

AVL Binary Tree

version 2.1

Generated by Doxygen 1.8.9.1

Wed May 20 2015 18:44:41

Chapter 1

Sprawozdanie

1.1 Zadanie Laboratoryjne

Zadaniem była implementacja drzewa binarnego typu AVL bądź czerwono-czarnego i zastosowanie używanego wcześniej frameworka do zbadania złożoności czasowej zapisywania i odnajdywania elementów w tym drzewie.

1.2 Implementacja

Zaimplementowano drzewo AVL (patrz: [documentation.pdf](#)) w którym możliwy jest zapis elementu o określonym, unikatowym kluczu, odnalezienie tego elementu jak również usunięcie wszystkich elementów drzewa i usunięcie go.

1.3 Złożoność czasowa

Przy wykorzystaniu używanego wcześniej stopera dostępnego w bibliotece libcommon.a za pośrednictwem plika nagłówkowego Observer.h dokonano pomiarów czasu dodawania i wyszukiwania elementów dla drzewa o rozmiarach do tysiąca elementów (wyniki dostępne w pliku logfile.txt i observerData.txt w katalogu AVL).

Niestety, pomiar czasu jest nie dość dokładny - algorytm wykonuje zarówno zapisywanie jak i wyszukiwanie w jednakowym czasie 2.18428e-307 s.

1.4 Możliwe rozszerzenia

Zastosowana struktura obiektu AVLTree pozwala na łatwe dopisywanie metod takich jak usuwanie konkretnego elementu czy wyszukiwanie elementu ze zwracaniem wskaźnika na niego.

Stosunkowo łatwo będzie przekształcić drzewo w strukturę przechowującą dane w formacie string (bądź dowolnym innym) w formie podobnej do tablicy haszującej, ale z jeszcze szybszym od tamtej dostępem do konkretnych elementów. W takiej sytuacji struktura Node byłaby modyfikacją klasy Cell z wykonanego na jednym z poprzednich laboratoriów hashArray.