DP on trees

Page

Date

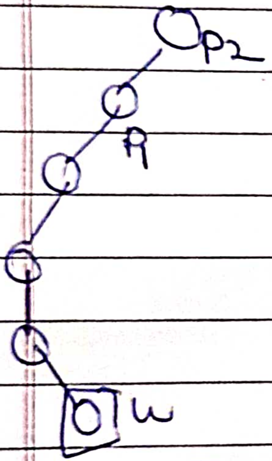
Binary Lifting

Given a tree, find which node is K level above.

for every node : u
$\Rightarrow up(u, 2^i)$

$\Rightarrow$ And a query (u, K)

$\Rightarrow K = (0101\,0110)_2 = 2^4 + 2^2 + 2^1$



$K = 5$

$P_1 = up(u, 2) \Rightarrow 2^2$

$P_2 = up(P_1, 0) \Rightarrow 2^0$

1. To find $up[root][x] = -1$
2. $up(u, 0) = par(u)$ [use DFS]

$up(u, x) = up[up[u, x-1], x-1]$

Date LCA in O(logN) Page

```
int lca(int u, int v)

        if(lvl[u] < lvl[v])
            swap(u,v).

        u= lift_node(u, lvl[u]- lvl[v]);

        if(u==v)
            return u;

        for(int i=19; i>=0;i--){
            if(up[u][i] != up[v][i]){
                u= up[u][i];
                v= up[v][i];
            }
        }
        return lift_node(u,1);
}

int lift_node(int node, int jump_req){
    for(i=19; i>=0; i--){
        if(jump_req==0 || node==-1)
        break;

        if(jump_req>= (1<<i)){
            jump_req -=(1<<i);
            node=up[node][i];
        }
    }
    return node;
}
```

Date   Page

```
void dfs (int src, int par, int level){

        level[src]= level;
        for(int child: tree[src]){
            if( child!=par)
               dfs(child, src, level+1);
            }
        }
}


void binary_lifting(int src, int par){

        up[src][0]=par;
        for(int i=1; i<20; i++){
            if(up[src][i-1]!=-1){
                up[src][i]=up[up[src][i-1]][i-1];
            else
                up[src][i]=-1;
        }
        for(int child: tree[src]){
            if(child!=par){
                binary_lifting(child, src);
            }
        }
}
```