

Zadanie: PJK

PJKSort



XIV obóz informatyczny, grupa zaawansowana, dzień 3. Dostępna pamięć: 128 MB. 18.01.2017

Przemek nieustannie pracuje nad obozową sprawdzarką, aby działała niezawodnie i najsprawniej, jak to możliwe. Ostatnio wymyślił nowy sposób na trzykrotne przyspieszenie działania sprawdzarki. Jednym z warunków koniecznych do działania tego sposobu jest posortowanie wszystkich do tej pory sprawdzonych zgłoszeń po ich rezultacie – pierwsze w kolejności powinny wystąpić wszystkie niezaakceptowane zgłoszenia, a następnie te zaakceptowane.

Przemek wymyślił ostatnio nowy algorytm sortujący swoje zgłoszenia i nazwał go PJKSort. Okazuje się, że procesor posiada pewną funkcję, która pozwala wykonywać pewne operacje jednocześnie *, dzięki czemu można znacznie zaoszczędzić na czasie algorytmu. Dokładniej, dla danego ciągu zero-jedynkowego procesor potrafi za pomocą tylko jednej operacji jednocześnie zamienić miejscami każdą jedynkę z następnym elementem ciągu, jeżeli jest nim zero. W tym przypadku jedynki oznaczają zgłoszenia zaakceptowane, a zera zgłoszenia odrzucone. Algorytm PJKSort polega na wykonywaniu opisanej powyżej funkcji procesora tak długo, aż ciąg będzie posortowany, tzn. wszystkie zera będą przed wszystkimi jedynkami.

Jedynym problemem jest zbadanie, czy PJKSort jest rzeczywiście tak szybkim algorytmem, jak się Przemkowi wydaje. W celu zbadania tego Przemek przygotował listę zgłoszeń uporządkowaną po czasie nadesłania i zastanawia się, ile iteracji wykona jego algorytm dla tego ciągu, gdzie każda iteracja wywołuje opisaną funkcję procesora. Ponadto, aby jak najlepiej przetestować algorytm, postanowił dokonywać małych zmian w ciągu, polegających na odwróceniu rezultatu dowolnego zgłoszenia (zaakceptowane zgłoszenie staje się niezaakceptowane i na odwrót). Po każdej takiej operacji należy ponownie podać, ile iteracji wykona algorytm dla zmodyfikowanego ciągu. Przemek prosi Cię o pomoc, ponieważ jeszcze implementuje PJKSort.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite n, q ($1 \leq n, q \leq 500\,000$), oznaczające odpowiednio liczbę zgłoszeń oraz liczbę operacji odwrócenia rezultatu zgłoszenia.

W drugim wierszu wejścia znajduje się n liczb a_i ($a_i \in \{0, 1\}$), oznaczające rezultaty kolejnych zgłoszeń. Wartość 1 oznacza zgłoszenie zaakceptowane, a 0 zgłoszenie niezaakceptowane.

W każdym z kolejnych q wierszy znajduje się jedna liczba całkowita p ($1 \leq p \leq n$), oznaczająca pozycję zgłoszenia, którego rezultat zostanie odwrócony.

Wyjście

Na wyjściu powinno pojawić się $q+1$ wierszy. i -ty wiersz powinien zawierać jedną liczbę całkowitą, oznaczającą liczbę wykonanych iteracji algorytmu PJKSort po zaaplikowaniu pierwszych $i-1$ zamian dla początkowego ciągu.

Przykład

Dla danych wejściowych:

```
6 2
1 1 0 0 1 0
4
1
```

poprawnym wynikiem jest:

```
4
4
3
```

*W rzeczywistości, procesory nie wykonują jednocześnie operacji opisanych w zadaniu.