Algorytmy tekstowe

Adam Dyda

Marzec 2021

1 Algorytmy

Pierwsze laboratoria dotyczyły sposobów wyszukiwania wzorca w tekście. Zajęliśmy się trzema algorytmami rozwiązującymi ten problem:

- algorytm naiwny
- automat skonczony
- algorytm Knutha-Morrisa-Pratta

2 Porównanie złożoności

Powyższe algorytmy posiadają różne złozoności obliczeniowe, algorytm naiwny cechuje sie najgorszą złożonością obliczeniową która w najgorszym wypadku wynosi $O(m(n-m+1))$, gdzie m jest długościa wzorca który chcemy znalezc a n dlugościa tekstu który przeszukujemy. Złożoności algorytmu atuomatu skonczonego i algorytmu KMP wynoszą $O(n)$ jednak trzeba zaznaczyc że w przypadku tych algorytmów wymagany jest wcześniejszy preprocessing.

3 Wykonane testy

3.1 Wyszukiwanie wzorca w ustawie

Szybkości działania algorytmów dla znajdywania wzorca "art" w załączonej ustawie prezentują się następująco:

Naive algorithm time: 0.0482029914855957 [s]
Finite automata time: 0.035314083099365234 [s]
KMP algorithm time: 0.04389023780822754 [s]

3.2 Najgorszy przypadek dla algorytmu naiwnego

Najgorszy przypadek dla algorytmu naiwnego wystepuje gdy we wzorcu i w tekscie wszystkie litery sa takie same. Szybkośći działania algorytmów dla wzorca "AAA...A" powtórzone 100000 razy, i dla tekstu "AAA...A" powtórzone 1000000 razy:

Naive algorithm time: 7.927556276321411 [s]
Finite automata time: 1.0640978813171387 [s]
KMP algorithm time: 0.4513130187988281 [s]

3.3 Najgorszy przypadek tworzenia tablicy przejścia

Najgorszy przypadek dla tworzenia tablicy przejscia wystepuje gdy w tekscie jest duzo unikalnych znaków. Szybkośc dzialania algorytmów dla wzorca "abcdefghijklmnoprstuwxyz" powtórzonego 30 razy:

Automata build time: 2.3946692943573 [s]
Prefix function build time: 0.0003020763397216797 [s]