

Tablica asocjacyjna, haszująca oraz drzewa binarne  
BST - Laboratorium nr 7 z PAMSI

Justyna Klijewska

22 04 2014

## 1 Wstęp

Celem ćwiczenia było przetestowanie implementacji (zmierzenie poszczególnych czasów) za pomocą:

- tablicy asocjacyjnej (wykonanej w tym przypadku na liście)
- tablicy haszującej
- drzewa binarnego BST. Program został przetestowany w środowisku Windows.

## 2 Teoria

### Drzewo binarne

W teorii grafów jest to drzewo, w którym stopień każdego wierzchołka wynosi maksymalnie 2. W najgorszym przypadku jego złożoność jest liniowa - dzieje się tak, kiedy drzewo składa się tylko z jednej gałęzi. W przypadku optymistycznym tak konstruujemy drzewo, aby zawsze po lewej stronie znajdowały się elementy nie większe, a po prawej większe. Następnie wyszukując dany element odrzucamy zawsze po połowie naszych elementów, co wpływa na szybki czas.

### Tablica haszująca

Ponieważ zmieszczenie wszystkich kluczy i elementów dużej tablicy w pamięci byłoby niemożliwe, stosujemy tablicę haszującą. Dzięki temu możemy uzyskać szybki dostęp do przechowywanych informacji. W najprostszym przypadku wartość funkcji mieszającej, obliczona dla danego klucza, wyznacza dokładnie indeks szukanej informacji w tablicy. Jeżeli miejsce wskazywane przez obliczony indeks jest puste, to poszukiwanej informacji nie ma w tablicy. W ten sposób wyszukiwanie elementu ma złożoność czasową  $O(1)$ . Jednak w sytuacji tej pojawia się problem **kolizji**, to znaczy przypisania przez funkcję mieszającą tej samej wartości dwóm różnym kluczom. Pesymistyczny przypadek to taki, kiedy mamy liniową złożoność, a optymistyczny kiedy jest ona zależna tylko od czasu wykonywania funkcji haszującej.

### 3 Tabele pomiarowe, wykresy

Tablica asocjacyjna

Ilość danych	Czas [s]
10	0
100	0.001
1000	0.002
10000	0.031
100000	0.400
1000000	3.921

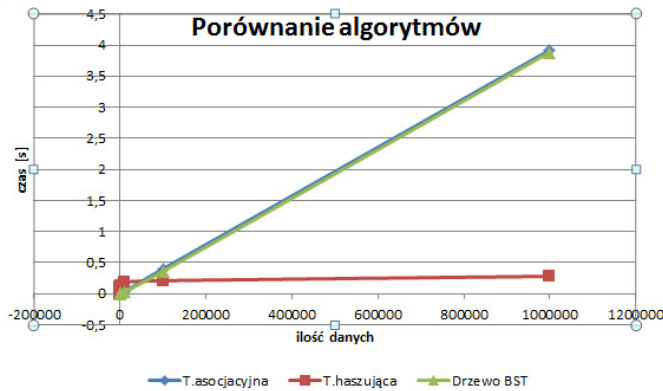
Tablica haszująca

Ilość danych	Czas [s]
10	0
100	0.1
1000	0.13
10000	0.16
100000	0.21
1000000	0.28

Drzewo binarne BST

Ilość danych	Czas [s]
10	0
100	0.001
1000	0.002
10000	0.029
100000	0.362
1000000	3.887

Wykres dla powyższych algorytmów



Rysunek 1: Wykres zależności liczby elementów od czasu implementacji dla powyższych algorytmów.

## 4 Wnioski

- Przedstawione powyżej algorytmy cechują się dużą szybkością (nawet dla milionowego rozmiaru danych wykonanie algorytmu nie trwa długo).
- Z powodu szybkiego wyszukiwania powyższe algorytmy wykorzystuje się przy tworzeniu słowników.
- Przy dużej ilości danych, można zauważyć, że tablica haszująca jest efektywniejsza niż drzewo binarne.
- Powyższe algorytmy mają złożoność obliczeniową:

Algorytm	Złożoność Pesymistyczna	Złożoność Typowa	Złożoność Optimistyczna
T. asocjacyjna		$O(\log_2 n)$	
Drzewo binarne BST	$O(n)$		$O(\log_2 n)$
T. haszująca	$O(n)$		$O(1)$