

Algorytmy i struktury danych, Teleinformatyka, I rok

Raport z laboratorium nr: 3

Imię i nazwisko studenta: Amadeusz Gunia

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

```
1. for i in range (N-2):
2.     for j in range (N-2):
3.         x=(i*N)+j
4.         T[x]=='A' and T[x+1]=='B' and T[x+2]=='C' and
5.         T[x+N]=='B' and T[x+2*N]=='C':
6.             licznik=licznik+1
7.
```

Uzasadnij swój wybór.

Uważam, że ten fragment kodu jest najważniejszy, gdyż w prosty i krótki sposób przedstawia działanie algorytmu naiwnego, który według mnie jest najprostszym algorytmem do wyszukiwania wzorca.

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.

Wykonałem oba algorytmy dla każdego rozmiaru macierzy $N \times N$. W algorytmie Karpa-Rabina używałem różnych liczb pierwszych do dzielenia modulo. Najlepszy wynik uzyskałem dla liczby większej niż wartość haszu wzorca (w kodzie, który przesłałem $q=11111111$). Wynika z tego, że można pominąć dzielenie modulo, gdyż im mniejsze wartości haszów, tym dłużej trwa algorytm, ponieważ ma wtedy więcej tekstu do sprawdzenia. Uzyskane wyniki pokazują, że czas wykonania obu algorytmów jest bardzo podobny. Ponadto w algorytmie Karpa-Rabina przed dopasowywaniem tekstu do wzorca występuje tzw. proces przygotowania, który trwa o wiele dłużej niż samo dopasowanie. Podsumowując, stwierdzam, że algorytm naiwny jest prosty i zrozumiały nawet dla zwykłego użytkownika, w przeciwieństwie do algorytmu Karpa-Rabina gdzie problemem może być haszowanie, a ponadto łączny czas trwania algorytmu Karpa-Rabina jest dłuższy niż czas trwania algorytmu naiwnego, więc uważam, że algorytm naiwny jest lepszy.

Poniżej załączam zrzuty ekranu z Pythona oraz wykresy.

Czas tworzenia i przygotowania tablicy z tekstem: 0.4400978088378906

W macierzy o wielkosci 1000 x 1000 liczba wystapien wzorca wyniosla: 6

Czas wykonania algorytmu "naiwnego": 0.3899383544921875

Czas tworzenia tablicy z tekstem: 0.5860855579376221

Czas przygotowania algorytmu "Rabina-Karpa": 1.746213674545288

W macierzy o wielkosci 1000 x 1000 liczba wystapien wzorca wyniosla: 6

Czas wykonania algorytmu "Rabina-Karpa": 0.3209514617919922

N	liczba wystąpień wzorca	naiwny [s]	pre-KR [s]	Karp-Rabin [s]
1000	6	0,3899	1,7462	0,3209
2000	40	1,6360	7,2717	1,4883
3000	57	3,2552	15,6638	3,0263
4000	97	5,4252	27,6120	5,4670
5000	161	8,9985	43,5278	9,0398
8000	393	24,1054	141,6763	23,3576

