

```

1  #include <stdio>
2  #include <vector>
3  #include <cstring>
4
5  using namespace std;
6
7  int deep[80010];
8  int cnt;
9  int pos[40010];
10 bool visit[40010];
11 vector<int> adj[40010];
12 vector<int> cost[40010];
13 int n,m;
14 char buf[100];
15 int data[320010];
16
17 void dfs(int now,int len) {
18     deep[cnt] = len;
19     pos[now] = cnt;
20     cnt ++;
21     visit[now] = true;
22     for (int i = 0; i < adj[now].size(); i++) {
23         int t = adj[now][i];
24         int v = cost[now][i];
25         if (!visit[t]) {
26             dfs(t,len + v);
27             deep[cnt] = len;
28             pos[now] = cnt;
29             cnt ++;
30         }
31     }
32 }
33
34 void init(int now,int left,int right) {
35     if (left == right) { data[now] = deep[left]; return; }
36     int mid = (left + right) >> 1;
37     init(now + now, left, mid);
38     init(now + now + 1, mid + 1, right);
39     data[now] = data[now + now];
40     if (data[now + now + 1] < data[now])
41         data[now] = data[now + now + 1];
42 }
43
44 int query(int now,int left,int right,int l,int r) {
45     if (l <= left && right <= r) return data[now];
46     if (l > right || r < left) return 0x7fffffff;
47     int mid = (left + right) >> 1;
48     int tl = query(now + now, left, mid, l, r);
49     int tr = query(now + now + 1, mid + 1, right, l, r);
50     return tl > tr ? tr : tl;
51 }
52

```

```

53 int main() {
54     scanf("%d%d",&n,&m);
55     for (int i = 0; i < m; i++) {
56         int f,t,v;
57         scanf("%d%d%d",&f,&t,&v);
58         scanf("%s",buf);
59         adj[f].push_back(t);
60         cost[f].push_back(v);
61         adj[t].push_back(f);
62         cost[t].push_back(v);
63     }
64     dfs(1,0); init(1,0,cnt - 1);
65     int k; scanf("%d",&k);
66     for (int i = 0; i < k; i++) {
67         int f,t;
68         scanf("%d%d",&f,&t);
69         f = pos[f]; t = pos[t];
70         if (f > t) {int tmp = f; f = t; t = tmp;}
71         int LCA = query(1, 0, cnt - 1, f, t);
72         printf("%d\n",deep[f] + deep[t] - LCA * 2);
73     }
74     return 0;
75 }

```