

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define make make_pair
5  #define MAXN 100005
6  map< int, pair<int,int> >mp;
7  vector<int>ed[MAXN],cs[MAXN];
8  int n,N,a[MAXN],L[MAXN],st[MAXN],fn[MAXN],T[MAXN],P[MAXN][20];
9  ll tree[8*MAXN];
10 void DFS(int u,int lev,int pr,int w)
11 {
12     st[u] = ++N;
13     T[u] = pr;
14     L[u] = lev;
15     a[u] = w;
16     for(int i=0; i<ed[u].size(); i++){
17         int v = ed[u][i];
18         int w = cs[u][i];
19         if(v==pr)continue;
20         DFS(v,lev+1,u,w);
21     }
22     fn[u] = ++N;
23 }
24 void LCA_init()
25 {
26     memset(P,-1,sizeof(P));
27     for(int i=1; i<=n; i++)P[i][0] = T[i];
28     for(int j=1; (1<<j)<=n; j++){
29         for(int i=1; i<=n; i++){
30             if(P[i][j-1]!=-1){
31                 P[i][j] = P[P[i][j-1]][j-1];
32             }
33         }
34     }
35 }
36 int LCA(int u,int v)
37 {
38     if(L[u]<L[v])swap(u,v);
39     int log = (int)log2(n);
40     for(int i=log; i>=0; i--){
41         if(P[u][i]!=-1 && (L[u]-(1<<i))>=L[v]){
42             u = P[u][i];
43         }
44     }
45     if(u==v)return u;
46     for(int i=log; i>=0; i--){
47         if(P[u][i]!=-1 && P[u][i]!=P[v][i]){
48             u = P[u][i];
49             v = P[v][i];
50         }
51     }
52     return T[u];
53 }
54 void update(int nd,int b,int e,int x,int v)
55 {
56     if(b==x&&e==x){ tree[nd]=v; return; }
57     int lf=2*nd, rg=2*nd+1, md=(b+e)/2;
58     if(x<=md)update(lf,b,md,x,v);
59     else update(rg,md+1,e,x,v);
60     tree[nd] = tree[lf]+tree[rg];
61 }
62 ll query(int nd,int b,int e,int x,int y)
63 {
64     if(b>y || e<x)return 0;
65     if(b>=x&&e<=y)return tree[nd];
66     int lf=2*nd, rg=2*nd+1, md=(b+e)/2;
67     ll p1 = query(lf,b,md,x,y);
68     ll p2 = query(rg,md+1,e,x,y);
69     return p1+p2;
70 }
71 }

```

```
73 int main()
74 {
75     scanf("%d",&n);
76
77     for(int i=1; i<n; i++){
78         int u,v,w; scanf("%d%d%d",&u,&v,&w);
79         ed[u].push_back(v);
80         ed[v].push_back(u);
81         cs[u].push_back(w);
82         cs[v].push_back(w);
83         if(u>v)swap(u,v);
84         mp[i] = make(u,v);
85     }
86
87     N = 0;
88     DFS(1,0,0,0);
89
90     LCA_init();
91
92     for(int i=1; i<=n; i++)
93     {
94         update(1,1,N,st[i],a[i]);
95         update(1,1,N,fn[i],-a[i]);
96     }
97
98     int Q; scanf("%d",&Q);
99     while(Q-->0)
100     {
101         int tp; scanf("%d",&tp);
102         if(tp==1)
103         {
104             int r,x; scanf("%d%d",&r,&x);
105             int u = mp[r].first;
106             int v = mp[r].second;
107             if(L[u]>L[v])swap(u,v);
108             a[v] = x;
109             update(1,1,N,st[v],x);
110             update(1,1,N,fn[v],-x);
111         }
112         else
113         {
114             int u,v; scanf("%d%d",&u,&v);
115             int lca = LCA(u,v);
116             ll ret1 = query(1,1,N,st[lca]+1,st[u]);
117             ll ret2 = query(1,1,N,st[lca]+1,st[v]);
118             ll ans = ret1 + ret2;
119             printf("%lld\n",ans);
120         }
121     }
122     return 0;
123 }
```