

All Contests > Assignment 03 | Introduction to Algorithms | Batch 03 > Can Go Again?

# Can Go Again?

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Submitted an hour ago • Score: 25.00

Status: Accepted

✓	Test Case #0	✓	Test Case #1	✓	Test Case #2
✓	Test Case #3	✓	Test Case #4	✓	Test Case #5
✓	Test Case #6	✓	Test Case #7	✓	Test Case #8
✓	Test Case #9	✓	Test Case #10	✓	Test Case #11
✓	Test Case #12	✓	Test Case #13	✓	Test Case #14
✓	Test Case #15	✓	Test Case #16	✓	Test Case #17
✓	Test Case #18	✓	Test Case #19	✓	Test Case #20
✓	Test Case #21	✓	Test Case #22	✓	Test Case #23
✓	Test Case #24	✓	Test Case #25	✓	Test Case #26
✓	Test Case #27	✓	Test Case #28	✓	Test Case #29
✓	Test Case #30	✓	Test Case #31	✓	Test Case #32
✓	Test Case #33	✓	Test Case #34	✓	Test Case #35
✓	Test Case #36	✓	Test Case #37	✓	Test Case #38
✓	Test Case #39	✓	Test Case #40	✓	Test Case #41
✓	Test Case #42	✓	Test Case #43	✓	Test Case #44
✓	Test Case #45	✓	Test Case #46	✓	Test Case #47
✓	Test Case #48	✓	Test Case #49	✓	Test Case #50
✓	Test Case #51	✓	Test Case #52	✓	Test Case #53
✓	Test Case #54	✓	Test Case #55	✓	Test Case #56
✓	Test Case #57				

## Submitted Code

Language: C++20

Open in editor

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define pii pair<pair<int, int>, int>
4
5 const int N = 1e3 + 13;
6 int n, m;
```

```
7 const long long INF = 1e18 + 8;
8 vector<pii> edgeList;
9 long long dist[N];
10
11 void initializeDistance()
12 {
13     for (int i = 1; i <= n; i++)
14     {
15         dist[i] = INF;
16     }
17 }
18
19 void BellmanFord(int src)
20 {
21     initializeDistance();
22
23     dist[src] = 0;
24
25     for (int i = 1; i < n; i++)
26     {
27         for (auto el : edgeList)
28         {
29             long long u = el.first.first;
30             long long v = el.first.second;
31             long long w = el.second;
32
33             if (dist[u] != INF and dist[u] + w < dist[v])
34             {
35                 dist[v] = dist[u] + w;
36             }
37         }
38     }
39 }
40
41 int main()
42 {
43
44     cin >> n >> m;
45     for (int i = 1; i <= m; i++)
46     {
47         long long u, v, w;
48         cin >> u >> v >> w;
49         edgeList.push_back({{u, v}, w});
50     }
51
52     // for(auto el : edgeList)
53     // {
54     //     cout<<el.first.first<<" "<<el.first.second<<": "<<el.second<<endl;
55     // }
56
57     int src;
58     cin >> src;
59     BellmanFord(src);
60
61     bool cycle = false;
62     for (auto el : edgeList)
63     {
64         long long u = el.first.first;
65         long long v = el.first.second;
66         long long w = el.second;
67
68         if (dist[u] != INF and dist[u] + w < dist[v])
69         {
70             cycle = true;
71             break;
72             dist[v] = dist[u] + w;
```

```
73     }
74 }
75
76 int q;
77 cin >> q;
78 while (q--)
79 {
80     int dest;
81     cin >> dest;
82     if (cycle == false)
83     {
84         if (dist[dest] == INF)
85         {
86             cout << "Not Possible" << endl;
87         }
88         else
89         {
90             cout << dist[dest] << endl;
91         }
92     }
93 }
94
95 // for (int i = 1; i <= n; i++)
96 // {
97 //     // cout << "D of " << i << ": " << dist[i] << endl;
98 // }
99 // cout<<"cycle = "<<cycle<<endl;
100 if (cycle)
101     cout << "Negative Cycle Detected" << endl;
102
103 return 0;
104 }
```