

Zadanie: KAS

Kasyno



XXXII OI, etap I. Plik źródłowy kas.* Dostępna pamięć: 512 MB.

14.10–18.11.2024

Uwaga: W tym zadaniu poznasz wynik punktowy swoich zgłoszeń dopiero po zakończeniu zawodów.

Po wątpliwej jakości inwestycji w zakłady bukmacherskie Bajtazar postanowił odzyskać stracone pieniądze w pobliskim kasynie. Do gustu przypadła mu specyficzna maszyna oferująca grę w zgadywanie liczby. Zasady gry są dość nietypowe.

Przed grą maszyna losuje pewną liczbę x ($1 \leq x \leq n$) z jednostajnym rozkładem (czyli prawdopodobieństwo wylosowania każdej liczby jest takie samo i wynosi $\frac{1}{n}$). Bajtazar może wrzucić do maszyny jednego bajtalara i podać liczbę y ($1 \leq y \leq n$). Wtedy maszyna daje mu w wyniku największy wspólny dzielnik liczb x oraz y , czyli $\text{nwd}(x, y)$. Po wrzuceniu bajtalara, zamiast podawać liczbę y , Bajtazar może też (lekko) szturchnąć maszynę, co spowoduje ponowne wylosowanie liczby x , również z jednostajnym rozkładem. Możesz założyć, że kolejne losowania są niezależne od siebie.

Trzecią możliwą akcją po wrzuceniu jednego bajtalara jest próba zgadnięcia liczby x , co kończy grę (niezależnie od tego, czy próba była udana). Gdy Bajtazar zgadnie liczbę x , wygrywa grę i rozpoczyna kolejną grę. W przeciwnym wypadku Bajtazar przegrywa i smutny opuszcza kasyno, nie odebrawszy nawet swojej ewentualnej wygranej.

Bajtazar zna liczbę n i ma do dyspozycji 10^7 bajtalarów. Każda akcja (zadanie pytania, szturchnięcie, próba zgadnięcia) wymagają wrzucenia jednego bajtalara. Pomóż Bajtazarowi w wygraniu jak największej liczby rozgrywek, zanim skończy mu się budżet, bez przegrywania ani jednej rozgrywki.

Komunikacja

To zadanie jest interaktywne. Należy napisać program, który będzie grał z maszyną, używając do tego dostarczonej biblioteki. Aby użyć biblioteki, należy wpisać w swoim programie:

- C++: `#include "kaslib.h"`
- Python: `from kaslib import DajN, Pytaj, Szturchnij, Odpowiedz`

Biblioteka udostępnia następujące funkcje:

- C++: `long long DajN()`
Python: `DajN()`
Zwraca parametr n z treści zadania, który jest stały i równy 10^{18} . Ta funkcja nie ma wpływu na działanie maszyny. Nie trzeba jej wywoływać i jest dostarczona tylko i wyłącznie dla wygody implementacji rozwiązań.
- C++: `long long Pytaj(long long y)`
Python: `Pytaj(y)`
Daje w wyniku $\text{nwd}(x, y)$ dla aktualnej wartości x .
- C++: `void Szturchnij()`
Python: `Szturchnij()`
Ustawia x na losową (z jednostajnym rozkładem) wartość ze zbioru $\{1, 2, \dots, n\}$.
- C++: `void Odpowiedz(long long y)`
Python: `Odpowiedz(y)`
Jeżeli $x = y$ zwiększa liczbę wygranych gier o 1 i natychmiast zaczyna kolejną rozgrywkę. W przeciwnym wypadku natychmiastowo kończy wykonanie z werdyktem „błędna odpowiedź”.

Należy założyć, że po wykorzystaniu limitu 10^7 zapytań wykonanie programu automatycznie się zakończy. **Zakończenie działania programu z innego powodu skutkować będzie werdyktem „błędna odpowiedź”.** Innymi słowy, Twój program musi wykonać 10^7 zapytań, aby otrzymać niezerową liczbę punktów.

Użycie którejkolwiek funkcji z niepoprawnymi argumentami będzie skutkowało otrzymaniem werdyktu „błędna odpowiedź”.

Twój program nie może otwierać żadnych plików ani używać standardowego wejścia i wyjścia. Może on korzystać ze standardowego wyjścia diagnostycznego (`stderr`), jednak pamiętaj, że zużywa to cenny czas.

Celowe próby wpłynięcia na wewnętrzne działanie biblioteki oceniane są zakazane.

Uwaga: Podane na górze ograniczenie pamięci dotyczy tylko Twojego rozwiązania, a zatem nie wlicza pamięci wykorzystywanej przez bibliotekę.

Eksperymenty

W sekcji „Pliki i testy” w SIO możesz znaleźć przykładowe pliki `kaslib.h` w C++ i `kaslib.py` w Pythonie, które implementują biblioteczke zgodną z treścią zadania. Wystarczy umieścić odpowiedni plik w folderze ze swoim rozwiązaniem i skompilować/uruchomić swoje rozwiązanie wraz z biblioteką np. za pomocą jednego z następujących poleceń:

- C++: `g++ -O3 -static -o kas kas.cpp -std=c++20`
- Python: `python3 kas.py`

Po uruchomieniu programu, jeśli jego przebieg będzie pomyślny, dostaniemy komunikat o liczbie wygranych. Losowość w tej biblioteczce jest zawsze taka sama, ale można ją ręcznie zmieniać, o czym mowa w komentarzach w biblioteczce. Zachęcamy do zapoznania się z nią.

Przykładowy przebieg programu

Wywołana funkcja	Wynik	Opis
—	—	początek interakcji – maszyna wylosowała $x = 7$
Pytaj(9)	1	$nwd(x, 9) = 1$
Szturchnij()	—	maszyna wylosowała nową wartość $x = 6$
Pytaj(2)	2	$nwd(x, 2) = 2$
Pytaj(3)	3	$nwd(x, 3) = 3$
Odpowiedz(6)	—	wygraliśmy – maszyna wylosowuje x na kolejną grę

Ocenianie

Liczba punktów, które Twoje rozwiązanie otrzyma na danym teście, jest określana w następujący sposób. Niech w będzie liczbą wygranych gier. Wtedy liczba punktów to $\lfloor 100 \cdot \frac{\log_2(1 + \min(2000, w))}{\log_2(1 + 2000)} \rfloor$ procent punktów przypisanych do tego testu.

W tym zadaniu jest tylko jedna grupa testów, w której znajduje się dokładnie 10 testów.

Ciąg liczb losowanych przez maszynę w jednym teście jest zawsze taki sam w każdym wykonaniu programu na tym teście. Ciąg dla każdego testu został wylosowany jak w treści.

Przykład

Test przykładowy 0 w SIO ma taką samą strukturę jak każdy z pojedynczych testów właściwych.