AiSD, Lab 04, Zadanie k-inwersje (mini omówienie)

Tomasz Waleń

Wydział Matematyki, Informatyki i Mechaniki, UW

Definicja problemu

Dla zadanej permutacji $A = a_1, \ldots, a_n$ i liczby k.

Należy obliczyć liczbę różnych ciągów $i_1, \ldots, i_k$ takich, że:

- $1 \leq i_1 < i_2 < \cdots < i_k \leq n$
- $a_{i_1} > a_{i_2} > \ldots > a_{i_k}$

Ponieważ wynik może być bardzo dużą liczbą, stąd szukamy jego wartości modulo 10^9.

Możliwe rozwiązania

- dla $k = 1$ problem jest prosty, odpowiedzią jest zawsze n,
- dla dużych k (np. $k = 10$) sprawdzanie wszystkich podciągów będzie zbyt wolne (wymaga czasu $O(n^k)$)
- potrzebujemy szybszego rozwiązania niż $O(n^2)$

Programowanie dynamiczne

Niech $T[i][l]$ oznacza liczbę podciągów malejących długości l, których ostatnim elementem jest a_i.

- $T[i][1] = 1$,
- $T[i][l] = \sum_{\substack{j<i \\ a_j>a_i}} T[j][l-1]$.

Wynik to $\sum_{i=1}^{n} T[i][k] \mod 10^9$.

Programowanie dynamiczne dla $k = 2$

- $T[i][1] = 1$,
- $T[i][2] = \sum_{\substack{j<i \\ a_j > a_i}} T[j][1] = \sum_{\substack{j<i \\ a_j > a_i}} 1 = |\{a_j : a_j > a_i \text{ oraz } j < i\}|$.

Czyli $T[i][2]$ możemy policzyć korzystając z algorytmu:

Algorytm 1: (meta) Algorytm wyznaczania $T[i][2]$

$\mathcal{T}$ =pusta struktura
foreach $i = 1, \ldots, n$ **do**
    $T[i][2] = \mathcal{T}$.QUERY($\{$ile jest elementów większych niż $a_i\}$)
    $\mathcal{T}$.MODIFY($\{$dodaj element $a_i\}$)
end

Magiczna struktura $\mathcal{T}$ – nasze oczekiwania

- struktura będzie służyła do utrzymywania dynamicznego ciągu $X = x_1, \ldots, x_n$,
- strukturę inicjalizujemy przy użyciu $\mathcal{T}$.INIT(n) co powoduje utworzenie stałego ciągu $x_1, \ldots, x_n$ takiego, że $x_i = 0$,
- ciąg możemy modyfikować przy użyciu $\mathcal{T}$.MODIFY(i, w), co oznacza $x_i := w$,
- możemy zadawać pytania postaci $\mathcal{T}$.QUERY(l, r), co oznacza $\sum_{i=l}^{r} x_i$
- struktura wymaga inicjalizacji i pamięci $O(n)$, zapytania $modify$ i $query$ są wykonywane w czasie $O(\log n)$,
- struktura spełniające nasze wymagania to drzewo przedziałowe.

Programowanie dynamiczne dla $k = 2$ przy użyciu $\mathcal{T}$

Algorytm 2: Algorytm wyznaczania $T[i][2]$

$\mathcal{T}.\textsc{Init}(n)$
foreach $i = 1, \ldots, n$ **do**
    $T[i][2] = \mathcal{T}.\textsc{Query}(a_i + 1, n)$
    $\mathcal{T}.\textsc{Modify}(a_i, 1)$
end

Programowanie dynamiczne dla dowolnego k przy użyciu $\mathcal{T}$

Algorytm 3: Algorytm wyznaczania $T[i][k]$

$T[i][1] = 1$ dla $i = 1, \ldots, n$
foreach $l \in 2, \ldots, k$ **do**
    $\mathcal{T}$.INIT(n)
    foreach $i = 1, \ldots, n$ **do**
        $T[i][l] = \mathcal{T}$.QUERY$(a_i + 1, n)$
        $\mathcal{T}$.MODIFY$(a_i, T[i][l-1])$
    end
end

Dziękuję za uwagę!