```cpp
/***Farthest Nodes in a Tree (II):
Given a tree (a connected graph with no cycles), you have to find
the cost to go to the farthest node from each node.
The edges of the tree are weighted and undirected.
***/
#include<bits/stdc++.h>
using namespace std;
#define mx 30005
vector<int>ed[mx], cost[mx];
bool vis[mx];
int n, maxx, node;
int res[mx], resx[mx], resy[mx];
void reset(){
    for(int i=0; i<=n; i++){
        ed[i].clear(); cost[i].clear(); vis[i]=false;
    }
}
void dfs(int u, int weight, int r[]){
    vis[u]=true;
    r[u]=weight;

    for(int i=0; i<ed[u].size(); i++){
        int v=ed[u][i];
        if(vis[v]==false){
            int w=cost[u][i];
            if(weight+w>=maxx){
                maxx=weight+w;
                node=v;
            }
            dfs(v,weight+w,r);
        }
    }
    vis[u]=false;
}

int main(){
    int t; scanf("%d",&t);
    for(int kase=1; kase<=t; kase++){
        scanf("%d",&n);
        reset();
        for(int i=1; i<n; i++){
            int u,v,w; scanf("%d%d%d",&u,&v,&w);
            ed[u].push_back(v);
            ed[v].push_back(u);
            cost[u].push_back(w);
            cost[v].push_back(w);
        }

        maxx=0;
        dfs(0,0,res);

        maxx=0;
        int x=node;
        dfs(x,0,resx);

        int y=node;
        dfs(y,0,resy);

        for(int i=0; i<n; i++){
            res[i]=max(resx[i],resy[i]);
        }

        printf("Case %d:\n",kase);
        for(int i=0; i<n; i++) printf("%d\n",res[i]);
    }
    return 0;
}
```