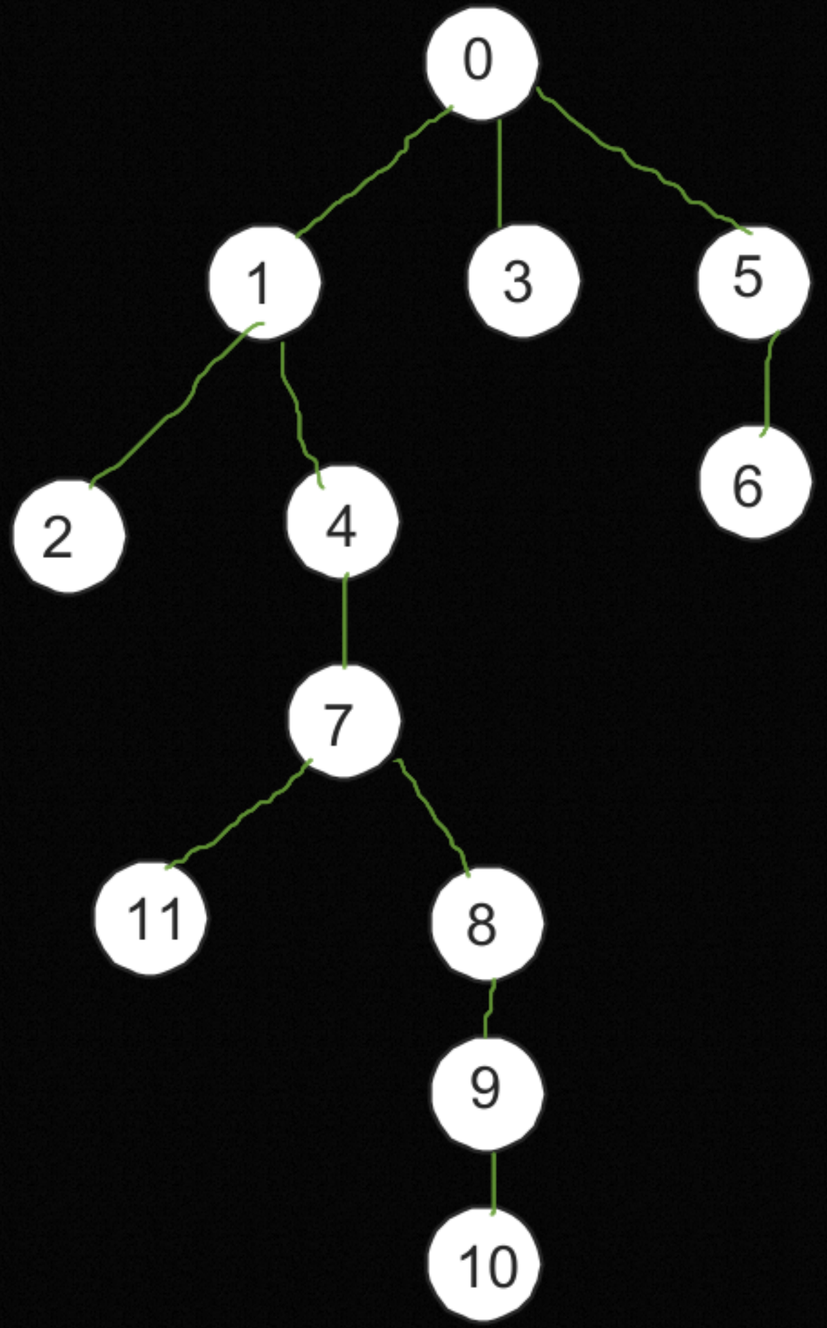
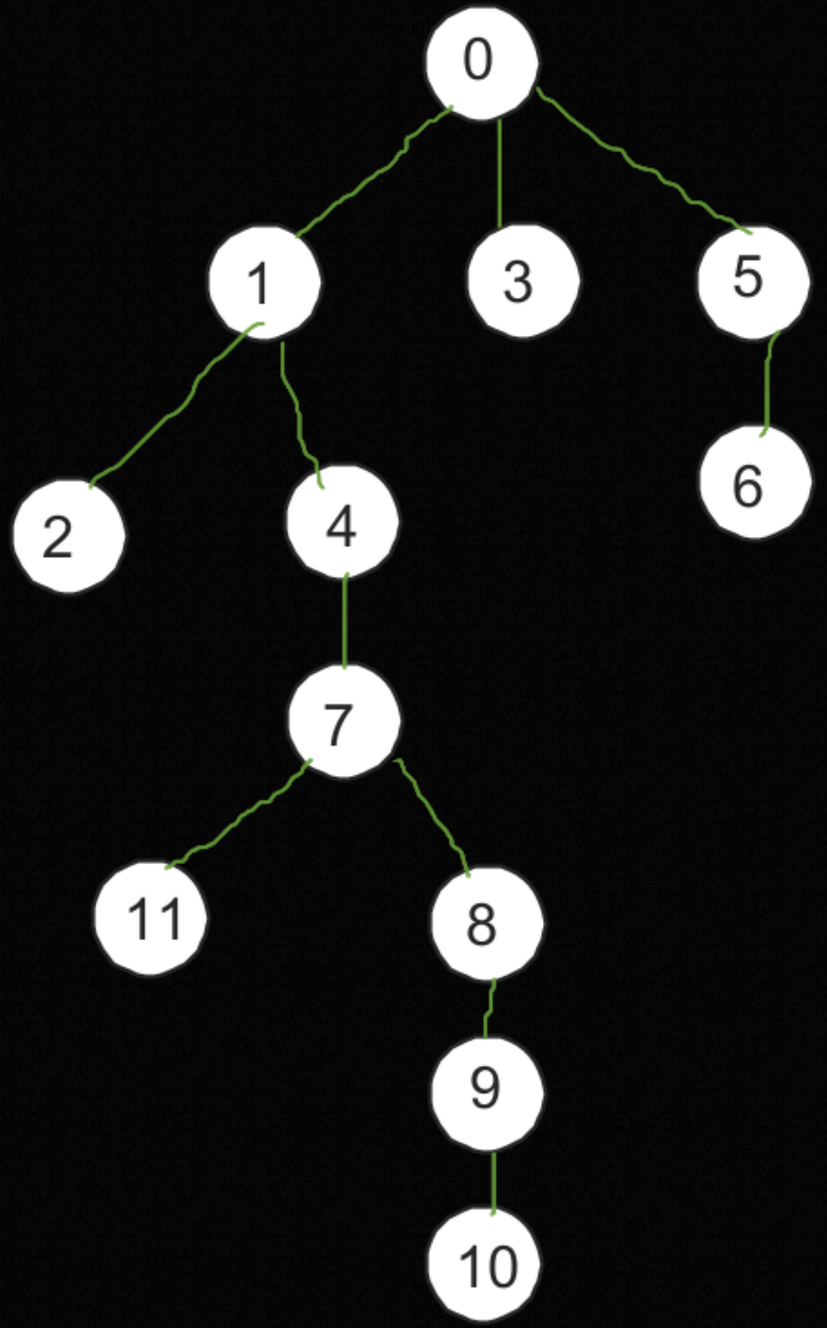
Binary Lifting



Find Kth ancestor of a given node U

Qry(10,0)=10
Qry(10,1)=9
Qry(10,2)=8
Qry(10,3)=7
Qry(10,5)=1

Qry(6,2)=0
Qry(11,3)=1
Qry(1,0)=0
Qry(7,3)=0

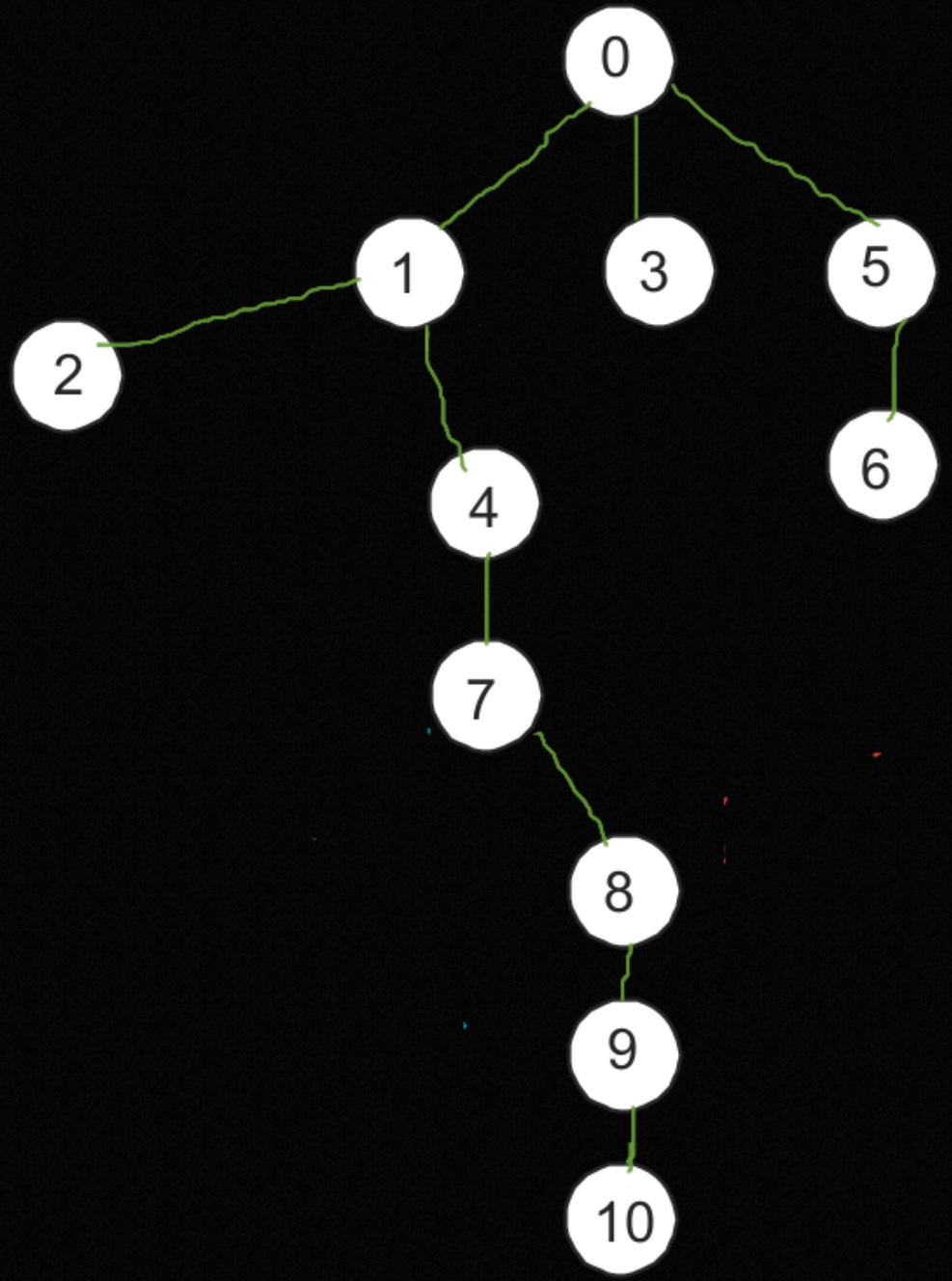


Up[u][j] = 2^j th ancestor of u

Up[u][0] = 2^0 or 1st ancestor of U
Up[u][0] = parent[u]

Up[4][0] = 1
Up[9][0] = 8
Up[11][0]= 7

Up[9][1] = 7 ; 2^1=2 ; Qry(9,2) = 7
Up[9][2]= 1 ; 2^2= 4 ; Qry(9,4) = 1

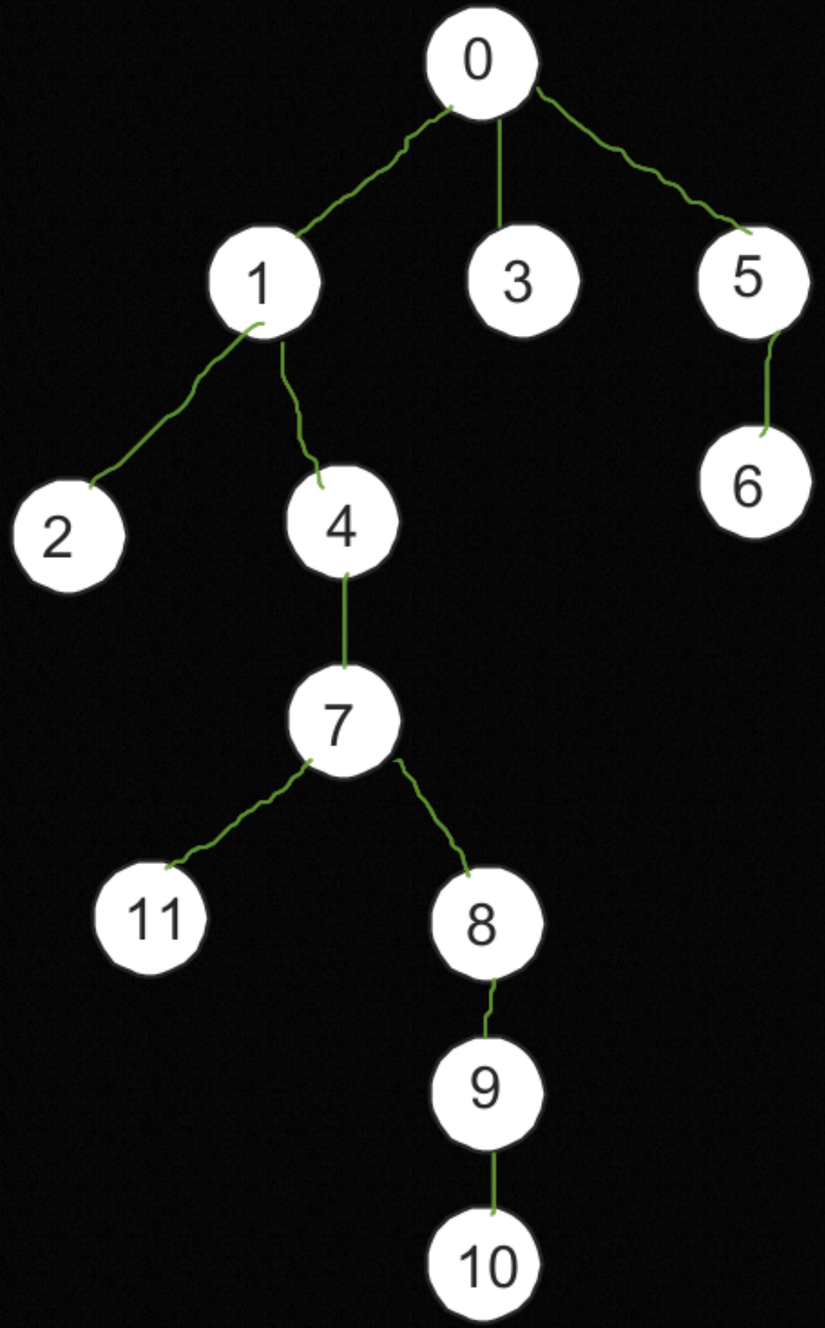


Up[u][j] = 2^j th ancestor of u

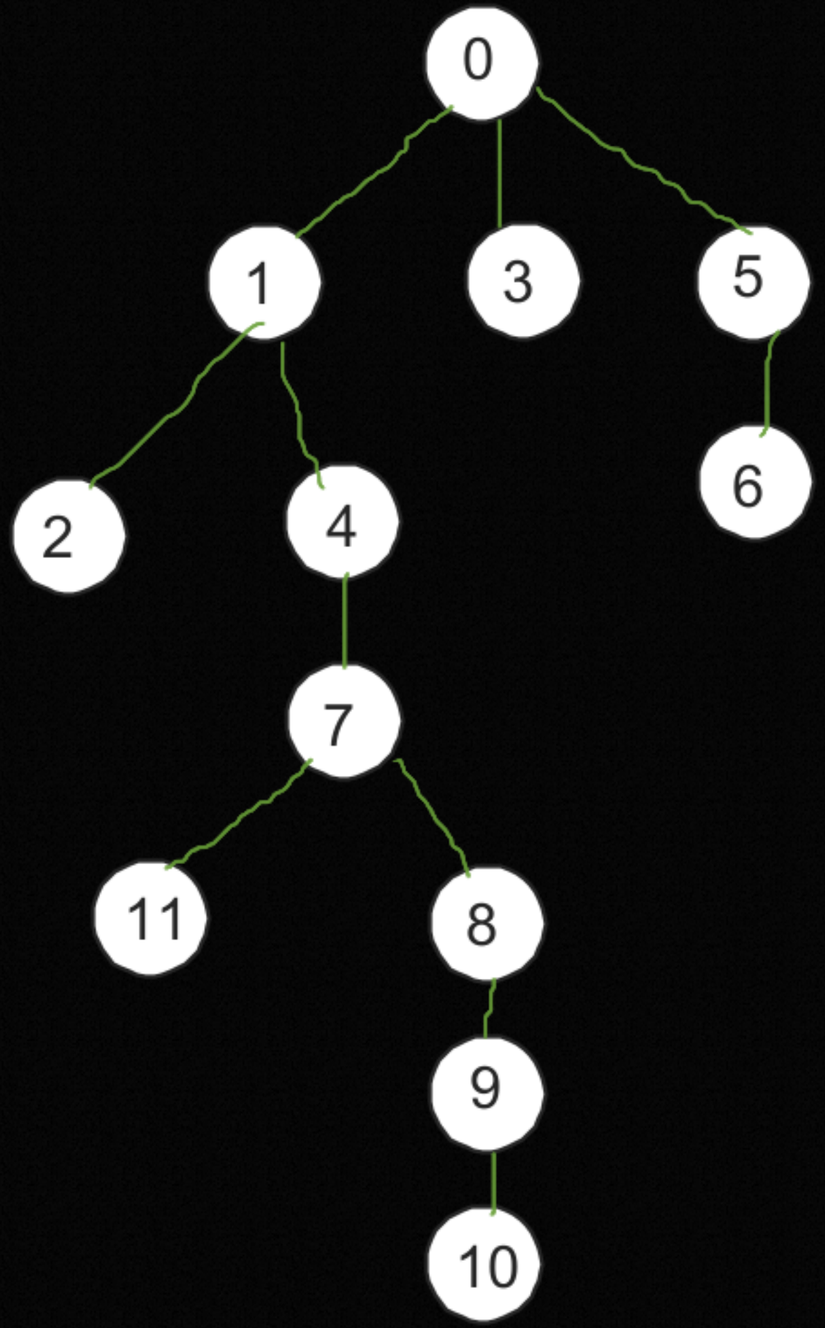
Up[7][1] = 1 ; 2^1=2 ; Qry(7,2) = 1
Up[9][1] = 7 ; 2^1=2 ; Qry(9,2) = 7
Up[9][2]= 1 ; 2^2= 4 ; Qry(9,4) = 1

Up[9][2] = Up[7][1]
Up[9][2] = Up[Up[9][1]] [1]

Up[u][j] = Up[Up[u][j-1]] [j-1]




```
// up[v][j] is 2^j -th ancestor of node v
 parent[0] = 0;
 for(int v = 0; v < n; v++) {
      up[v][0] = parent[v];
      for(int j = 1; j < LOG; j++) {
           up[v][j] = up[ up[v][j-1] ][j-1];
      }
 }
```

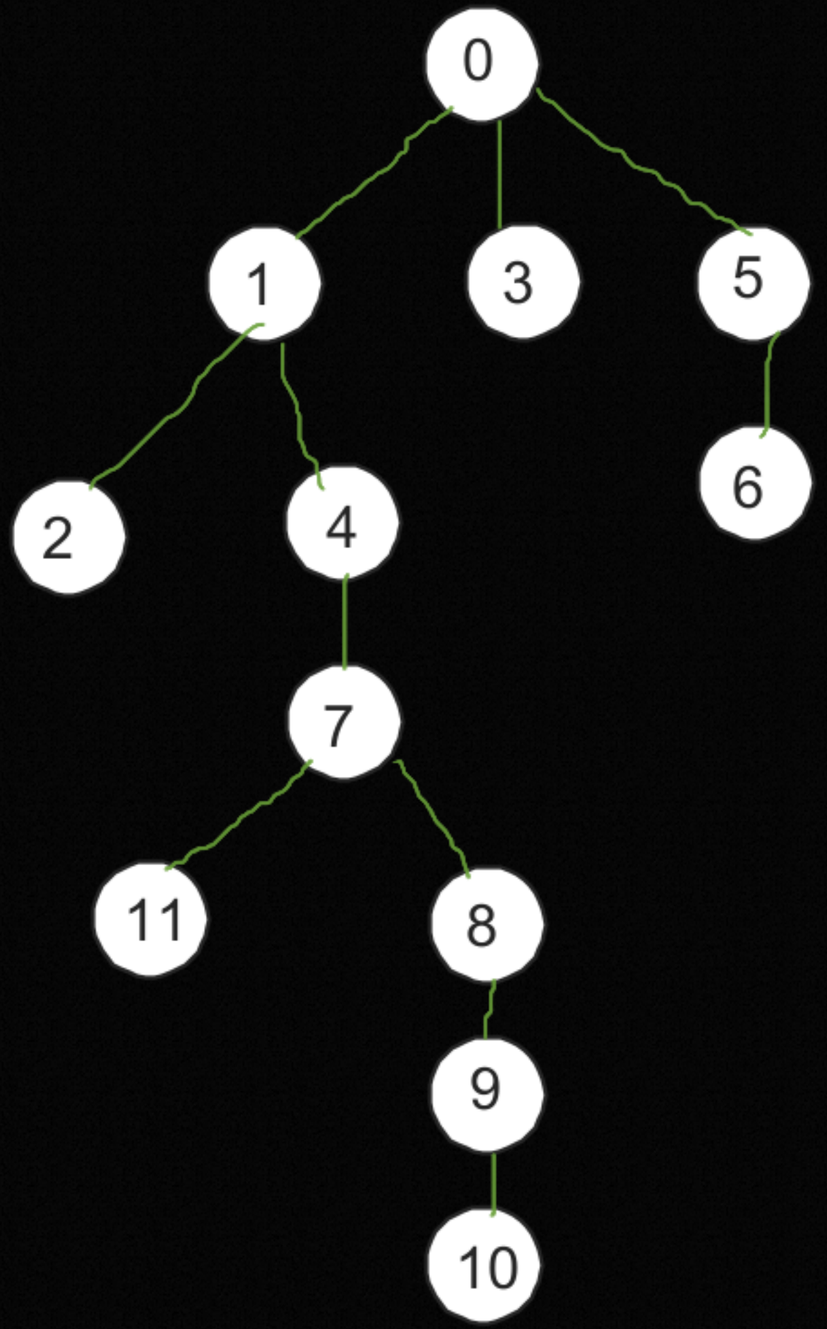


// qry (v , k) = kth parent of v
up [v][j] = qry (v , 2^j)
up [8][2] = qry (8, 4) = 0
up [10][1] = qry (10 , 2) = 8

qry (10 , 6) = 6th parent of 10 = 0

Binary Form of 6 => 0110

qry(10,6) = up[10][1]=> up[8][2]
= 8 => 0



qry (10 , 6) = 6th parent of 10 = 0

Binary Form of 6 => 0110
=> 2^2 + 2^1

qry(10,6) = up[10][1]=> up[8][2]
= 8 => 0

```
int getKthAncestor(int node, int k) {

if(depth[node] < k) {
    return -1;
}
for(int j = 0; j <LOG ; j++) {
    if(k & (1 << j)) {
        node = up[node][j];
    }
}
return node;

}
```