Sprawozdanie

Autorzy: Angelika Ostrowska, Kacper Straszak

Data: 06.05.2023

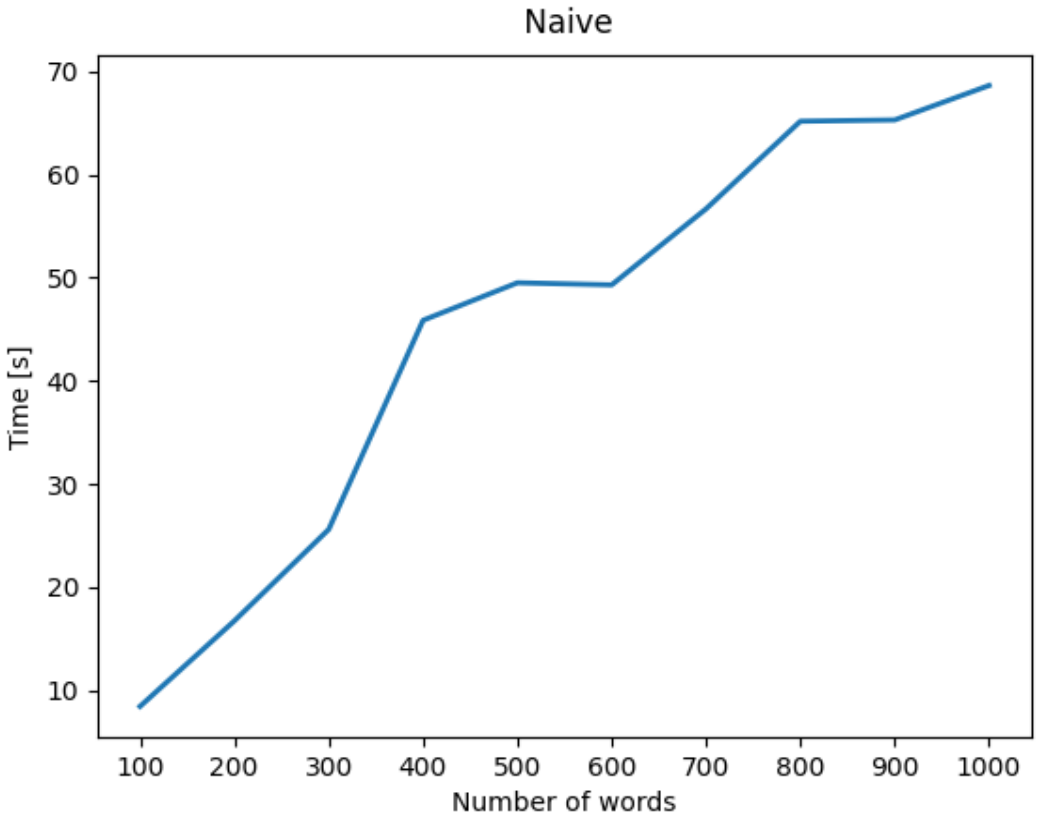
AISDI Zad5

Algorytmy wyszukiwania wzorca:

Algorytmy wyszukiwania wzorca przeszukują dany tekst w celu znalezienia w nim podanego ciągu znaków. Rozwiązaniem są indeksy miejsc w tekście, w których rozpoczyna się znaleziony wzorzec.

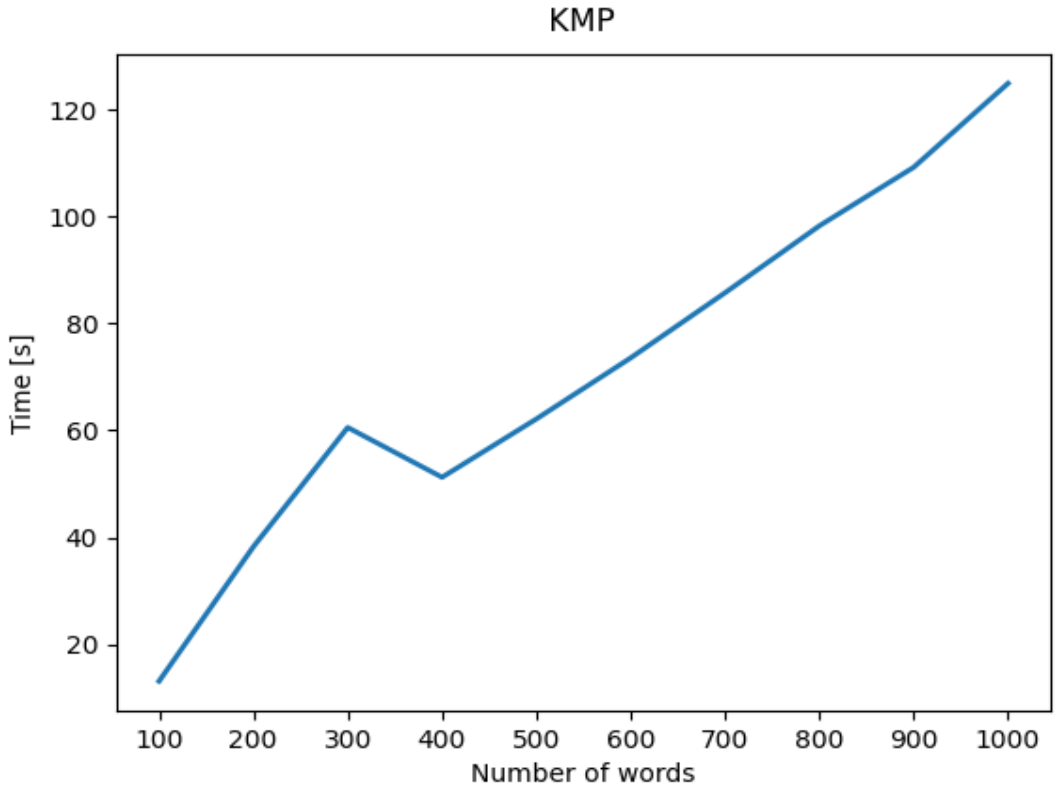
Algorytm naiwny:

Najprostszy algorytm, sprawdza on kolejne litery w tekście i porównuje z pierwszym znakiem wzorca. Jeśli znaki są równe, to kolejny znak tekstu porównywany jest z kolejnym znakiem wzorca. Proces powtarza się dopóki nie zostanie znaleziony znak w tekście, który nie jest identyczny z danym znakiem wzorca, lub kiedy skończą się znaki wzorca do porównywania. W obu przypadkach kolejne sprawdzanie zacznie się od znaku następującego po pierwszym, który zgadzał się ze wzorcem. Dodatkowo w drugim przypadku zapisany zostanie indeks pierwszego sprawdzanego znaku (został znaleziony wzorzec w tekście).



Algorytm KMP:

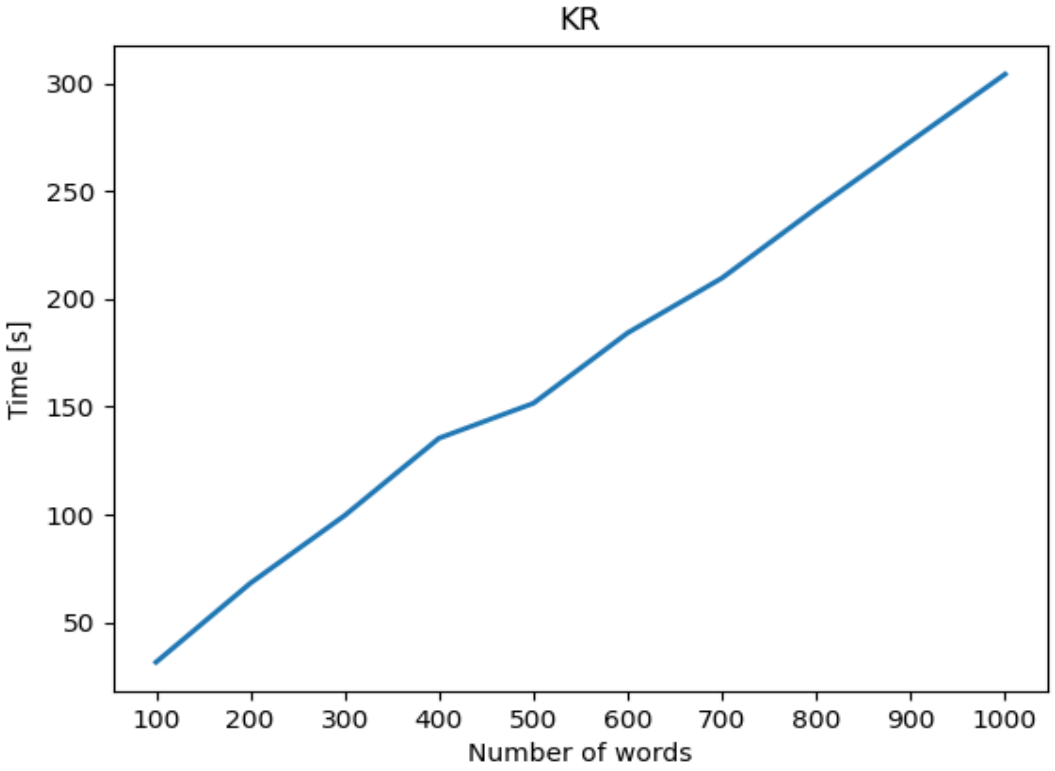
Algorytm KMP początkowo działa na tej samej zasadzie co algorytm naiwny - szuka pierwszego znaku który pokryje się z pierwszym znakiem wzorca. Zmiana występuje, gdy przynajmniej 2 znaki tekstu zostaną porównane ze wzorcem.
Po przerwaniu porównywania (znaki różne) lub po znalezieniu wzorca kolejne sprawdzanie nie zaczyna się od indeksu znaku następującego po pierwszym sprawdzanym elemencie.
Podczas przechodzenia po tekście już częściowo pokrywającym się z wzorcem, zapisywany jest kolejny potencjalny początek wzorca (indeks znaku takiego jak pierwszy znak wzorca) jeśli taki istnieje. Kolejne sprawdzenie zaczyna się od niego lub (jeśli takiego znaku nie było) od miejsca, gdzie skończyło się poprzednie sprawdzanie.



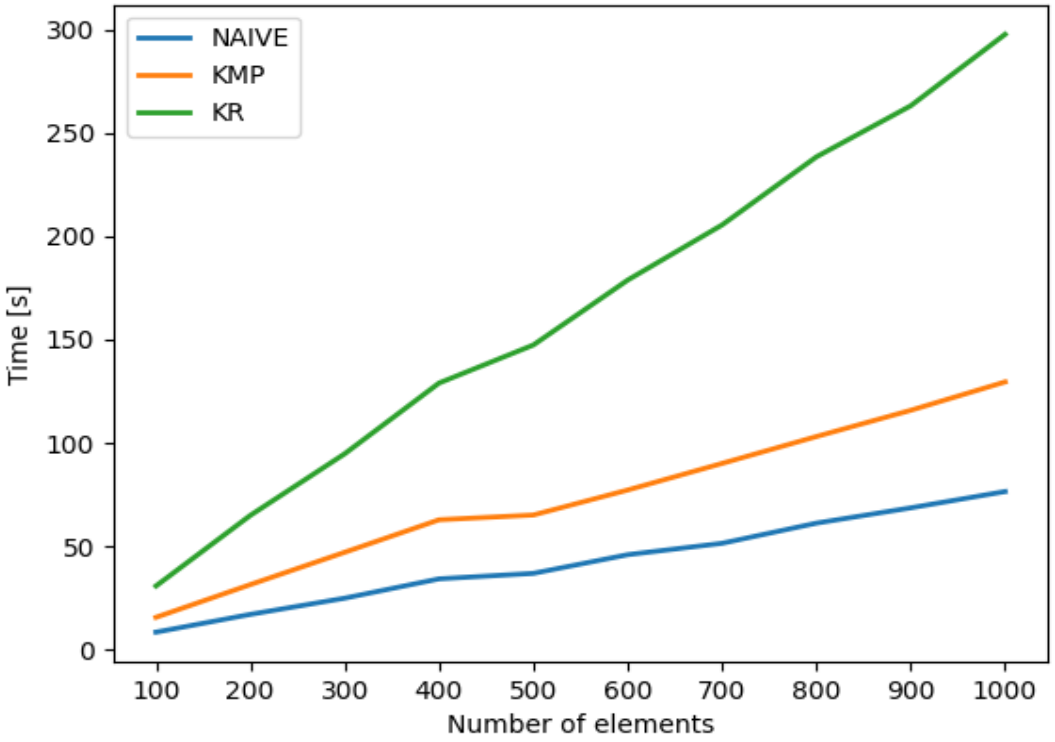
Algorytm KR:

Algorytm KR zapisuje hash wzorca wytworzony przy pomocy specjalnie zbudowanej funkcji i porównuje go z hashami kolejnych okienek o wielkości takiej jak wzorzec. Jeśli hashe są różne to okienko przesuwa się o jeden znak tekstu. Jeśli znaleziony zostanie taki sam hash dla wzorca i okienka, to kolejne znaki są sprawdzane, żeby zobaczyć czy tekst w okienku i wzorzec są takie same. Jeśli są to zostaje zapisany indeks początku okienka, w przeciwnym razie okienko przesuwa się.
Dodatkowe sprawdzenie jest potrzebne, ponieważ więcej niż jeden ciąg znaków może generować ten sam hash. Dlatego ważny jest wybór funkcji hashującej. My zastosowaliśmy sumowanie wartości ascii kolejnych znaków ciągu. Może to

powodować dosyć częste otrzymywanie takich samych hashy dla różnych ciągów, ale okazało się szybsze niż użycie innej testowanej przez nas funkcji (sumowanie kodów ascii znaków pomnożonych przez 10^(pozycja w oknie)).



Porównanie wykresów:



Działanie KR jest najdłuższe przez to, że muszą zostać wytworzone hashe, ale w tym samym czasie algorytm ten jest najbardziej powtarzalny, długość jego działania nie jest zależna od ułożenia znaków w tekście jak w pozostałych dwóch algorytmach. Algorytm naiwny okazuje się najszybszy, ale w tym samym czasie jest najbardziej nieregularny i nieprzewidywalny w działaniu przy różnych tekstach.