Informatyczny kącik olimpijski (136): *Theater Tickets*

Tym razem omówimy zadanie „Theater Tickets”, które pojawiło się w konkursie *Zinc 2018* organizowanym przez firmę Codility.

Zadanie: *Dany jest n-elementowy ciąg liczb naturalnych $a = (a_1, a_2, \ldots, a_n)$ z przedziału od 1 do n. Oblicz, ile różnych trzyelementowych podciągów występuje w ciągu a? Dwa podciągi uznajemy za różne, jeśli różnią się na przynajmniej jednej pozycji. Przykładowo $(1, 2, 1, 1)$ ma trzy różne podciągi: $(1, 2, 1)$ (występujący dwa razy), $(1, 1, 1)$ (występujący raz) oraz $(2, 1, 1)$ (występujący raz).*

Niech $a_{l:p}$ oznacza podsłowo $a_l, a_{l+1}, \ldots, a_p$.

Rozwiązanie $O(n^3)$
Pierwsze rozwiązanie będzie polegało na prostym zliczeniu podciągów trzyelementowych. Zauważmy, że wszystkich trzyelementowych ciągów o wartościach z przedziału $[1; n]$ jest n^3, gdyż każdy z trzech elementów można wybrać na n sposobów. Niech $T[x][y][z] = 1$, jeśli (x, y, z) jest podciągiem a oraz $T[x][y][z] = 0$ w przeciwnym przypadku. Aby uzupełnić tablicę zliczającą T, wystarczy dla każdej trójki indeksów $1 \leqslant i < j < k \leqslant n$ zaktualizować $T[a_i][a_j][a_k] := 1$. Ta faza zajmuje czas $O(n^3)$. Wynikiem jest liczba komórek tablicy zliczającej T o wartości 1. Rozwiązanie działa w czasie i pamięci $O(n^3)$.

Rozwiązanie $O(n^2)$
Policzmy dla każdego prefiksu, ile zawiera on różnych dwuelementowych podciągów. Niech $P2[i]$ oznacza tę wartość dla i-elementowego prefiksu $a_{1:i}$. Wyniki będziemy obliczali od najkrótszych do najdłuższych prefiksów. Oczywiście $P2[1] = 0$. Zastanówmy się zatem, jak wyznaczyć $P2[i]$ dla $i > 1$. Otóż $P2[i] = P2[i-1] + l$, gdzie l oznacza liczbę takich dwuelementowych podciągów, które nie występują w $a_{1:i-1}$, ale występują w $a_{1:i}$, czyli takich, których drugim elementem jest a_i. Rozważmy więc takie dwuelementowe podciągi (a_j, a_i), że $1 \leqslant j < i$, i zliczmy te, które nie występują w $a_{1:i-1}$. Aby sprawdzić, które dwuelementowe podciągi wystąpiły wcześniej, można, podobnie jak w poprzednim rozwiązaniu, skorzystać z tablicy zliczającej. Wynik dla każdego prefiksu liczymy w czasie $O(n)$, prefiksów jest $O(n)$, zatem $P2$ obliczamy w czasie $O(n^2)$.

Przejdźmy teraz do wyznaczenia liczby różnych trzyelementowych podciągów a. Na początku dla każdej wartości od 1 do n zapamiętajmy numer ostatniej pozycji, na której ta wartość występuje w a. Niech $ost[x]$ oznacza takie największe i, że $a_i = x$. Podciągi będziemy zliczali grupami, biorąc pod uwagę wartość ostatniego elementu. Otóż policzymy, ile jest podciągów, których ostatni element to odpowiednio $1, 2, \ldots, n$, a na koniec zsumujemy te wyniki. Zastanówmy się, jak dla ustalonego z wyznaczyć liczbę różnych podciągów w a postaci (x, y, z) dla $1 \leqslant x, y \leqslant n$. Jeśli z nie występuje w a, to nie ma takich podciągów. W przeciwnym przypadku weźmy ostatnie wystąpienie z w a, które znajduje się na pozycji $ost[z]$. Trzeci element mamy ustalony, zaś dwa pierwsze elementy możemy wybrać na $P2[ost[z]-1]$ sposobów, co jest równe liczbie różnych trzyelementowych podciągów kończących się liczbą z. Całkowita liczba trzyelementowych podciągów to $\sum_{z=1}^{n} P2[ost[z]-1]$, co możemy obliczyć w $O(n)$, znając $P2$. Natomiast całe rozwiązanie działa w czasie i pamięci $O(n^2)$.

Rozwiązanie $O(n)$
Spróbujmy przyspieszyć pierwszą fazę poprzedniego rozwiązania (obliczanie $P2$). Na początku dla każdego prefiksu policzmy, ile zawiera on różnych wartości (podciągów jednoelementowych). Niech $P1[i]$ oznacza tę wartość dla i-elementowego prefiksu, czyli $a_{1:i}$. Wyniki będziemy wyznaczali w kolejności rosnącej długości prefiksów. Oczywiście $P1[i] = 1$ (mamy tylko jeden element). Zastanówmy się teraz, jak wyznaczyć $P1[i]$ dla $i > 1$. Jeśli a_i występowało wcześniej, wtedy $P1[i] = P1[i-1]$. W przeciwnym przypadku $P1[i] = P1[i-1] + 1$. Do sprawdzania, czy a_i występowało wcześniej, możemy wykorzystać tablicę zliczającą.

Przejdźmy teraz do obliczenia $P2$. Podobnie jak wcześniej, wartości tej tablicy będziemy obliczali od najkrótszych do najdłuższych prefiksów. Dodatkowo niech $pop[x]$ dla $1 \leqslant x \leqslant n$ oznacza numer ostatniej pozycji spośród przejrzanych elementów, na której wystąpił x.

Najpierw dla jednoelementowego prefiksu ustawiamy $P2[1] = 0$ (nie ma podciągów dwuelementowych) oraz zapisujemy informację $pop[a_1] = 1$ (ostatnie wystąpienie wartości a_1 na pozycji numer 1). Następnie przeglądamy kolejne prefiksy. Załóżmy, że obliczamy $P2[i]$ dla $i > 1$. Otóż $P2[i]$ to $P2[i-1]$ powiększone o liczbę takich dwuelementowych podciągów $a_{1:i}$, które nie występują w $a_{1:i-1}$. Szukane podciągi są postaci (x, a_i) – drugi element ma wartość a_i. Wszystkich takich podciągów w $a_{1:i}$ jest $P1[i-1]$, gdyż na tyle sposobów można wybrać x. Powinniśmy jednak odjąć te podciągi, które występują również w $a_{1:i-1}$. Jeśli a_i występuje pierwszy raz, wtedy nie musimy nic odejmować. Jeśli natomiast a_i występowało wcześniej, to weźmy jego poprzednie wystąpienie na pozycji $pop[a_i]$. Zauważmy, że jest ono drugim elementem $P1[pop[a_i]-1]$ podciągów (na tyle sposobów można wybrać element stojący przed poprzednim wystąpieniem a_i). Zatem otrzymaliśmy, że: $P2[i] = P2[i-1] + P1[i-1] - P1[pop[a_i]-1]$. Po obliczeniu $P2[i]$ możemy zaktualizować $pop[a_i] = i$. Wyznaczyliśmy $P2$ w czasie $O(n)$. Druga faza algorytmu jest analogiczna do tej opisanej w sekcji *Rozwiązanie* $O(n^2)$ i działa w czasie $O(n)$, co daje nam pełne rozwiązanie o złożoności czasowej $O(n)$.

Bartosz ŁUKASIEWICZ