Informatyczny kącik olimpijski (138): *Yet Another Substring Reverse*

Tym razem omówimy zadanie *Yet Another Substring Reverse*, które pojawiło się na portalu `www.codeforces.com`.

Zadanie: *Dane jest słowo $s = s_1 s_2 \ldots s_n$ o długości n, zawierające wystąpienia części liter alfabetu angielskiego A. W słowie s należy odwrócić jedno podsłowo, tzn. zastąpić je zapisem od prawej do lewej, aby otrzymać możliwie najdłuższe różnorodne podsłowo, czyli takie, które nie zawiera powtórzeń liter. Jaka jest długość najdłuższego różnorodnego podsłowa, które można uzyskać? Przykładowo dla $s =$ „bbaacc” wynikiem jest 3. Po odwróceniu fragmentu od trzeciej do szóstej litery otrzymujemy „bbacca”, którego najdłuższy różnorodny fragment to „bac”.*

Niech $s_{l:p}$ oznacza podsłowo $s_l s_{l+1} \ldots s_p$.

Na samym początku zauważmy, że długość różnorodnego podsłowa nie przekroczy $|A|$.

Najdłuższe różnorodne podsłowo $O(n|A|)$
Załóżmy przez chwilę, że nie wykonujemy odwrócenia i chcemy znaleźć długość najdłuższego różnorodnego podsłowa. Dla każdej pozycji początkowej wyznaczmy najdłuższe różnorodne podsłowo, które się w niej zaczyna. Formalnie dla ustalonego $1 \leqslant i \leqslant n$ znajdźmy takie największe k_i, że $s_{i:k_i}$ jest różnorodne. Zacznijmy od jednoliterowego podsłowa s_i i rozszerzajmy je o kolejne litery, dopóki nie nastąpi powtórzenie. Do sprawdzania, czy jakaś litera wystąpiła wcześniej, można wykorzystać tablicę zliczającą, którą po każdym rozszerzeniu aktualizujemy. Opisane rozwiązanie działa w czasie $O(n|A|)$, gdyż od każdej pozycji początkowej wykonamy co najwyżej $|A|$ rozszerzeń w prawo.

Najdłuższe różnorodne podsłowo $O(n + |A|)$
W tym rozwiązaniu, podobnie jak wcześniej, dla każdej pozycji $1 \leqslant i \leqslant n$ chcemy wyznaczyć takie największe k_i, że $s_{i:k_i}$ jest różnorodne. Skorzystajmy z metody gąsienicy. Otóż, tym razem nie będziemy dla każdej pozycji początkowej rozpoczynali przeszukiwania od słów jednoliterowych. Zauważmy, że jeśli $s_{i-1:k_{i-1}}$ jest różnorodne, to również $s_{i:k_{i-1}}$ jest różnorodne. Zatem nie ma potrzeby sprawdzania słów, które kończą się wcześniej niż k_{i-1}. Do sprawdzania powtórzeń wykorzystujemy tablicę zliczającą. Przy rozszerzaniu słowa z prawej strony dodajemy literę, a przy przejściu do kolejnej pozycji początkowej usuwamy literę z tablicy zliczającej. Rozwiązanie działa w czasie $O(n + |A|)$, gdyż każda litera zostanie dokładnie raz dodana i raz usunięta z tablicy zliczającej.

Rozwiązanie $O(n^2(n + |A|))$
Opisaliśmy metody znajdowania najdłuższego różnorodnego podsłowa, zatem możemy przejść do pierwszego rozwiązania zadania. Rozważmy odwrócenie każdego z $\frac{n(1+n)}{2}$ podsłów. Dla każdego przypadku znajdźmy najdłuższe różnorodne podsłowo za pomocą jednej z wyżej opisanych metod i wybierzmy najlepszy wynik. W zależności od wykorzystanej metody otrzymamy rozwiązanie $O(n^3|A|)$ lub $O(n^2(n + |A|))$.

Rozwiązanie $O(n^2|A|^3)$
Zauważmy, że możemy zredukować zadanie do znalezienia dwóch różnorodnych podsłów, które mają rozłączne zbiory liter oraz największą sumaryczną długość. Formalny dowód tego faktu pozostawiamy Czytelnikowi, a teraz pokażemy, jak dwa podsłowa, $s_{l_1:p_1}$ oraz $s_{l_2:p_2}$, niemające wspólnych liter umieścić obok siebie za pomocą jednego odwrócenia. Załóżmy bez straty ogólności, że $s_{l_1:p_1}$ występuje na lewo od $s_{l_2:p_2}$, wówczas należy odwrócić $s_{p_1+1:p_2}$. Teraz $s_{l_1:p_1+p_2-l_2+1}$ jest różnorodne.

Znajdźmy wszystkie różnorodne podsłowa, których jest $O(n|A|)$. Następnie rozważmy każdą z $O(n^2|A|^2)$ par takich podsłów. Jeśli podsłowa mają rozłączne zbiory liter, wtedy razem tworzą różnorodne podsłowo. Sprawdzenie, czy zbiory liter są rozłączne, możemy wykonać metodą zliczenia w czasie $O(|A|)$. Spośród par tworzących różnorodne podsłowa wybieramy tę, której sumaryczna długość fragmentów jest największa. Otrzymaliśmy rozwiązanie o złożoności czasowej $O(n^2|A|^3)$.

Rozwiązanie $O(|A|(n|A| + 2^{|A|}))$
Zauważmy, że kolejność liter w podsłowie nie ma znaczenia, zatem każde różnorodne podsłowo możemy rozważać jako zbiór liter. Spośród wszystkich podsłów o długości co najwyżej $|A|$ znajdźmy te, które są różnorodne, i zbiory ich liter oznaczmy jako zbiór R. Przykładowo dla $s =$ „bab”, $R = \{\{'a'\}, \{'b'\}, \{'a', 'b'\}\}$. Wyznaczenie R zajmuje czas $O(n|A|^2)$, zaś moc R wynosi co najwyżej $2^{|A|}$. Zadanie redukuje się teraz do znalezienia dwóch rozłącznych zbiorów w R, których suma ma największą moc.

Dla każdego zbioru $r \in R$ znajdźmy taki zbiór $r' \in R$, że $r \cap r' = \emptyset$ oraz $|r \cup r'|$ jest maksymalne. Innymi słowy dla każdego zbioru znajdźmy inny zbiór, który tworzy z nim najlepszy wynik. Wówczas odpowiedzią w zadaniu jest maksimum z rozważonych przypadków. Niech $W[a]$ dla każdego $a \subseteq A$ oznacza moc największego takiego zbioru $r \in R$, że $r \subseteq a$. Teraz $r \in R$ tworzy najdłuższe różnorodne podsłowo o długości $|r| + W[\overline{r}]$, gdzie $\overline{r}$ oznacza dopełnienie r.

Pozostało nam jeszcze wyznaczyć wartości W, które będziemy obliczali w kolejności niemalejących mocy zbiorów. Załóżmy, że obliczamy $W[a]$ i mamy obliczone $W[b]$ dla wszystkich $b \subseteq a$. Jeśli $a \in R$, wówczas $W[a] = |a|$. W przeciwnym przypadku $W[a] < |a|$, czyli największy podzbiór a należący jednocześnie do R nie zawiera jakiegoś elementu a, zatem $W[a] = \max_{e \in a} W[a \setminus e]$. Obliczenie $W[a]$ zajmuje czas $O(|A|)$, wartości tablicy W jest $O(2^{|A|})$, zatem czas potrzebny do obliczenia W wynosi $O(2^{|A|}|A|)$. Całe rozwiązanie działa w czasie $O(|A|(n|A| + 2^{|A|}))$.

Bartosz ŁUKASIEWICZ