

# LCA

<http://www.spoj.com/problems/LCA/>

```
#include<bits/stdc++.h>

using namespace std;
#define log 11
const int MAXN = 1030;
vector<int> adj[MAXN];
int pa[MAXN][log], P[MAXN], depth[MAXN], n;

void pre_process(){

    //pa[i][j] --> 2^jth parent of node i

    //Initialize every element of parent matrix
    //with -1
    for(int i = 0;i<n;i++){
        for(int j = 0;j < log;j++){
            pa[i][j] = -1;
        }

        //first ancestor of every node i is its
        //parent P[i]
        for(int i = 0;i<n;i++)
            pa[i][0] = P[i];
    }
```

```

//bottom up dynammic programming
//for filling pa[i][j]
for(int i = 0;i<n;i++){
    for(int j = 1;(1<=j) < n;j++){
        if(pa[i][j-1] != -1)
            pa[i][j] = pa[pa[i][j-1]][j-1];
    }
}
return;
}

```

```

//Need to do dfs for calculating
//depth and Parent of every matrix
void dfs(int cur, int prev, int d){
    P[cur] = prev;
    depth[cur] = d;
    //cout<<cur<<" "<<depth[cur]<<endl;
    for(int i = 0;i<adj[cur].size();i++){
        int v = adj[cur][i];
        if(v != prev){
            dfs(v,cur,d+1);
        }
    }
}
return;
}

```

```

int lca(int u, int v){

    //cout<<u<<" "<<depth[u]<<endl<<v<<" "<<depth[v]<<endl;

    if(depth[u] < depth[v])
        swap(u,v);
    int diff = depth[u] - depth[v];
    for(int i = 0;i<log;i++){
        if((diff >> i) & 1)
            u = pa[u][i];
    }
    //cout<<u<<" "<<v<<endl;
    if(u == v)
        return u;
    for(int i = log-1;i>=0;i--){
        if(pa[u][i] != pa[v][i]){
            u = pa[u][i];
            v = pa[v][i];
            //cout<<i<<" "<<u<<" "<<v<<endl;
        }
    }
    return pa[u][0];
}

int main(){
    int t,sz,u,v,ch,q,cnt = 0;
    cin>>t;

```

```

while(t--){
    cnt++;
    cin>>n;
    for(int i = 0;i<n;i++){
        cin>>sz;
        for(int j = 1;j<=sz;j++){
            cin>>u;
            P[u-1] = i;
            adj[i].push_back(u-1);
            adj[u-1].push_back(i);
        }
    }
    dfs(0,-1,0);
    pre_process();
    //for(int i = 0;i<n;i++)cout<<i<<" "<<depth[i]<<"
    "<<P[i]<<" "<<pa[i][1]<<endl;
    cin>>q;
    cout<<"Case "<<cnt<<": \n";
    for(int i = 0;i<q;i++){
        cin>>u>>v;
        cout<<lca(u-1,v-1) + 1<<"\n";
    }
    for(int i = 0;i<n;i++)
        adj[i].clear();
}
}

```