O rozkładzie słów na słowa Lyndona

Łukasz GRZĄDKO[*]

W tym artykule rozwiążemy problem rozkładu słowa na najmniejszą liczbę słów Lyndona (zwanych też słowami pierwszymi). Problem ten jest inspirowany zadaniem *Jan* z pierwszej edycji Potyczek Algorytmicznych, która odbyła się w roku 2005. Jakub Radoszewski w artykule *Słowa pierwsze*, *Delta* 12/2010, podał kluczowe własności słów Lyndona, które wykorzystamy podczas konstrukcji algorytmu. Przypomnijmy, że słowo Lyndona to takie słowo, że przy dowolnym jego obrocie cyklicznym zawsze otrzymuje się słowo późniejsze od niego leksykograficznie. Obrót cykliczny polega na przeniesieniu początkowego fragmentu słowa na jego koniec, np. dla słowa `baca` mamy następujące obroty: `acab`, `caba`, `abac`. Oczywiście słowo `baca` nie jest słowem Lyndona, gdyż `acab` występuje wcześniej w słowniku. Słowo `abac` jest już słowem Lyndona.

Słowo `abaabab` również nie jest słowem Lyndona, ale można je rozłożyć na słowa Lyndona. Trywialny jest rozkład na słowa jednoliterowe $\mathtt{a}\cdot\mathtt{b}\cdot\mathtt{a}\cdot\mathtt{a}\cdot\mathtt{b}\cdot\mathtt{a}\cdot\mathtt{b}$ (przy czym $\cdot$ oznacza sklejenie słów), ale istnieją też bardziej oszczędne rozkłady, np. $\mathtt{ab}\cdot\mathtt{aab}\cdot\mathtt{ab}$ lub $\mathtt{ab}\cdot\mathtt{aabab}$.

Dalej skorzystamy z trzech twierdzeń. Opisują one znane własności słów Lyndona. Dowody tych twierdzeń można znaleźć w przywołanym artykule z *Delty* 12/2010.

Twierdzenie 1. *Słowo s jest słowem Lyndona wtedy i tylko wtedy, gdy jest leksykograficznie wcześniejsze niż dowolny właściwy sufiks s.*

Twierdzenie 2. *Słowo s jest słowem Lyndona wtedy i tylko wtedy, gdy s jest jednoliterowe lub istnieją słowa Lyndona s_1 oraz s_2, takie że $s = s_1 s_2$ oraz $s_1 < s_2$.*

Twierdzenie 3. *Dowolne słowo s przedstawia się jednoznacznie jako sklejenie pewnej liczby słów Lyndona $s = l_1 l_2 \ldots l_k$, takich że $l_1 \geqslant l_2 \geqslant \ldots \geqslant l_k$. Będziemy też pisać, że rozkład ten spełnia warunek monotoniczności.*

Zauważmy, że w naszym problemie nie ma wymagania, że słowa Lyndona w rozkładzie mają spełniać warunek monotoniczności $l_1 \geqslant l_2 \geqslant \ldots \geqslant l_k$, tylko że ich liczba k ma być jak najmniejsza. Okazuje się jednak, że jeśli znajdziemy rozkład spełniający warunek monotoniczności, to k będzie najmniejsze i *vice versa*:

Dowód. ($\Rightarrow$) Przypuśćmy, że mamy rozkład $s = l_1 l_2 \ldots l_k$ długości k spełniający warunek monotoniczności $l_1 \geqslant l_2 \geqslant \ldots \geqslant l_k$, ale k nie jest najmniejsze. Wtedy istnieje optymalny rozkład $s = l'_1 l'_2 \ldots l'_m$ długości $m < k$. Rozkład długości m nie może spełniać warunku $l'_1 \geqslant l'_2 \geqslant \ldots \geqslant l'_m$, gdyż z jednoznaczności (twierdzenie 3) jest tylko jeden taki rozkład. Stąd dla pewnego $i < m$ mamy $l'_i < l'_{i+1}$. Ale wtedy można takie słowa skleić w jedno słowo Lyndona l'' (twierdzenie 2) i otrzymamy rozkład $s = l'_1 l'_2 \ldots l'_{i-1} l'' l'_{i+2} \ldots l'_m$ o długości $m-1$, co jest sprzeczne z minimalnością m. Uzyskaliśmy sprzeczność, skąd wynika, że k jest najmniejsze.

($\Leftarrow$) Chcemy wykazać, że jeśli k jest najmniejsze oraz $s = l_1 l_2 \ldots l_k$ jest rozkładem słowa s na słowa Lyndona, to $l_1 \geqslant l_2 \geqslant \ldots \geqslant l_k$. Przypuśćmy, że k jest najmniejsze, lecz rozkład ten nie spełnia warunku monotoniczności, czyli dla pewnego i mamy $l_i < l_{i+1}$. Z twierdzenia 2 możemy otrzymać krótszy rozkład, długości mniejszej niż k. Otrzymana sprzeczność dowodzi tezy. $\triangle$

Rozwiązując zatem nasz problem, będziemy konstruować rozkład $s = l_1 l_2 \ldots l_k$ spełniający warunek monotoniczności $l_1 \geqslant l_2 \geqslant \ldots \geqslant l_k$.

Jak znaleźć słowo l_i? Bez straty ogólności możemy się skupić na wyznaczeniu słowa l_1. Jeśli s jest słowem Lyndona, to wystarczy przyjąć $l_1 = s$.
W przeciwnym przypadku rozważamy kolejne prefiksy s w kolejności malejących długości. Kiedy pewien prefiks s jest słowem Lyndona, znaleźliśmy l_1

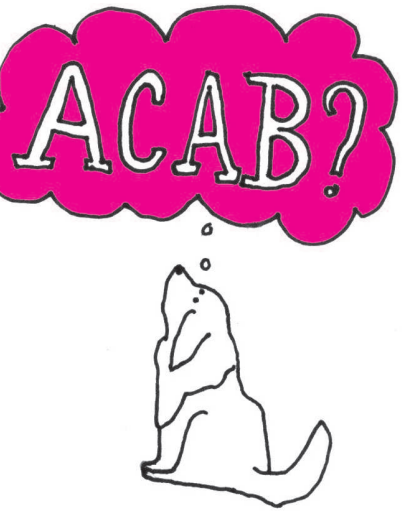
Niech s będzie tablicą, w której zapiszemy badane słowo. Każda kolejna komórka s będzie zawierać kolejną literkę tego słowa. Np. dla $s = \mathtt{abaabab}$, $s[0] = \mathtt{a}$, $s[6] = \mathtt{b}$. Definiujemy również długość słowa s jako $|s|$. Dla powyższego słowa $|s| = 7$.

Przez $s[i..j]$ oznaczymy podsłowo, czyli spójny fragment słowa zawierający litery występujące od pozycji i do pozycji j. Dla przykładu $s[1..3] = \mathtt{baa}$.

Jeśli $i = 0$, to podsłowo nazwiemy prefiksem słowa s, a jeśli $j = |s| - 1$, to nazwiemy je sufiksem. Jeśli nie zachodzi jednocześnie $i = 0$ i $j = |s| - 1$, to dany prefiks (sufiks) nazwiemy właściwym.

Dla dwóch słów s i t powiemy, że s jest wcześniejsze niż t w porządku leksykograficznym (co oznaczymy przez $s < t$), jeśli s jest prefiksem właściwym t lub dla pewnego i jest $s[i] < t[i]$ oraz dla wszystkich $j < i$ zachodzi $s[j] = t[j]$. Dla przykładu $\mathtt{ab} < \mathtt{abc}$, $\mathtt{abc} < \mathtt{ac}$, $\mathtt{ac} < \mathtt{b}$.

Obrotem cyklicznym słowa s długości n jest każde słowo postaci $s(k) = s[k..n-1]s[0..k-1]$. Słowo s jest słowem Lyndona, jeśli $s < s(k)$ dla wszystkich $1 \leqslant k < n$.



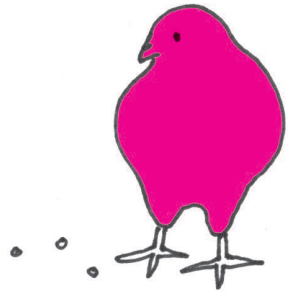
[*]pracownik firmy Nokia

Rozwiązanie zadania F 872.
Energia wiązania sześcianu o boku o długości L zawierającym N cząsteczek wynosi $3\varepsilon N^3$, bo każda cząsteczka ma 6 najbliższych sąsiadów i z każdym dzieli jedno wiązanie. Do oddzielenia dwóch sąsiednich warstw cząsteczek potrzebna jest energia εN^2, ale po rozdzieleniu powstają dwie kwadratowe powierzchnie swobodne cieczy o boku o długości L każda. Mamy więc $\rho L_p L^3 = 3\varepsilon N^3$ oraz $2\gamma L^2 = \varepsilon N^2$. Stąd już łatwo wyznaczamy:

$$\frac{N}{L} = \frac{\rho L_p}{6\gamma}.$$

Po podstawieniu danych liczbowych otrzymujemy $(N/L)^3 \approx (6{,}4 \cdot 10^7/\mathrm{cm})^3 \approx \approx 2{,}6 \cdot 10^{23}$ cząsteczek w cm^3. Dokładna wartość to 1/18 liczby Avogadro, czyli $3{,}345 \cdot 10^{22}$ cząsteczek w cm^3, a więc otrzymaliśmy niezły wynik.

Najsłabszym ogniwem naszego rozumowania jest założenie o charakterze wiązań (więcej w artykule K. Rejmera „Wiązania wodorowe" w *Delcie* 5/2014). Lepsze oszacowanie otrzymujemy dla substancji tworzącej kryształy o wiązaniach kowalencyjnych, jak np. krzem (Si) o $L_p = 13{,}7$ J/g, $\gamma = 1{,}41$ N/m – dla powierzchni (w 100°C) – oraz $\rho = 2{,}32$ g/cm^3, dla którego otrzymujemy wartość $5{,}3 \cdot 10^{22}$ atomów w cm^3 wobec dokładnej wartości $2{,}151 \cdot 10^{22}$ – tu rozbieżność wynika z nieco bardziej skomplikowanej postaci sieci krystalicznej niż przyjęta w naszych obliczeniach.



```
p := 0;
j := 1;
while p < n do
  if s[j] > s[p] then
    j := j + 1
  else if s[j] = s[p] then
    i := p;
    while s[j] = s[i] do
      i := i + 1;
      j := j + 1;
    if s[j] < s[i] then
      m := (j − p) div (j − i);
      wypisz m razy podsłowo
                  s[p..p + j − i − 1];
      p := p + (j − i) · m;
      j := p + 1;
    else
      j := j + 1;
  else
    wypisz podsłowo s[p..j − 1];
    p := j;
    j := j + 1;
```

i dostajemy rozkład $s = l_1 t$. Zauważmy, że prefiks jednoliterowy jest zawsze słowem Lyndona, więc ten proces na pewno zakończy się powodzeniem. Powtarzając powyższe rozumowanie dla słowa t, wyznaczamy l_2 itd.

Opisany proces wyznacza nam rozkład słowa s na słowa Lyndona $s = l_1 l_2 \ldots l_k$. A co z warunkiem monotoniczności? Przypuśćmy, że dla pewnego i jest $l_i < l_{i+1}$. Skoro algorytm znalazł słowo l_i, szukając od „końca", nie mógłby pominąć słowa $l_i l_{i+1}$, które (z twierdzenia 2) jest dłuższym słowem Lyndona. Dlatego dla każdego i mamy $l_i \geqslant l_{i+1}$.

Otrzymany proces podziału na słowa Lyndona jest jednakże nieefektywny. Sprawdzenie „siłowe", czy podsłowo długości m jest słowem Lyndona, zajmuje czas z grubsza $O(m^2)$. Wyznaczenie wszystkich słów l_i będzie wymagać n takich sprawdzeń, zatem powyższy algorytm działa w czasie $O(n^3)$. Pokażemy, że nasz problem można rozwiązać w istotnie lepszej złożoności czasowej.

Wprowadzimy w tym celu trzy lematy pomocnicze.

Lemat pomarańczowy. *Jeśli $s[i..j]$ jest słowem Lyndona dla pewnych $i \leqslant j$ oraz $s[j+1] > s[i]$, to $s[i..j+1]$ jest słowem Lyndona.*

Dowód. Z tego że $s[i] < s[j+1]$ oraz z definicji porządku leksykograficznego wynika, że $s[i..j] < s[j+1]$. Dalej to i twierdzenie 2 implikują, że $s[i..j+1]$ jest słowem Lyndona. △

Lemat kurczaczkowy. *Niech $s = pay$, $t = pb$, gdzie p jest wspólnym prefiksem słów s i t, natomiast a i b są pojedynczymi literami spełniającymi $a < b$. Jeśli s jest słowem Lyndona, to również t jest słowem Lyndona.*

Dowód. Przypuśćmy, że t nie jest słowem Lyndona. Wtedy na podstawie twierdzenia 1 mamy dla pewnego $0 < i \leqslant |p|$, że $p[0..|p|-1]b \geqslant p[i..|p|-1]b$ (jeśli $i = |p|$, to słowo $p[i..|p|-1]$ jest puste). Weźmy najmniejszy indeks $k < |p| - i$ taki, że $p[k] > p[i+k]$. Wtedy dla $j < k$ zachodzi $p[j] = p[i+j]$. Ale to przeczy temu, że s jest słowem Lyndona. W przypadku gdy nie ma takiego k, wtedy $p[|p|-i] > b > a$ oraz $p[j] = p[i+j]$ dla $j < |p| - i$. To również przeczy temu, że s jest słowem Lyndona.

Otrzymane sprzeczności dowodzą, że t jest słowem Lyndona. △

Lemat winogronowy. *Jeśli $s[i..j]$ jest słowem Lyndona oraz $s[i] > t$ dla pewnego niepustego słowa t, to $s[i..j]t$ nie jest słowem Lyndona.*

Dowód. Ponieważ $s[i..j]t > s[i]$ oraz $s[i] > t$, więc $s[i..j]t > t$. Z twierdzenia 1 wynika, że $s[i..j]t$ nie jest słowem Lyndona. △

Jesteśmy już gotowi do przedstawienia lepszego algorytmu podziału słowa s na minimalną liczbę słów Lyndona. Na marginesie przedstawiliśmy jego zapis w pseudokodzie. Dla uproszczenia załóżmy, że słowo s kończy się dodatkową literą (strażnikiem) mniejszą leksykograficznie od pozostałych liter słowa s.

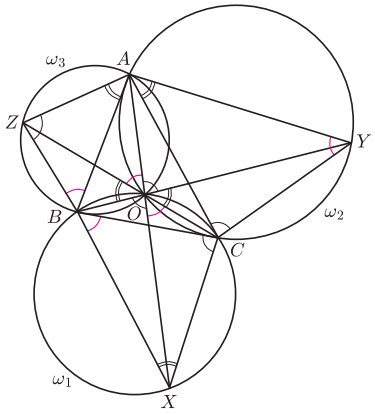
Zmienna p używana w algorytmie oznacza początek aktualnie przetwarzanego słowa (i jednocześnie początek kolejnego słowa Lyndona w rozkładzie). Przed każdym obrotem zewnętrznej pętli będzie spełniony następujący niezmiennik pętli: słowo $s[p..j-1]$ jest słowem Lyndona.

Oczywiście dla $p = 0$, $j = 1$ słowo $s[0]$ jest słowem jednoliterowym i tym samym również słowem Lyndona. Niezmiennik przed pierwszym wejściem do pętli jest prawdziwy. Omówimy teraz trzy przypadki, które mogą wystąpić w pętli, w zależności od tego, która z liter $s[j]$, $s[p]$ jest mniejsza.

Przypadek 1. $s[j] > s[p]$

Niezmiennik pętli gwarantuje, że słowo $s[p..j-1]$ jest słowem Lyndona. Zatem z lematu pomarańczowego wynika, że słowo $s[p..j]$ też jest słowem Lyndona i po prostu zwiększamy j o jeden.

Rozwiązanie zadania M 1444.
Oznaczmy kąty przy wierzchołku O jak następuje: $\sphericalangle AOZ = \sphericalangle COX = \alpha$, $\sphericalangle AOY = \sphericalangle BOX = \beta$, $\sphericalangle COY = \sphericalangle BOZ = \gamma$.



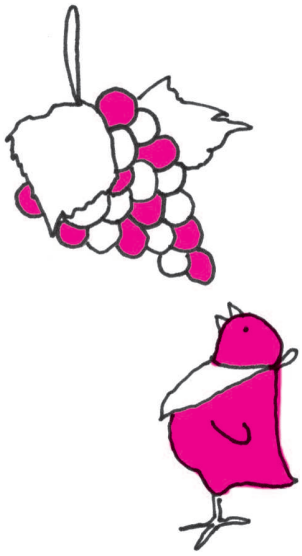
Zauważmy, że $\alpha + \beta + \gamma = 180^\circ$ oraz na mocy tw. o kątach wpisanych, $\sphericalangle CBX = \alpha$, $\sphericalangle BCX = \beta$, więc kąty w trójkącie BCX wynoszą α, β, γ. Analogicznie jest dla trójkątów BZA i YCA. Zatem są to trójkąty podobne do trójkąta XCB, w szczególności

$$\frac{AY}{CY} = \frac{BX}{BC} \quad \text{i} \quad \frac{BZ}{AZ} = \frac{BC}{CX}.$$

Mnożąc te równania stronami, dostajemy

$$\frac{AY}{CY} \cdot \frac{BZ}{AZ} = \frac{BX}{CX},$$

co daje tezę.



W trakcie pisania tego artykułu autor przypadkowo odnalazł w literaturze bardzo podobny algorytm, który w 1983 roku opublikował Jean-Pierre Duval (*Factorizing words over an ordered alphabet*, J. Algorithms 4, no. 4, 363–381).

Przypadek 2. $s[j] = s[p]$

Dopóki $s[i] = s[j]$, zwiększamy $i = i + 1$ oraz $j = j + 1$. Zauważmy, że po wykonaniu tej pętli słowo $s[p..p + j - i - 1]$ o długości $j - i$ powtarza się jako spójny fragment pewną liczbę razy, dając słowo okresowe.

Z niezmiennika pętli wynika, że słowo $s[p..p + j - i - 1]$ jest słowem Lyndona. Kolejne spójne fragmenty długości $j - i$ są również słowami Lyndona.

Rozkład podsłowa $s[p..j - 1]$ przedstawia się następująco: $x_1 x_2 \dots x_m x_{m+1}$, gdzie $x_1 = x_2 = x_3 = \dots = x_m$; dla $k \leqslant m$ słowo x_k ma długość $j - i$ oraz $x_k = s[p..p + j - i - 1]$. Słowo $t = x_{m+1}$ jest słowem $s[p + (j - i) \cdot m..j - 1]$, gdzie $m = \lfloor (j - p)/(j - i) \rfloor$. Słowo t jest prefiksem właściwym słowa x_m. Może się zdarzyć, że t jest puste. Słowo t również nie musi być słowem Lyndona.

Mamy teraz dwie możliwości: (a) $s[j] < s[i]$, (b) $s[j] > s[i]$.

Dla (a) wypisujemy rozkład $x_1 x_2 \dots x_m$. Rozkład ten spełnia warunek monotoniczności, czyli $x_1 \geqslant x_2 \geqslant \dots \geqslant x_m$ i jest to najkrótszy rozkład słowa $s[p..p + (j - i) \cdot m - 1]$ na słowa Lyndona. Mamy też $x_m = ts[i]y$. Zauważmy, że t jest prefiksem x_m. Kolejne słowo Lyndona P jest prefiksem słowa t albo słowo $ts[j]$ jest prefiksem P. W pierwszym przypadku zachodzi $P \leqslant t < x_m$. Widać więc, że kolejne słowo P spełnia warunek monotoniczności. Rozważmy teraz drugi przypadek. Łatwo zauważyć, że $ts[j] < ts[i]y$. Z definicji porządku leksykograficznego wynika dalej, że $P < ts[i]y = x_m$. Również tutaj P spełnia warunek l. Słowo to będzie znalezione w następnej fazie. Kładziemy zatem $p = p + (j - i) \cdot m$, $j = p + 1$.

Dla (b) wykażemy, że słowo $s[p..j]$ jest słowem Lyndona.

Mamy tutaj taki rozkład jak w poprzednim przypadku, tj. $x_1 = x_2 = \dots = x_m$ oraz x_{m+1}, przy czym $x_{m+1} = ts[j]$ oraz $x_1 = ts[i]y$, gdzie t jest wspólnym prefiksem, być może pustym. Z niezmiennika pętli wiemy, że $x_1, x_2, \dots, x_m$ są słowami Lyndona. Przypuśćmy, że t jest niepuste.

Ponieważ $ts[i]y$ jest słowem Lyndona, to również $ts[j]$ jest słowem Lyndona, co wynika z lematu kurczaczkowego. Mamy również $ts[i]y < ts[j]$. Stąd $ts[i]yts[j]$ jest słowem Lyndona na podstawie twierdzenia 2. Doklejając kolejne słowa, dostajemy, że $s[p..j] = (ts[i]y)^m ts[j]$ jest słowem Lyndona. Zwiększamy zatem j o jeden.

Przypadek 3. $s[j] < s[p]$

Z lematu winogronowego wynika, że słowo $s[p..j]$ nie jest słowem Lyndona. Łatwo też zauważyć, że słowo $s[p..k]$ dla $k > j$ nigdy nie będzie słowem Lyndona. Wypisujemy zatem $s[p..j - 1]$ i ustawiamy $p = j$ oraz $j = j + 1$.

Twierdzenie 4. *Rozkład uzyskany w powyższym algorytmie spełnia warunek monotoniczności.*

Dowód. Przed każdym wejściem do pętli słowo $s[p..j - 1]$ jest słowem Lyndona. Jedynymi momentami algorytmu, w których wypisywane są słowa rozkładu, są przypadki (2a) i (3). W przypadku (2a) rozkład $x_1 = \dots = x_m$ spełnia warunek monotoniczności.

Analizując przypadek (2a), można zaobserwować, że dowolny prefiks P słowa $s[j..n]$ będący słowem Lyndona spełnia $x_m > P$. W przypadku (3) mamy $s[p..j - 1] > s[j..n]$, więc kolejne słowo Lyndona x (prefiks $s[j..n]$) będzie spełniać $x < s[p..j - 1]$. $\triangle$

Twierdzenie 5. *Złożoność czasowa powyższego algorytmu to $O(n)$.*

Dowód. Liczba przebiegów pętli wynosi co najwyżej n. Wypisanie słów rozkładu jest liniowe względem n, wynika to wprost z algorytmu. Pozostałe operacje powodują, że wskaźnik j zwiększa się o jeden, oprócz przypadku (2a), w którym „cofa" się on o co najwyżej $j - i$ pozycji. Jednak sumaryczny koszt „cofnięć" jest niewiększy niż suma długości słów Lyndona z rozkładu, a więc również liniowy względem n. $\triangle$