

F. Podciągi

Dostępna pamięć: 128 MB

Dany jest ciąg A liczb naturalnych dodatnich, niekoniecznie różnych. Z tego ciągu usuwamy jeden *spójny* podciąg. Możemy w ten sposób otrzymać różne ciągi; ich zbiór oznaczmy przez $\mathcal{R}(A)$. Można też usunąć pusty ciąg, czyli $A \in \mathcal{R}(A)$. Teraz dla dowolnego ciągu B liczb naturalnych dodatnich zdefiniujmy $\text{NRP}(B)$ jako długość najdłuższego *spójnego* podciągu B , który jest (ściśle) rosnący. Twoim zadaniem jest znalezienie takiego ciągu B w zbiorze $\mathcal{R}(A)$, dla którego wartość $\text{NRP}(B)$ jest największa.

Przykładowo dla ciągu $A = (5, 3, 4, 9, 2, 8, 6, 7, 1)$, najlepszym rozwiązaniem jest usunięcie spójnego podciągu $(9, 2, 8)$. Otrzymujemy wtedy ciąg $(5, 3, 4, 6, 7, 1)$, w którym najdłuższy podciąg rosnący $(3, 4, 6, 7)$ ma długość równą 4.

Uwaga: W tym zadaniu zabronione jest używanie tych konstrukcji STL-a, których nazwy zawierają: `set`, `map`, `heap`, `priority_queue`, `lower_bound`, `upper_bound` lub `binary_search`. Niedozwolone jest też wykorzystanie gotowych odpowiedników tych konstrukcji w innych językach programowania.

Specyfikacja danych wejściowych

W pierwszym wierszu danych wejściowych znajduje się liczba naturalna $1 \leq T \leq 10$, będąca liczbą różnych ciągów. W kolejnych $2 \cdot T$ wierszach znajduje się opis T ciągów A_1, A_2, \dots, A_T , po dwa wiersze na każdy ciąg:

1. Pierwszy wiersz opisu ciągu A_i zawiera liczbę naturalną $1 \leq n_i \leq 2 \cdot 10^6$ będącą długością ciągu A_i .
2. Drugi wiersz opisu ciągu A_i zawiera n_i liczb naturalnych dodatnich, oddzielonych pojedynczymi spacjami, nie większych niż 10^9 , będących kolejnymi wyrazami ciągu A_i .

Specyfikacja danych wyjściowych

Twój program powinien wypisać T wierszy, po jednym dla każdego ciągu A_i . W i -tym wierszu powinna znaleźć się jedna liczba naturalna dodatnia równa $\max_{B \in \mathcal{R}(A_i)} \text{NRP}(B)$.

Przykład A

Wejście:

```
1
9
5 3 4 9 2 8 6 7 1
```

Wyjście:

```
4
```

Przykład B

Wejście:

```
1
7
1 2 3 10 4 5 6
```

Wyjście:

```
6
```

Przykład C

Wejście:

```
2
3
3 3 3
4
4 3 2 1
```

Wyjście:

```
1
1
```