Algorytm SA-IS

Notacja

Jako S będziemy oznaczać słowo, którego tablicę sufiksową mamy obliczyć. Przyjmujemy, że S kończy się znakiem \$, który nie występuje nigdzie indziej w słowie i w porządku leksykograficznym jest najmniejszy.

Dodatkowo niech n będzie długością słowa S, a znaki będą numerowane od 1. Wtedy $S_i = S[i..n]$ dla $i \in \{1, ..., n\}$ jest sufiksem S rozpoczynającym się na i-tej pozycji.

Algorytm

Ogólny sposób działania SA-IS polega na sortowaniu sufiksów na podstawie tablicy sufiksowej krótszego słowa S', obliczanej rekurencyjnie. Słowo S' powstaje przez wybór pewnych podsłów S, i przepisaniu każdego z nich na pojedynczy znak zachowując kolejność leksykograficzną. Aby osiągnąć czas $O(n)$, obie te czynności wykonywane są za pomocą *sortowania indukowanego*.

Aby przedstawić dokładne działanie algorytmu, koniecznie są następujące pojęcia:

Definicja 1. *S_i jest sufiksem typu S (typu L) gdy $S_i < S_{i+1}$ ($S_i > S_{i+1}$).*
$S_n = \$$ jest typu S.
Typ znaku $S[i]$ określany jest jako ten sam co typ sufiksu S_i.

Definicja 2. *Znak $S[i]$, $i \in \{2, ..., n\}$, jest LMS-znakiem (leftmost S-type) gdy $S[i-1]$ i $S[i]$ są odpowiednio typu L i typu S.*

Definicja 3. *LMS-podsłowo to 1. $S[n] = \$$; lub 2. podsłowo $S[i..j]$, gdzie $S[i]$, $S[j]$ to kolejne LMS-znaki w S.*

Podsłowa początkowo wybierane z S to właśnie LMS-podsłowa. Sortowanie indukowane zarówno dla LMS-podsłów (celem zredukowania podsłów do S'), jak i sufiksów S, działa opierając się na podziale sufiksów na typ S i typ L.

Całość algorytmu zawiera się w poniższych krokach:

1. Określ typ każdego z sufiksów S
2. Wyznacz LMS-podsłowa S

3. Posortuj znalezione LMS-podsłowa za pomocą sortowania indukowanego

4. Przepisz każde LMS-podsłowo na literę odpowiadającą jego pozycji w porządku i utwórz z nich słowo S' zachowując kolejność występowania w S

5. Oblicz tablicę sufiksową SA' słowa S'

6. Wyznacz tablicę sufiksową SA słowa S za pomocą sortowania indukowanego na podstawie SA'

W kroku 5, jeżeli nie występują dwa identyczne LMS-podsłowa (wszystkie znaki S' są unikalne), SA' można uzyskać bezpośrednio jako odwrotność permutacji reprezentowanej przez S'; w przeciwnym przypadku, jest ona pozyskiwana rekurencyjnie.

Wyznaczanie LMS-podsłów

Typ każdego z sufiksów można ustalić, przeglądając S od końca:

1. \$ jest typu S

2. jeśli $S[i] = S[i+1]$, typ S_i jest ten sam co S_{i+1}

3. wpp. $S[i] \neq S[i+1]$, a typ S_i zależy bezpośrednio od nich.

Tablica przechowująca typy sufiksów S pozwala wskazać pozycje, na których zaczynają się kolejne LMS-podsłowa. W dalszej części algorytmu LMS-podsłowa będą reprezentowane przez indeks ich pierwszego znaku.

Redukcja do mniejszego problemu

Na LMS-podsłowach jest zdefiniowany następujący porządek: porównując parami kolejne znaki obu podsłów, jeśli się różnią, decyduje ich kolejność leksykograficzna; w przepiwnym przypadku znak typu L jest mniejszy od znaku typu S. Ten porządek odpowiada temu, że jeśli S_i jest typu S, S_j typu L, i pierwsze litery są sobie równe, to $S_j < S_i$.

Sortowanie indukowane pracuje na wynikowej tablicy SA, podzielonej na kubełki dla każdego znaku w S. W trakcie sortowania wyłania się dodatkowy

podział każdego kubełka na te przenaczone dla sufiksów różnego typu w kolejności L, S; ale nie jest on reprezentowany bezpośrednio. Dla każdej litery w tablicy B rozmiaru alfabetu przechowywany jest wskaźnik na pewne miejsce w odpowiadającym kubełku. Wskazywane miejsce zależy od etapu sortowania.

Samo sortowanie odbywa się w trzech krokach:

1. wyznacz koniec każdego z kubełków. Wstaw każde z LMS-podsłów do jego kubełka (odp. pierwszemu znakowi podsłowa);
2. wyznacz początek każdego z kubełków. Dla każdego $i = 1, ..., n$: jeśli $c = S[SA[i] - 1]$ jest typu L, wstaw $SA[i] - 1$ do kubełka c;
3. wyznacz koniec każdego z kubełków. Dla każdego $i = n, ..., 1$: jeśli $c = S[SA[i] - 1]$ jest typu S, wstaw $SA[i] - 1$ do kubełka c.

Dokładniej, wyznaczenie początku (końca) kubełka c oznacza zapisanie odpowiedniego indeksu do $B[c]$, natomiast wstawianie elementu odbywa się przez zapisanie go do $SA[B[c]]$ i przesunięcie $B[c]$ o jeden w lewo (prawo).

S' jest utworzone przez ułożenie LMS-podsłów tak, jak występują w S i zamianę każdego z nich na kolejne liczby zgodnie z porządkiem uzyskanym z powyższego sortowania (identyczne podsłowa otrzymują tę samą liczbę).

Sortowanie wszystkich sufiksów

Procedura odtworzenia tablicy sufiksowej S przez sortowanie indukowane jest niemal identyczna z sortowaniem LMS-podsłów. Jedyna różnica występuje w pierwszym kroku: na końce kubełków należy wstawić LMS-podsłowa (indeksy pierwszych znaków) w zgodzie z kolejnością uzyskaną w SA'. Po jego wykonaniu zawartość SA jest tablicą sufiksową S.

Poprawność sortowania

Rozważmy ostatni krok całego algorytmu:

Lemat 1. *Mając dane posortowane wszystkie sufiksy typu L, krok 3 sortuje wszystkie sufiksy S w czasie $O(n)$.*

Dowód. Pokażemy przez indukcję, że w momencie przeglądania $SA[i]$, $S_{SA[i]}$ został już zapisany na swoim miejscu.

Dla $i = n$, największy sufiks musi być typu L, zatem był posortowany w kroku 2.

Dla $i < n$, gdy wszystkie $SA[i+1], ..., SA[n]$ są uzupełnione poprawnie, załóżmy że sufiks, który powinien znajdować się w $SA[i]$ leży pod $SA[k]$, $k < i$. Ponieważ sufiksy typu L zostały posortowane, w $SA[i]$, $SA[k]$ znajdują się sufiksy typu S. Sufiksy są już w swoich kubełkach, więc $SA[i] = c\alpha$, $SA[k] = c\beta$ dla jakiegoś c. $c\beta < \beta < \alpha$, zatem poprawne pozycje β, α znajdują się wśród już wstawionych i przeglądniętych $SA[i+1], ..., SA[n]$, a $c\alpha$ powinno było zostać wstawione na koniec swojego kubełka przed $c\beta$, co jest sprzeczne z obecnym stanem tablicy SA. □

Gdyby najpierw posortować sufiksy typu S, a następnie wykonać krok 2, wszystkie sufiksy typu L również zostały by posortowane, co można wykazać w analogiczny sposób. Ze względu na to, że tylko LMS-sufiksy biorą udział w ich sortowaniu, wystarczy zacząć od wstawienia do SA LMS-sufiksów w kolejności leksykograficznej, aby uzyskać ten sam rezultat. Tym samym, o ile tablica SA' wyznacza poprawny porządek LMS-sufiksów, ostatni etap algorytmu jest poprawny.

Niech $P[i]$ wskazuje na początek i-tego LMS-podsłowa. Wtedy

Lemat 2. *$S'_i < S'_j$ jest równoważne z $S_{P[i]} < S_{P[j]}$.*

Dowód. Jeśli $S'[i] \neq S'[j]$, to znak świadczący o różnicy w odpowiadających im LMS-podsłowach wyznacza kolejność sufiksów zaczynających się na tych samych pozycjach, natomiast porządek na LMS-podsłowach jest zgody z porzadkiem leksykograficznym na sufiksach. W przeciwynm wypadku, zauważmy, że $S'[i] = S'[j]$ oznacza równą długość odpowiednich LMS-podsłów, zatem możliwe jest przeprowadzenie powyższego rozumowania na pierwszym różniącym się znaku sufiksów w S'. □

Pozostaje wykazać, że pierwsze użycie sortowania indukowanego sortuje LMS-podsłowa. Mechanizm dowodu jest podobny, z kilkoma różnicami: w pierwszym kroku chcemy otrzymać poprawny porządek na LMS-podsłowach obciętych do pierwszego znaku. Wystarczy, że zostaną wstawione do odpowiednich kubełków, co zapewnia krok 1. Te z kolei wystarczą do posortowania sufiksów typu L obciętych do pierwszeko LMS-znaku w kroku 2. Analogicznie dzieje się dla sufiksów typu S w kroku 3, z wyjątkiem LMS-sufiksów, które są rozważane do miejsca wystąpiena *drugiego* LMS-znaku, czyli LMS-podsłów.

Złożoność

Zarówno wyznaczanie LMS-podsłów, jak i sortowanie indukowane działa w oczywisty sposób w czasie $O(n)$. Ponieważ LMS-podsłów jest nie więcej niż połowa długości S (z wyjątkiem \$ w środku każdego z nich musi się znajdować przynajmniej jeden znak typu L), całkowity czas wykonania można określić równaniem $T(n) = T(\lfloor n/2 \rfloor) + O(n)$, które daje $O(n)$.