Zmiennoliterowe słowo II (rozwiązanie)

Autor zadania: **Karol Pokorski**
Opracowanie: **Vladyslav Rachek, Mateusz Olszewski**
Opis rozwiązania: **Lech Duraj**

Olimpiada Informatyczna Juniorów

W zadaniu dane jest słowo S długości $N \leq 1\,000\,000$ składające się wyłącznie z małych liter alfabetu angielskiego. Dla tego słowa musimy wyznaczyć dwie liczby:

1. Liczbę trójek (i, j, k) takich że $i < j < k$ oraz $T = S[i]S[j]S[k]$ jest zmiennoliterowym słowem.
2. Liczbę różnych zmiennoliterowych podsłów $T = S[i]S[j]S[k]$ słowa S.

Do rozwiązania będziemy potrzebowali pomocniczej tablicy P określonej następująco: $P[\alpha][j]$ to liczba wystąpień α-tej litery w początkowym fragmencie słowa $S[1..j]$. Zakładamy przy tym, że litery numerujemy od 0 – litera `a` to 0, `b` to 1 itd. Na przykład dla słowa `abacaddd` mamy $P[0][4] = 2$, co oznacza, że litera `a` występuje 2 razy wśród pierwszych 4 znaków słowa. Dla wygody przyjmujemy też $P[\alpha][0]$ dla każdego α. Jest to tak zwana *tablica sum prefiksowych*, opisywana już w rozwiązaniu zadania *„... albo psikus!"* w turze otwartej.

Aby teraz rozwiązać punkt 1. naszego zadania, weźmy pewną literę $S[j]$ w słowie S i zastanówmy się, na ile sposobów można dobrać wartości $i < j$ oraz $k > j$ tak, aby $S[i] \neq S[j]$ oraz $S[j] \neq S[k]$. Rozważenie wszystkich możliwych i oraz k to czas $O(N)$, co dla wszystkich j dawałoby łącznie $O(N^2)$ – za dużo. Na szczęście, dzięki tablicy P możemy znacznie przyśpieszyć tę część algorytmu.

Niech $S[j] = \alpha$. Jako wartość i możemy dobrać dowolną spośród liter $S[1], \ldots, S[j-1]$, która *nie* jest równa α – ale skoro wszystkich liter jest $j-1$, a liter α jest $P[\alpha][j-1]$, to prawidłowych wyborów jest dokładnie $L_j = j - 1 - P[\alpha][j-1]$. Z kolei wyborów k takich, żeby $S[k] \neq \alpha$ jest $R_j = (N - j) - (P[\alpha][N] - P[\alpha][j])$, jako że wszystkich liter po j-tej jest $N - j$, a wśród nich jest $P[\alpha][N] - P[\alpha][j]$ liter α. Podsumowując, dla każdego $j = 1, 2, \ldots, N$ liczba trójek (i, j, k) tworzących słowo zmiennoliterowe równa jest $L_j \cdot R_j$, a obie te wielkości możemy wyznaczyć w czasie stałym mając tablicę P. Po zsumowaniu wyników dla wszystkich j dostajemy poszukiwaną wartość.

Czas na punkt 2. Spróbujmy rozważyć wszystkie możliwe zmiennoliterowe słowa (`aba`, `abc`, `abd`, ...), których jest $26 \cdot 25 \cdot 25 = 16250$. Dla każdego z nich mamy możliwość rozstrzygnięcia w czasie liniowym, czy jest podciągiem S – taki algorytm też już wystąpił na tej Olimpiadzie, w zadaniu *„OIJ"*. Rozwiązanie to działa w czasie liniowym, wywołanie go dla każdego trójliterowego słowa to $16250 \cdot N$ operacji (formalnie, $O(N \cdot \Sigma^3)$, gdzie Σ to rozmiar używanego alfabetu).

Wciąż możemy jednak obniżyć złożoność. Zauważmy, że jeśli szukamy np. słowa `cde`, to warunek „`cde` występuje gdzieś w S" możemy sformułować inaczej: „pomiędzy pierwszym wystąpieniem `c` oraz ostatnim wystąpieniem `e` da się znaleźć co najmniej jedną literę `d`". Istotnie, skoro `cde` jest podciągiem S, to litera `d` da się znaleźć pomiędzy *jakimś* wystąpieniem `c`, a *jakimś* wystąpieniem `e`, więc tym bardziej pomiędzy pierwszym `c` a ostatnim `e`. Zatem możemy próbować ułożyć algorytm następująco:

- Dla każdej litery α znajdujemy indeks jej pierwszego wystąpienia $F[\alpha]$ w słowie S – jeśli α nie występuje w S, to $F[\alpha] = -1$.
- Analogicznie, dla każdej litery znajdujemy indeks jej ostatniego wystąpienia $L[\alpha]$ w S (lub -1, jeśli nie występuje).
- Dla każdej trójki liter (α, β, γ) takiej, że $\alpha \neq \beta$ i $\beta \neq \gamma$ sprawdzamy, czy między indeksami $i = F[\alpha]$ i $j = L[\gamma]$ choć raz występuje β – mając tablicę P, wystarczy po prostu sprawdzić, czy $P[\beta][i] < P[\beta][j]$.

Szukanie pierwszego wystąpienia możemy dla wszystkich liter rozwiązać jedną pętlą: wpisujemy dla wszystkich liter wartość F równą -1, po czym analizujemy kolejne znaki S od początku. Dla $j = 1, 2, \ldots, N$ jeśli litery $\alpha = S[j]$ jeszcze nie widzieliśmy (czyli $F[\alpha] = -1$), wpisujemy $F[\alpha] = j$. Analogicznie prosto rozwiązujemy dla wszystkich liter wyznaczanie wartości $L[]$. Obie pętle działają w czasie $O(N)$, a ostatni punkt – iterowanie się po trójkach liter – w czasie $O(\Sigma^3)$. Łączna złożoność to zatem $O(N + \Sigma^3)$.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".