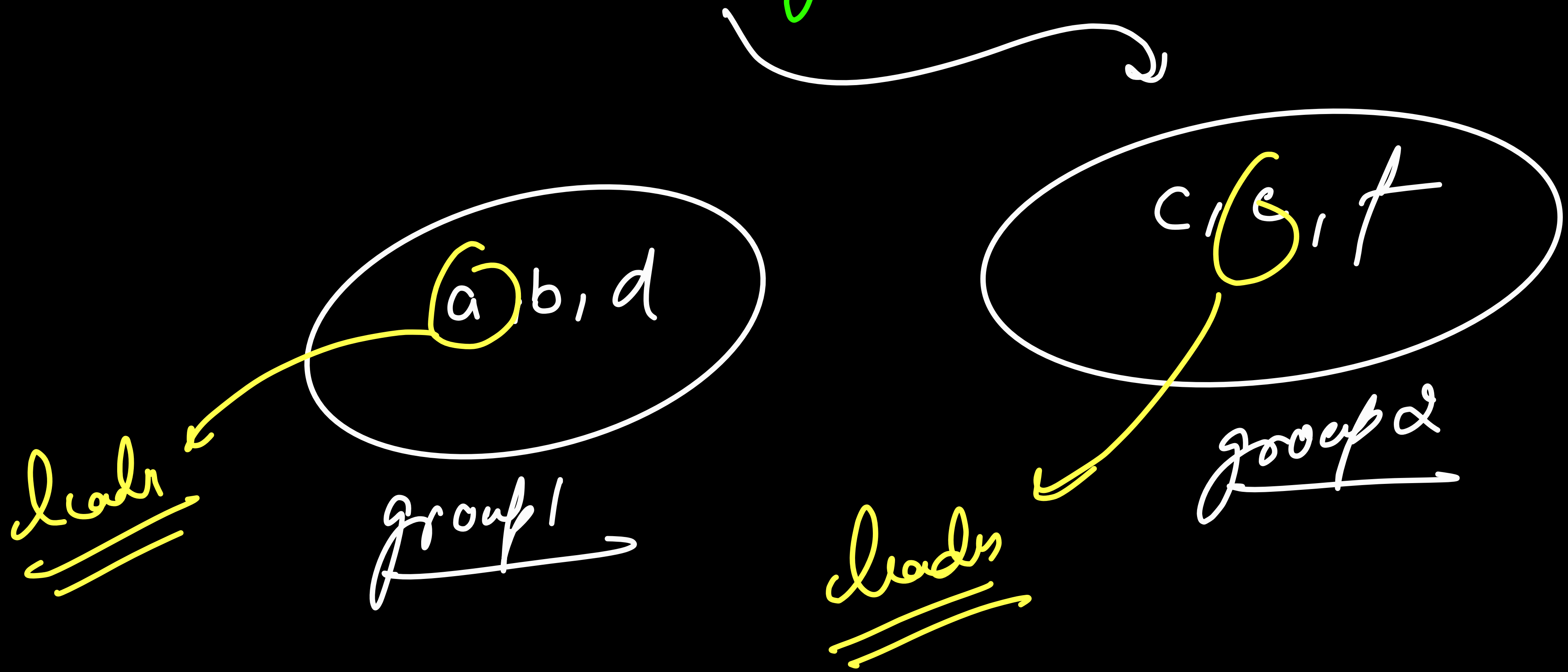
Disjoint Set Union (DSU)

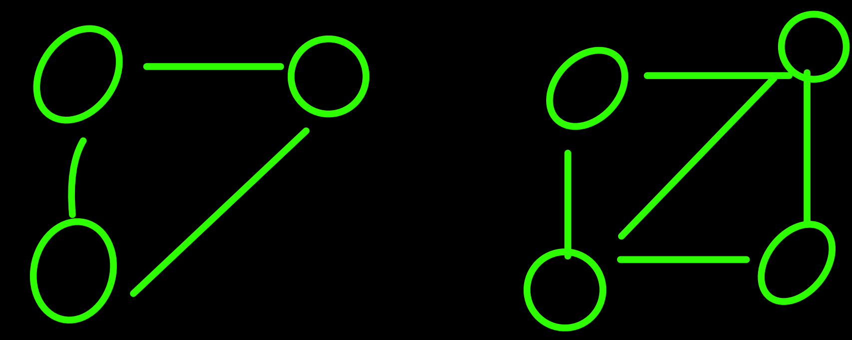
↳ Linear DS → stacks, queues, arrays, ll etc.

↳ Non-linear DS → Hashmaps
↳ Trees → BT, BST, heaps

clustering / grouping → You will be having some elements in diff groups / & you need to add them / segregate them & identify the clusters. and sometimes we might need to identify the group any element belongs to.

a, b, c, d ...... f, g, h





Terminologies

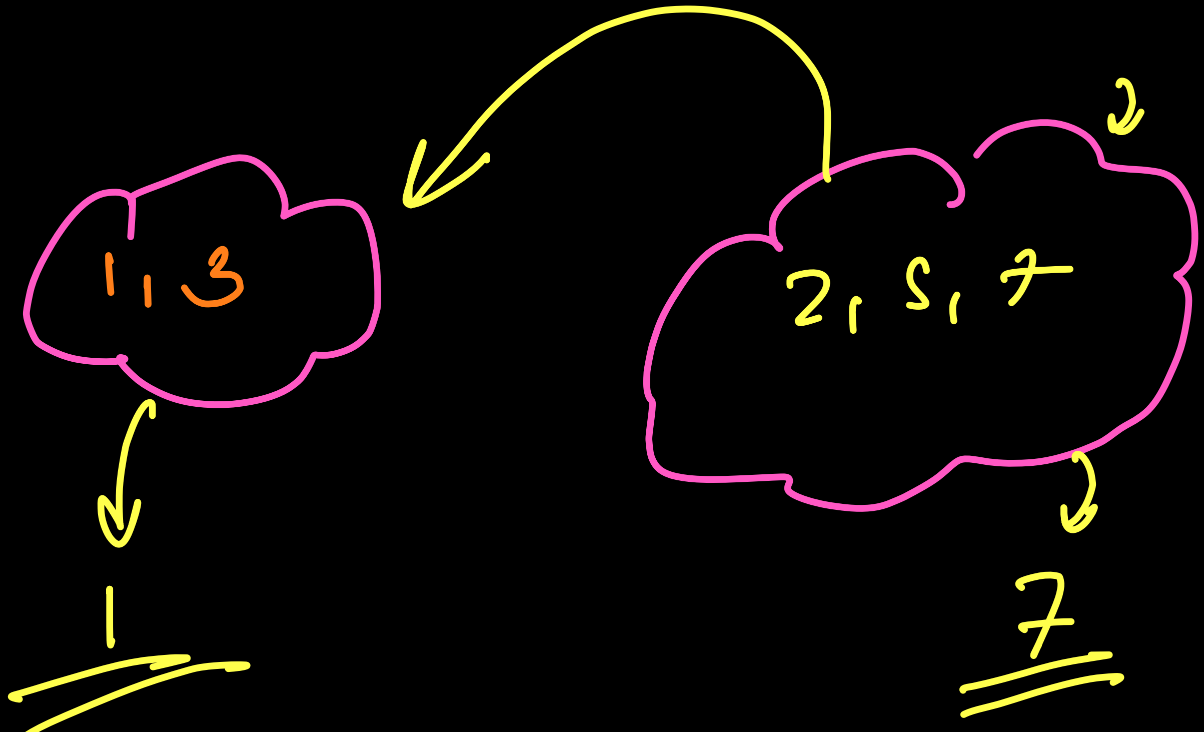
① Leader / parent of the group → to uniquely identify a group we will pick any element from the group & make it the representative / leader / parent of the group.

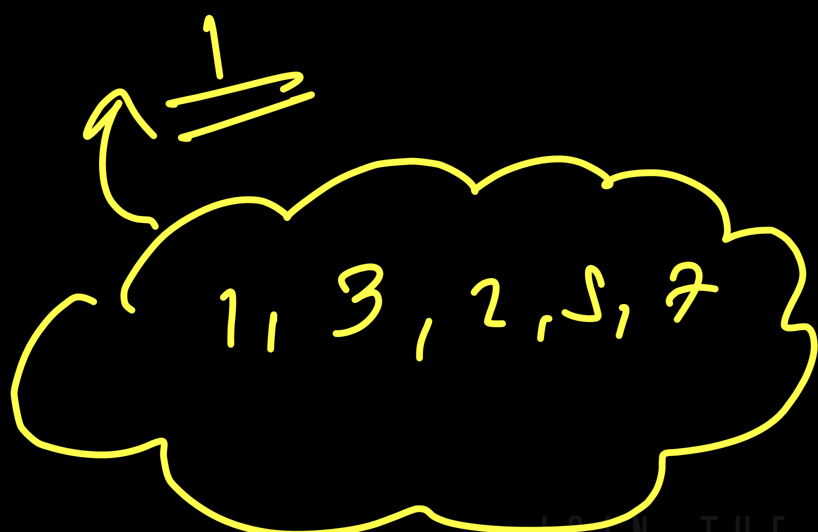
DSU

Q→ what func^n dsu need to support ??

① Union (a, b) → adds the group where element b belongs to the group where element A belongs or vice-a-versa.

ex → union (1, 5)





② find(x) / get(x) : this will be used to find which group x belongs to. we will return the parent of the group that x belongs to.

2, 3, 1, 6 → parent = 3

get(2) → 3

get(3) → 3

Approach → Represent every group as a set.

2

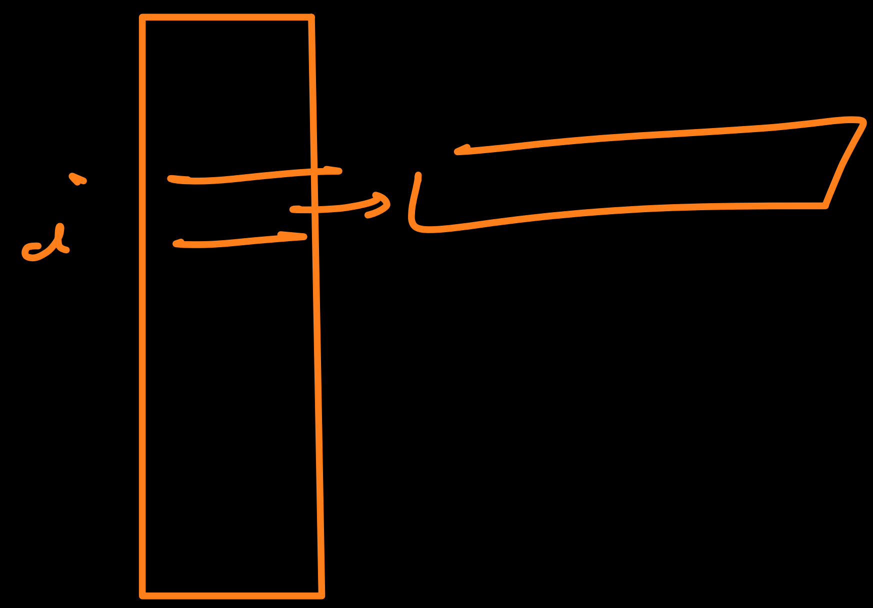
2, 3, 1, 6, 5

← set

7, 10

N/2

N/2



↳ array of sets

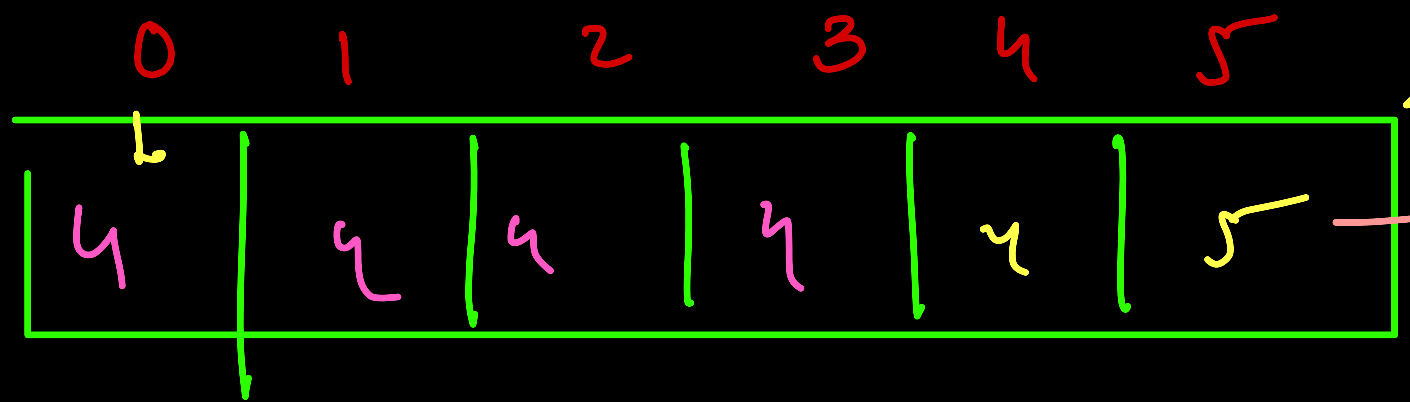
O(n log n) → O(n)

approach 2

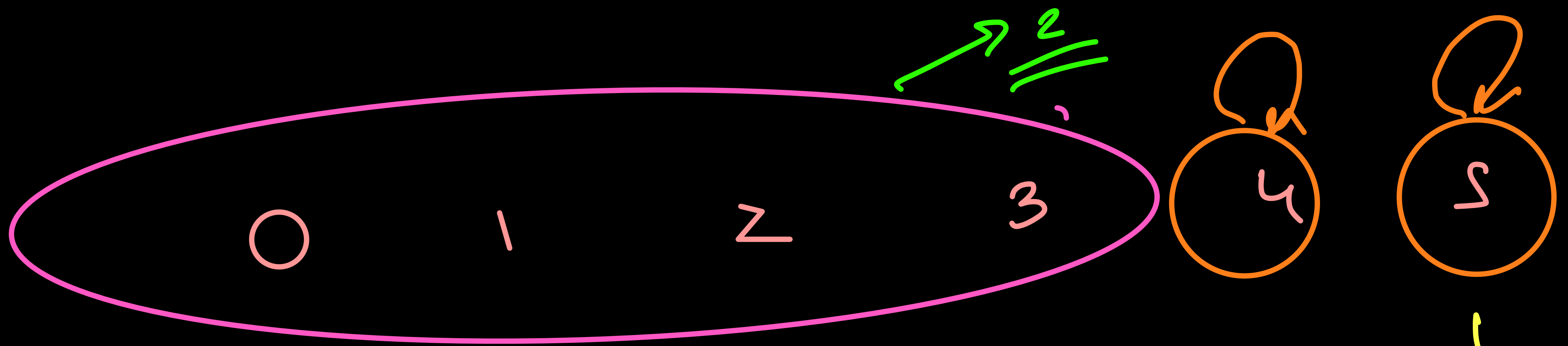
Can we use arrays ??

par

indexes ← represent actual element



→ values => par[i] denotes which group i belong to.



initially every element belongs to diff own grps.

union (0,1)

par[1] → p

union (2,3)

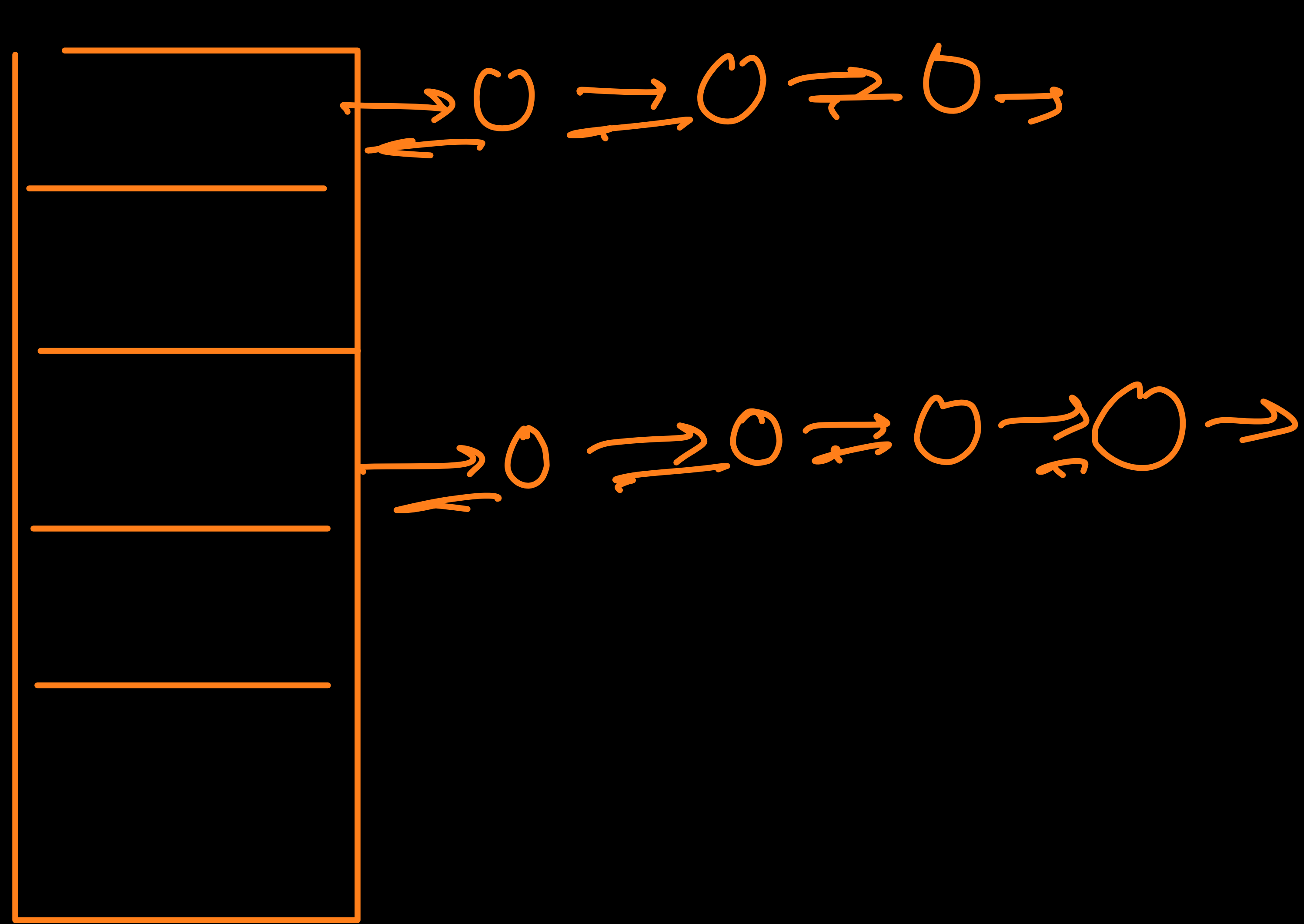
union (2,1)

union (4,3)

```
int find(x) {
    return par[x];          → O(1)
}

void union(a, b) {
    a = find(a)             → O(n)
    b = find(b)
    for(i=0; i<n; i++)
        if(par[i] == b) {
            par[i] = a
        }
}
```

$\underbrace{n \times n}_{n} \rightarrow O(n^2)$

LL



find $O(n)$

union $(O(1))$

0	1	2	3	4	5	6
3	3	3	3	4	5	6

n

$$\frac{n \times n}{n} \rightarrow O(n)$$

union (0,1)

union (2,1)

un (3,2)

un (4,3)

n

n union $\rightarrow$ n

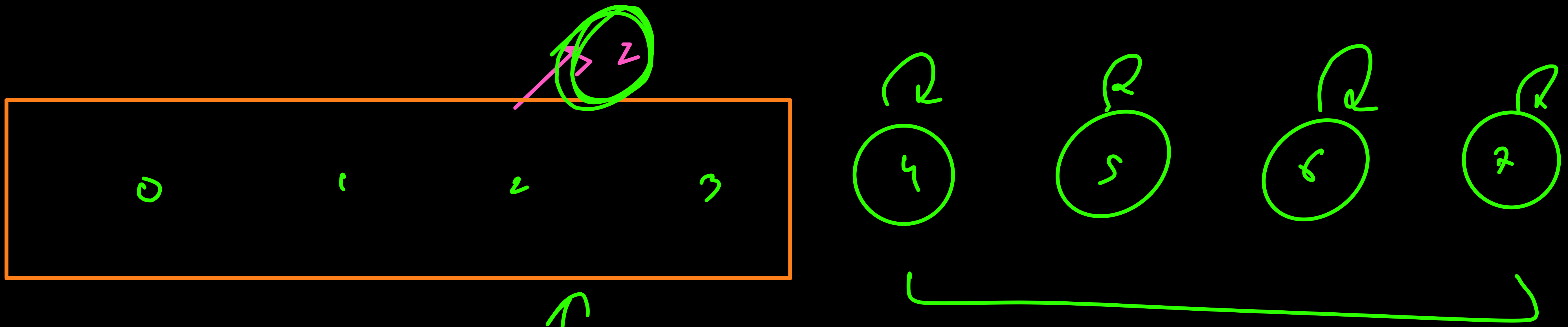
Union By Size → smaller → larger

2 ← 4 — 5

0	1	2	3	4	5	6	7
2	2	2	2	~~4~~ 2	4	6	7

size →

size[1] =+ size[2]



7 → 5

union(0,1)

un(2,3)

un(1,2)

1
2
4
8
⋮
2^k

$K \approx \log n$

$n \rightarrow \log n$

$2^k \leq n \rightarrow k \leq \log n$

$n \times \log n \longrightarrow O(\log n)$

n union

Q: if for an element we need to find parent, what is one of the good D.S. to represent child & parent ?? → Tree

	0	1	2	3	4	5	6	7
→ par	1	3	3	3	5	3	6	7
→ size	1	2	1	6	1	2	1	1

union (0,1)

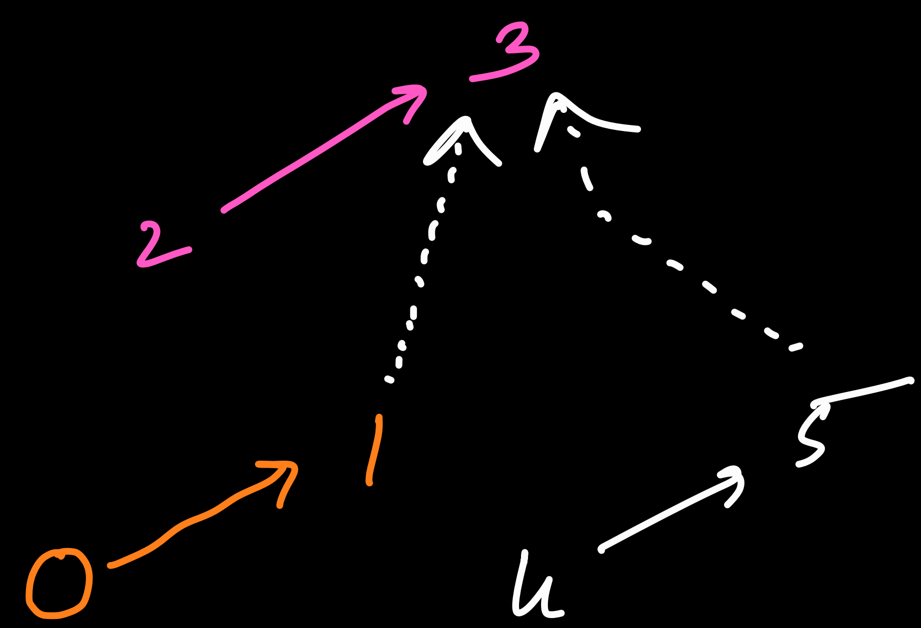
union (2,3)

union (2,0)

find (0)

un (4,5)

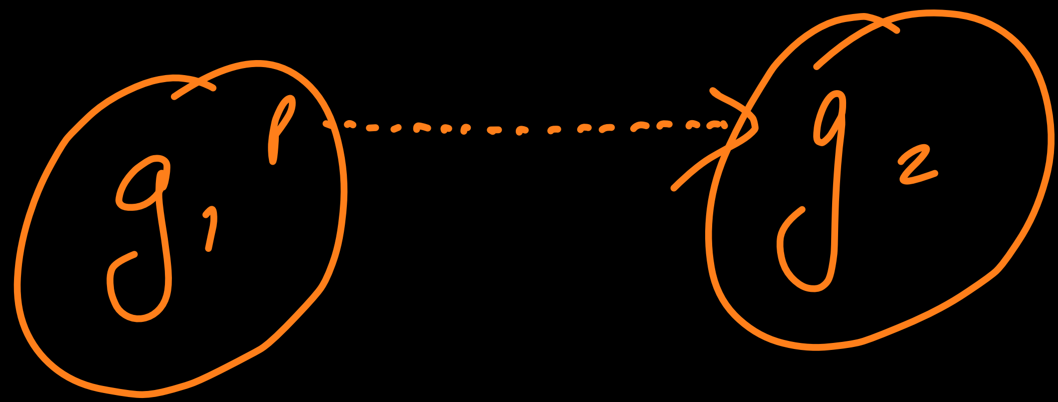
un (0,4)



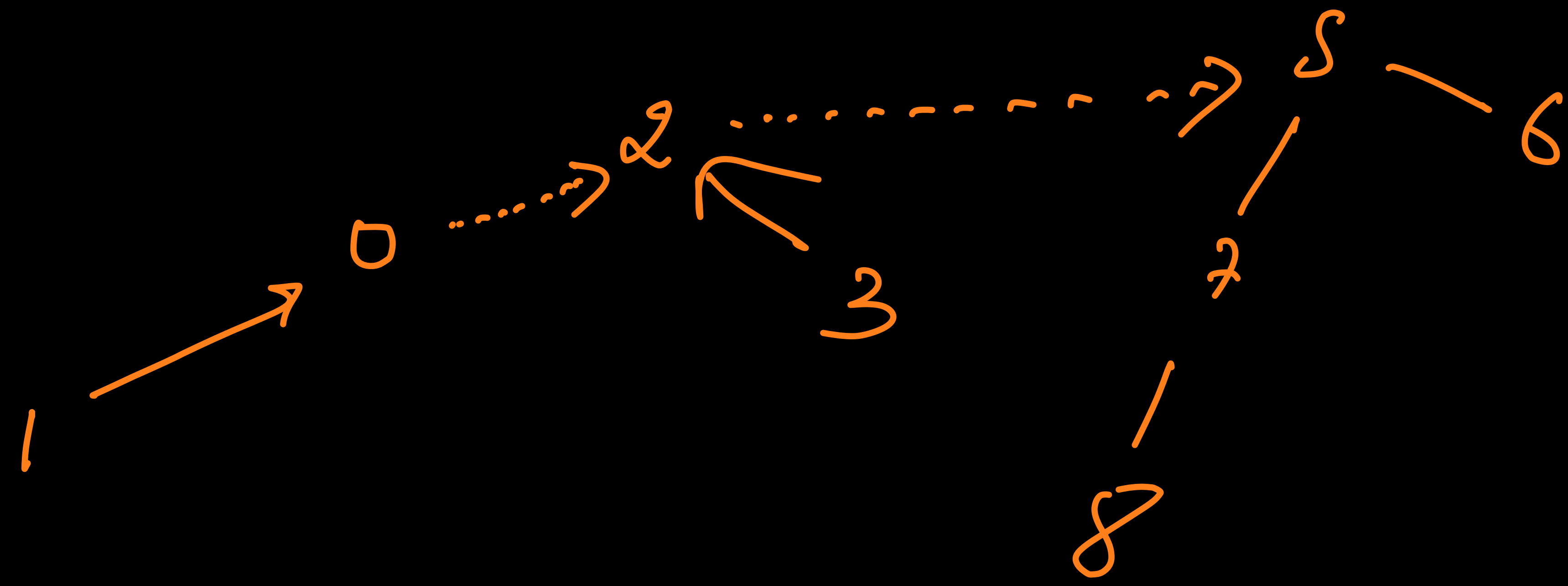
union by Rank

```
void union ( a, b ) {
    a = find (a)
    b = find (b)
    if ( sz [b] < sz [a] ) {
        sz [a] += sz [b]
        par [b] = a
    } else {
        sz [b] += sz [a]
        par [a] = b
    }
}
```

```
int find(x) {
    if(par[x] == x) return x;
    return find(par[x])
}
```



$Sz_{g_1} < Sz_{g_2}$



0	1	2	3	4	5	6	7
1	2	2	2	2	4	6	7

0	1	2	3	4	5	6	7
1	2	4	1	2	1	1	1

Rank

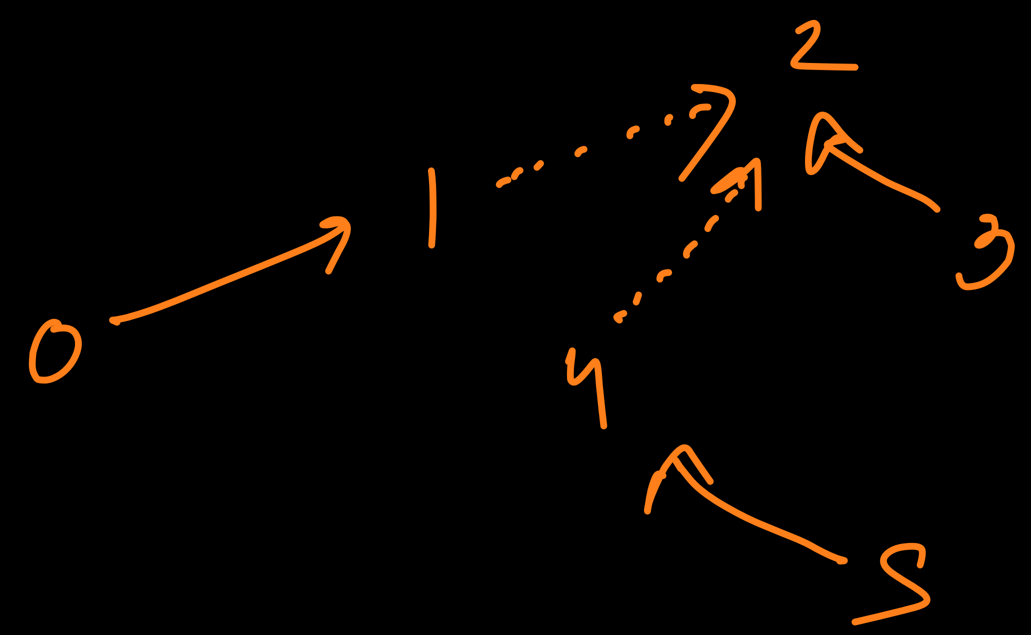
union (0, 1)

union (2, 3)

union (1, 3)

union (4, 5)

union (2, 4)



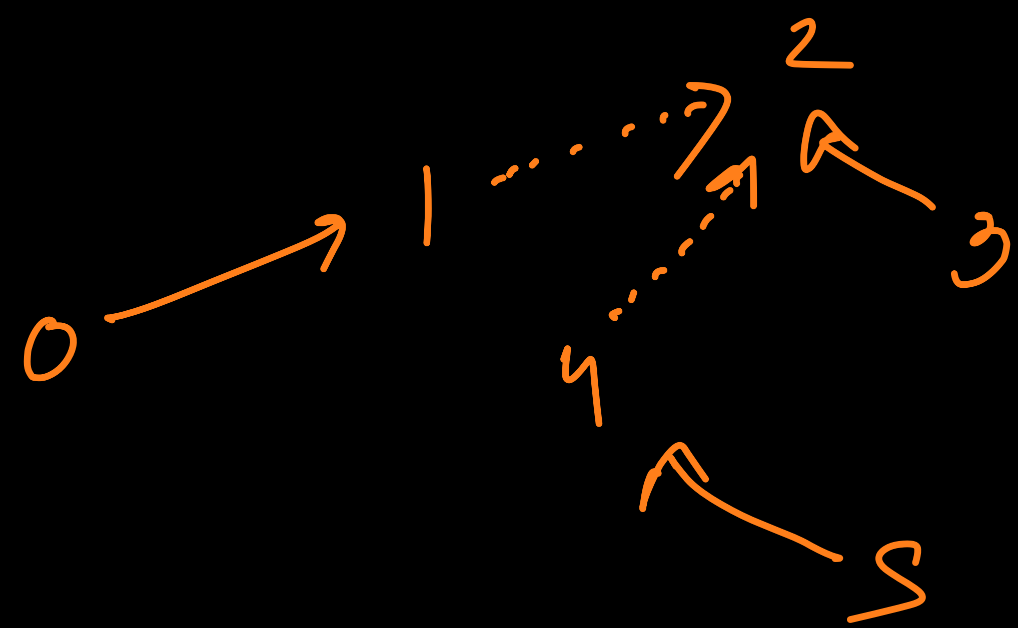
```
void union(a, b) {

    a = find(a)
    b = find(b)
    if( rank[a] <= rank[b])
        par[a] = b
        rank[b]++
    } else {
        par[b] = a
        rank[a]++

    }
}
```

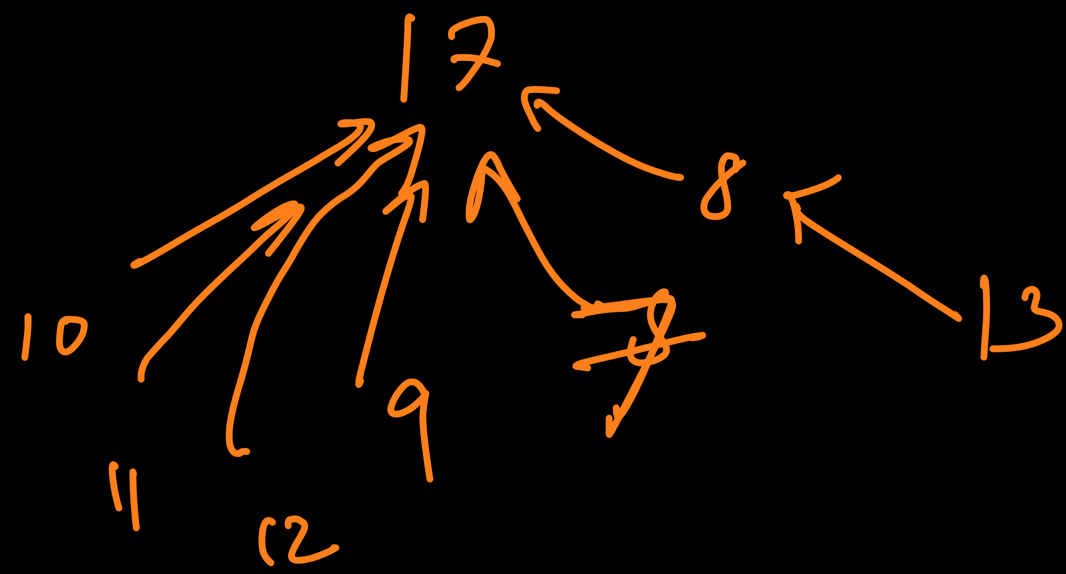
$O(\log n)$

union by size / union by rank with path compression

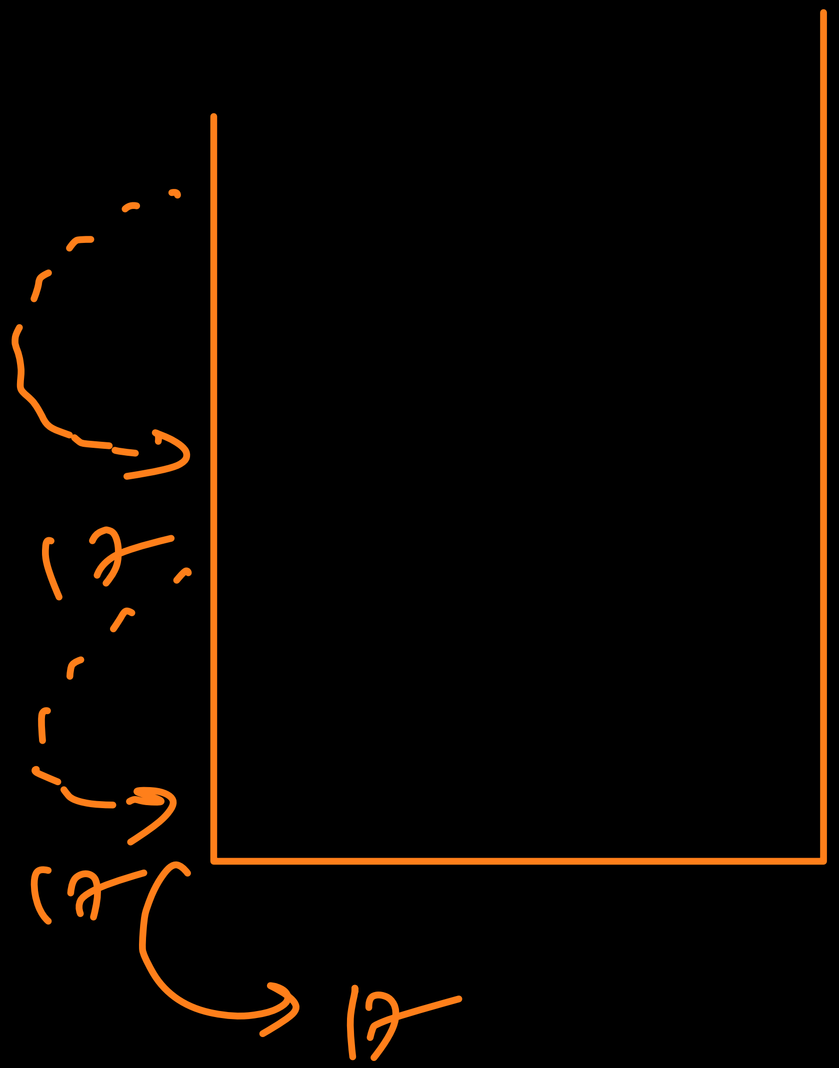
$O(\log^* n)$

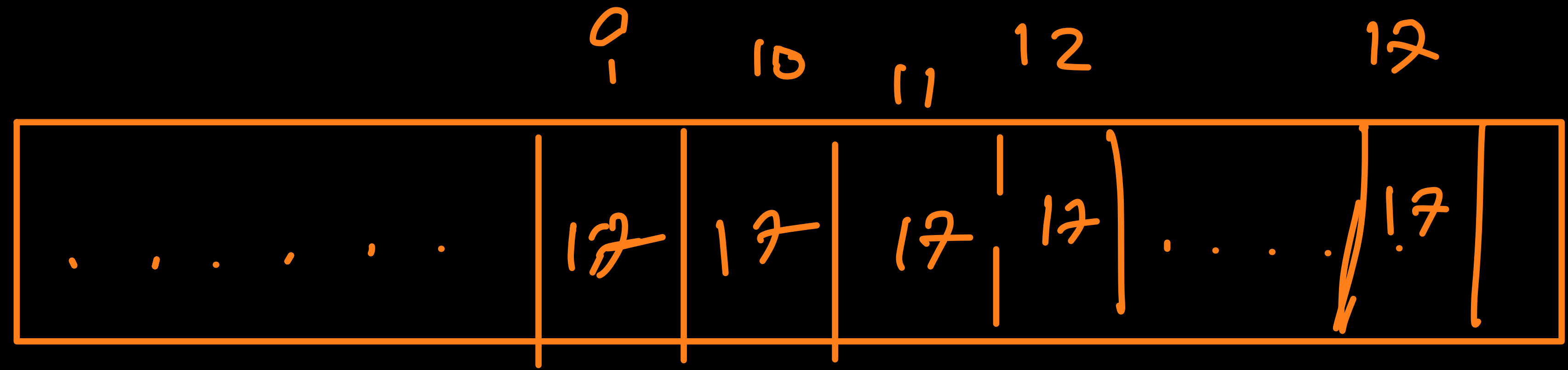
union (5, 12)





```
1)    int find(x) {
2)        if ( par[x] == x) return x;
3)    return  par[x] = find(par[x])
4)    }
```





inverse ackermann func^n

$n = \log_2 n$

$\log^* n$ → this represents that if you have a value of n, and you repeatedly apply $\log_2 n$ on this value then in how many ops you can reduce it to ≤ 1

$n = 65536$

$\hookrightarrow \log n = 16$

$\rightarrow \log_2(65536) = 16$

2^{16}

$n = 65536$

$n = \log_2 ?$??

$\log^* n \rightarrow \approx 5$

$\log_2 \overset{(1)}{(65536)} \rightarrow \log_2 \overset{(2)}{16} \rightarrow \log_2 \overset{(3)}{4} \rightarrow \log_2 \overset{(4)}{2}$

$\rightarrow \log_2 \overset{(5)}{(1)} \rightarrow 0$

$n = 2^{32}$

$\log^* n$? ?

(1)

$\log_2 2^{32} \to \log 2^{32} \to \log_2 5$

$\approx \log_2 2 \to \log_2 1 \to 0$

JOIN THE DARKSIDE