

# Disjoint Set

Date / /  
Page No.

① find Parent

② Union()  $\xrightarrow{\text{rank}(a) \geq \text{size}(b)}$  2 ways of implementing Union()

Union By Rank array:

(1, 2)

(2, 3)

(4, 5)

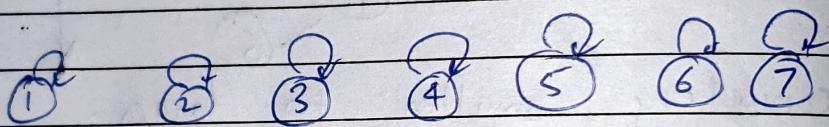
(6, 7)

(5, 6)

(3, 7)

rank  $\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix}$

Parent  $\begin{bmatrix} 1 & 1 & 4 & 6-7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix}$



Rank[i] : denotes the no. of nodes beneath it. If it will change further going keep reading.

Parent[i] : just right above guy of i. // Def. changes as we further in algo.

Both Rank & Parent def. changes as we do multiple optimization in order to obtain constant Time complexity.

Union(u, v)

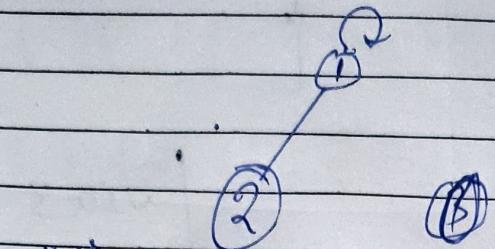
1 - find ultimate Parent of u, v; let call them Pu and Pv

2 - find rank of Pu, Pv

3 - Connect smaller rank to larger rank always.

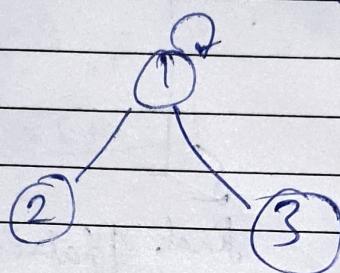
$$(1,2) \Rightarrow P_{u(1)} = 1 ; P_{v(2)} = 2$$

$$r_1 = 0 \quad r_2 = 0$$



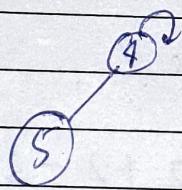
$$(2,3) \Rightarrow P_{u(2)} = 1 ; P_{v(3)} = 3$$

$$r_1 = 1 ; r_3 = 0$$



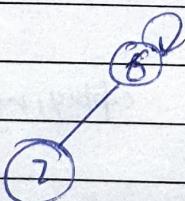
$$(4,5) \Rightarrow P_{u(4)} = 4 ; P_{v(5)} = 5$$

$$r_2 = 0 \quad r = 0$$



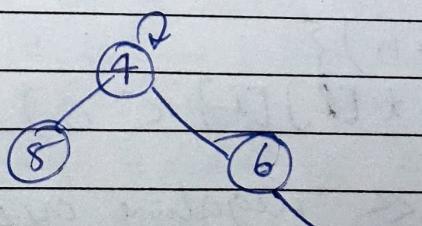
$$(6,7) \Rightarrow P_{u(6)} = 6 ; P_{v(7)} = 7$$

$$r = 0 \quad r = 0$$



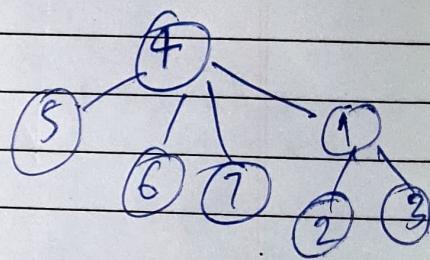
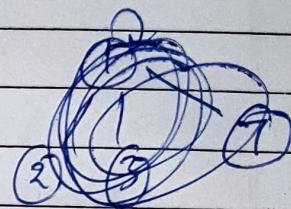
$$(5,6) \Rightarrow P_{u(5)} = 4 ; P_{v(6)} = 6$$

$$r = 1 \quad r = 0$$



$$(3,7) \Rightarrow P_{u(3)} = 1 , P_{v(7)} = 4$$

$$r_1 = 1 \quad r_2 = 2$$



here path

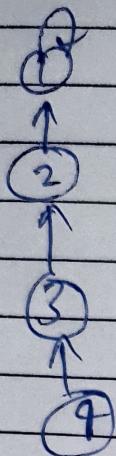
compression  
takes place  
at calling Part 7)

That's why  
1 parent got  
changed to 4

Q. at this moment how to know 1 & 7 belongs to same component 2. (given by find parc)

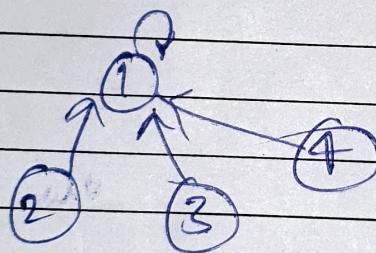
## Path Compression :-

let say a component at a time is as:



now find Parent(4) = ?

after calling parent function component will like :



find\_Par(u)

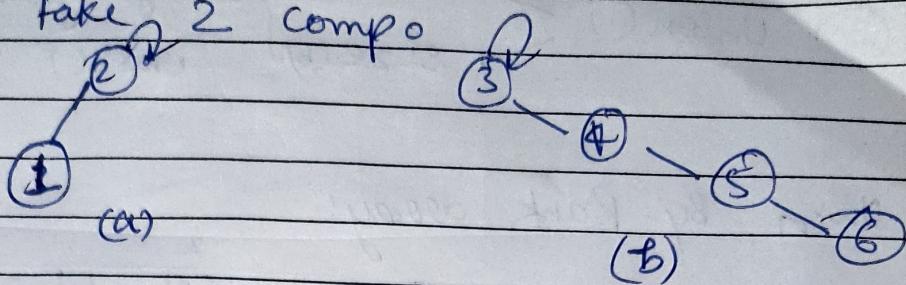
{ if  $u == \text{Par}(u)$ ) return u

? find\_Par(Par[u]);

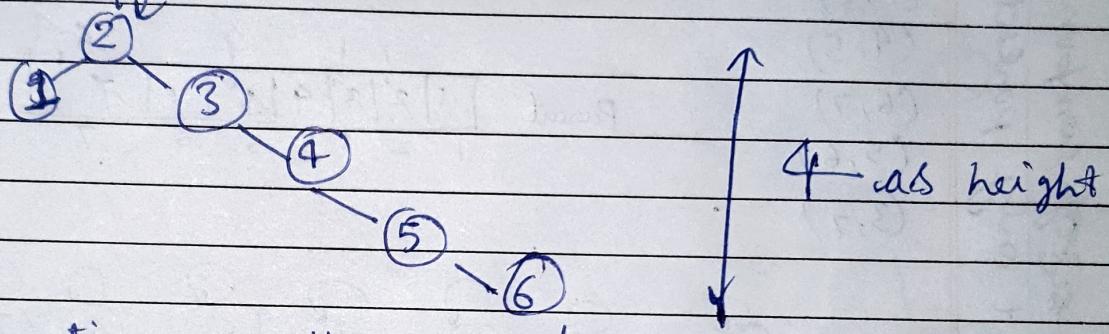
Note => on Path compression Rank won't change (Video 61-46 (20:25 timestamp))

Q. why connect smaller to larger?

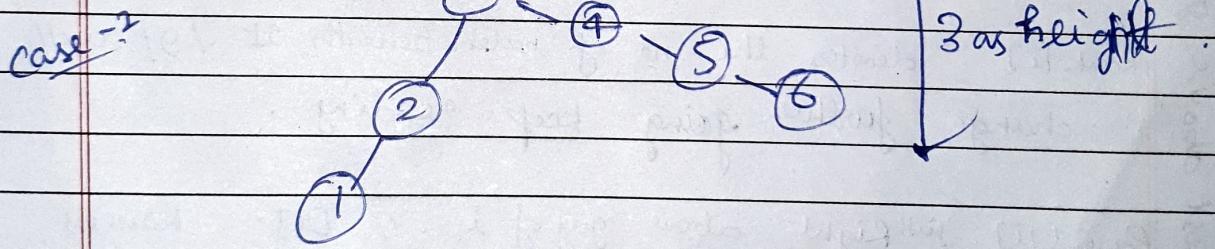
let's take 2 compo.



case 1 connecting larger to smaller.



connecting smaller to larger.



Hence, we can observe height remains is unnecessarily increasing case-2; Hence we to increase the optimization of query answer/response using case-2.

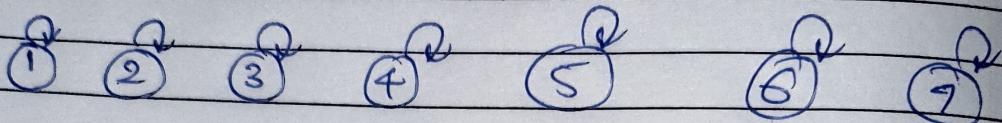
# Union By Size:

Date / /  
Page No. \_\_\_\_\_

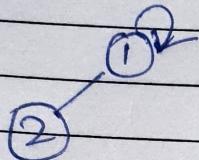
Initially

	3		7	4	6	
1	2	1	1	4	2	1

Parent =)	4	1	2	1	8	4	8	4	6	7	64
	1	2	3	4	5	6	7				

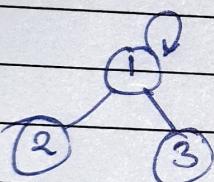


(1, 2)  $\Rightarrow$  Parent  $P_u = 1$ ,  $P_v = 2$   
 $S_u = 1$ ;  $S_v = 1$



(2, 3)

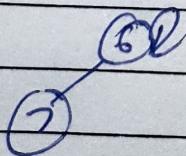
$P_u = 1$ ,  $P_v = 3$   
 $S_u = 2$ ;  $S_v = 1$



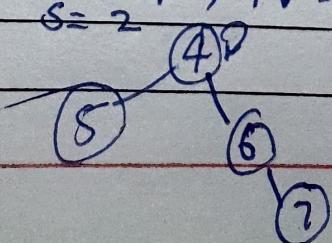
(4, 5)  $P_u = 4$ ,  $P_v = 5$

(4)  
(5)

(6, 7)  $P_u = 6$ ,  $P_v = 7$   
 $S_u = 1$        $S_v = 1$



(5, 6)  $P_u = 4$ ,  $P_v = 6$  ( $S_u = 2$ )



(3,7)  $P_a = 1 ; P_v = 4$

$S = 3$

$S = 4$

