Ireneusz Okniński

Jakub Jażdżyk

AISDI – Wyszukiwanie wzorca

Struktura projektu.

Projekt składa się z modułów:

KMP_algorithm.py – implementacja algorytmu Knutha-Morrisa_Pratta

KR_algorithm.py – implementacja algorytmu Karpa-Rabina

N_algorithm.py – implementacja algorytmu naiwnego

tests.py – testy poprawności działania algorytmów

benchmark.py – testy wydajności i rysowanie wykresu

Sprawdzenie poprawności implementacji.

Sprawdzamy poprawność działania algorytmu dla przypadków:

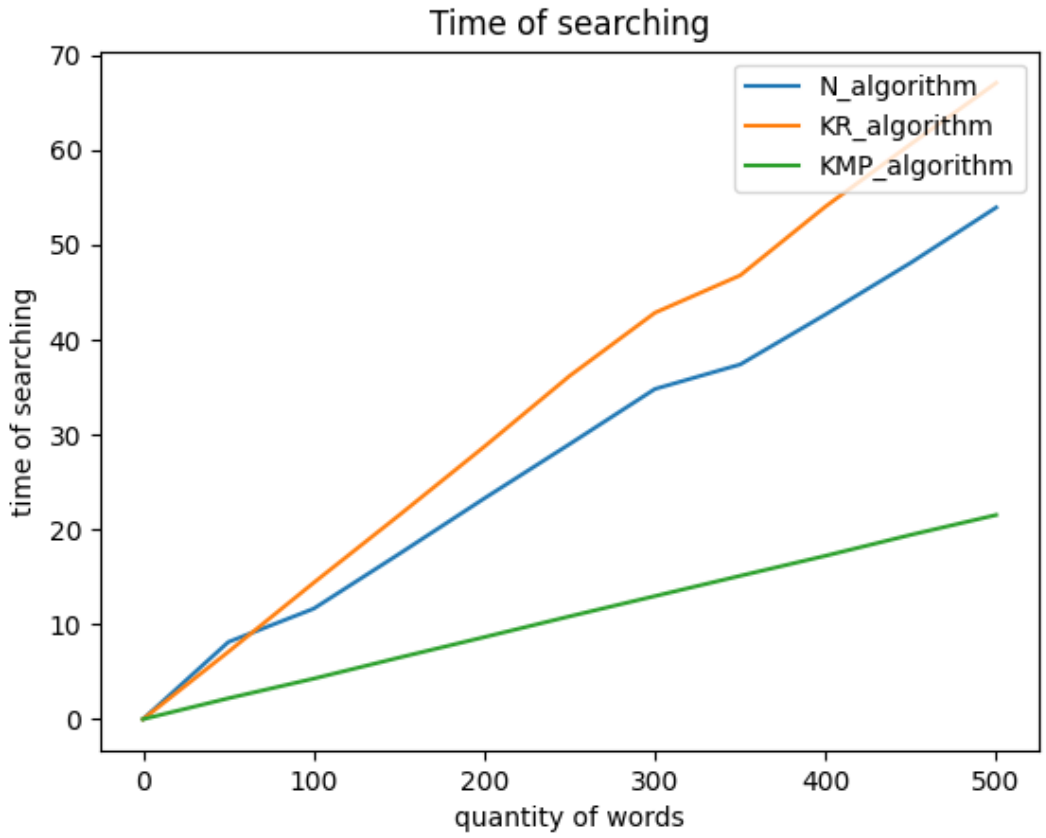
- tekst i wzorzec są równe
- wzorzec jest dłuższy od tekstu
- tekst nie zawiera wzorca
- poprawność wyszukiwania algorytmem naiwnym
- losowo generowane dane (5 zestawów)

```
===EQUAL===
Naive: [0], KR: [0], KMP: [0]

===PATTERN LONGER===
Naive: [], KR: [], KMP: []

===NOT CONTAINS===
Naive: [], KR: [], KMP: []

===NAIVE===
Naive: [1, 3]
Naive: [0, 5, 6]

===RANDOM===
Naive: [38, 45, 95, 108, 130], KR: [38, 45, 95, 108, 130], KMP: [38, 45, 95, 108, 130]
Naive: [3, 51, 64, 74, 81, 97, 121], KR: [3, 51, 64, 74, 81, 97, 121], KMP: [3, 51, 64, 74, 81, 97, 121]
Naive: [15, 58], KR: [15, 58], KMP: [15, 58]
Naive: [56, 136, 141], KR: [56, 136, 141], KMP: [56, 136, 141]
Naive: [104, 116], KR: [104, 116], KMP: [104, 116]
```

Testy dają poprawne rezulataty.

Porównanie algorytmów wyszukiwania wzorca.

Zmierzyliśmy czas wyszukiwania pierwszych (0, 50, 100, … 500) słów „Pana Tadeusza” w całym tekście. Z wyników pomiarów utworzyliśmy wykres.



Najszybszy był algorytm KMP, a najwolniejszy algorytm KR. W tekście „Pana Tadeusza” znajduje mało powtórzonych i nakładających się wzorców. Dlatego też wszystkie algorytmy mają złożoność bliską złożoności optymistycznej O(n).