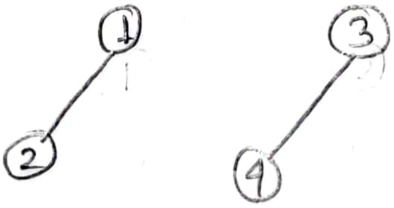
	0	1	2	3	4	5	6	7	8	9	10	11	12
	1	~~1~~ ~~2~~ 3		1					~~1~~ 2				
height	~~0~~	~~0~~	0	~~0~~	0	0	0	0	~~0~~	0	0	0	0

Now, let's do Union by height

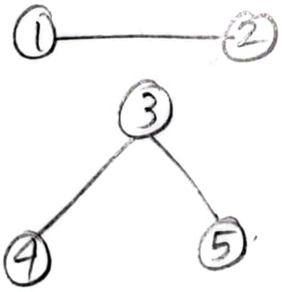
(i) Union (1,2)



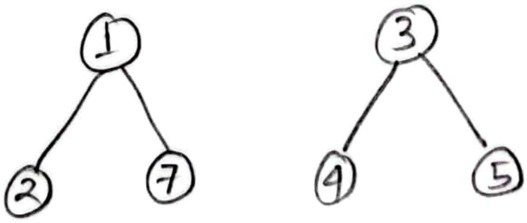
(ii) Union(3,4)



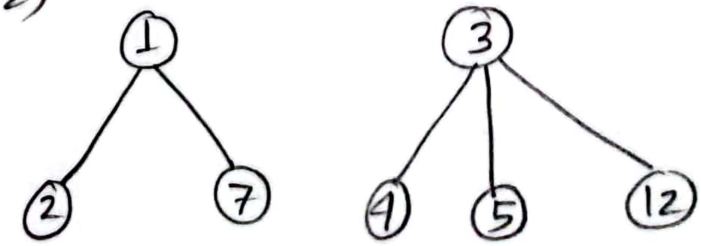
(iii) Union(3,5)



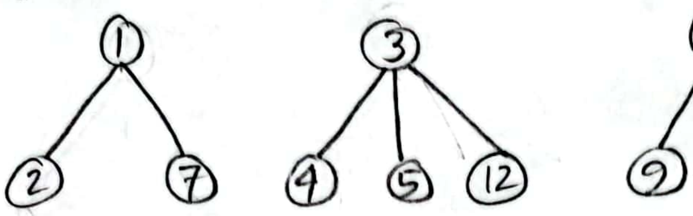
(iv) Union(1,7)



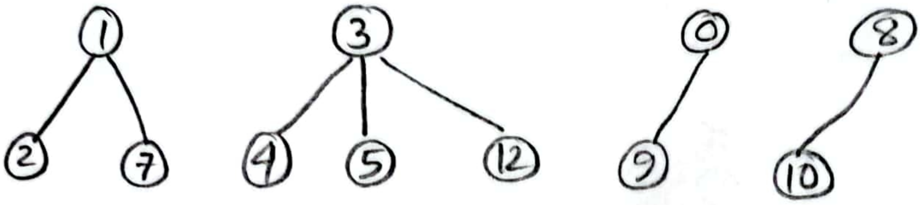
(v) Union(3,12)



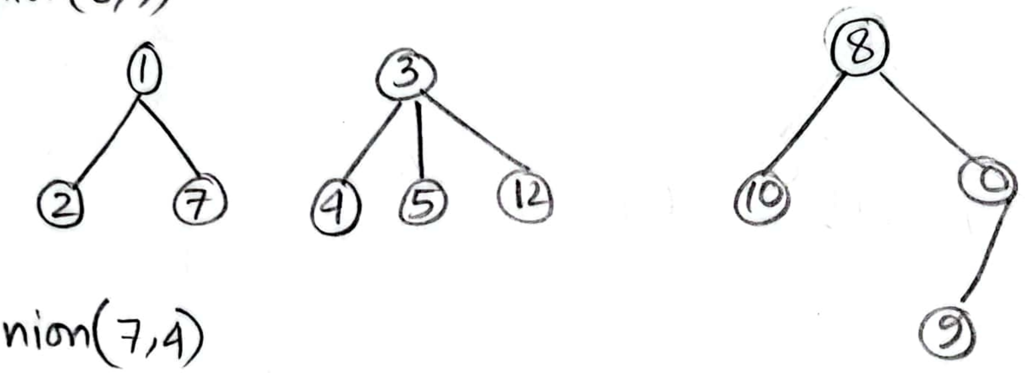
(VI) Union (0,9)



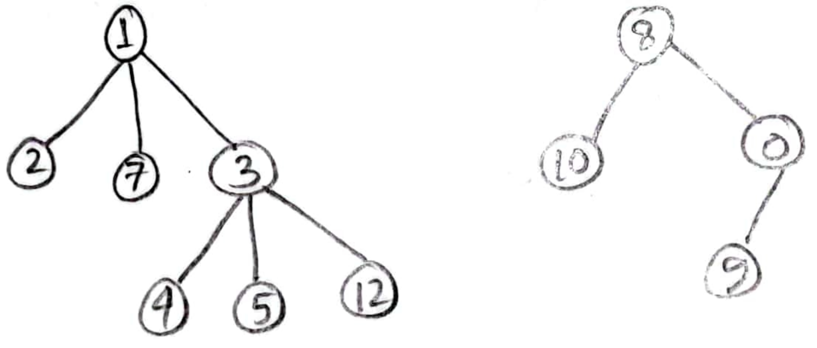
(VII) Union (8, 10)



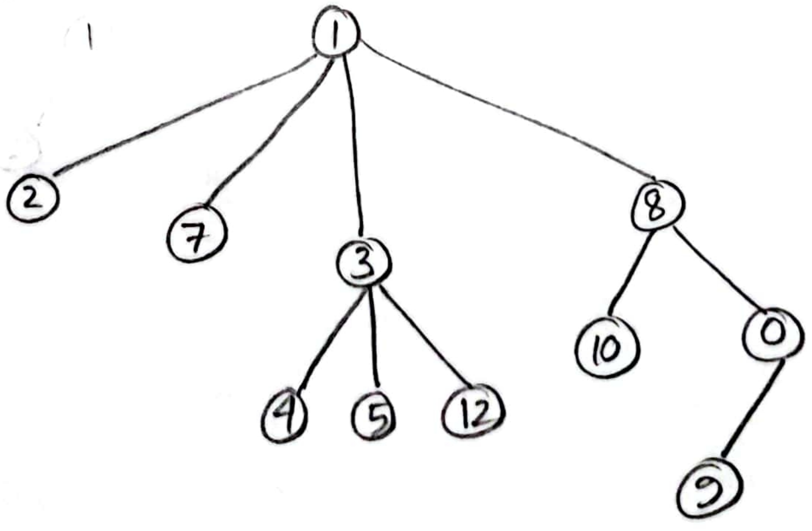
(VIII) Union(8,9)



(IX) Union(7,4)



(X) Union(2,9)



So, the parent and height array looks like,

parent =

0	1	2	3	4	5	6	7	8	9	10	11	12
8	1	1	1	3	3	6	1	1	0	8	11	3

height =

0	1	2	3	4	5	6	7	8	9	10	11	12
1	3	0	1	0	0	0	0	2	0	0	0	0

Ⓒ

parent =

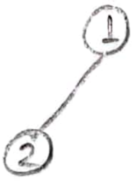
~~0~~ 8	~~1~~ 3	~~2~~ ~~1~~ 3	3	~~4~~ 3	~~5~~ 3	6	~~7~~ 1	~~8~~ 3	~~9~~ ~~0~~ 8	~~10~~ 8	11	~~12~~ 3
0	1	2	3	4	5	6	7	8	9	10	11	12

size =

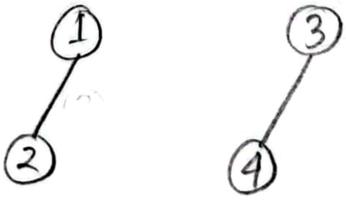
~~1~~ 2	~~1~~ ~~2~~ 3	1	~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~7~~ 11	1	1	1	1	~~1~~ ~~2~~ 4	1	1	1	1
0	1	2	3	4	5	6	7	8	9	10	11	12

Now, let's find the unions by size with path compression.

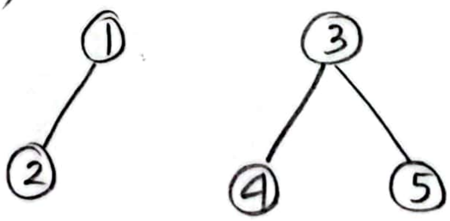
(i) Union (1,2)



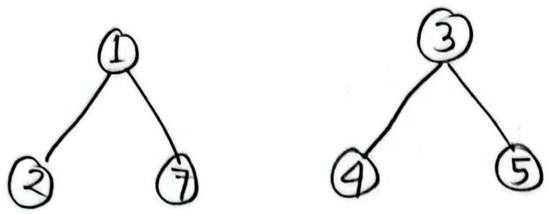
(ii) Union (3,4)



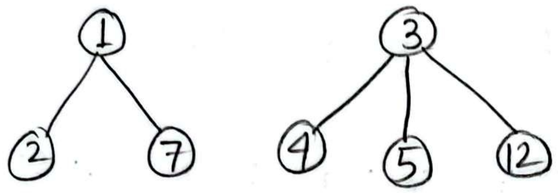
(iii) Union (3,5)



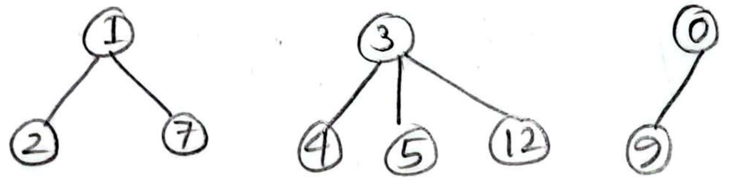
(iv) Union(1,7)



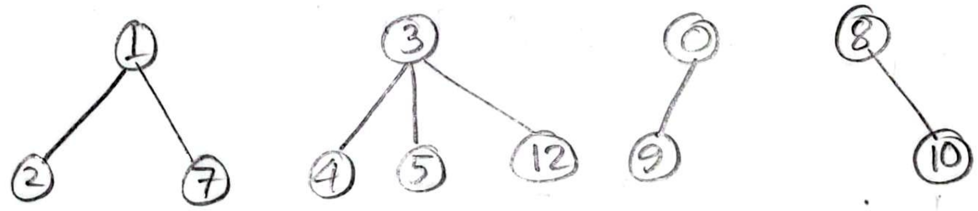
(v) Union (3,12)



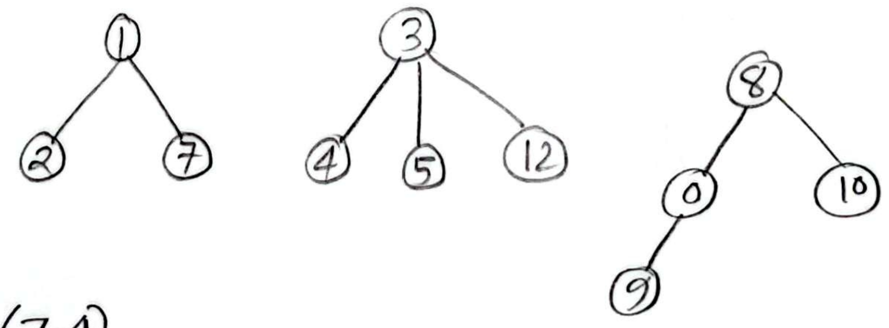
(vi) Union (0,9)



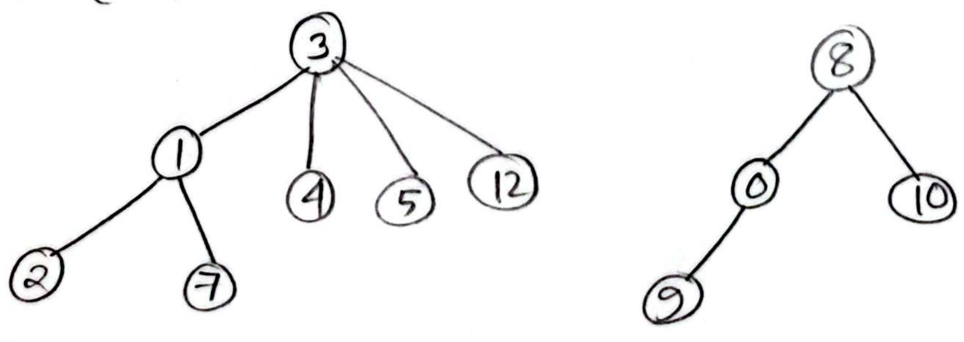
(vii) Union (8,10)



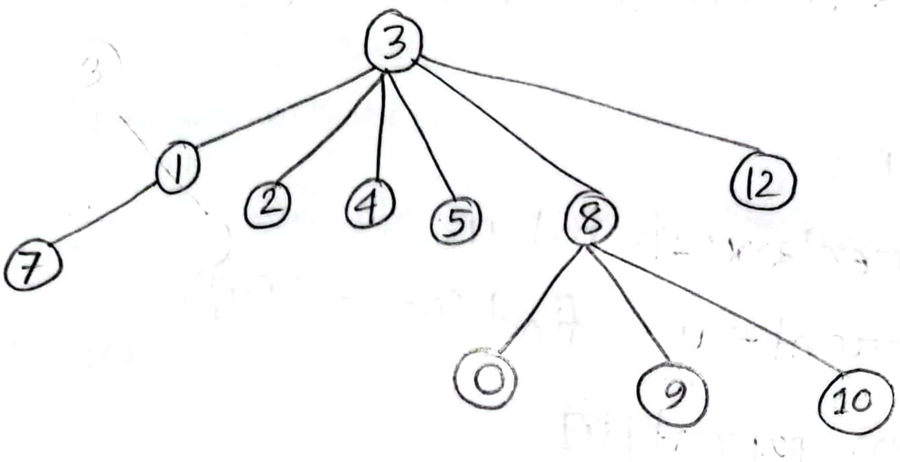
(viii) Union (8,9)



(ix) Union(7,4)



⊗ Union(2,9)



Finally, the parent and size arrays are,

parent =

0	1	2	3	4	5	6	7	8	9	10	11	12
8	3	3	3	3	3	6	1	3	8	8	11	3

size =

0	1	2	3	4	5	6	7	8	9	10	11	12
2	3	1	11	1	1	1	1	4	1	1	1	1

Answer to the question no-2

The pseudocode for the problem is given below:

```
def find(u);
    if parent→u is not u;
        parent→u = find(parent[u])
    return parent[u]

def union(u,v);
    parent_u = find(u)
    parent_v = find(v)
    if parent_u is not parent_v; then
        if rank→parent_u ≥ rank→parent_v; then.
            parent[parent_v] = parent_u
            rank→ parent_u += rank→ parent_v
        else;
            parent[parent_u] = parent_v
            rank→parent_v += rank→ parent_u
        return True
    return False
def connected_component(u,v);
    parent = []
    rank = []
    for i in range(n);
        parent.append (i)
        rank. append (1)
```

```
count = 0
for every u,v in edges;

    x=union(u,v)
    if x; then
        count += 1

return count
```