

Can Go?

Problem

Submissions

Leaderboard

Discussions

Problem Statement

You will be given N numbers of nodes, E numbers of edges in a graph. For each edge you will be given A , B and W which means there is a connection from A to B only and for which you need to give W cost. The value of nodes could be from 1 to N .

You will be given a source node S . Then you will be given a test case T , for each test case you will be given a destination node D and a cost DW . You need to tell if you can go to the destination from source using atmost DW cost.

Input Format

- First line will contain N and E .
- Next E lines will contain A , B and W .
- Next line will contain source node S .
- Next line will contain T , the number of test cases.
- For each test case, you will get D and DW .

Constraints

1. $1 \leq N \leq 10^3$
2. $1 \leq E \leq 10^6$
3. $1 \leq S \leq N$
4. $1 \leq T \leq 10^3$
5. $1 \leq D \leq N$
6. $0 \leq W, DW \leq 10^9$

Output Format

- Output "YES" or "NO" for each test case if it is possible to go from S to D using atmost DW cost.

Sample Input 0

```
5 7
1 2 10
1 3 2
3 2 1
2 4 7
3 4 2
4 5 5
2 5 2
1
5
```

```
1 0
2 5
3 1
4 4
5 6
```

Sample Output 0

```
YES
YES
NO
YES
YES
```



Submissions: [325](#)

Max Score: 20

Difficulty: Easy

Rate This Challenge:



[More](#)

C++20



```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define lli long long int
4
5 const lli M = 1e18;
6 vector<pair<int,lli>> v[1005];
7 lli dis[1005];
8
9 class cmp{
10 public:
11     bool operator()(pair<int,lli> a, pair<int,lli> b){
12         return a.first > a.second;
13     }
14 };
15
16 void dijkstra(int src){
17     priority_queue<pair<int,lli>, vector<pair<int,lli>>, cmp> pq;
18     pq.push({src,0});
19     dis[src] = 0;
20     while(!pq.empty()){
21         pair<int,lli> parent = pq.top();
22         pq.pop();
23         int node = parent.first;
24         lli cost = parent.second;
25
26         for(pair<int,lli> child: v[node]){
27             int childNode = child.first;
28             lli childCost = child.second;
29             if(cost+childCost < dis[childNode]){
30                 dis[childNode] = cost+childCost;
31                 pq.push({childNode, dis[childNode]});
32             }
33         }
34     }
35 }
36
37 }
38
```

```

39 int main(){
40
41     int n,e;
42     cin >> n >> e;
43     while(e--){
44         int a,b;
45         lli c;
46         cin >> a >> b >> c;
47         v[a].push_back({b,c});
48     }
49
50     for(int i = 1; i <= n; i++){
51         dis[i] = M;
52     }
53
54     int src;
55     cin >> src;
56     dijkstra(src);
57
58     int test;
59     cin >> test;
60     while(test--){
61         int d;
62         lli dw;
63         cin >> d >> dw;
64         if(dis[d] <= dw){
65             cout << "YES" << endl;
66         }else{
67             cout << "NO" << endl;
68         }
69     }
70
71     return 0;
72 }

```

Line: 1 Col: 1

 [Upload Code as File](#) ☐ [Test against custom input](#)

Run Code

Submit Code