American University of Armenia
CS 121 Data Structures A

Homework Assignment 6.1

1. (3 points) Your task is to write a Java/C++ program that reads a piece of text and counts the number of occurrences of each word. In order to achieve this do the following.

(a) Implement the map ADT in two different ways: one based on an unsorted arraylist/vector and another based on a sorted arraylist/vector. Your implementations can be based on the code given in the textbook.

(b) Write a function that reads a piece of text, uses a map to maintain the statistics about word occurrences, and prints all the words along with the numbers of occurrences.

(c) Test your program on a few small examples using both variants of maps. Do both variants work the same way? In what way are the produced outputs different?

(d) What is the worst-case running time of your program? Justify your answer.

2. (4 points) Your task is to implement an AVL tree by directly inheriting from the `LinkedBinaryTree` class from the previous homework. Note that nodes in an AVL tree store an additional piece of information: the current height of the node. You may need to override some of the functions from the parent `LinkedBinaryTree< Entry<K,V> >` class in your `AVLTree<K,V>` class. In addition, you need to implement functions that:

(a) performs a recursive search in the AVL tree; parameters: a position and a key; return type: a position

(b) inserts a key-value pair into the AVL tree; parameters: a key and a value; return type: value/iterator

(c) removes the entry with given key; parameters: a key; return type: value/none

(d) rebalances the subtree rooted at a given position by applying trinode restructuring and returns the position of the new root of this subtree.

Since your insertion/removal functions need to rebalance the tree, they need to make use of the rebalancing function.

3. (2 points) Implement the word-counting program from Task 1 using a map based on an AVL tree. You only need to add a map class that relies on an AVL tree using adapter pattern. What is the advantage of this implementation?