# Задача Short on three

Header C++ `shortonthree.h`

Отборът има $N$ играчи, подредени в редица и номерирани от 0 до $N-1$ и играч $i$ има умение $v_i$. Треньорът стои в левия край на редицата. Той приема всяка топка, подадена към него, но не я връща, защото треньорът не участва в играта. В румънския футбол съществува стара команда, отправяна от тъчлинията – "short on two", която означава къс пас към най-близкия съотборник. Треньорът смята, че това е твърде голяма привилегия за играча, който просто се намира наблизо. Затова в неговия отбор командата е "short on three" – пасът отново е къс, но винаги трябва да бъде насочен към играч, който е строго по-добър.

Една **група** се състои от играчите, заемащи няколко **последователни** позиции. В началото на тренировъчното упражнение всеки играч от групата има топка в краката си, а треньорът застава непосредствено вляво от групата. След това всеки играч подава топката, с която е започнал упражнението, и задържа всяка топка, която получи. Даден играч подава топката към най-близкия играч отляво в рамките на групата, чието умение е строго по-голямо от неговото. Ако няма такъв играч, топката се подава към треньора.

Формално, нека играчите в групата бъдат номерирани от 0 до $m-1$, от ляво надясно, като $b_k$ обозначава нивото на умение на $k$-тия играч и нека треньорът заема позиция $-1$. За всяко $i$ от 0 до $m-1$ позицията в рамките на групата, към която играч $i$ подава своята топка, се определя по следния начин:

$$f(i) = \begin{cases} \max\{\, j \mid 0 \le j < i \text{ and } b_j > b_i \,\}, & \text{ако такъв } j \text{ съществува,} \\ -1, & \text{в противен случай.} \end{cases}$$

Например, при нива на умение 5 1 3 2 получаваме $f = -1, 0, 0, 2$: играчът с номер 2 пропуска този с номер 1, чието умение не е по-добро от това на играч номер 2.

Треньорът нарича една група **дисциплинирана**, ако последователността $f(0), f(1), \ldots, f(m-1)$ има формата на **планина**, т.е. съществува позиция $p$, за която $0 \le p \le m-1$, така че:

$$f(0) \le f(1) \le \cdots \le f(p) \ge f(p+1) \ge \cdots \ge f(m-1).$$

Той никога не е обяснявал защо е така, а и никой не е питал два пъти, защото треньорът не обича шегите.

През сезона се извършват $Q$ **трансфера**, които се прилагат последователно. Всеки трансфер се описва с две цели числа $pos$ и $val$, представящи окончателна замяна на играча, намиращ се на позиция $pos$ в редицата, с нов играч с ниво на умение $val$, който заема същото място.

### Задача

След всеки трансфер треньорът иска да знае максималния брой играчи, които могат да участват в една дисциплинирана група.

### Детайли по имплементацията

Тази задача изисква имплементация на функции. Вашата програма не трябва да чете от стандартния вход или да записва в стандартния изход. Трябва да включите хедър файла `shortonthree.h` и да имплементирате следната функция, без да имплементирате main функцията:

```
void init(int N, std::vector<int> v);
int transfer(int pos, int val);
```

Функцията `init` се извиква точно веднъж в началото.

- $N$: броят на играчите.
- $v$: вектор с дължина $N$, в който $v[i]$ представлява нивото на умение на играч $i$, за $0 \le i < N$.

След първоначалното извикване на функцията `init`, функцията `transfer` се извиква точно $Q$ пъти – по веднъж за всеки трансфер, в реда, в който те се извършват.

- *pos*, *val*: играчът на позиция *pos* (номерацията започва от 0) се заменя с нов играч, чието ниво на умение е *val*.
- Функцията трябва да връща максималния брой играчи в дисциплинирана група **след** извършването на съответния трансфер.

## Примерен грейдър

Примерният грейдър чете входните данни от стандартния вход в следния формат:

- ред 1: $N$ $Q$;
- ред 2: $v_0$ $v_1$ ... $v_{N-1}$;
- ред $3 + i$ ($0 \le i < Q$): *pos val*, описващи трансфер $i + 1$.

Примерният грейдър извиква `init`, след което извиква `transfer` по веднъж за всеки трансфер и извежда получените $Q$ стойности на стандартния изход – по една на ред.

Грейдърът, използван за оценяване на решението, може да се различава от примерния грейдър.

## Ограничения

- $1 \le N, Q \le 200\,000$.
- $1 \le v_i, val \le 1\,000\,000\,000$.
- $0 \le pos \le N - 1$.

| # | Точки | Ограничения |
|---|---|---|
| 1 | 9 | $N, Q \le 50$. |
| 2 | 15 | $N, Q \le 400$. |
| 3 | 18 | Всяко $v_i$ и всяко *val* е 1 или 2. |
| 4 | 26 | $N, Q \le 10\,000$. |
| 5 | 32 | Няма допълнителни ограничения. |

## Пример

| Вход | Изход |
|---|---|
| 6 2<br>1 2 1 3 2 5<br>1 3<br>3 1 | 4<br>6 |

Примерният грейдър прави следните извиквания:

```
init(6, {1, 2, 1, 3, 2, 5});
transfer(1, 3);
transfer(3, 1);
```

- След първия трансфер нивата на умение на играчите са 1 3 1 3 2 5. Групата, заемаща номера от 2 до 5, с нива на умение 1 3 2 5, дава $f = -1, -1, 1, -1$, което представлява планина. Може да се докаже, че няма дисциплинирана група от 5 или 6 играчи. Например, за цялата редица получаваме $f = -1, -1, 1, -1, 3, -1$, което не представлява планина. Следователно `transfer(1, 3)` връща 4.
- След втория трансфер нивата на умение на играчите са 1 3 1 1 2 5, като цялата редица представлява една дисциплинирана група с $f = -1, -1, 1, 1, 1, -1$. Следователно `transfer(3, 1)` връща 6.