# Algorytmika, 7. ćwiczenia

2008-04-10

## 1 Plan

- Rownowazność cyliczna 2 słów za pomocą wyszukiwania wzorca
- Minimalne slowo pokrywające za pomoca tablicy P z algorytmu KMP
- Przykład, że KMP może wisieć nad 1 pozycją w tekście m razy,
- KMP z tablicą P' (znana tez pod nazwą NEXT), ew. pokazać, że wtedy wisi nad 1 pozycja $\leq \log n$ razy.
- algorytm Aho-Corasick
- algorytm Bakera (Wyszukiwanie wzorca 2D)

## 2 Równoważność cykliczna 2 słów

Dane dwa słowa $x$ i $y$, należy sprawdzić, czy $x$ jest równoważne $y$ (z dokładnością do przesunięcia cyklicznego).
Rozwiązanie: trzeba sprawdzić czy $y$ występuje w $x^2$.

## 3 Minimalne slowo pokrywające

Dla danego tekstu $T$, należy znaleźć najkrótsze słowo, którego wszystkie wystąpienia w całości pokrywają tekst.
(rozwiązanie z wazniak.mimuw.edu.pl)

- oblicz tablicę $P$ dla tekstu $T$,
- niech $S[i]$ długość najkrótszego słowa pokrywającego $T[1..i]$,
- niech $Zakres[j] = k$ jeśli za pomocą słowa długości $j$ można pokryć $T[1..k]$,
- rozważmy następujący algorytm:

```
for i = 2 to n do
    Zakres[i] = S[i] = i ;
end
for i = 2 to n do
    if P[i] > 0 and i − Zakres[S[P[i]] ≤ S[P[i]] then
        S[i] = S[P[i]]; Zakres[S[P[i]] = i ;
    end
end
```

- możemy udowodnić, że dla dowolnego $i$, mamy $S[i] = i$ lub $S[i] = S[P[i]]$: powiedzmy, że warunek nie jest spełniony, czyli $S[i]! = S[P[i]]$ i $S[i]! = i$:
  - $S[i] < S[P[i]]$ – sprzeczność w takim wypadku istniałoby słowo pokrywające również dla $P[i]$ o długości $S[i]$,
  - $P[i] < S[i] < i$ – sprzeczność w takim wypadku $P[i]$ ma złą wartość,
  - $S[P[i]] < S[i] \leq P[i]$ — niech $y$ słowo pokrywające $T[1..P[i]]$ ($y = T[1..S[P[i]]]$), niech $x$ słowo pokrywające $T[1..i]$ oraz $T[1..P[i]]$, czyli mamy $|x| > |y|$. Wiemy, że $y$ pokrywa słowo $x$ ($y$ jest prefiksem $x$, $y$ jest sufiksem $x$, oraz $x$ pokrywa $T[1..P[i]]$ i $T[1..i]$)

    Teraz, ponieważ $x$ pokrywa $T[1..i]$, wiec pozycja $T[i - |y|]$ jest pokryta przez $x$ wiec jest pokryta przez $y$ konczace sie przed $i$. Stąd $zakres[S[P[i]]] \geq i - |y| = i - S[P[i]]$, a wiec warunek $S[i] = S[P[i]] = |y|$.

# 4 KMP – ciekawostki

- przykład, że KMP może spędzić nad jedną pozycją $O(m)$ razy: $T = a^{m-1}b$, $W = a^m$,
- możemy zdefiniować tablicę $P'$ jako "ulepszoną" tablicę prefikso-sufiksów, $P'[i] = \max(-1, \{j : W[1..j] = W[(i-j+1)..i]$ **and** $W[j+1] \neq W[i+1]\})$
- przykład, że KMP' (dla tablicy $P'$) może spędzić nad jedną pozycją $\Omega(\log n)$ razy:
  - dla KMP' $delay = O(\log m)$: Z lematu o okresowości można udowodnić, że jeśli $P'[i] = j$, $P'[j] = k$, to $i \geq k + j$, czyli $P'[F_i] \leq i$. Jeśli $i \leq k + j$ to można udowodnić, że $W[k + 1] = W[j + 1]$.
  - niech $F_0 = a$, $F_1 = ab$, $F_{n+1} = F_n + F_{n-1}$, $F'_n$ oznacza $F_n$ z usuniętymi dwoma ostatnimi znakami, rozważmy $T = F'_n cc$, $W = F_n$

Lemat o okresowości: jeśli słowo $x$ ma okresy $p$ i $q$, takie, że $p + q \leq |x|$ to $nwd(p, q)$ jest również okresem $x$ (wystarczy pokazać, że (dla $p > q$) $p - q$ jest również okresem $x$).

# 5 Algorytm Aho-Corasick

Wyszukiwanie wielu wzorców w czasie liniowym (do sumy długości wzorców i przeszukiwanego tekstu).

Dane są wzorce $W_1, \ldots, W_k$.

- Przygotowujemy drzewo TRIE zawierające wzorce $W_1, \ldots, W_k$,
- Dla każdego węzła drzewa $L(v)$ oznacza napis powstały z konkatenacji etykiet na ścieżce z korzenia do $v$,
- Dla każdego węzła musimy obliczyć $f(v)$ (failure function), $f(v) = x$ jeśli $L(x)$ jest najdłuższym sufiksem $L(v)$
- Dla każdego węzła oblicz $out(v)$ (zbiór wzorców które są rozpoznawane po osiągnięciu $v$) (początkowo $out(v) = \{i\}$ jeśli $W_i$ kończy się w $v$, potem idąc od korzenia uzupełniamy $out(v) := out(v) + out(f(v))$)

# 6 Algorytm Bakera

Wyszukiwanie dwuwymiarowych wzorców.

Dany jest tekst $T[1..n, 1..n]$ oraz wzorzec $W[1..m, 1..m]$ należy wyznaczyć pary $(i, j)$ t.że $T[i..(i+m-1), j..(j+m-1)] = W$.

- przygotuj zbiór wzorców $\{W_i = W[i, 1..m] : 1 \leq i \leq m\}$ (kolejne kolumny wzorca),
- za pomocą algorytmu Aho-Corasick znajdź wystąpienia wzorców $W_i$ w poszczególnych kolumnach tekstu, wynikiem niech będzie tablica $A[1..n, 1..n]$ t.że $A[i, j] = k$ jeśli $A[i, j..(j+m)] = W_k$ lub $A[i, j] = -1$ wpp.
- za pomocą algorytmu KMP w poszczególnych wierszach tabli $A$ odszukaj wystąpienia ciągu $1, 2, \ldots, m$ (jeśli kolumny $W_i$ się powtarzają, to ciąg będzie trochę inny)