

Raport z laboratorium nr: 7

Imię i nazwisko autora: Grzegorz Piśkorski

Nr indeksu: 405879

1. Najważniejszy fragment kodu z laboratoriów:

```
for row in range(n - m):
    sum1 = 0
    for i in range(m):
        if ord(Text[row][i]) < 65:
            sum1 += int(Text[row][i])
        else:
            sum1 += (ord(Text[row][i]) - 55)
    hash_table[row].append(sum1)

for i in range(1, n - m):
    if ord(Text[row][i - 1]) < 65:
        sum1 -= int(Text[row][i - 1])
    else:
        sum1 -= (ord(Text[row][i - 1]) - 55)
    if ord(Text[row][i + m - 1]) < 65:
        sum1 += int(Text[row][i + m - 1])
    else:
        sum1 += (ord(Text[row][i + m - 1]) - 55)
    hash_table[row].append(sum1)
```

Uzasadnienie wyboru:

Wybrałem fragment kodu z algorytmu Rabina-Karpa, w którym to tworzymy naszą listę z sumami odpowiednich fragmentów tekstu. Moim zdaniem jest to najważniejsza część kodu, ponieważ jest kluczową częścią wyżej wspomnianego algorytmu. Bez tej listy, algorytm po prostu by nie działał.

Podsumowanie

Pomiary czasowe:

=== 1000 ===

Naiwne podejście

Czas wykonania całego algorytmu: 0.07317399978637695

Rabin Karp

Czas wykonania całego algorytmu: 0.8612525463104248

Czas wykonania preprocessingu: 0.7831463813781738

Czas szukania wzorca: 0.07810616493225098

Wyniki obu algorytmow: 6 6

=== 2000 ===

Naiwne podejscie

Czas wykonania calego algorytmu: 0.2655618190765381

Rabin Karp

Czas wykonania calego algorytmu: 2.6500906944274902

Czas wykonania preprocessingu: 2.353287696838379

Czas szukania wzorca: 0.29680299758911133

Wyniki obu algorytmow: 40 40

=== 3000 ===

Naiwne podejscie

Czas wykonania calego algorytmu: 0.5681052207946777

Rabin Karp

Czas wykonania calego algorytmu: 5.935931205749512

Czas wykonania preprocessingu: 5.293835401535034

Czas szukania wzorca: 0.6420958042144775

Wyniki obu algorytmow: 57 57

=== 4000 ===

Naiwne podejscie

Czas wykonania calego algorytmu: 1.0153858661651611

Rabin Karp

Czas wykonania calego algorytmu: 10.75268268585205

Czas wykonania preprocessingu: 9.517292022705078

Czas szukania wzorca: 1.2353906631469727

Wyniki obu algorytmow: 97 97

=== 5000 ===

Naiwne podejście

Czas wykonania całego algorytmu: 1.7443454265594482

Rabin Karp

Czas wykonania całego algorytmu: 17.8665828704834

Czas wykonania preprocessingu: 16.035855531692505

Czas szukania wzorca: 1.8307273387908936

Wyniki obu algorytmów: 161 161

=== 8000 ===

Naiwne podejście

Czas wykonania całego algorytmu: 4.134145975112915

Rabin Karp

Czas wykonania całego algorytmu: 44.0252149105072

Czas wykonania preprocessingu: 39.587244272232056

Czas szukania wzorca: 4.4379706382751465

Wyniki obu algorytmów: 393 393

Jak widzimy mimo, że algorytm Rabina-Karpa ma teoretycznie lepszą złożoność to jednak wykonuje się znacznie dłużej niż algorytm naiwny. Ale jest tak dlatego, że nasz wzorzec jest bardzo krótki. Najprawdopodobniej, dla dużo dłuższego wzorca zauważylibyśmy znaczną poprawę. Niestety (dopiero dzisiaj to zauważyłem) mój algorytm Rabina-Karpa nie jest tym oryginalnym algorytmem. Inaczej wygląda w nim „funkcja hashująca”. Mimo to jednak wolę wysłać to co mam, a nie zostać z niczym. Mój algorytm działa i jest podobny, ale jednak ta funkcja się różni.