

Zadanie: POS

Poszukiwania [B]



POTYCZKI ALGORYTMICZNE

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.

Bajtazar jest astronomem, pracującym w projekcie Poszukiwania Pozabajtockiej Inteligencji. Jego radio-teleskop od wielu lat zbiera sygnały z odległej planety. Sygnały te mają dość charakterystyczną postać – są ciągami liczb zapisanymi w systemie binarnym. Bajtazar myśli, że zebrane sygnały mogą pochodzić od pozabajtockiej cywilizacji, jednak do tej pory nie udało mu się ich rozszyfrować.

Ostatnio bajtocy matematycy znaleźli pewien interesujący ciąg liczbowy. Bajtazar przyjrzał mu się i stwierdził, że wygląda on znajomo. Czyżby pozabajtocka cywilizacja wysyłała właśnie fragment tego ciągu?

Dane są dwa ciągi liczbowe: (s_i) dla $i = 1, \dots, S$ (sygnał nadawany przez kosmitów) oraz (m_i) dla $i = 1, \dots, M$ (ciąg znaleziony przez matematyków). Znajdź wszystkie takie liczby całkowite p , $0 \leq p \leq M - S$, że dla każdego $i = 1, \dots, S$ mamy $s_i = m_{p+i}$.

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:

```
#include "poszukiwania.h"
```

Biblioteka udostępnia cztery funkcje:

- `SignalLength()` – zwraca długość sygnału S ($1 \leq S \leq 2 \cdot 10^9$).
- `SeqLength()` – zwraca długość ciągu znalezionej przez matematyków M ($S \leq M \leq 2 \cdot 10^9$).
- `SignalAt(i)` – zwraca s_i , i -ty wyraz ciągu będącego sygnałem ($1 \leq i \leq S$, $0 \leq s_i \leq 10^9$).
- `SeqAt(i)` – zwraca m_i , i -ty wyraz ciągu znalezionej przez matematyków ($1 \leq i \leq M$, $0 \leq m_i \leq 10^9$).

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

```
long long SignalLength();
long long SeqLength();
long long SignalAt(long long i);
long long SeqAt(long long i);
```

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 wartości p , które spełniają warunek z treści zadania.

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

Przykładowy przebieg programu

Dla przebiegu programu:

Wywołanie funkcji	Zwrócona wartość
SignalLength();	5
SignalAt(1);	1
SignalAt(2);	2
SignalAt(3);	3
SignalAt(4);	2
SignalAt(5);	1
SeqLength();	11
SeqAt(1);	2
SeqAt(2);	1
SeqAt(3);	2
SeqAt(4);	3
SeqAt(5);	2
SeqAt(6);	1
SeqAt(7);	2
SeqAt(8);	3
SeqAt(9);	2
SeqAt(10);	1
SeqAt(11);	2

poprawnym wynikiem jest:

2

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:
 - SignalLength() zwraca 10^7 ,
 - SeqLength() zwraca $2 \cdot 10^7$,
 - SignalAt(i) zwraca resztę z dzielenia $(i - 1)$ przez 7,
 - SeqAt(i) zwraca resztę z dzielenia $(i - 1)$ przez 7;
- 0c:
 - SignalLength() zwraca $2 \cdot 10^7$,
 - SeqLength() zwraca $2 \cdot 10^8$,
 - SignalAt(i) zwraca $(i - 1) + 10^7$,
 - SeqAt(i) zwraca resztę z dzielenia $(i - 1)$ przez $4 \cdot 10^7$.