Maciej Zajkowski EADS LAB3 – AVL Tree 23.05.2021

Task: To design AVL Tree class and write two functions that will use it.

Public methods description:

AVL_TREE();

AVL_TREE(const AVL_TREE& a);

~AVL_TREE();

bool operator ==(const AVL_TREE&);

int getcount(); // returns number of elements

bool CheckParameter(const Key key); // checks if there exist element witch such key if exists returns true otherwise returns false.

bool Insert(Key newKey, Info newinfo); // inserts key and balances tree in 4 ways right -> left rotation right rotation left rotation and left->right rotation if there is already such key in tree returns false increases number_of_elements

bool Removal(Key where); - removes element like in BST and balances it in similar way like in insertion

bool GetInfo(Key where, Info& info); returns true if key exist and changes info input in data stored in element

bool ChangeInfo(Key where, Info& info);- returns true if key exist and changes info

bool Print(bool flag); if true prints tree from left to right else from right to left

bool ReturnSmallest(Key& key, Info& info); - returns parameters of smallest element. If head is empty returns false.

bool Visual_Print(); - prints every element due to hight starting from head (basically drawing of tree)

AVL_TREE<Key, Info>& operator = (AVL_TREE<Key, Info>& b) – self explanatory

bool ReturnBiggest(Key& key, Info& info); - same thing but for biggest element

Testing :

Test 1 - tests every type of printing

Test 2 - tests right rotation for insertion

Test 3 – tests left rotation for insertion
Test 4 - tests right - left rotation for insertion
Test 5 – tests left - right rotation for insertion
Test 6 - tests right rotation for deletion
Test 7 – tests left rotation for deletion
Test 8 - tests right - left rotation for deletion
Test 9 – tests left - right rotation for deletion
Test 10 – tests operator ==
Test11 – tests copy constructor

Test 12 – tests assignment operator

Test 13 – tests getcount

Test 14 - getting and changing info

Function counter – it works almost properly but it considers “(word” or “word,” as different from “word”;