

Sprawozdanie LAB7 - Drzewa

Arkadiusz Ziółkowski

20.05.2015r

1 Cel ćwiczenia

Celem ćwiczenia jest implemetacja drzewa binarnego (BST) oraz drzewa AVL. Następnie należy zbadać złożoności obliczeniowe zapisu danych do wspomnianych struktur, oraz złożoność ich przeszukiwania.

2 Teoretyczna złożoność obliczeniowa

2.1 Drzewo binarne BST

1. Zapis danych

- Ponieważ nowe węzły drzewa są umieszczane jako liście to w celu zapisania nowej wartości w drzewie należy wykonać tyle porównań ile wynosi wysokość drzewa.
- Wysokość drzewa binarnego wyraża się wzorem $h = \log_2 n$ więc średnia złożoność obliczeniowa zapisu do drzewa wyraża się w $O(\log_2 n)$.
- Należy zwrócić uwagę, że w drzewie BST nie ma mechanizmów wymuszających zachowanie struktury drzewa wobec czego drzewo w pesymistycznym wypadku może przyjąć strukturę listy w której pesymistyczna złożoność obliczeniowa zapisu wyrażać się będzie w $O(n)$ ponieważ w celu umieszczenia nowej danej trzeba będzie przejść całą "listę" od korzenia aż do liścia.

2. Przeszukiwanie

- Przeszukując drzewo zaczynamy od jego korzenia więc schodząc coraz niżej ilość elementów do przeszukania maleje dwukrotnie. Wobec tego złożoność obliczeniowa przeszukiwania drzewa wyraża się w $O(\log_2 n)$.
- Ponieważ, w przypadku zapisu ustanieje możliwość degradacji struktury drzewa do listy to pesymistyczna złożoność obliczeniowa przeszukiwania wyraża się w $O(n)$.

2.2 Drzewo AVL

1. Zapis danych

- Ponieważ drzewo AVL posiada mechanizm wymuszający zachowanie struktury drzewa którego złożoność obliczeniowa jest pomijalnie mała (wykonanie max dwóch rotacji) to zapis do tego drzewa będzie wyrażał się w $O(\log_2 n)$, ponieważ wysokość drzewa nigdy nie przekroczy wartości $\log_2 n + 1$.

2. Przeszukiwanie

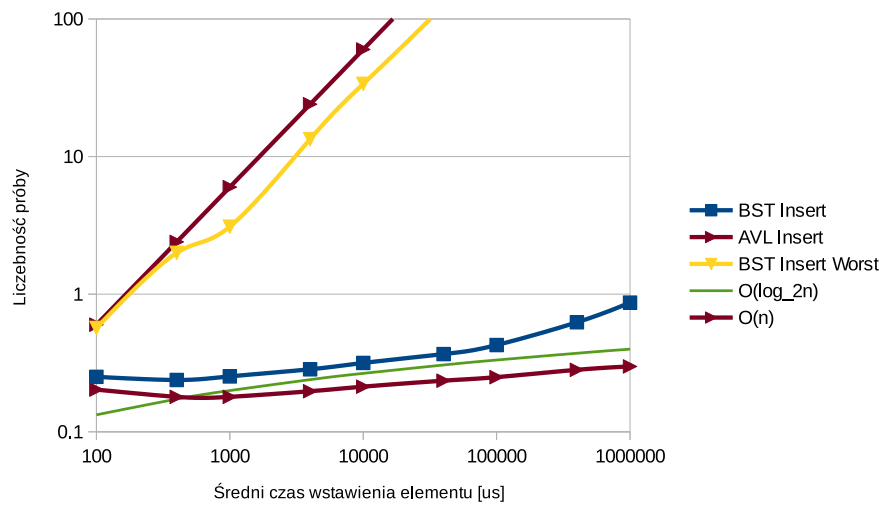
- Zgodnie z powyższym odczyt z takiego drzewa również wyraża się w $O(\log_2 n)$.

3 Wyniki pomiarów

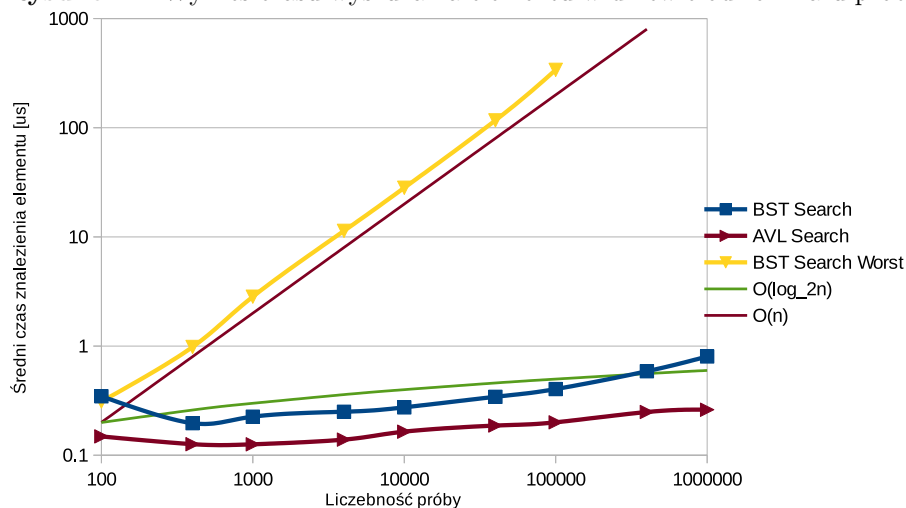
Pomiary czasu wykonywania algorytmów zostały wykonane na podstawie losowo wygenerowanych danych wejściowych po 50 razy dla każdej ich liczności. Dodatkowo został wygenerowany przypadek pesymistyczny polegający na wczytywaniu danych ułożonych w kolejności rosnącej.

Ilość elementów	Średni czas dla zapisu / znalezienia pojedynczego elementu [us]					
	Zapis			Wyszukanie		
	BST	BST pesym.	AVL	BST	BST pesym.	AVL
10^2	0.2508	0.5680	0.2026	0.3466	0.3100	0.1486
$4 * 10^2$	0.2379	1.9980	0.1794	0.1964	0.9810	0.1261
10^3	0.2525	3.0940	0.1795	0.2254	2.8328	0.1258
$4 * 10^3$	0.2845	13.4006	0.1971	0.2501	11.3613	0.1389
10^4	0.3162	33.7410	0.2176	0.2743	28.2026	0.1642
$4 * 10^4$	0.3669	125.6100	0.2351	0.3426	116.9067	0.1868
10^5	0.4269	347.2000	0.2494	0.4034	339.3180	0.2000
$4 * 10^5$	0.6248	-	0.2820	0.5877	-	0.2486
10^6	0.8682	-	0.2982	0.8024	-	0.2610

Tabela 1. Wyniki pomiarów



Rysunek 1. Wykres czasu wyszukiwania elementu w drzewie od rozmiaru próby



Rysunek 2. Wykres czasu wyszukiwania pojedynczego elementu od rozmiaru próby

4 Wnioski

- Zgodnie z wykresem zaprezentowanym na rysunku nr 1 złożoność obliczeniowa algorytmu zapisu elementu do drzewa binarnego (BST) jest większa od logarytmicznej, a dla pesymistycznego przypadku charakteryzuje się złożonością liniową.
- Na rysunku nr 1 widać, że złożoność obliczeniowa algorytmu zapisu elementu do drzewa AVL wyraża się w $O(\log_2 n)$.
- Z rysunku nr 2 wynika, że średnia złożoność wyszukiwania elementu w drzewie binarnym BST jest większa od logarytmicznej, natomiast dla przypadku pesymistycznego jest ona liniowa.

- Na tym samym wykresie widać, iż złożoność obliczeniowa algorytmu wyszukiwania elementu w drzewie AVL wyraża się w $O(\log_2 n)$.