Cheapest Routes X81287_en

We have collected abundant information about the local roads and accommodations in a region that we will traverse. Our plan is to go from city A to city B and we would like to spend the least possible money. For each road connecting two cities u and v we know the cost $\omega(u,v) = \omega(v,u)$ to travel along that road (tolls, fuel, meals during the journey, ...). Every time we go from a city u to one of its neighbors v we must stop at v and spend there one night; we know the cost $\omega'(v)$ of stopping at each city v (the cost added by A and B to our route is 0, since they are our initial and final points). All costs, of vertices and of edges, are non-negative. Thus the cost of the route

$$P = [A, v_1, \ldots, v_n, B]$$

is

$$\text{cost}(P) = \omega(A, v_1) + \omega(v_1, v_2) + \ldots + \omega(v_n, B) + \omega'(v_1) + \ldots + \omega'(v_n).$$

Write a program in C++ which, given an undirected weighted graph with non-negative costs at the vertices and at the edges, and two vertices A and B, returns the cost of the cheapest route to go from A to B, or an indication that not such route exists.

Input

All data in the input are non-negative integers. The input starts with two integers $2 \le n \le 10000$ and m, $0 \le m \le 20n$. After that, a sequence of non-negative integers $\omega'(0), \ldots, \omega'(n-1)$ of the weights $\omega'(u)$ of the n vertices of the graph. Then the input contains a sequence of the m edges in the graph as triplets of the form $\langle u, v, \omega(u,v)\rangle$. Vertices u and v are integers in $\{0, \ldots, n-1\}$ and the weights $\omega(u,v)$ are non-negative integers. You can assume that there are no two different edges connecting the same pair of vertices nor any edge connecting a vertex to itself. Finally, there is a sequence of pairs $\langle A_i, B_i\rangle$, with each A_i and B_i denoting vertices of the graph ($0 \le A_i, B_i < n$).

Output

For each pair $\langle A_i, B_i\rangle$ in the input sequence the program writes the cost δ of the cheapest route between A_i and B_i. with the format `c(`A_i`, `B_i`) = `δ. If no route exists between A_i and B_i the program writes `c(`A_i`, `B_i`) = +oo`. The ouput for each case is ended with a newline (`endl`).

Sample input 1

```
6 8
3 6 10 15 5 2
0 1 2  1 2 7  2 3 2
0 2 1  1 3 4  2 4 8
3 4 2  3 0 5
0 4
1 4
2 4
3 1
4 1
2 5
2 2
```

Sample output 1

```
c(0,4) = 19
c(1,4) = 21
c(2,4) = 8
c(3,1) = 4
c(4,1) = 21
c(2,5) = +oo
c(2,2) = 0
```

Problem information

Author: Conrado Martinez

Generation: 2026-01-25T22:36:23.768Z