```
pair<int,int> RMQ[2*N+5][20];
int start[maxn],time=0;
int dep[maxn];
void DFS(int u,int dad)
{
    start[u]=++time;
    RMQ[time][0]= {dep[u],u};
    for(int v:g[u])
        if(v!=dad)
    {
        dep[v]=dep[u]+1;
        DFS(v,u);
        RMQ[++time][0]={dep[u],u};
    }
}
void BuildRMQ()
{
    int lim=int(log2(time))+1;
    for(int k=1;k<=lim;k++)
        for(int i=1;i+(1<<k)-1<=time;++i)
        RMQ[i][k]=min(RMQ[i][k-1],RMQ[i+(1<<(k-1))][k-1]);
}
int  LCA(int u,int v)
{
    int x=start[u],y=start[v];
    if(x>y) swap(x,y);
    int k=int(log2(y-x+1));
    return min(RMQ[x][k],RMQ[y-(1<<k)+1][k]).second;

}
```