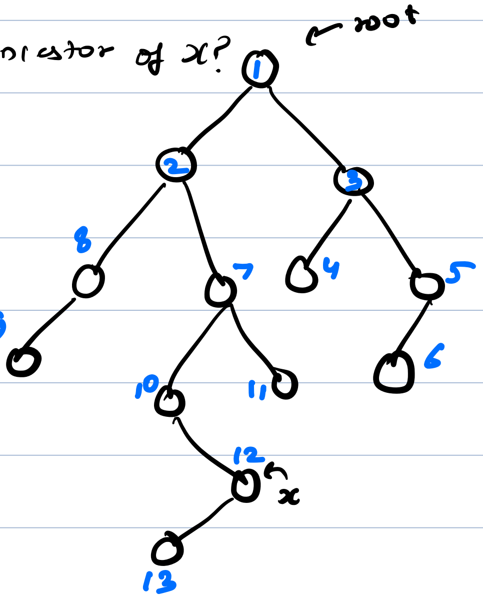
3rd Oct
Mon
Articulation

6th Oct
Thursday
AVL Trees

Binary lifting.

What is kth ancestor of x?



par[ ]

10^5

```
while(k--)
{   x = par[x]
```

Tc O(N)

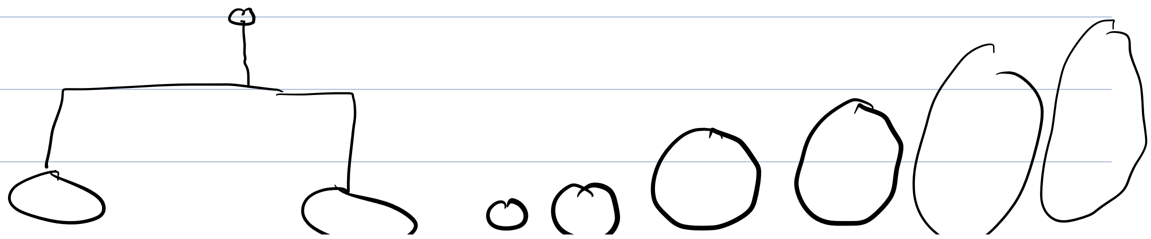
Sc O(N)

Maintain 1st par, 2nd par, 3rd par, . . . nth par

Tc : O(1)

Sc : O(N²)

1 − 50



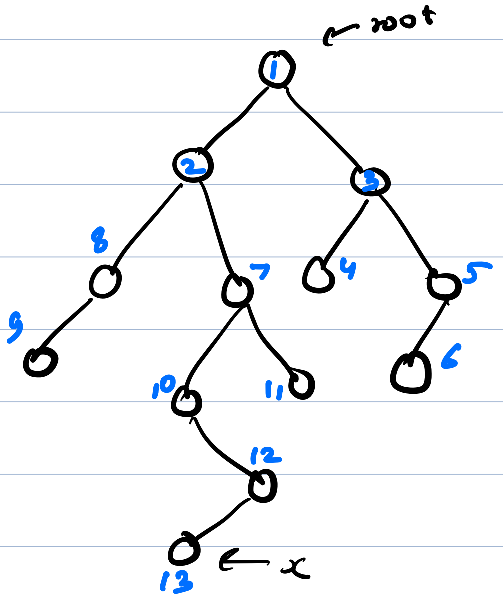
1 2 4 8 16 32

log 50 = 6

50
-32
18
-16
2
-2
0

110010

Maintain 1st par, 2nd par, 4th par, 8th par, 16th pos, 32 parth, 64 parth ...



10th ancestor.

TC: (log N)

SC: NlogN

up[N][log N]

up[i][0]	2^0 par of i
up[i][1]	2^1 par of i
up[i][2]	2^2 par of i
up[i][j]	2^j par of i

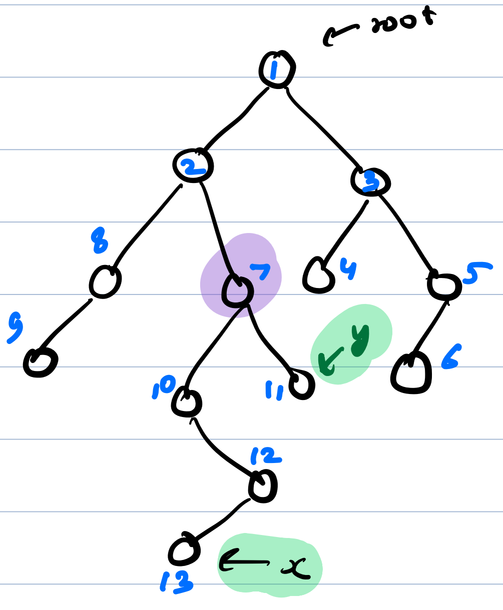
```
void dfs ( int curr, int par)
{
        up[curr][0] = par ,

                up[curr][1] = up[ up[curr][0] ][0]
                up[curr][2] = up[ up[curr][1] ][1]
                up[curr][3] = up[ up[curr][2] ][2]

        for ( i=1, i < log N ; i++)
        {       up[ curr] [i] = up[ up[curr][i-1] ][i-1]

        for( auto j : adj [curr])
        {
                if( j != par)
                        dfs( j, curr)
```

$$\begin{tikzcd} \text{up[curr][2]} \\ \text{up[curr][0]} \arrow[u] \\ \text{curr} \arrow[u] \end{tikzcd}$$

↑ No of Nodes

```
int kthanc (x, kth, N )
{
    for( i = logN-1 ; i ≥ 0 ; i--)
    {
        if( (k & (1<<i)) != 0 )
            x = up[x][i]
    }
    return x
}
```

LCA = lowest common ancestor.



LCA(x, y) = 7

LCA(11, 9) = 2

LCA(8, 6) = 1

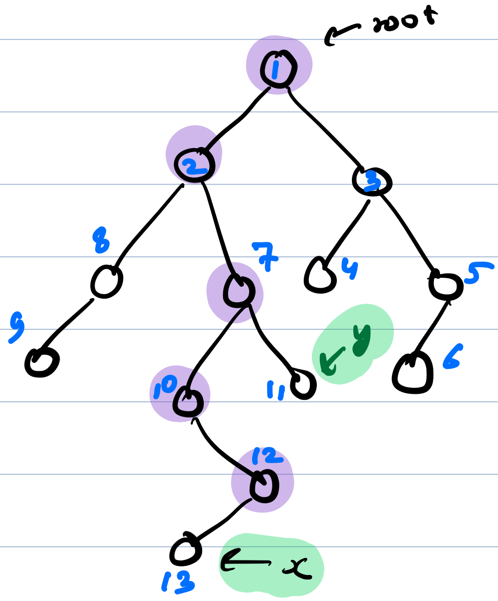
Check is x is a ancestor of y or not

⇓

If y is in subtree of x
then x is ancestor.

If x is an ancestor of y

```
bool IsAscentor ( x , y )
{
    if ( intime[y] ≥ intime[x]
            && outime[y] ≤ outime[x]
                return True
    else
                return False
}
```



12 → ✗

10 → ✗

7 → ✓

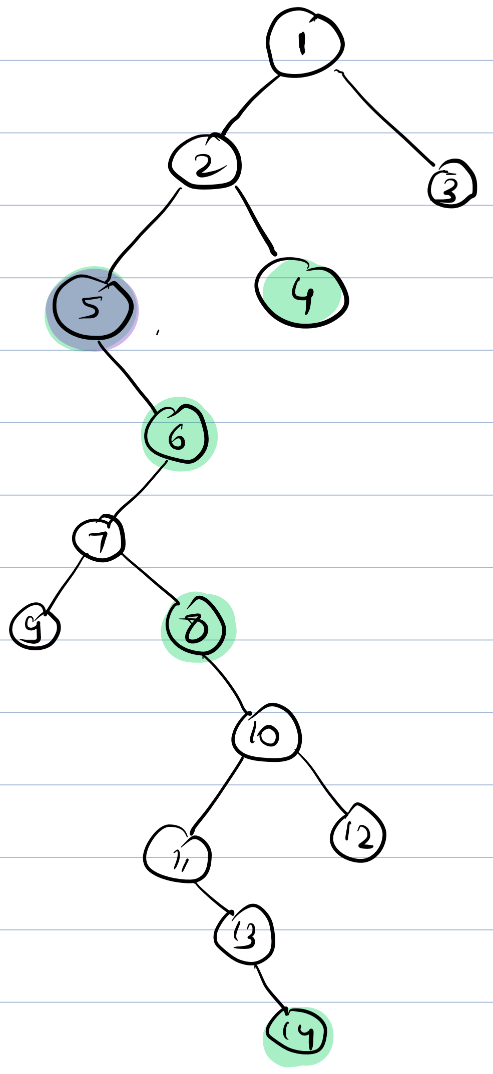
2 → ✓

1 → ✓

Goal is go as much up as you can

S.T node is not an ancestor y.

Once this is done, you are standing at child of LCA.



LCA(14,4)

$2^3 = 8$ ✗

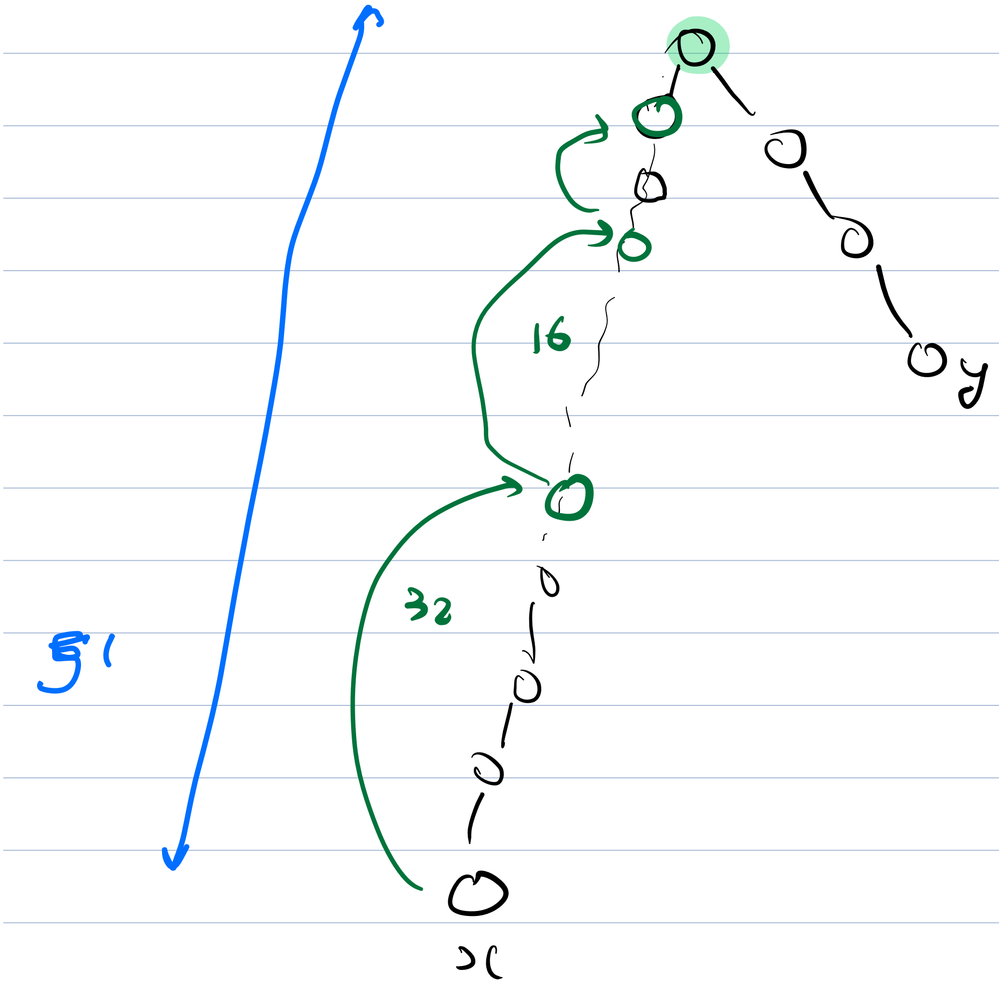
$2^2 = 4$ ✓

$2^1 = 2$ ✓

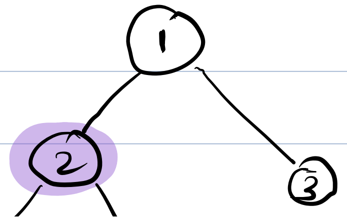
$2^0 = 1$

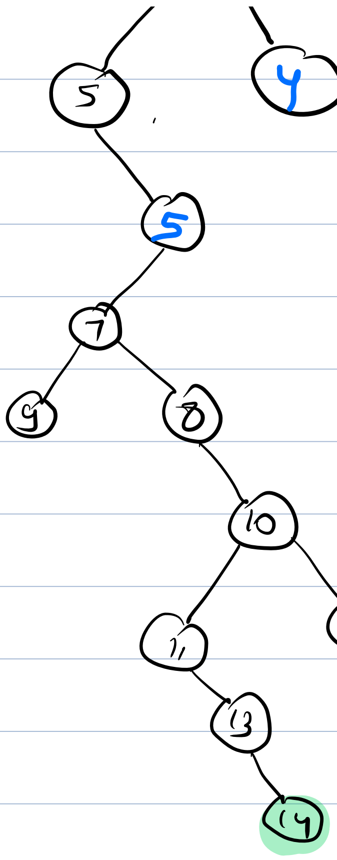
50

32	16	8	4	2	1
✓	✓	✗	✗	✓	✗



64	✗
32	✓
16	✓
8	✗
4	✗
2	✓
1	✗





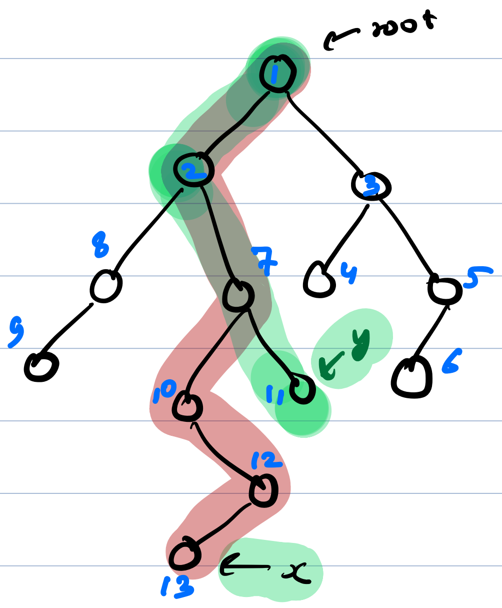
LCA(2,14)

= 2

```
int lca( int x, int y)
{
        if( is_ancestor(x, y))
                    return x

        if( is_ancestor( y,x ))
                    return y.

        for( i = logN-1 ; i ≥0 ; i-- )
        {
                temp = up[x][i]
        if ( is_ancestor ( temp, y ) == 0 )
            x = up[x][i]

    return up[x][0]
```

TC: log N.

Q. Tree, calculate distance b/w x & y.



dist(x,y) = 5

$6 + 4 - 2 \times 3$

$= 4$

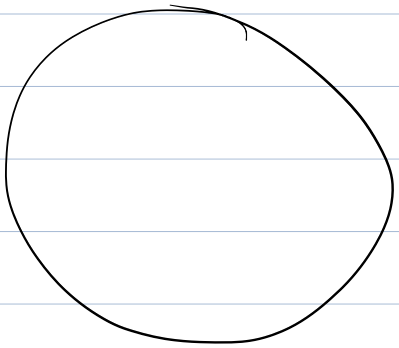
level[] → distance from root.

level[x] + level[y] − 2 level[LCA]+1

↓ root → x   ↓ root − y.

Doubt.

```
void dfs(curr, par)
{
        level[curr] = 1+ level[par]
```



```
}
```

3rd ——— 5th oct

mon   Thur

Articulation points

AVL Tree