Zadanie G
Antykwariat

Zgłoszeń: 54
Zaakceptowanych: 8

Pierwsze rozwiązanie:
uwr01 [Rzepecki, Górniak, Agrawal]
01:02:36*

Autor: Krzysztof Maziarz

Mamy dany ciąg n liczb $a_1, ..., a_n$ z przedziału $[1, M]$. Dostajemy dużo zapytań (L, R), dla każdego musimy podać liczbę spójnych przedziałów naszego ciągu które mają wszystkie wartości w przedziale $[L, R]$.

Mamy dany ciąg n liczb $a_1, ..., a_n$ z przedziału $[1, M]$. Dostajemy dużo zapytań (L, R), dla każdego musimy podać liczbę spójnych przedziałów naszego ciągu które mają wszystkie wartości w przedziale $[L, R]$.

W zadaniu mieliśmy jeszcze jeden bardzo ważny warunek: elementy ciągu a_i zostały wygenerowane *losowo* (niezależnie i jednostajnie z przedziału $[1, M]$). Dokładniej, użyliśmy do tego funkcji *lrand48* – ale do rozwiązania nie trzeba wiedzieć jak ta funkcja działa.

Spostrzeżenie: jeśli wylosujemy n liczb z $[1, M]$, to prawdopodobieństwo że ostatnia z ich jest największa nie przekracza $\frac{1}{n}$.

Spostrzeżenie: jeśli wylosujemy n liczb z $[1, M]$, to prawdopodobieństwo że ostatnia z ich jest największa nie przekracza $\frac{1}{n}$.

Z tego wynika, że jeśli czytamy n losowych liczb, i utrzymujemy obecne maksimum, to oczekiwana liczba zmian tego maksimum to

$$\sum_{i=1}^{n} \frac{1}{i} = \mathcal{O}(\log n)$$

Spostrzeżenie: jeśli wylosujemy n liczb z $[1, M]$, to prawdopodobieństwo że ostatnia z ich jest największa nie przekracza $\frac{1}{n}$.

Z tego wynika, że jeśli czytamy n losowych liczb, i utrzymujemy obecne maksimum, to oczekiwana liczba zmian tego maksimum to

$$\sum_{i=1}^{n} \frac{1}{i} = \mathcal{O}(\log n)$$

Oczywiście jeśli utrzymujemy parę (min, max), to oczekiwana liczba zmian któregokolwiek komponentu tej pary to również $\mathcal{O}(\log n)$.

Czytamy ciąg od prawej do lewej. Gdy jesteśmy na pozycji i, utrzymujemy stos zawierający takie pozycje j, na których zmienia się maksimum ciągu $a[i, ..., j]$. Analogicznie utrzymujemy drugi stos dla minimum.

Czytamy ciąg od prawej do lewej. Gdy jesteśmy na pozycji i, utrzymujemy stos zawierający takie pozycje j, na których zmienia się maksimum ciągu $a[i, ..., j]$. Analogicznie utrzymujemy drugi stos dla minimum.

Łatwo możemy aktualizować nasze stosy gdy przesuwamy i, a z obserwacji z poprzedniego slajdu ich rozmiar zawsze będzie $\mathcal{O}(\log n)$.

Czytamy ciąg od prawej do lewej. Gdy jesteśmy na pozycji i, utrzymujemy stos zawierający takie pozycje j, na których zmienia się maksimum ciągu $a[i, ..., j]$. Analogicznie utrzymujemy drugi stos dla minimum.

Łatwo możemy aktualizować nasze stosy gdy przesuwamy i, a z obserwacji z poprzedniego slajdu ich rozmiar zawsze będzie $\mathcal{O}(\log n)$.

Stosy pozwalają nam rozważyć wszystkie przedziały o lewym końcu w i, i pogrupować je po parze (min, max) na $\mathcal{O}(\log n)$ grup. Grupując tak dla każdego i otrzymujemy podział wszystkich $\mathcal{O}(n^2)$ podprzedziałów na $\mathcal{O}(n \log n)$ grup (każda grupa ma to samo minimum i maksimum).

Zamiatamy wartości od M do 1. Gdy rozważamy wartość x, to najpierw dodajemy do naszej struktury wszystkie grupy o $min = x$, a następnie odpowiadamy na zapytania o $L = x$.

Zamiatamy wartości od M do 1. Gdy rozważamy wartość x, to najpierw dodajemy do naszej struktury wszystkie grupy o $min = x$, a następnie odpowiadamy na zapytania o $L = x$.

Nasza struktura musi obsługiwać operację dodania grupy o zadanym max i pewnej liczności, oraz zapytania o sumę liczności grup o $max \leq x$. Do tego wystarczy proste drzewo przedziałowe.

Zamiatamy wartości od M do 1. Gdy rozważamy wartość x, to najpierw dodajemy do naszej struktury wszystkie grupy o $min = x$, a następnie odpowiadamy na zapytania o $L = x$.

Nasza struktura musi obsługiwać operację dodania grupy o zadanym max i pewnej liczności, oraz zapytania o sumę liczności grup o $max \leq x$. Do tego wystarczy proste drzewo przedziałowe.

Wykonujemy $\mathcal{O}(n \log n)$ operacji na drzewie, więc złożoność rozwiązania to $\mathcal{O}(n \log^2 n)$.