Farthest Nodes in a Tree

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


using namespace std;
#define ll long long
ll node;
vector<ll>graph[30005],cost[30005];
ll dp[30005];
ll st;
ll mx;
int dfs(ll nn,ll cst)
{
   if(graph[nn].size() == 0)
      return dp[nn] = cst;


   if(dp[nn]!=-1) return dp[nn];
   ll ans = 0;
   for(int i=0; i<graph[nn].size(); i++)
   {
      ans+=dfs(graph[nn][i],cst +cost[nn][i]);
   }
   if(ans>=mx)
   {
      mx = ans;
      st = nn;
   }
   return dp[nn] = ans;
}
int main()
```

```
{
    ll t,frm,to,costt;
    scanf("%lld",&t);
    while(t--)
    {
        scanf("%lld",&node);
        for(int i=0; i<node-1; i++)
        {
            scanf("%lld %lld %lld",&frm,&to,&costt);
            graph[frm].push_back(to);
            graph[to].push_back(frm);
            cost[frm].push_back(costt);
            cost[to].push_back(costt);
        }
        memset(dp,-1,sizeof(dp));
        mx = 0;
        ll k = dfs(0,0);
        k = dfs(st,0);
        cout<<dp[st]<<endl;
    }

    return 0;
}
```