Sprawozdanie PAMSI

24.04.2016 Zimoń Robert

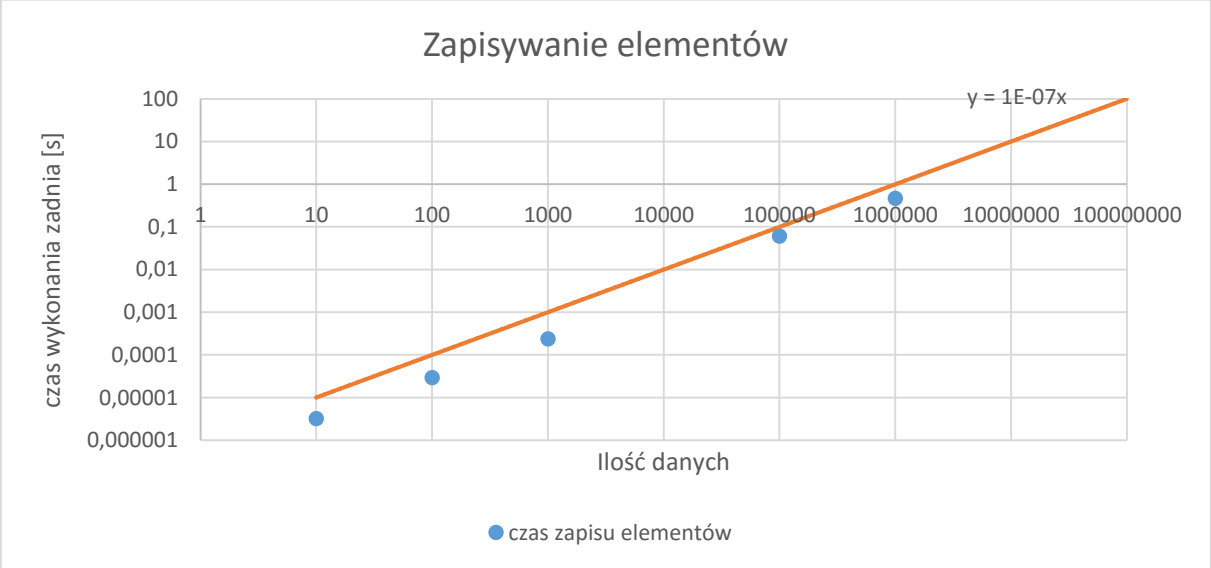
Tablica z haszowaniem 218682

Wprowadzenie:

Celem ćwiczenia było zaimplementowania i przetestowanie tablicy z haszowaniem. Badania poprawności jej działania zostały przeprowadzone dziesięciokrotnie dla każdej zadanej ilości elementów. Ilość bucketów z danymi została ustalona tak by zawsze było ich 10 razy mniej niż danych, co oznacza, że przy prawidłowym, równomiernym ich podzieleniu w każdym z pojemników było by po 10 elementów.

Badania:

1) Pierwsze z badań sprawdzało czas zapisu informacji do tablicy z haszowaniem, a wyniki zostały przedstawione na poniższym wykresie:



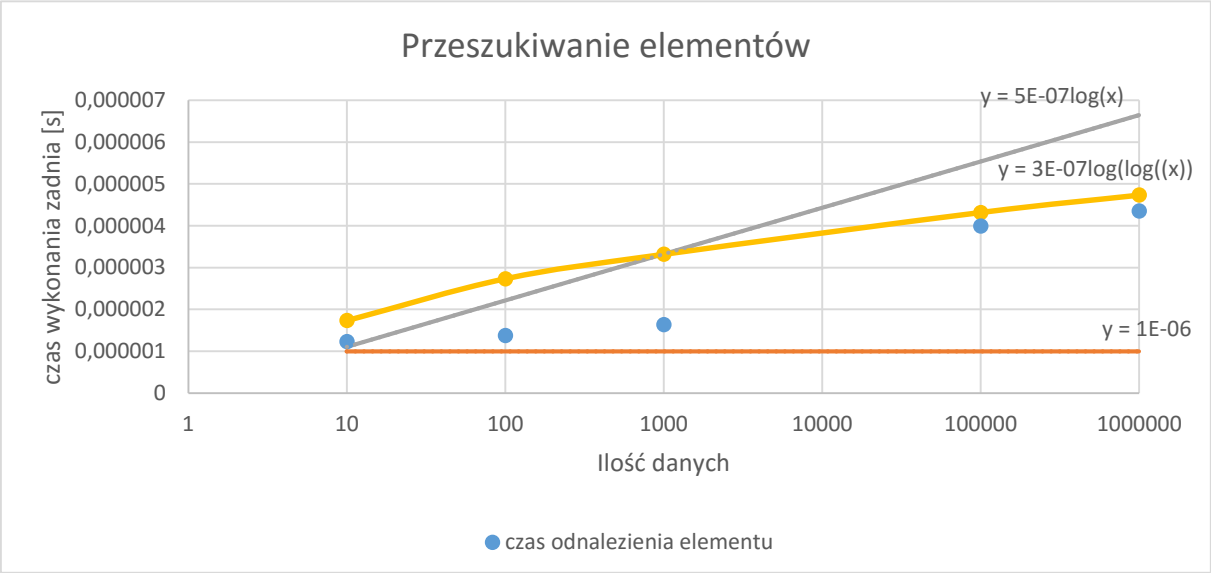
(ilość danych i czas przedstawione w skali logarytmicznej).

Zgodnie z teorią czas zapisu elementów jest liniowy co związane jest z czasem zapisu danych do poszczególnych zaimplementowanych struktur, które przechowują dane w tablicy z haszowaniem, w przypadku tego algorytmu była to lista zbudowana na tablicy dynamicznie alokującej zarezerwowaną pamięć. Wpływ na czas zapisu może mieć również funkcja haszująca, jednak w zaimplementowanym algorytmie ma on złożoność stałą O(1), więc nie zwiększa on rzędu złożoności czasowej w notacji dużego O pozostałych algorytmów.

Podsumowanie:

- algorytm poprawie zapisuje wprowadzone dane
- czas zapisu zależny jest od czasu zapisu elementów do struktury przechowującej dane takiej jak lista
- Złożoność tego algorytmu w notacji O wynosi O(n) – czas zależy liniowo od ilości danych

2) Drugie z badań sprawdzało czas odszukania informacji w tablicy z haszowaniem, a wyniki zostały przedstawione na poniższym wykresie:



(z powodu bardzo małego przyrostu czasu na wykresie został on przedstawiony w skali liniowej, a ilość danych jak poprzednio - w skali logarytmicznej).

Zgodnie z teorią czas dostępu do elementu powinien być stały i nie zależeć od ilości danych. Przeprowadzone badania wykazały jednak, że w przypadku zaimplementowanego algorytmu czas odbiega od liniowego i zbliża się do zależności podwójnie logarytmicznej $O(\log_2(\log_2(n))$. Powodem tego może być niedoskonałość algorytmu haszującego powodując nierównomierne przydzielanie miejsca lub całkowitym niewykorzystaniem części z bucketów.

Podsumowanie:

- algorytm poprawie wyszukuje dane
- czas wyszukiwania nie powinien być zależny od ilości danych
- złożoność zaimplementowanego algorytmu wyszukiwania jest trudna do oszacowania, dla małej ilości danych zbliżona jest do stałej złożoności O(1), przy większych ilościach następuje skok do złożoności $O(\log_2(\log_2(n))$
- niezgodność teoretycznego czasu wyszukania elementu z faktycznym czasem wykonania zaimplementowanego algorytmu przeszukiwania związana jest prawdopodobnie z niedoskonałością algorytmu haszującego