

X

NPTEL

souravsharma2468@gmail.com ▼

Courses » Design and Analysis of Algorithms[Announcements](#)**[Course](#)**[Ask a Question](#)[Progress](#)[FAQ](#)**Register for
Certification exam****Course
outline****How to access
the portal****Week 1:
Introduction****Week 1:
Analysis of
algorithms****Week 1 Quiz****Week 2:
Searching and
sorting****Week 2 Quiz****Week 2
Programming
Assignment****Week 3:
Graphs****Week 3 Quiz****Week 3
Programming
Assignment****Week 4:
Weighted
graphs****Week 4 Quiz****Week 4
Programming
Assignment**

Week 4 Programming Assignment

Due on 2019-03-05, 23:59 IST

- Select your language (C/C++/Java/Python2/Python3)
- Paste your code into the submission window.
- There are some public test cases and some (hidden) private test cases.
- "Compile and run" will evaluate your submission against the public test cases.
- "Submit" will evaluate your submission against the hidden private test cases and report a score on 100. There are 20 private testcases in all, each with equal weightage. You will only get a score on 100. You will not get feedback on which private testcases passed or failed.
- Ignore warnings about "Presentation errors".

Padayatra

(IOI Training Camp, Bangalore, 2004)

The exit polls indicate that the sitting MLA is likely to lose the upcoming election. The party high command instructs him to undertake a padayatra through his constituency to boost his popularity. To maximize coverage of the constituency, the party decides that he should choose a circular route that returns to the starting point without using any road twice. The route need not visit all the towns and villages in the constituency.

Being averse to physical exercise, the MLA would like to minimize the distance that he has to walk. He has a helicopter at his disposal, so he can begin the padayatra at any town or village.

The task is to help him find the shortest circular route. You are guaranteed that there is always at least one circular route.

Solution Hint

Given an edge (i,j) with weight $W(i,j)$, the shortest cycle from i to i via j can be found by temporarily deleting the edge (i,j) , finding the shortest path from j to i , and then adding $W(i,j)$ to the length of this path. Do this systematically to find the shortest cycle in the graph.

Input format

The first line of input is an integer N , $1 \leq N \leq 1000$, the number of roads in the constituency. The constituency has no more than 300 towns and villages connected by roads.

The next N lines specify the roads. Each line consists of three space

<div><div></div><div>Week 4 Programming Assignment</div></div>
<div>Week 5: Data Structures: Union-Find and Heaps</div>
<div>Week 5: Divide and Conquer</div>
<div>Week 5 Quiz</div>
<div>Week 6: Data Structures: Search Trees</div>
<div>Week 6: Greedy Algorithms</div>
<div>Week 6 Quiz</div>
<div>Week 6 Programming Assignment</div>
<div>Week 7: Dynamic Programming</div>
<div>Week 7 Quiz</div>
<div>Week 7 Programming Assignment</div>
<div>Week 8: Linear Programming and Network Flows</div>
<div>Week 8: Intractability</div>
<div>Week 8 Quiz</div>
<div>Download</div>
<div>TEXT TRANSLATION</div>

The next N lines specify the roads. Each line consists of three space separated integers S , T and D where S is the starting point of the road, T is the ending point and D is the length of the road. Each road is a two-way road and is listed exactly once, in one of the two directions.

Output format

A single line with the integer distance around the shortest possible circular route.

[Facebook](#)

[Twitter](#)

[YouTube](#)

[LinkedIn](#)

[Google Plus](#)

Test data

There are at most 300 towns and villages and at most 1000 roads.

Example

Sample Input

```
4
3 1 5
1 2 4
3 2 3
1 5 44
```

Sample output 1

12

Sample Test Cases		
	Input	Output
Test Case 1	4	12
	3 1 5	
	1 2 4	
	3 2 3	
	1 5 44	
Test Case 2	7	77
	3 1 11	
	6 1 12	
	6 4 52	
	6 2 17	
	5 2 14	
	4 5 13	
	3 4 10	
	10	
	6 1 13	
Test Case 3	6 3 17	116
	5 3 18	
	8 5 11	
	8 7 19	
	6 8 85	
	8 4 92	
	7 2 15	
	4 2 13	
	4 1 10	
	139	
Test Case 4	1 2 93027	20297

243 6 3134
 245 6 31434
 246 6 25285
 6 250 86504
 6 251 69331
 252 6 87408
 6 253 84534
 254 6 96547
 6 257 93512

Due Date Exceeded.

20 out of 20 tests passed.

You scored 100.0/100.

Your last recorded submission was :

```

1  #include<iostream>
2  #include<cmath>
3  #include<vector>
4  #include<algorithm>
5  #include<list>
6  #include<iterator>
7  #include<set>
8  #include<limits.h>
9  #include<cstdlib>
10 #define INF 99999999
11
12 using namespace std;
13
14 struct Edge
15 {
16     int u;
17     int v;
18     int w;
19 };
20
21 class Graph
22 {
23     int V ;
24     list< pair<int,int> >*adj;
25     vector<Edge> edge;
26
27 public :
28     Graph( int j ):V(j)
29     {
30         adj=new list< pair<int,int> >[V];
31     }
32
33     void add_edge(int u,int v,int w){
34         adj[u].push_back(make_pair(v,w));
35         adj[v].push_back(make_pair(u,w));
36         Edge e ;
37         e.u=u,e.v=v,e.w=w;
38         edge.push_back(e);
39     }
40
41     void remove_edge(int u,int v,int w)
42     {
43         adj[u].remove(make_pair(v,w));
44         adj[v].remove(make_pair(u,w));
45     }
46
47     int dijkstra(int u,int v);
48     int fmc();
49
50 int Graph::dijkstra(int u,int v)
51 {
52     set< pair<int, int> > st;
53     vector<int> dist(V, INF);
54     st.insert(make_pair(0, u));
55     dist[u] = 0;
56
57     while (!st.empty())
58     {
59
60         pair<int, int> tmp = *(st.begin());
61         st.erase(st.begin());
62         int u = tmp.second;
63
64         list< pair<int, int> >::iterator i;
```



```

65     for(i=adj[u].begin();i!=adj[u].end();++i)
66     {
67         int v=(*i).first;
68         int weight=(*i).second;
69
70         if (dist[v]> dist[u]+weight)
71         {
72             if (dist[v] != INF)
73                 st.erase(st.find(make_pair(dist[v], v)));
74
75             dist[v] = dist[u] + weight;
76             st.insert(make_pair(dist[v], v));
77         }
78     }
79 }
80 return dist[v] ;
81 }
82 int Graph :: fmc ( )
83 {
84     int min_cycle = INT_MAX;
85     int E = edge.size();
86     for(int i=0;i<E;i++)
87     {
88         Edge e=edge[i];
89         remove_edge(e.u,e.v,e.w) ;
90         int dist=dijkstra(e.u,e.v);
91         min_cycle=min(min_cycle, dist + e.w);
92         add_edge(e.u,e.v,e.w);
93     }
94     return min_cycle ;
95 }
96
97
98
99 int main(){
100     ios_base::sync_with_stdio(false);
101     cin.tie(0);
102     cout.tie(0);
103     int n;
104     cin>>n;
105     Graph g(300);
106     for(int i=0;i<n;++i){
107         int s,t,w;
108         cin>>s>>t>>w;
109         g.add_edge(s-1,t-1,w);
110     }
111
112     int k=g.fmc();
113
114     cout<<k<<endl;
115     return 0;
116 }
117
118

```



End