

Zadanie: WAG

Waga medianowa



Trening przed BOI, dzień pierwszy. Plik źródłowy `wag.*` Dostępna pamięć: 64 MB. 15.07.2020

Danych jest $N = 2K - 1$ monet, z których każda ma inną wagę. Twoim zadaniem jest wyznaczyć K -tą w kolejności (środkową) monetę. Do dyspozycji masz jedynie *wagę medianową*: urządzenie, któremu podajesz trzy monety, a ono wskazuje, która jest spośród nich środkową.

Wykonując jak najmniej ważeń, znajdź K -tą monetę.

Działanie programu

Twój program nie powinien czytać żadnych danych, ani wypisywać żadnych danych. Zamiast tego zostanie skompilowany z odpowiednią biblioteką oceniającą, która będzie symulować odpowiedzi wagi medianowej. Aby użyć biblioteki, należy wpisać na początku programu:

```
#include "grader.h"
```

Biblioteka udostępnia następujące funkcje:

- `int init()` – zwraca liczbę N . Powinna zostać użyta dokładnie raz, na samym początku działania programu.
- `int median(int x, int y, int z)` – pyta o medianę monet numer x , y i z . Zwraca jedną z liczb x , y , z – numer monety, która ma środkową wagę spośród trzech podanych. **Uwaga:** Monety numerowane są od 0 do $N - 1$.
- `void answer(int a)` – podanie odpowiedzi. Wywołując tę funkcję kończysz program informując jednocześnie bibliotekę, że Twoim zdaniem środkowa jest moneta o numerze a .

Jakakolwiek komunikacja powinna odbywać się tylko za pośrednictwem powyżej podanych funkcji.

Rozwiązanie będzie kompilowane wraz z biblioteką przy użyciu polecenia:

```
g++ -O2 -static grader.cpp wag.cpp -lm
```

Ocenianie i limity

Dla każdego testu ustalony jest limit pytań (wywołań funkcji `median()`), które wolno Ci zadać. Otrzymasz punkty za dany test, jeśli podasz prawidłową odpowiedź oraz zmieścisz się w tym limicie.

W testach wartych przynajmniej 50% punktów limit będzie ustawiony na co najmniej $(\frac{N-1}{2})^2$ wywołań.

W testach wartych przynajmniej 80% punktów limit będzie ustawiony na co najmniej $50 \cdot N$ wywołań.

We wszystkich testach limit będzie ustawiony na co najmniej $8 \cdot N$ wywołań.

Liczba monet nigdy nie przekracza 200 000.

Biblioteka ustala wagi monet przed wykonaniem Twojego programu i nie zmienia ich w trakcie działania programu.

Przykładowa biblioteka

Do dyspozycji masz przykładową bibliotekę składającą się z plików `grader.h` i `grader.cpp`. Po skompilowaniu swojego programu wraz z biblioteką podanym wyżej poleceniem otrzymasz wykonywalny program, który:

- Wczytuje dane ze standardowego wejścia: w pierwszym wierszu liczbę monet N , w drugim ciężary kolejnych monet. Aby program działał poprawnie, ciężary muszą być liczbami całkowitymi z przedziału $[0, N - 1]$ oraz żadne dwie monety nie mogą mieć tego samego ciężaru.
- Rozgrywa komunikację Twojego programu z biblioteką.
- Wypisuje na standardowe wyjście DOBRZE i liczbę wykonanych zapytań, jeśli Twój program podał poprawną odpowiedź – moneta o numerze podanym w funkcji `answer()` ma ciężar $\frac{N-1}{2}$. W przeciwnym wypadku wypisuje ZLE.

Przykład interakcji

W teście 0 (przykładowym) są 3 monety o wagach kolejno 2, 0, 1. Wywołanie `median(0, 1, 2)` zwraca indeks 2 – środkowa jest moneta o numerze 2 i wadze 1. Możemy zatem już po jednym zapytaniu udzielić prawidłowej odpowiedzi `answer(2)`.