Algorytmy i złożoność obliczeniowa

Laboratorium 12. Algorytmy tekstowe.

1. Algorytmy tekstowe

Algorytmy tekstowe mają decydujące znaczenie przy wyszukiwaniu informacji typu tekstowego, ten typ informacji jest szczególnie popularny w informatyce, np. w edytorach tekstowych i wyszukiwarkach internetowych. Tekst jest ciągiem symboli. Przyjmujemy, że jest on zadany tablicą $x[1,\ldots,n]$, elementami której są symbole ze skończonego zbioru *A* (zwanego alfabetem). Liczba $n = |x|$ jest długością (rozmiarem) tekstu.

Typowe reprezentacje tekstów to reprezentacja listowa i reprezentacja tablicowa. Podstawowym problemem dotyczącym tekstów jest problem wyszukiwania wzorca. Problem ten polega na znalezieniu wszystkich wystąpień tekstu *x*, zwanego wzorcem w tekście *y*. Przyjmujemy, że $|x| = m,\ |y| = n,\ n \geq m$.

Wyszukiwanie wzorca było wszechstronnie badane ze względu na duże zastosowanie praktyczne. Poniżej zostaną zaprezentowane podstawowe algorytmy dla problemu wyszukiwania wzorca takie jak : algorytm N (algorytm „naiwny”), Algorytm KMP (Knutha-Morrisa-Pratta), Algorytm GS' (wersja algorytmu Galila-Seiferasa dla pewnej klasy wzorców) - więcej algorytmów zostało omówione na wykładzie.

2. Problem wyszukiwania wzorca

Niech x, będzie wzorcem, a y tekstem, w którym szukamy wzorca. Dla słowa z przyjmijmy $z[i..j] = z[i]z[i+1]\ldots z[j]$, gdzie $i \leq j$. Mówimy, że x występuje w y na pozycji i, gdy $y[i..i+m-1] = x$ - inaczej mówiąc, gdy począwszy od i-tej pozycji, wzorzec „pasuje” do tekstu. Z tego powodu problem wyszukiwania wzorca jest również nazywany problemem dopasowywania wzorca.

Przykład

-alfabet: {a, b}

-tekst: abbbaabaabba

-wzorzec: aaba

wystąpienia:

1 abbbaabaaabaa

2 abbbaabaabaa

2.1. Algorytm N(„naiwny”)

Schemat najbardziej bezpośredniego algorytmu, zwanego „naiwnym”, wygląda następująco:

```
begin
    i := 1;
    while i <= n - m + 1 do
    begin
        if x[1..m] = y[i..i + m -1] then write(i);
        i := i +1; (przesunięcie =1)
    end;
end;
```

Pełny algorytm otrzymamy, rozpisując instrukcję sprawdzenia równości tekstów.

```
begin
    i := 1;
while i <= n - m + 1 do
    begin
        j := 0; while x[j+1] = y[i+j] do  j := j + 1;
        if j = m then write(i); i := i +1; {przesunięcie =1}
    end;
end;
```

Przykład

teskt=bbabbbaabb, n=10

wzorzec=aab, m=3

-i=1 nie nastąpi wejście do zagnieżdżonej pętli while, nie jest również spełniony warunek if.
- i=2 nie nastąpi wejście do zagnieżdżonej pętli while, nie jest również spełniony warunek if.
- i=3 nastąpi wejście do zagnieżdżonej pętli while, ale dla j=1 nastąpi wyjście, po tym wyjściu warunek if będzie niespełniony,
- i=4..6 nie nastąpi wejście do zagnieżdżonej pętli while, nie jest również spełniony warunekif.
- i=7 nastąpi wejście do zagnieżdżonej pętli while, która wykonywana będzie aż do j=3, po tym wyjściu warunek if jest spełniony tzn. znaleziono dopasowanie.
- i=8 nie nastąpi wejście do zagnieżdżonej pętli while, nie jest również spełniony warunek if.
- i=9 nie jest spełniony główny warunek, algorytm zostaje zakończony.