EJOY

EJOY 2067

Task 2. Bribe

⏳0.67 sec. 💾256 MB

Bobi has a graph, but he does not know what it is. He can ask the jury whether in the graph there is a connection between vertices u and v ($1 \leq u, v \leq N$) with a direct edge between them, and if so, what its weight is. Bobi's goal is to find a minimum spanning tree of his graph. The problem is that the jury is bored and decides to mess with you. The jury may lie to you about whether there is an edge between two vertices. At the beginning the chance that it lies to you is 67%. Bobi is rich and has infinite money with which he can bribe the jury not to lie to you. The price of one bribe is 1 lev, and one bribe reduces the chance that it lies to you by 1%. You have to help Bobi give as little money as possible (even though he is rich) and at the same time find the most minimal spanning tree that he can.

Task

Write a program **`bribe`**, containing a function `solve`, which will be compiled with the program of the jury and will communicate with it, by asking questions about edges and bribing the jury. Your program must find the minimum spanning tree with the information that it has.

Implementation details

The function `void solve(int n)`, which you have to write, will be called only once by the program of the jury and will receive as arguments the number of vertices in the graph N and Bobi's money M. For communication with the program of the jury you are provided with the following functions:

```
int query(int u, int v);
void bribe();
void submit(int l);
```

On every call of the function `query`, it will return the weight of the edge if the jury claims that it exists, and otherwise it returns -1. The function `bribe` performs the described bribe. Once you find the minimum spanning tree, your function must call `submit` and pass as an argument the sum of the weights in the found minimum spanning tree. After that the execution of your function will be terminated.

Your program `bribe.cpp` must implement the function `solve`. It may also contain other code and functions necessary for your work, but it must not contain the main function `main`. Also, you must not read from the standard input or print to the standard output. Your program must include the header file `bribe.h` through a directive to the preprocessor:

```
#include "bribe.h"
```

Constraints and scoring

- $1 \leq N \leq 1000$
- $1 \leq$ weight of an edge ≤ 10000
- The points for a given test are computed by some stupid calculations, which took too much time to come up with and will not be explained here.

EJOY 2067

Sample communication

Function of the participant	Program of the jury
	`solve(5)`
`query(1, 2)`	1
`query(2, 3)`	-1
`bribe()`	
`query(3, 4)`	1
`query(4, 5)`	1
`submit(4)`	

Explanation: In the example the graph is a path graph, and on the second query the jury lies to us.

Local testing

You are provided with the file `Lgrader.cpp`, which you can compile together with your program in order to test it. On startup the program will read from the standard input the value of N and then the number of edges M. On the next M lines it reads 3 numbers each – the two vertices that it connects, and the weight. After that the communication which takes place will be printed. You may modify the provided file as you wish.