Bajthomix®

WWI 2024 – grupa 3
Dzień 5 – 20 sierpnia 2024

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

Bajtałko rezygnuje. Wasz klasowy kucharz zawiesza chochlę na kołku, a wyjazdy na żagle już nigdy nie będą takie same. Pech chciał, że dopiero co dojechaliście do Bitorzewa. Gdy wszystkim wydaje się, że całą wyprawę szlag trafił i smak słynnej zupy ryżowej Bajtałka będziecie znali już tylko ze wspomnień, pojawia się on: Bitar. Nie przychodzi jednak z pustymi dłońmi. W rękach trzyma Bajthomix®.

Bajthomix® to urządzenie, które zrewolucjonizowało świat bajtockich kulinariów. Jego użytkowanie jest bajecznie proste: żeby wykonać przepis, dla każdego z n dostępnych składników (a może być ich nawet 128!) wybierasz, czy ma się znaleźć on w Twojej potrawie, czy nie. Gdy zadecydujesz o wszystkich składnikach, wciskasz guzik `dodaj`.

To jednak nie wszystko: nie musisz wcale zatrzymywać się na jednym dodanym daniu – w Bajthomixie® można przygotowywać nawet k przepisów naraz! Gdy wprowadzisz wszystkie pożądane przepisy do systemu, wciskasz wielki zielony guzik `let_him_cook` i *voilà*.

Ano właśnie żadne *voilà*. Bajthomix® przyniesiony przez Bitara jest popsuty i wypluwa zupełnie inne dania niż chcecie. Udało wam się ustalić, że usterka jest bardzo specyficzna: dla każdego składnika, o który poprosiliście w danym przepisie, maszyna aktywuje w tym przepisie jakiś inny składnik. Co więcej, każdy składnik jest aktywowany jedynie przez wybranie dokładnie jednego konkretnego (innego lub tego samego) składnika. Innymi słowy, jeżeli dodawany przez nas przepis przedstawimy jako ciąg binarny $a_0, a_1, ..., a_{n-1}$ (dla i-tego indeksu 0 oznacza nieużycie danego składnika, a 1 - użycie), to niewdzięczne urządzenie wielofunkcyjne ugotuje przepis, który można przedstawić jako ciąg $a_{p_0}, a_{p_1}, ..., a_{p_{n-1}}$, gdzie $p_0, p_1, ..., p_{n-1}$ to permutacja liczb od 0 do $n-1$. Permutacja ta na i-tym indeksie przechowuje wiadomość, przez który składnik aktywowany jest składnik i. Co ważne, permutacja ta jest identyczna niezależnie od tego, jaki przepis przyrządzamy!

Dania przyrządzone przez Bajthomix® są... ciekawe, więc czasami ciężko powiedzieć, co jest czym. Na szczęście Bajthomix® oferuje funkcje rozpoznawania potraw. Gdy już wciśniesz guzik `let_him_cook` i zakończy się proces gotowania, możesz złożyć zapytania postaci: *czy zostało upichcone danie o składnikach* $b_0, b_1, ..., b_{n-1}$ (mówimy tutaj o ostatecznych użytych składnikach, niekoniecznie tych wybranych przez użytkownika). Oczywiście ta funkcja również jest wadliwa, więc możesz z niej skorzystać co najwyżej m razy.

Bitar wpadł na genialny pomysł: jak poznacie permutację $p_0, p_1, ..., p_{n-1}$, to będziecie w stanie przygotowywać takie dania, jakie Wam się tylko zamarzy. Twoim zadaniem jest więc ją poznać.

Chcielibyście dzisiaj zjeść coś na kolację, spieszy Wam się - dlatego też zielony przycisk `let_him_cook` możecie wcisnąć tylko raz.

Komunikacja

To jest zadanie interaktywne. Twoim zadaniem jest napisanie programu, który będzie komunikował się z biblioteką oceniającą. Innymi słowy, Twój program nie będzie czytał z wejścia ani wypisywał na wyjście. Zamiast tego powinien korzystać z następujących funkcji dostarczonych przez bibliotekę:

- `void init(int & n, int & k, int & m)` – tę funkcję Twój program powinien wykonać dokładnie raz na początku działania programu, przed wywoływaniem pozostałych funkcji. Funkcja wczytuje do podanych zmiennych wartości n, k i m oznaczające odpowiednio: liczbę różnych składników dostępnych w Bajthomixie®, liczbę unikalnych dań, jakie możecie przygotować oraz liczbę zapytań o upichcone dania, które możecie wykonać.
- `void dodaj(string a)` – funkcja ta powoduje dodanie do Bajthomixa® przepisu, reprezentowanego ciągiem binarnym $a_0, a_1, ..., a_{n-1}$, gdzie $a_i = 0$ oznacza, że nie chcemy wykorzystać składnika numer i, a $a_i = 1$ oznacza, że chcemy wykorzystać składnik numer i. Funkcji tej możesz użyć co najwyżej k razy. Jeżeli dodasz dany przepis więcej niż raz, to nic się nie stanie, ale operacja ta będzie wliczała się do limitu.
- `void let_him_cook()` – funkcji tej możesz użyć tylko raz, powoduje ona przyrządzenie dodanych przepisów, po przepermutowaniu ich zgodnie z niejawną permutacją $p_0, p_1, ..., p_{n-1}$. Po użyciu tej funkcji nie możesz już używać funkcji `dodaj`.

- `bool zapytaj(string b)` – funkcja ta zwraca wartość binarną, oznaczającą, czy w Bajthomixie® zostało przygotowane danie o składnikach, które można przedstawić jako ciąg binarny $b_0, b_1, ..., b_{n-1}$. Funkcję tę możesz wywołać co najwyżej m razy. Funkcji tej możesz używać dopiero po wywołaniu funkcji `let_him_cook`.
- `void odpowiedz(vector<int> p)` – tę funkcję Twój program powinien wykonać dokładnie raz, na końcu swojego działania. Wektor `p` powinien zawierać permutację, której używa Bajthomix® do pomieszania składników potraw.

Niepoprawna komunikacja poskutkuje werdyktem `Zła odpowiedź`.

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.

Kompilacja

Aby program się skompilował, a odpowiednie funkcje były dostępne, należy załączyć nagłówek `#include "bthlib.h"`. Program należy skompilować razem z biblioteką `bthlib.cc`. Można to zrobić za pomocą polecenia:
`g++ -O3 twoj_program.cpp bthlib.cc -o twoj_program`.
Wszystkie potrzebne pliki, razem z przykładowym (błędnym) rozwiązaniem znajdują się w archiwum `bth_dlazaw.zip`, które można znaleźć w zakładce "Pliki". Archiwum to należy wypakować do folderu z kodem źródłowym swojego programu.
Uwaga: Biblioteka `bthlib.cc` zamieszczona w zakładce "Pliki" **może różnić się** od biblioteki sprawdzającej Twoje zgłoszenia w systemie.

Przykład

W pierwszym z dwóch testów przykładowych $n = 4$, $k = 16$, $m = 16$, natomiast permutacja $p = [2, 0, 3, 1]$.

Funkcja	Zwrócona wartość	Opis
`init(n,k,m)`	–	Funkcja wczytuje do podanych zmiennych wartości $n = 4, k = 16, m = 16$.
`dodaj("1000")`	–	Dodajemy do Bajthomixa® przepis "1000"
`dodaj("0011")`	–	Dodajemy do Bajthomixa® przepis "0011"
`let_him_cook()`	–	Pozwalamy mu gotować.
`zapytaj("1000")`	`false`	W Bajthomixie® nie ma dania "1000".
`zapytaj("0100")`	`true`	W Bajthomixie® jest danie "0100"(powstało poprzez przepermutowanie przepisu "1000").
`zapytaj("1010")`	`true`	W Bajthomixie® jest danie "1010"(powstało poprzez przepermutowanie przepisu "0011").
`odpowiedz({2,0,3,1})`	–	Zgadujemy, że szukana permutacja wynosi $\{2, 0, 3, 1\}$. Po wywołaniu tej funkcji powinniśmy zakończyć działanie programu.

Ocenianie

Podzadanie	Ograniczenia	Limit czasu	Liczba punktów
1	$n = 8, k = 256, m = 256$, dla co najwyżej dwóch indeksów i: $p_i \neq i$	2 s	20
2	$n = 32, k = 320, m = 1024$	2 s	18
3	$n = 32, k = 1024, m = 320$	2 s	11
4	$n = 128, k = 1792, m = 1792$	2 s	21
5	$n = 128, k = 896, m = 896$	2 s	30