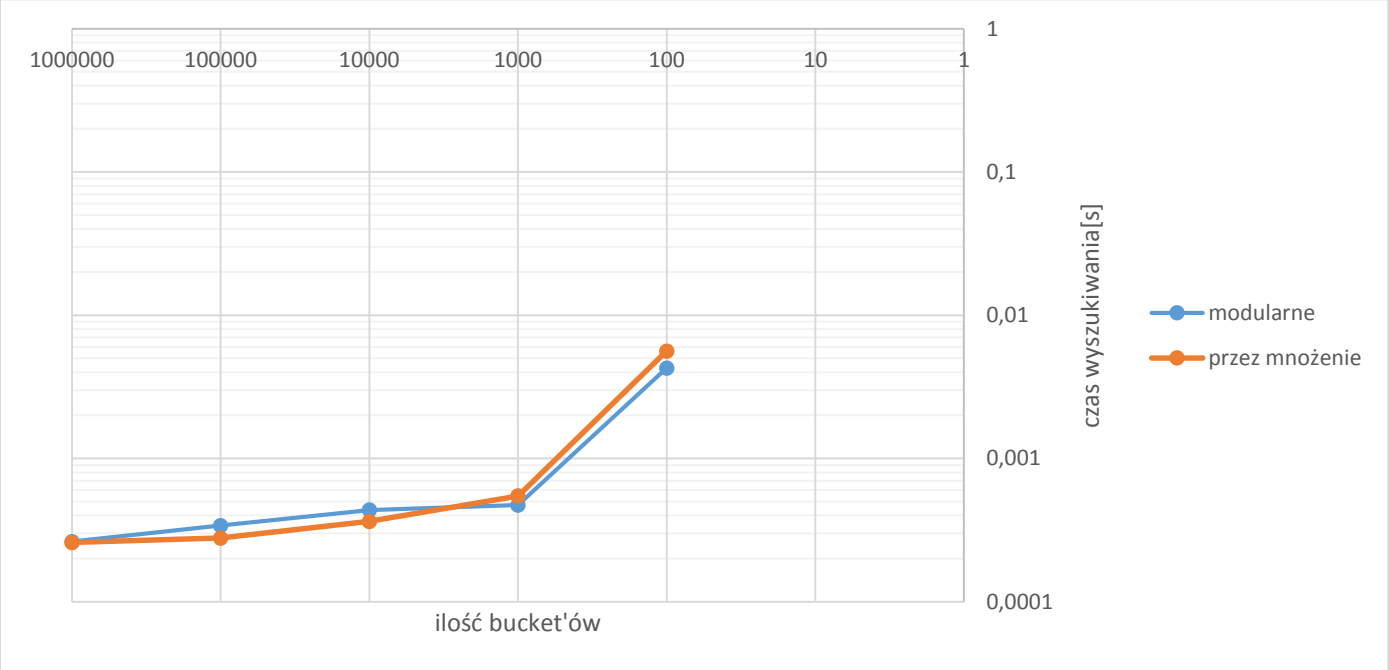
Sprawozdanie

Implementacja Tablicy asocjacyjnej

Zadanie polegało na zaimplementowaniu struktury danych odwzorowującą tablicę asocjacyjną z funkcją haszowania. Dla takiej struktury należało zmierzyć czas jednej z operacji słownikowych- wyszukiwanie.

Dla tablicy asocjacyjnej z idealną funkcją haszującą wszystkie operacje słownikowe takie jak wstawianie, usuwanie, szukanie powinny mieć złożoność O(1). W mojej implementacji posłużyłem się dwoma funkcjami haszującymi, których realizację zaczerpnąłem z książki T. H. Cormen'a „wprowadzenie do algorytmów" . Mianowicie haszowanie modularne oraz haszowanie przez mnożenie, które są jednymi z prostych funkcji nieidealnych. W swojej strukturze, zaprojektowanej na potrzeby badań, posłużyłem się kluczami, które od początku były generowane jako liczby całkowite. Problem kolizji rozwiązałem metodą łańcuchową tworząc listy jednokierunkowe. Czas wyszukiwania był mierzony względem ilości bucket'ów w tablicy. Przeprowadzono dwa testy, dla haszowania modularnego oraz przez mnożenie. Za każdym razem w tablicy było 1 000 000 elementów. Poniżej znajdują się tablica z danymi oraz wykres.

Liczba bucket'ów	Czas „modularne"[s]	Czas „Przez mnożenie"[s]
1000000	0,000263	0,000259
100000	0,000340	0,000278
10000	0,000437	0,000364
1000	0,000473	0,000548
100	0,004275	0,005623



Wnioski:

Przy implementacji tablicy z haszowaniem zauważyłem, że problemem nie jest sama funkcja haszująca, a prezentacja kluczy za pomocą liczb naturalnych. Generowane w programie klucze nigdy się nie powtarzają, co za tym idzie, przy takiej samej ilości bucket'ów w tablicy oraz danych funkcja haszująca przypisze klucze równomiernie. Przy zmniejszaniu ilości kubełków, będzie zwiększać się proporcjonalnie rozmiar list. W praktyce trzeba zastanowić się nad dobrą funkcją przypisującą kluczom ich odwzorowanie w zbiór liczb naturalnych. Funkcja dodająca do siebie jedynie

kody ASCII poszczególnych znaków klucza nie jest dobrą funkcją, ponieważ dla przykładu z grupy 3 znaków można stworzyć 6 różnych kluczy, które prezentować będzie ta sama liczba naturalna. Budowanie funkcji, która zapewniałaby jak największą unikalność liczb naturalnych prezentujących klucze, ma ogromny wpływ na usprawnienie mechanizmów tablicy asocjacyjnej.