

Задача 2. Едно

Имаме скрит масив от N бита – a_0, \dots, a_{N-1} . В една заявка можем да изберем подмножество от позиции в редицата и да обърнем битовете на тези позиции. Обръщане на бит със стойност 0 го прави 1, а обръщане на бит 1 го променя на 0. След всяка заявка получаваме дължината на най-дългия последователен подмасив, който се състои единствено от 1-ици в новата редица. **Промените от заявките се запазват, тоест обърнат бит от предишна заявка ще остане обърнат докато не е обърнат отново.**

Искаме да намерим местоположението във финалната редица на най-дългия подмасив от единици (или който и да е от тях, ако съществува повече от един). Напишете програмата, която да свърши това с възможно най-малко заявки.

Детайли по имплементацията

Има T подтеста за всеки тест.

Вашата програма трябва да съдържа функция със следния прототип:

```
std::pair<int, int> find_longest_subarray_of_ones(int n);
```

Тя ще бъде повикана T пъти на тест и ще получи числото N като параметър – дължината на скритата редица. Функцията трябва да връща $pair\{l, r\}$, където l е индексът на най-лявата част на търсения подмасив, а r е индексът на най-дясния му елемент.

Заявките ще се правят чрез функцията *flip_bits*:

```
int flip_bits(const std::vector<bool> &flips);
```

Тя получава вектор от N бита $flips_0, \dots, flips_{n-1}$, където $flips_i = 1$ означава, че a_i ще бъде обърнат. След като обърне избраните битове, функцията връща дължината на най-дългия подмасив от единици в скритата редица.

Вие трябва да изпратите файла *ones* към системата. Той трябва да съдържа функцията *find_longest_subarray_of_ones*. Той може да съдържа и друг код и функции, които са необходими за работата Ви, но не трябва да съдържа главната функция *main*. Също така не трябва да четете от стандартния вход или да пишете на стандартния изход. Програмата ви трябва да включва хедър файла *ones.h* чрез инструкция към препроцесора:

```
#include "ones.h"
```

Локално тестване

Даден Ви е файла *Lgrader.cpp*, който може да бъде компилиран заедно с програмата Ви, за да я тествате. Той прочита броя T на подтестовете за дадения тест, последван от описание за всеки от тях. За всеки подтест, първо ще бъде дадено числото N , последвано от $a_0 \dots a_{N-1}$ на следващия ред. Ако работи правилно, вашата програма ще изведе 1 и максималния брой заявки, използвани за подтест. Иначе, ще бъде изведено само числото 0.

Ограничения

$$T = 5$$

$$1 \leq N \leq 10^4$$

$$0 \leq a_i \leq 1$$

Задача 2. Ones

Оценяване

Резултатът, който ще получите за задачата, зависи от максималния брой заявки Q , които програмата Ви е използвала, за да реши един подтест. Резултатът се изчислява по формулата:

$$\left(\frac{85}{\max(85, Q)}\right)^{0.294} \times 100$$

Примерна комуникация

Нека началният масив да е 1, 0, 1. Тогава един начин, по който комуникацията би протекла е:

| Вашата функция | Програмата на журито |
|---|---|
| | Извиква <code>find_longest_subarray_of_ones(3)</code> |
| Извиква <code>flip_bits({0, 0, 0})</code> | Връща стойност 1 |
| Извиква <code>flip_bits({0, 1, 0})</code> | Връща стойност 3 |
| Връща стойност <code>{0, 2}</code> | |