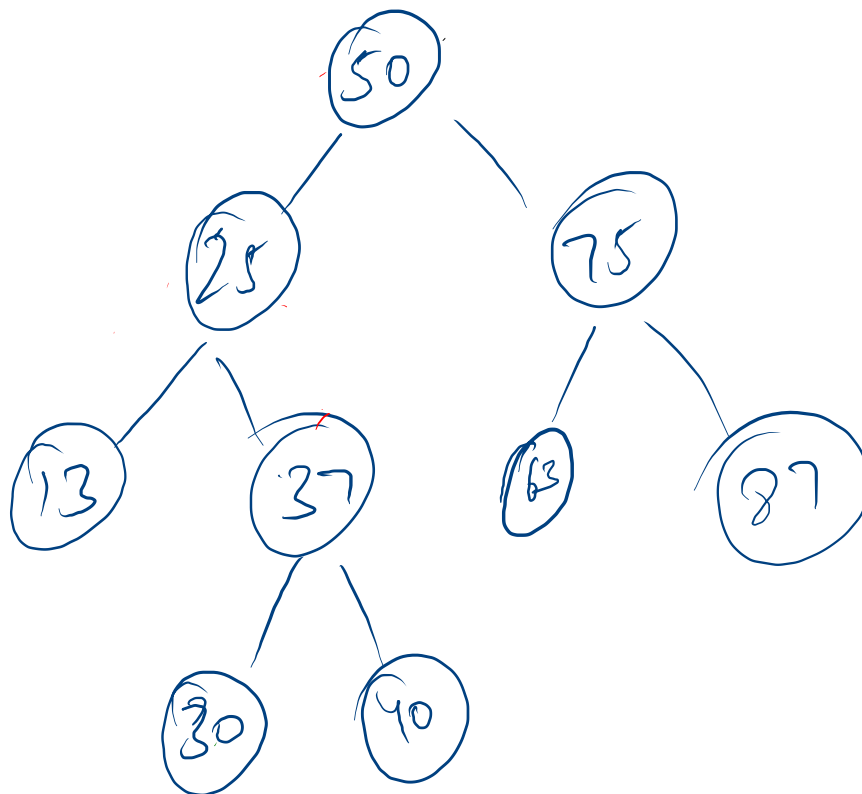


LCA



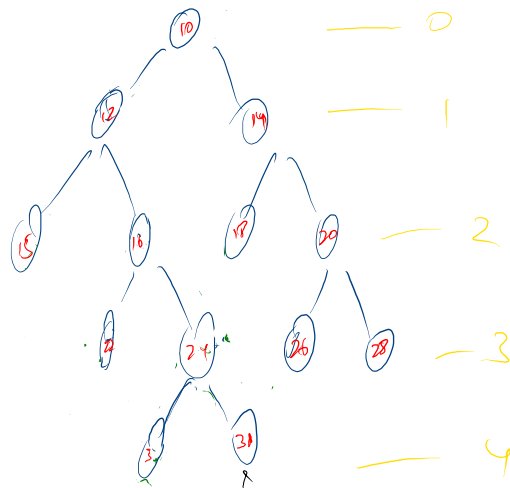
$O(h)$

$p = 37$

$q = 40$

node	parent
12	10
14	10
15	12
16	12
17	14
18	14
19	16
20	16
21	20
22	20
23	24
24	24
25	24
26	24
27	24
28	26
29	26

node	level
10	0
12	1
14	1
15	2
16	2



Preprocessing $\rightarrow O(n)$
time \leftrightarrow space $O(n)$
 \downarrow 500+

boolean flag1: false true
boolean flag2: false true
false

P = 16, Q = 27

$$\begin{cases} n & n \\ !n & n \\ n & !n \\ !n & !n \end{cases}$$

n1 = 15, n2 = 30

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

3 2 4 0 14 10 -1 -2 4 4 -3 8 6 -7 -3 8

4	7	16	1	0	0	0	0									
-3	15		0	1	1	1	1	1								
-1	14		0	1	1	1	1	0								
6	13		0	1	1	0	1									
13	12		0	1	1	0	0									
-3	11		0	1	0	1	1									
8	10		0	1	0	1	0									
4	9		0	1	0	0	1									
3	0	8	0	1	0	0	0									
1	7		0	0	1	1	1									
4	6		0	0	1	1	0									
14	5		0	0	1	0	1									
9	4		0	0	1	0	0									
4	3		0	0	0	1	1									
5	2		0	0	0	1	0									
3	1		0	0	0	0	1									
0	0		0	0	0	0	0									

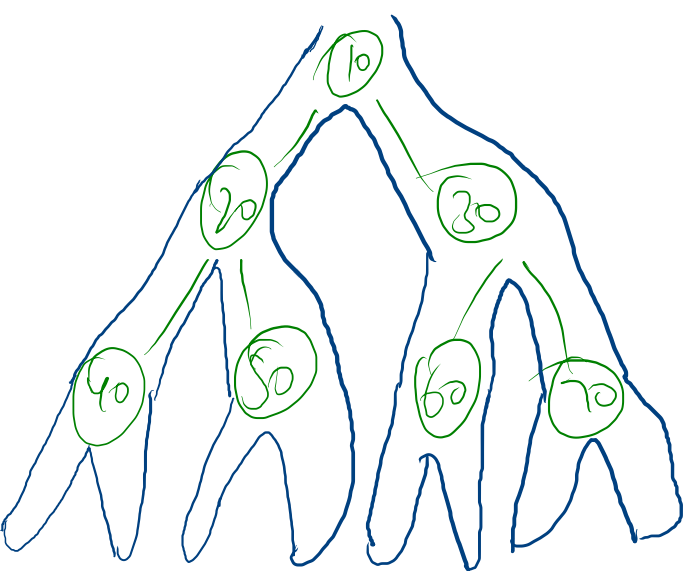
update

$$idn = idn + (idn \& -idn)$$

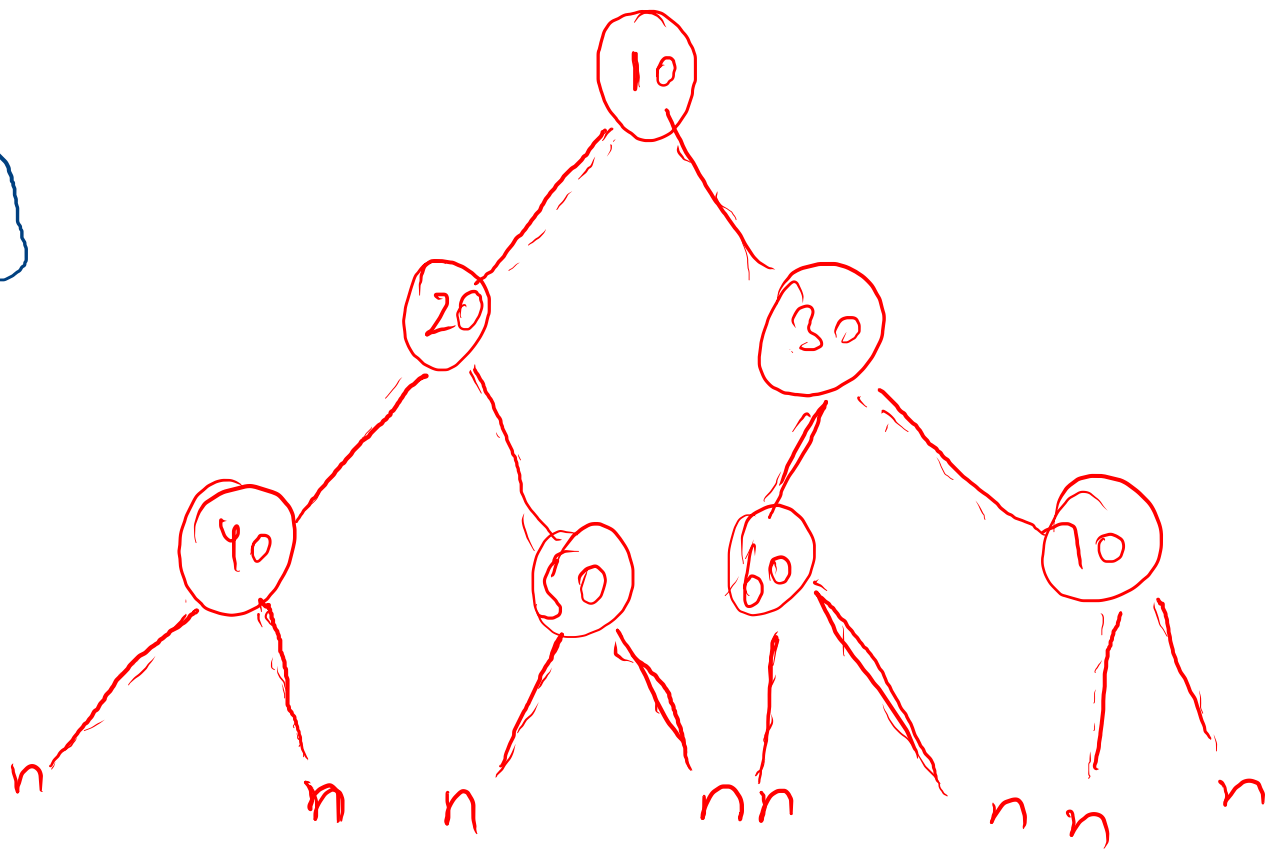
```

f 5 void update(int arr[],
               int idn,
               int delta) {
    arr[idn] += delta;
    while (idn < arr.length) {
        arr[idn] += delta;
        idn = idn + (idn & -idn);
    }
}

```



10 - 20 - 40 - # - # - 50 - # - # - 30 - 60 - # - # -
 70 - # - # -

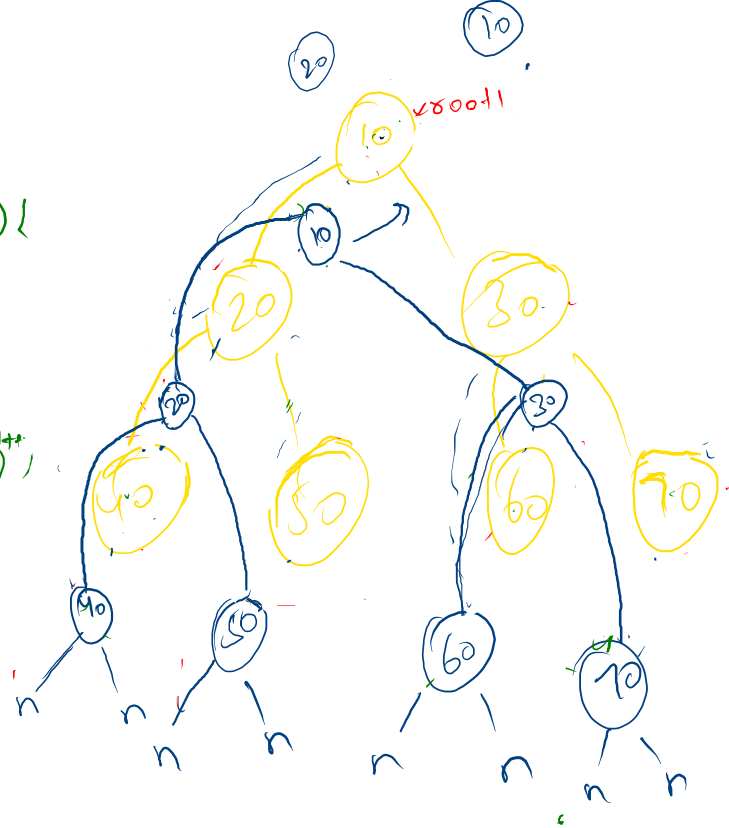


⑪ random pointer assign.

```

P S void rpa(Node root) {
    if (root == null)
        return;
    if (root->rand != null)
        root->left->rand = root->rand->left;
    rpa(root->left->left);
    rpa(root->right);
}

```

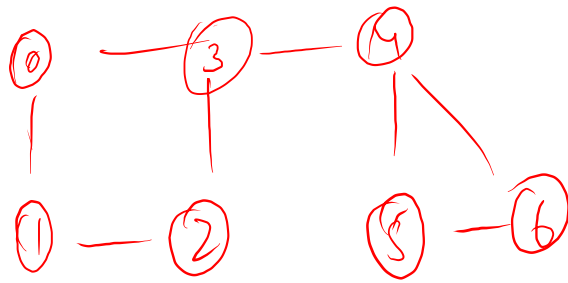


```

P S node correction (Node root) {
    if (root == null)
        return;
    Node ls = correction(root->left->left);
    Node rs = correction(root->right);
    Node temp = root->left;
    root->left = root->left->left;
    temp->left = ls;
    temp->right = rs;
    return temp;
}

```

Eulerian Path & Circuit
 undirected
 directed



undirected (degree)

↳ eulerian circuit \Rightarrow degree \rightarrow even

↳ eulerian Path \Rightarrow $\left. \begin{matrix} \text{degree} \rightarrow \text{even} \\ n-2 \text{ nodes} \end{matrix} \right\}$ 2 nodes odd

directed \rightarrow indegree or outdegree

↳ eulerian circuit \Rightarrow

↳ eulerian Path \Rightarrow

indegree = outdegree

$n-2$ nodes

indegree = outdegree



① indegree = outdegree + 1

② outdegree = indegree + 1