



Zadanie: SAB

Mały sabotaż w wielkim mieście [A]

Potyczki Algorytmiczne 2015, runda 4. Dostępna pamięć: 128 MB.

1.10.2015

Uwaga! To jest zadanie *rozproszone*. Zanim zaczniesz je rozwiązywać, zapoznaj się z informacjami dotyczącymi tego typu zadań dostępnymi w serwisie internetowym zawodów.

Wielkie miasto Megabajtopolis jest nietypowe – zostało zbudowane na wielkim zbiorniku wodnym i składa się z wysepek połączonych mostami. Niektóre z mostów przebiegają nad innymi mostami, inne łączą wyspy po przeciwnych krańcach miasta. Z lotu ptaka wygląda to jak wielka, chaotyczna sieć.

W mieście znalazła się grupka sabotażystów, którzy za cel obrali sobie zmianę władzy. Obalenie administracji prezydenta Megabajtazara będzie wymagało zatrzęsienia opinią publiczną – najlepiej byłoby, gdyby udało się pokazać jego nieporadność w sytuacji kryzysowej. Nasi (anty)bohaterowie postanowili wysadzić jeden z mostów w mieście (na wysadzenie większej liczby nie mają środków). Taki sabotaż można by uznać za udany tylko wtedy, gdyby nie było żadnej innej drogi (być może biegnącej przez wiele mostów), która łączyłaby wysępki uprzednio połączone wysadzonym mostem. Jak wiele mostów powinni brać pod uwagę sabotażyści?

Dane wejściowe

Twój program nie może używać standardowego wejścia. Zamiast tego dane testowe zostaną mu udostępnione za pośrednictwem dostarczonej biblioteki interaktywnej. Aby jej użyć, wpisz w swoim programie (C/C++):

```
#include "sabotaz.h"
```

Biblioteka udostępnia cztery funkcje:

- `NumberOfIsles()` – zwraca N ($1 \leq N \leq 200\,000$) – liczbę wysepek w Megabajtopolis. Wysepki są ponumerowane liczbami od 0 do $N - 1$.
- `NumberOfBridges()` – zwraca M ($1 \leq M \leq 10^8$) – liczbę mostów w Megabajtopolis. Mosty są ponumerowane liczbami od 0 do $M - 1$.
- `BridgeEndA(i)` – zwraca numer pierwszej z wysepek połączonych mostem i .
- `BridgeEndB(i)` – zwraca numer drugiej z wysepek połączonych mostem i .

Dwie wysępki może łączyć więcej niż jeden most (w tym wypadku sabotażyści mogą wysadzić tylko jeden z nich). Może się zdarzyć, że oba końce mostu leżą na tej samej wysepce – w takiej sytuacji most prowadzi przez jezioro położone na tej wysepce. Wysadzenie takiego mostu nie pomoże sabotażystom w osiągnięciu celu.

Udostępnionym funkcjom odpowiadają następujące deklaracje w językach C lub C++:

```
long long NumberOfIsles();  
long long NumberOfBridges();  
long long BridgeEndA(long long);  
long long BridgeEndB(long long);
```

W dziale *Pliki* w systemie SIO2 znajduje się archiwum zawierające przykładowe pliki bibliotek oraz (niepoprawne) rozwiązania ilustrujące sposób ich użycia.

Wyjście

Twój program powinien wypisać na wyjście liczbę mostów, których wysadzenie spowoduje, że sabotaż będzie można uznać za udany.

Komunikacja

Podczas oceny Twojego programu system sprawdzający uruchomi jednocześnie wiele jego instancji, każdą na osobnym komputerze. Instancje powinny komunikować się za pomocą biblioteki `message`. W tym celu w programie w języku C bądź C++ należy umieścić wiersz:

```
#include "message.h"
```

Instrukcja dotycząca używania tej biblioteki jest dostępna w serwisie internetowym zawodów w zakładce *Zadania rozproszone*.

Ograniczenia liczby i rozmiaru wysyłanych wiadomości

- Liczba wiadomości wysyłanych przez pojedynczą instancję nie może przekroczyć 1000.
- Sumaryczny rozmiar wiadomości wysyłanych przez jedną instancję nie może przekroczyć 8MB.

Przykładowy przebieg programu

Dla przebiegu programu:

Wywołanie funkcji	Zwrócona wartość
NumberOfIsles();	4
NumberOfBridges();	4
BridgeEndA(0);	0
BridgeEndB(0);	1
BridgeEndA(1);	1
BridgeEndB(1);	2
BridgeEndA(2);	2
BridgeEndB(2);	0
BridgeEndA(3);	2
BridgeEndB(3);	3

poprawnym wynikiem jest:

1

Testy przykładowe

Po wysłaniu rozwiązania od razu poznasz wyniki Twojego programu na poniższych testach, uruchamianych na 100 komputerach:

- 0a: test przykładowy z treści zadania;
- 0b: $N = 200\,000$, $M = 10^8$. Most i łączy wyspy $(i \bmod N)$ oraz $((i + 1) \bmod N)$.