

# Algorytmy i struktury danych, Teleinformatyka, I rok

## Raport z laboratorium nr: 7

Imię i nazwisko studenta: Mateusz Warzecha

nr indeksu: 410846

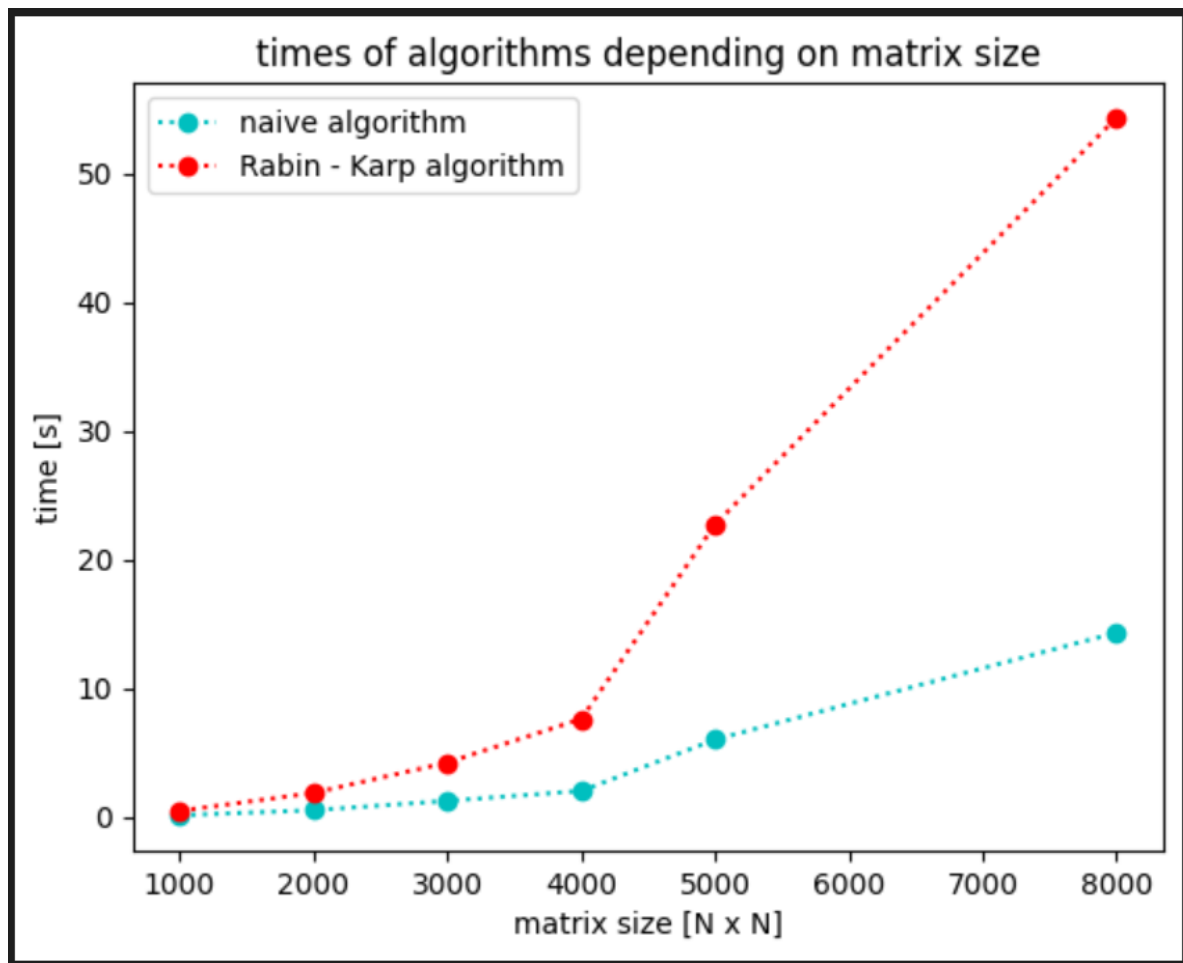
1. W pole poniżej wklej najważniejszy (według Ciebie) fragment kodu źródłowego z zajęć (maksymalnie 15 linii).

```
1. def naive_search(filename):  
2.     new_file=open('LAB_07\\patterns\\'+filename, 'r').readlines()  
3.     coordinates=[]  
4.     for x in range(len(new_file)-2):  
5.         for i in range(len(new_file[x])-2):  
6.             if new_file[x][i]==chr(65) and new_file[x][i+1]==chr(66) and  
               new_file[x][i+2]==chr(67) and new_file[x+1][i]==chr(66) and  
               new_file[x+2][i]==chr(67):  
7.                 coordinates.append(str(x)+', '+str(i))  
8.     print(coordinates)
```

Wybrałem te 8 linijek kodu ponieważ zawiera się w nich funkcja naiwnego szukania wzorca. Jest ona nie zbyt skomplikowana, a nawet szybko sobie radzi z przeszukiwaniem wzorców, co jest ciekawe w porównaniu do jej prostoty

2. Podsumuj wyniki uzyskane podczas wykonywania ćwiczenia. Co ciekawego zauważyłeś? Czego się nauczyłeś? Jeśli instrukcja zawierała pytania, odpowiedz na nie. Do sprawozdania możesz dodać wykresy jeśli jest taka potrzeba.

Co ciekawe algorytm naiwny jest bardziej wydajny od algorytmu Rabina-Karpa. Jest to prawdopodobnie spowodowane większą liczbą procesów które wykonuje algorytm Karpa-Rabina sprawdzając czy klucz się zgadza, a dopiero potem naiwnie porównuje elementy. Na poniższym wykresie wyraźnie widać że dla większej ilości elementów algorytm Karpa-Rabina jest o wiele mniej wydajny.



\*wykres stworzony przy pomocy biblioteki matplotlib