Algorytmy tekstowe labolatorium I. Wyszukiwanie wzorca w tekście

Łukasz Kita

Marzec 2020

1 Zadanie 4.

W przypadku wyszukiwania słowa "art" w tekście ustawy algorytm KMP okazał się nieznacznie szybszy od pozostałych algorytmów. Jego wykonanie zajęło ok. 0,055s, przy znikomo małym czasie inicjalizacji. Natomiast pozostałe dwa algorytmy (naiwny i wykorzystujący maszynę stanów skończonych) wykonały się w podobnym czasie ok. 0,077s.

2 Zadanie 5.

Czas inicjalizacji algorytmów: wykorzystującego maszynę stanów skończonych oraz algorytmu KMP jest znikomo krótki w porównaniu z czasem wykonania się algorytmu. Jest to spowodowane faktem, że wzorzec jest krótki. Algorytm naiwny wykonał się w czasie 75s, co czyni go szybszym od algorytmu opartego o maszynę stanów skończonych, którego wykonanie zajęło 82s. Jest to spowodowane faktem, że przetwarzany tekst jest tekstem naturalnym i nie zawiera wielu dopasować wzorca, natomiast sam wzorzec jest krótki. Ponownie najszybszy, lecz nieznacznie, okazał się algorytm KMP, którego wykonanie zajęło 70s.

3 Zadanie 6.

Proponowany jest dowolny tekst (np. przykładowy tekst zawierający fragment polskiej wikipedii) oraz wzorzec składający się z dużej liczby takich samych znaków. Wówczas to algorytm naiwny będzie działał w czasie $O(n \cdot m)$ gdzie n to długość tekstu, a m to długość wzorca, natomiast pozostałe algorytmy będą działać w czasie liniowym. Ponadto inicjalizacja algorytmu KMP i algorytmu wykorzystującego maszynę skończoną będzie się odbywać również w czasie liniowym ze względu na długość wzorca: O(m), co spowoduje, że całkowite wykonanie się tych algorytmów będzie zdecydowanie krótsze od wykonania się algorytmu naiwnego.

4 Zadanie 7.

Proponowany jest wzorzec zawierający dużą liczbę rożnych symboli, powtórzonych wielokrotnie, np. ciąg składający się z powtórzonych wielokrotnie ciągów "$qwertyuiop[]asdfghjkl;' zxcvbnm,./$". W tym przypadku inicjalizacja algorytmu wykorzystującego maszynę stanów skończonych wymaga przejścia po całej długości tekstu oznaczonej jako n i dla każdej pozycji w tekście wymaga przejścia po wszystkich m znakach występujących w tekście. Daje to złożoność obliczeniową $O(n \cdot m)$ W przypadku inicjalizacji algorytmu KMP nie ma konieczności wykonania się petli while wewnątrz pętli for, co skutkuje tym, że złożoność obliczeniowa jest liniowa względem długości wejścia: O(n).