Zadanie: SOR
Sortowanie komórkowe

Dostępna pamięć: 64 MB.

Sortowanie komórkowe jest bardzo ciekawym algorytmem o dużej jak na sortowanie złożoności czasowej. Algorytm ten działa krokowo, to znaczy wykonuje na danym ciągu pewien krok (sekwencję operacji), aż ciąg stanie się posortowany niemalejąco.

Krok sortowania wygląda tak, że analizujemy ciąg od lewej do prawej i na boku budujemy ciąg wynikowy kroku. Na początku do ciągu wynikowego wkładamy pierwszy element aktualnego ciągu, a następnie każdy kolejny element umieszczamy na początku ciągu pomocniczego, jeżeli poprzedni element oryginalnego ciągu był od niego większy, na końcu zaś, jeżeli poprzedni był od niego mniejszy. Dla przykładu, w jednym kroku algorytmu z ciągu: 5, 6, 2, 1, 4, 3 powstają kolejno ciągi pomocnicze:

- 5,
- 5, 6,
- 2, 5, 6,
- 1, 2, 5, 6,
- 1, 2, 5, 6, 4,
- 3, 1, 2, 5, 6, 4,

a ostatni z nich jest wynikiem działania tego kroku algorytmu.

Twoim zadaniem jest „odsortować” dany ciąg, czyli stwierdzić, ile różnych ciągów zmienia się w ten właśnie ciąg w jednym kroku algorytmu.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \leq n \leq 1000$). Drugi wiersz zawiera ciąg n parami różnych liczb całkowitych ze zbioru $\{1, \ldots, n\}$, reprezentujących ciąg, który należy odsortować.

Wyjście

Należy wypisać resztę z dzielenia przez 10^9 liczby różnych ciągów, które w jednym kroku sortowania komórkowego przechodzą na dany ciąg.

Przykład

Dla danych wejściowych:

```
4
1 2 3 4
```

poprawnym wynikiem jest:

```
8
```

natomiast dla danych wejściowych:

```
4
4 3 2 1
```

poprawnym wynikiem jest:

```
0
```

Wyjaśnienie do przykładu: W jednym kroku algorytmu sortowania na ciąg 1, 2, 3, 4 przechodzą ciągi:

- 1, 2, 3, 4,
- 4, 3, 2, 1,
- 2, 1, 3, 4,
- 3, 2, 1, 4,
- 2, 3, 1, 4,
- 2, 3, 4, 1,
- 3, 4, 2, 1,
- 3, 2, 4, 1,

natomiast na ciąg 4, 3, 2, 1 nie przechodzi żaden inny ciąg liczb.