

SPRAWOZDANIE LAB_7 (Tablica Asocjacyjna)

Cel: Zapoznanie się z Tablica Asocjacyjna i „Haszowaniem”. Porównanie różnych metod haszowania i ich wpływu na szybkość wyszukiwania.

Pomiary przeprowadzono dla dwóch różnych funkcji haszujących:

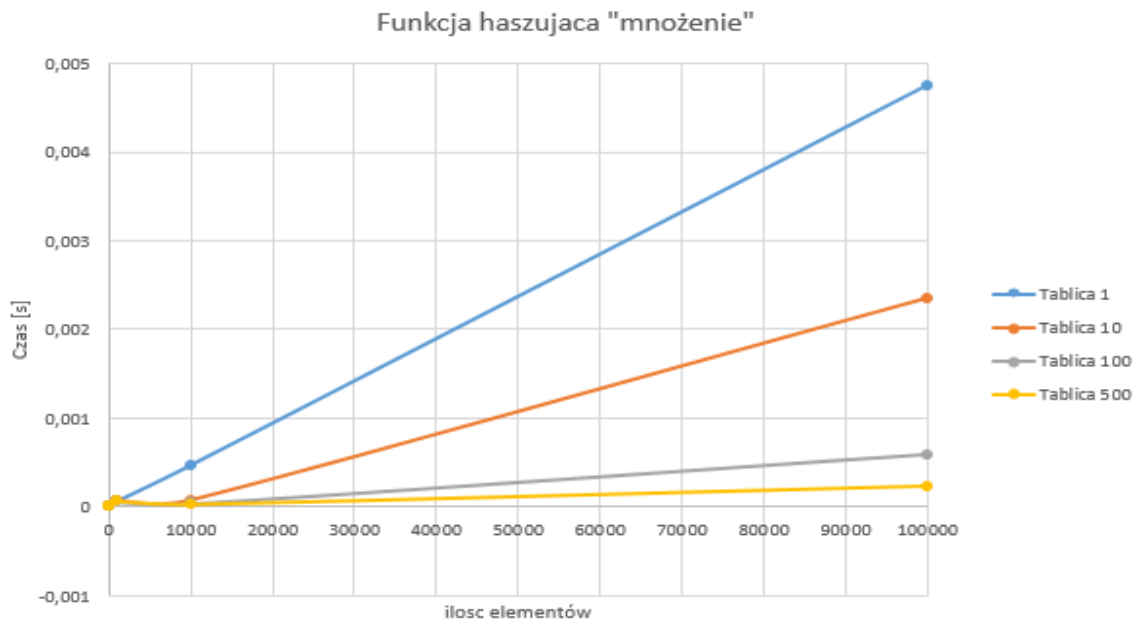
- metoda „MODULO” – dodanie kodów ASCII kolejnych liter i obliczenia modulo po wielkości tablicy,
- metoda „MNOŻENIE” - dodanie kodów ASCII kolejnych liter, pomnożenie przez stałą (zgodnie z książką Cormena: $A = (\sqrt{5} - 1)/2$), wyznaczenie części ułamkowej liczby. Na koniec mnożymy przez rozmiar tablicy i wyciągamy część całkowita z całości.

Ilość powtórzeń pomiarów to 20.

Wyszukiwany był losowo wygenerowany element.

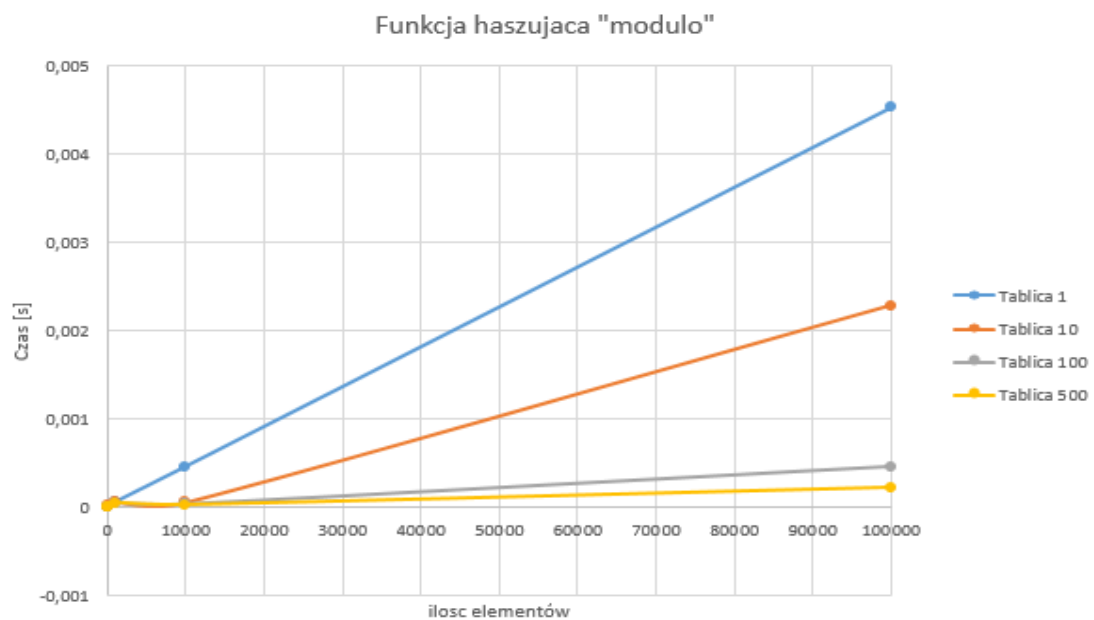
POMIARY DLA „MODULO”

Il.elem	Tablica 1	Tablica 10	Tablica 100	Tablica 500
10	0,0000032	0,0000035	0,0000038	0,0000018
100	0,0000092	0,0000082	0,0000065	0,000007
1000	0,0000667	0,0000758	0,0000578	0,0000654
10000	0,0004728	0,0000728	0,0000375	0,0000314
100000	0,004745	0,0023541	0,0005857	0,0002278



POMIARY DLA „MNOZENIE”

Il.elem	Tablica 1	Tablica 10	Tablica 100	Tablica 500
10	0,0000022	0,0000026	0,0000021	0,0000012
100	0,0000068	0,0000075	0,0000039	0,0000036
1000	0,0000501	0,0000628	0,0000413	0,0000458
10000	0,0004528	0,0000528	0,0000284	0,0000214
100000	0,0045231	0,0022781	0,0004557	0,0002141



Wnioski:

- Implementacja nie zapobiega duplikowaniu się elementów.

Podczas obserwacji wartości w tabeli można było zaobserwować że algorytm losowanie liczb „lubi” pewne wartości (haszu) i pojawiają się one znacznie częściej niż inne, co powodowało kolizje.

- Na podstawie danych widać, że zwiększanie wielkości tablicy pozytywnie wpływa na szybkość działania algorytmu, jednakże powstaje też wiele pustych „szufladek” tablicy, co zabiera pamięć.
- Według literatury szybkość obu funkcji jest zbliżona. W testach szybsza okazała się „MNOZENIE”
- Przy małych ilościach elementów złożoność odczytu jest zbliżona do $O(1)$, jednakże z powodu kolizji złożoność odczytu dla wielu elementu wynosi ok. $O(n)$.