

Задача A5. GAME
 1,5 сек.  256 MB

Клими и Никол разполагат с масив от N цели числа A , номерирани с числата от 0 до $N - 1$ и съдържащ **различни** цели стойности. Момичетата играят игра, като местят пионка по масива. В началото, пионката се намира в някоя клетка. Един ход на играта протича по следния начин:

1. Играчът, който е на ход, взема пионката и я премества в друга клетка, която е на разстояние най-много D от клетката, в която е бил. Формално, ако пионката е в момента в клетка x ($0 \leq x < N$), то играчът може да премести пионката в клетка y ($0 \leq y < N$), при условие че $|y - x| \leq D$ и $x \neq y$. Обърнете внимание, че това означава, че играчът **трябва** да премести пионката.
2. След като премести пионката в някоя клетка y , играчът прибавя A_y към своя резултат. Обърнете внимание, че резултатът се добавя **след** преместването на пионката.

Двете момичета се редуват да извършват ходове, като Клими играе първа. Играта свършва след точно 10^{100} хода, след което момичето с по-висок резултат печели. Ако техните резултати са равни, то Клими печели.

Момичетата не са много щастливи с времето, което е нужно, за да се завърши дори една игра, затова те искат от Вас да измислите оптимален начин на игра за различни сценарии.

Формално, ще Ви бъдат дадени началният масив A и стойност D и след това ще трябва да обработите Q заявки от два вида:

- Промени: Една стойност в масива се променя
- Въпроси: Пита се дали Клими (тази, която играе първа) печели играта при някои начални позиции на пионката при оптимална игра от двете момичета.

Промените се запазват за заявките, т.е. отговорът на всеки въпрос трябва да взема предвид всички досегашни промени по масива.

Детайли по имплементацията

Трябва да напишете три функции. Първата функция `init` трябва да има следния формат:

```
void init(std::vector<int> A, int D)
```

Тя ще бъде извикана веднъж в началото на теста със стойностите на масива A и стойността на D .

Другите две функции `updateValue` и `isWinning` трябва да имат следния формат:

```
void updateValue(int index, int newValue)
```

```
bool isWinning(int startIndex)
```

Общият брой извиквания към двете функции ще бъде равен на Q и всички тези извиквания ще бъдат направени след извикването на `init`.

Функцията `updateValue` трябва да обработи заявка за промяна и да промени стойността на A_{index} да бъде равна на `newValue`. Гарантирано е, че след всяка такава заявка, стойностите на масива A продължават да бъдат различни една от друга.

XV INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА, ШУМЕН 2024

Функцията `isWinning` трябва да обработи заявка за въпрос и да върне `true`, ако Клими може да спечели играта за текущия масив, при условие, че пионката започва от клетка `startIndex`. Иначе функцията трябва да върне `false`.

Вашата програма трябва да имплементира трите функции, но не трябва да съдържа функцията `main`. Освен това не трябва да чете от стандартния вход или да печата на стандартния изход. Вашата програма трябва да включва и хедърния файл `game.h` с инструкция към предпроцесора: `#include "game.h"`

Стига да спазвате тези условия, вашата програма може да съдържа всякакви помощни функции, променливи, константи и т.н.

Локално тестване

Предоставят Ви се файловете `Lgrader.cpp` и `game.h`, които може да компилирате заедно с вашата програма, за да проверите верността ѝ. Входният формат на грейдъра е:

- $N D$
- $A_0 A_1 \dots A_{N-1}$
- Q
- $\langle Q \text{ реда със заявките} \rangle$

Форматът на заявките е следният:

- `1 ind val` — заявка за промяна, която присвоява $A_{ind} = val$
- `2 ind` — заявка за въпрос за игра с пионка на стартова позиция `ind`

За всяка заявка от тип 2, грейдърът ще отпечата 0, ако вашата функция върне `false` и 1, ако вашата функция върне `true`

Ограничения

- $1 \leq N, Q \leq 2 \cdot 10^5$
- $1 \leq D \leq 25$
- $1 \leq A_i \leq 10^9$
- $0 \leq \text{index}, \text{startIndex} < N$
- $1 \leq \text{newValue} \leq 10^9$
- Стойностите на A са различни една от друга по всяко време, включително и след всяка заявка за промяна.

Подзадачи

Подзадача	Точки	N, Q	D	Допълнителна ограничения
1	8	$N, Q \leq 10$	$D \leq 25$	
2	18	$N, Q \leq 2 \cdot 10^3$	$D \leq 25$	Няма заявки за промяна.
3	16	$N, Q \leq 2 \cdot 10^5$	$D \leq 25$	Няма заявки за промяна
4	27	$N, Q \leq 10^5$	$D \leq 10$	
5	31	$N, Q \leq 2 \cdot 10^5$	$D \leq 25$	

Точките за дадена подзадача се получават само ако се преминат успешно всички тестове, предвидени за нея.

**XV INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS
ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА, ШУМЕН 2024**

Пример

Нека $A = [1, 7, 4, 9, 30, 2]$, $D = 2$, и $Q = 5$. Примерна поредица от извиквания е следната:

- Извикване на `init({1, 7, 4, 9, 30, 2}, 2)`
- Извикване на `isWinning(0)` което връща `true`
- Извикване на `isWinning(1)` което връща `false`
- Извикване на `updateValue(4, 8)`
- Извикване на `isWinning(0)` което връща `false`
- Извикване на `isWinning(1)` което връща `true`

Масивът след заявката за промяна става $[1, 7, 4, 9, 8, 2]$. Може да се покаже, че горните отговори на въпросите са верни, ако двете момичета играят оптимално.

Във формата на локалния грейдър, горният пример изглежда по следния начин:

Вход	Изход
6 2	1
1 7 4 9 30 2	0
5	0
2 0	1
2 1	
1 4 8	
2 0	
2 1	