

Фалшиви монети

Ави и Боби откриват съкровище от $N = 2^k$ монети (т.е. N е степен на 2), номерирани $0, 1, \dots, (N - 1)$. Те знаят, че имат $N - 2$ монети, тежащи 10г, 1 монета, тежаща 9г и една монета, тежаща 11г. Децата искат да разберат коя е 11 грамовата монета, защото смятат, че за разлика от останалите тя е направена от чисто злато. Ави и Боби разполагат единствено с везна (без теглилки) и ограничено време, за което могат да направят най-много Q претегляния.

Вашата задача е да напишете програма, която да помогне на Ави и Боби да открият по-тежката монета.

Детайли по имплементацията

Вашата програма трябва да съдържа в началото си `#include "fakecoins.h"`.

Вашата програма трябва да реализира следната функция:

- `int find_heavier(int N)`
 - N : броят на монетите на Ави и Боби. Гарантирано е, че $N = 2^k$.
 - Функцията трябва да връща на индекса на по-тежката монета (монетите са индексирани от 0).
 - Функцията се вика веднъж по време на изпълнението на програмата.

Вашата програма може да задава въпроси, използвайки следната функция:

- `int compare_coins(std::vector<int> a, std::vector<int> b)`
 - a : индексите a_0, a_1, \dots, a_{M-1} , на монетите, които Ави и Боби поставят на първото блюдо на везната. Трябва да задоволяват $0 \leq a_i < N$ за всяко $0 \leq i \leq M - 1$ и $a_i \neq a_j$ за всеки $0 \leq i < j \leq M - 1$.
 - b : индексите b_0, b_1, \dots, b_{M-1} , на монетите, които поставят на второто блюдо на везната. Трябва да задоволяват $0 \leq b_i < N$ за всяко $0 \leq i \leq M - 1$ и $b_i \neq b_j$ за всеки $0 \leq i < j \leq M - 1$.
 - Всяка монета може да се съдържа най-много в едно от блюдата, т.е. $a_i \neq a_j$ за всеки $0 \leq i, j \leq M - 1$.
 - Двете блюда трябва да съдържат равен брой монети, т.е. a и b трябва да са с равна дължина.
 - Нека монетите в първото блюдо тежат общо A , а тези във второто B . Функцията връща:
 - * -1 , ако $A < B$
 - * 0 , ако $A = B$
 - * 1 , ако $A > B$
 - Функцията може да се вика Q пъти за един тест.

В случай, че наруши ограниченията за параметри на `compare_coins`, ще получите съобщение „Invalid question asked.“.

В случай, че извикате функцията `compare_coins` повече от Q пъти, ще получите съобщение „Too many questions asked.“.

Пример

Грейдърът извършва следното извикване на функция:

- `find_heavier(4)` Ави и Боби имат $N = 4$ монети.

Програмата извършва следните извиквания на функции:

- `compare_coins([0], [1])` връща -1
- `compare_coins([2, 3], [1, 0])` връща 1
- `compare_coins([1], [2])` връща 0
- `compare_coins([1], [3])` връща -1

По-леката монета е с индекс 0, а по-тежката с индекс 3. Съответно `compare_coins` трябва да върне 3.

Подзадачи

Във всички подзадачи $N = 2^k$.

В някои подзадачи, грейдърът *не е адаптивен*, т.е. за всеки тест индексите на по-леката и по-тежката монета са фиксирани преди изпълнението му.

В останлите подзадачи грейдърът *е адаптивен*, т.е. индексите на по-леката и по-тежката монета могат да се определят в зависимост от въпросите, които Вашата програма задава. Разбира се, индексите са избират така, че да не противоречат на отговорите, които грейдърът е дал на Вашата програма.

1. (20 точки) $1 \leq N \leq 128$, $Q = 128$, гредърът *не е адаптивен*
2. (30 точки) $1 \leq N \leq 1024$, $Q = 30$, гредърът *не е адаптивен*
3. (40 точки) $1 \leq N \leq 1024$, $Q = 20$, гредърът *е адаптивен*
4. (10 точки) $1 \leq N \leq 1024$, $Q = 16$, гредърът *е адаптивен*

Локално тестване

За да можете да тествате решението си на компютъра си, Ви се предоставят файловете `Lgrader.cpp` и `fakecoins.h`, които да компилирате заедно с Вашето решение `fakecoins.cpp`.

Вход

- Ред 1: две цели числа N и Q
- Ред 2: две цели числа a и b : индексите съответно на по-леката и по-тежката монета.

Изход

- Ред 1:
 - „*Output is correct.*“, ако програмата е преминала успешно теста.
 - „*Invalid question.*“, ако програмата е задала въпрос, неотговарящ на гореописаните ограничения.
 - „*Too many questions asked.*“, ако програмата е извършила повече от Q извиквания на `compare_coins`.
 - „*Output isn't correct.*“, ако програмата не е надвишила максималния разрешен брой въпроси, но не е намерила правилно индекса на по-тежката монета.

Изпращане на тестове към системата

Можете да изпращате собствени тестове към системата.

Вход

- Ред 1: две цели числа N и Q
- Ред 2:
 - 0 a b , ако грейдърът не е адаптивен. a и b са съответно индексите на по-леката и по-тежката монета.
 - 1, ако грейдърът е адаптивен.

Изход

Форматът на изхода е същият като на локалния грейдър.