

# Zadanie: GDZ

## Gdzie jest jedyńka?



XX OI, etap III, dzień drugi. Plik źródłowy `gdz.*`. Dostępna pamięć: 128 MB.

14.03.2013

Bajtazar wymyślił sobie pewną permutację\*  $P$  liczb od 1 do  $n$ . Bajtazar chce, abyś zgadł(a), na którym miejscu tej permutacji znajduje się liczba 1. Abyś nie musiał(a) zgadywać w ciemno, Bajtazar będzie udzielać Ci odpowiedzi na pytania postaci:

- $f(i, j, d)$ : czy różnica między  $i$ -tym a  $j$ -tym elementem permutacji  $P$  jest podzielna przez  $d$ , tzn. czy  $d \mid P[i] - P[j]$ ?
- $g(i, j)$ : czy  $i$ -ty element permutacji  $P$  jest większy niż  $j$ -ty element permutacji  $P$ ?

W powyższych pytaniach  $i, j$  są dowolnymi indeksami ze zbioru  $\{1, 2, \dots, n\}$ , zaś  $d$  jest dowolną liczbą całkowitą dodatnią.

Podczas zgadywanki możesz dowolnie wiele razy użyć pytania typu  $f$ , natomiast liczba wykonanych pytań typu  $g$  musi być jak najmniejsza.

Napisz program komunikujący się z biblioteką dostarczoną przez Bajtazara, który rozwiąże tę zagadkę.

## Ocenianie

Niech  $M(n)$  będzie najmniejszą możliwą liczbą pytań typu  $g$ , za pomocą której da się znaleźć jedyńkę w permutacji o ustalonej długości  $n$ , niezależnie od tego, jaka jest ta permutacja. Twój program otrzyma punkty za dany test, tylko jeśli liczba pytań typu  $g$ , jaką wykona, nie przekroczy  $M(n)$ . Poza tym program musi zmieścić się w limicie czasowym, a więc wykonać rozsądną liczbę pytań typu  $f$  (choć niekoniecznie minimalną).

We wszystkich testach zachodzi warunek  $1 \leq n \leq 500\,000$ . W testach wartych 28% punktów zachodzi dodatkowy warunek  $n \leq 5\,000$ .

## Komunikacja

Aby użyć biblioteki, należy wpisać na początku programu:

- **C/C++:** `#include "cgdzlib.h"`
- **Pascal:** `uses pgdzlib;`

Biblioteka udostępnia następujące funkcje i procedury:

- **inicjuj** – zwraca liczbę  $n$ . Powinna zostać użyta dokładnie raz, na samym początku działania programu.
  - **C/C++:** `int inicjuj();`
  - **Pascal:** `function inicjuj: LongInt;`
- **f(i, j, d)** – zadaje bibliotece Bajtazara pytanie typu  $f$ . Jej wynikiem jest jedna liczba: 1, jeśli  $d \mid P[i] - P[j]$ , a 0 w przeciwnym przypadku.
  - **C/C++:** `int f(int i, int j, int d);`
  - **Pascal:** `function f(i, j, d : LongInt) : LongInt;`
- **g(i, j)** – zadaje bibliotece Bajtazara pytanie typu  $g$ . Jej wynikiem jest jedna liczba: 1, jeśli  $P[i] > P[j]$ , a 0 w przeciwnym przypadku.
  - **C/C++:** `int g(int i, int j);`
  - **Pascal:** `function g(i, j : LongInt) : LongInt;`
- **odpowiedz(k)** – odpowiada bibliotece Bajtazara, że jedyńka znajduje się na  $k$ -tej pozycji w permutacji (tzn. że  $P[k] = 1$ ). Uruchomienie tej procedury/funkcji **kończy działanie Twojego programu**.
  - **C/C++:** `void odpowiedz(int k);`
  - **Pascal:** `procedure odpowiedz(k : LongInt);`

Twój program **nie może** czytać żadnych danych (ani ze standardowego wejścia, ani z plików). **Nie może** również nic wypisywać do plików ani na standardowe wyjście. Może pisać na standardowe wyjście diagnostyczne (`stderr`) – pamiętaj jednak, że zużywa to cenny czas.

\*Permutacja liczb od 1 do  $n$  to taki ciąg o długości  $n$ , w którym każda liczba od 1 do  $n$  występuje dokładnie raz.

## Przykładowe wykonanie

Poniższa tabela zawiera przykładowy ciąg pytań do biblioteki Bajtazara prowadzący do odgadnięcia pozycji, na której znajduje się jedyńka.

Numer pytania	Wywołanie	Wynik	Wyjaśnienie
	inicjuj	5	$n = 5$
1	$f(1, 2, 2)$	0	$2 \nmid P[1] - P[2]$
2	$g(1, 2)$	0	$P[1] < P[2]$
3	$f(3, 2, 3)$	1	$3 \mid P[3] - P[2]$
4	$g(2, 5)$	1	$P[2] > P[5]$
5	$f(1, 3, 2)$	1	$2 \mid P[1] - P[3]$
6	$f(1, 4, 3)$	1	$3 \mid P[1] - P[4]$
	odpowiedz(4)		Odpowiadamy, że $k = 1$ .

Po drugim pytaniu wiemy, że  $P[2] \neq 1$ . Stąd po trzecim pytaniu zachodzi jedna z możliwości: ( $P[2] = 2$ ,  $P[3] = 5$ ) lub ( $P[2] = 4$ ,  $P[3] = 1$ ), lub ( $P[2] = 5$ ,  $P[3] = 2$ ). Czwarte pytanie eliminuje pierwszą z tych możliwości. Piąte pytanie pozwala teraz stwierdzić, że  $P[1] \in \{3, 4\}$ . Skoro więc w szóstym pytaniu mamy  $3 \mid P[1] - P[4]$ , to  $P[1] = 4$ ,  $P[4] = 1$ . Szukaną pozycją w permutacji jest  $k = 4$ .

Podana sekwencja pytań poprawnie znajduje pozycję jedyńki w permutacji, jednak nie uzyskalaby żadnych punktów za ten test, gdyż w dowolnej permutacji pięciu elementów jedynekę da się znaleźć za pomocą co najwyżej jednego pytania typu  $g$  (tj.  $M(5) = 1$ ). Zauważ, że liczba wykonanych pytań typu  $f$  nie gra tu roli.

## Eksperymenty

W katalogu `/home/zawodnik/rozw/gdz` na stacji roboczej dostępna jest przykładowa biblioteka, która pozwoli Ci przetestować poprawność formalną rozwiązania. Biblioteka wczytuje opis permutacji ze standardowego wejścia w następującym formacie:

- w pierwszym wierszu liczba całkowita  $n$  – długość permutacji;
- w drugim wierszu  $n$  pooddzielanych pojedynczymi odstępami liczb od 1 do  $n$  – kolejne elementy permutacji.

Przykładowe wejście dla biblioteki znajduje się w pliku `gdz0.in`. Po zakończeniu działania programu biblioteka wypisuje na standardowe wyjście informację, czy odpowiedź udzielona przez Twoje rozwiązanie była poprawna, oraz liczbę zadanych pytań typu  $g$ .

W tym samym katalogu znajdują się przykładowe rozwiązania `gdz.c`, `gdz.cpp` i `gdz.pas` korzystające z biblioteki. Rozwiązania te nie minimalizują liczby pytań typu  $g$ .

Do kompilacji rozwiązania wraz z biblioteką służą polecenia:

- **C:** `gcc -O2 -static cgdzlib.c gdz.c -lm -o gdz`
- **C++:** `g++ -O2 -static cgdzlib.c gdz.cpp -lm -o gdz`
- **Pascal:** `ppc386 -O2 -XS -Xt gdz.pas`

Plik z rozwiązaniem i biblioteka powinny znajdować się w tym samym katalogu.