

Harcerze w Plenerze

PREOI 2025

Dzień 1 - 25.01.2025

Kod zadania: **har**
Limit pamięci: **512 MiB**



W Bitockim Parku Narodowym, jest N szczytów, połączonych za pomocą M (oczywiście obustronnych) szlaków. Przejście każdego z nich zajmuje dokładnie minutę (przynajmniej według BTTK). Jest to niezwykle miejsce, która przyciąga co lata tysiące turystów - w tym Adasia i Alę. Adaś i Ala to harcerzy para, nic więc dziwnego że podczas wycieczki w góry postanowili zorganizować wyczyn.

Wyczyn jest następujący:

Najpierw Adaś maluje szlaki na A kolorów i idzie czekać na szczyt nr 0. Następnie Ala, startując z pewnego wierzchołka, musi do niego dojść. Jednak nie zna ona terenu i rozróżnia szlaki tylko po kolorze. Ala chodzi po parku w sposób następujący: za każdym wejściem na szczyt różny od 0, może wrócić szlakiem którym przyszła lub wybrać jeden z pozostałych (a konkretnie jego kolor). Jeżeli Ali nie uda się dojść do Adasia w ciągu $d + B$ minut, to wyczyn kończy się porażką (gdzie d to najszybszy możliwy czas dojścia z początkowego szczytu Ali do 0 (znając szlaki)). Pomóż im to zrobić.

Implementacja

Opisane poniżej funkcje, powinny zostać przez Ciebie zaimplementowane w jednym pliku.
Twój program będzie uruchamiany dwukrotnie i za każdym razem będzie wywoływał inne funkcje:

Pierwsze Uruchomienie:

- `std::vector<int> Pomaluj(int N, int M, int A, int B, std::vector<int> U, std::vector<int> V)`
Ta funkcja jest wywoływana dokładnie raz, na początku.
 - N to liczba szczytów N
 - M to liczba dróg M .
 - A to liczba kolorów szlaków A .
 - B to margines błędu Ali na dodatkowy czas dojścia
 - U oraz V to tablice długości M , gdzie $U[i]$ oraz $V[i]$ to miasta U_i oraz V_i połączone drogą i ($0 \leq i \leq M - 1$).
 - Funkcja powinna zwrócić tablicę x . Pole $x[i]$ oznacza kolor szlaku i ($0 \leq i \leq M - 1$). Powinna być to tablica długości M z wartościami z zakresu $\{0, 1, \dots, A - 1\}$ - w przeciwnym wypadku wykonanie programu skończy się błędem.

Drugie Uruchomienie:

- `void Init(int A, int B)`
Ta funkcja jest wywoływana dokładnie raz, na początku.
 - A to liczba kolorów szlaków A .
 - B to margines błędu Ali na dodatkowy czas dojścia
- `int Rusz(std::vector<int> y)`
Funkcja jest wywoływana gdy tylko Ala wejdzie na szczyt inny niż 0.
 - y to tablica długości A , w której pole $y[i]$ oznacza liczbę szlaków koloru i wychodzących ze szczytu w którym obecnie jest (poza tą którą właśnie przyszła (jeżeli takowa istnieje)).
 - Funkcja powinna zwrócić jedną liczbę z spełniającą $-1 \leq z \leq A - 1$. Jeżeli $z = -1$, to Ala zawraca, a w przeciwnym wypadku wybiera jeden ze szlaków o kolorze i . Jeżeli dla zwróconego z nie istnieje taka ścieżka to wykonanie programu kończy się błędem.

Program **musi** zaczynać się od linijki `#include "harlib.h"`



Ważne uwagi

- Jeżeli Ala w ciągu $d + B$ ruchów nie dojdzie do Adasia wykonanie programu kończy się błędem
- Jeżeli istnieje wiele szlaków o kolorze wyznaczonym przez Alę, to pewien zostanie wybrany (**niekoniecznie losowo**)

Program może używać zmiennych globalnych, ale ponieważ jest on uruchamiany dwa razy, uruchomienia te nie będą współdzielić zmiennych globalnych.

Dla uczestników korzystających z systemu operacyjnego Linux (lub WSL itp.) - do czego bardzo gorąco zachęcamy - w zakładce pliki został umieszczony plik `dlazaw.zip` (w kategorii (har)) znajduje się tam parę plików ale należy modyfikować jedynie plik `har.cpp` oraz korzystać z pliku `compile.sh` (poniżej wyjaśnienie).

Dla uczestników nie mających dostępu do Linux'a udostępniamy środowisko w przeglądarce. Aby z niego skorzystać, należy wejść w link, założyć konto (jeśli takowego się jeszcze nie ma) i w prawym górnym rogu kliknąć `Use this template -> Open in codespaces`.

Do dyspozycji jest skrypt `compile.sh`, aby z niego skorzystać należy wpisać komendę w terminal: `./compile.sh 0` albo `./compile.sh 1` - odpowiednio bez/z pomocniczymi informacjami. Skrypt ten stworzy plik `har.e`, który następnie należy opalić (komendą `./har.e`) i wpisać test.

Testy mają następującą strukturę:

N M A B S (gdzie S to numer początkowego szczytu Ali)

U_0 V_0

....

U_{M-1} V_{M-1}

Wyjście programu będzie dwojake, albo będzie to `ANSWER 1`, albo zacznie się od `ERROR` i odpowiedniej cyfry:

ERROR 1 - długość zwróconej listy kolorów nie wynosi m

ERROR 2 - kolor nie należy do przedziału $[0, a-1]$

ERROR 3 - przekroczono dozwoloną liczbę kroków

ERROR 4 - próbowano wrócić krawędzią, będąc w wierzchołku startowym

ERROR 5 - Rusz zwróciło liczbę poza przedziałem $[-1, a-1]$

ERROR 6 - Rusz zwróciło kolor bez żadnej krawędzi

Przykładowy test i wywołania

Wejście dla testu `har0a`:

```
7 6 2 6 1
0 2
0 4
1 2
1 3
1 5
4 6
```

Wyjście dla testu `har0a`:

Adaś		Ala	
Wywołanie	Wynik	Wywołanie	Wynik
Pomaluj(7,6,2,6,[0,0,1,1,1,4],[2,4,2,3,5,6])	[1,0,0,1,0,1]		
		Init(2,6)	
		Rusz([2,1])	0
		Rusz([0,0])	-1
		Rusz([1,1])	0
		Rusz([0,1])	1

Ograniczenia

- $2 \leq N \leq 20\,000$
- $1 \leq M \leq 20\,000$
- $1 \leq S \leq N - 1$ (S to szczyt od którego Ala zaczyna)
- $0 \leq U_i < V_i \leq N - 1$, dla $0 \leq i < M$
- $(U_i, V_i) \neq (U_j, V_j)$, dla $0 \leq i < j < M$
- Z każdego szczytu można przejść do każdego innego.

Ocenianie

Podzadanie	Ograniczenia	Limit czasu	Liczba punktów
1	$A = 4, B = 0, M = N - 1$	1 s	11
2	$A = 4, B = 0$	1 s	12
3	$A = 3, B = 0, M = N - 1$	1 s	13
4	$A = 3, B = 0$	1 s	14
5	$A = 2, B = 2N, M = N - 1, 6 \leq N \leq 500$	1 s	9
6	$A = 2, B = 12, M = N - 1$	1 s	25
7	$A = 2, B = 6, M = N - 1$	1 s	16