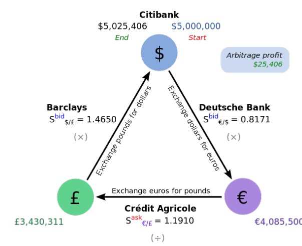
2 Detecting Anomalies in Currency Exchange Rates

Problem Introduction

You are given a list of currencies $c_1, c_2, \ldots, c_n$ together with a list of exchange rates: r_{ij} is the number of units of currency c_j that one gets for one unit of c_i. You would like to check whether it is possible to start with one unit of some currency, perform a sequence of exchanges, and get more than one unit of the same currency. In other words, you would like to find currencies $c_{i_1}, c_{i_2}, \ldots, c_{i_k}$ such that $r_{i_1,i_2} \cdot r_{i_2,i_3} \cdot r_{i_{k-1},i_k}, r_{i_k,i_1} > 1$. For this, you construct the following graph: vertices are currencies $c_1, c_2, \ldots, c_n$, the weight of an edge from c_i to c_j is equal to $-\log r_{ij}$. There it suffices to check whether is a negative cycle in this graph. Indeed, assume that a cycle $c_i \to c_j \to c_k \to c_i$ has negative weight. This means that $-(\log c_{ij} + \log c_{jk} + \log c_{ki}) < 0$ and hence $\log c_{ij} + \log c_{jk} + \log c_{ki} > 0$. This, in turn, means that



$$r_{ij}r_{jk}r_{ki} = 2^{\log c_{ij}} 2^{\log c_{jk}} 2^{\log c_{ki}} = 2^{\log c_{ij} + \log c_{jk} + \log c_{ki}} > 1\,.$$

Problem Description

Task. Given an directed graph with possibly negative edge weights and with n vertices and m edges, check whether it contains a cycle of negative weight.

Input Format. A graph is given in the standard format.

Constraints. $1 \le n \le 10^3$, $0 \le m \le 10^4$, edge weights are integers of absolute value at most 10^3.

Output Format. Output 1 if the graph contains a cycle of negative weight and 0 otherwise.

Time Limits.

language	C	C++	Java	Python	C#	Haskell	JavaScript	Ruby	Scala
time (sec)	2	2	3	10	3	4	10	10	6

Memory Limit. 512MB.

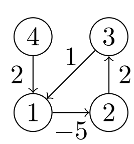
Sample 1.

Input:

```
4 4
1 2 -5
4 1 2
2 3 2
3 1 1
```

Output:

```
1
```



The weight of the cycle $1 \to 2 \to 3$ is equal to -2, that is, negative.