American University of Armenia
CS 121 Data Structures A

Homework Assignment 5

1. (6 points) Provide a full binary tree implementation using a linked structure. Your implementation can be based on the code given in the textbook. Complete your classes with functions:

(a) to calculate the depth of a given position;

(b) to calculate the height of a given position;

(c) to calculate the height of the tree itself;

(d) to print the elements at all the positions in the tree using preorder traversal (non-recursive using one stack);

(e) to print the elements at all the positions in the tree using postorder traversal (non-recursive using two stacks);

(f) to print the elements at all the positions in the tree using breadth-first traversal;

(g) to print the elements at all the positions in the tree using inorder traversal (non-recursive using one stack).

(h) Test your binary tree class and its functionality on a couple of examples. Briefly discuss the results.

2. (2 points) As per the discussion on page 301 of the C++ textbook, you can use inorder traversal to make a drawing of a binary tree. Write a program that first constructs a binary tree of 15 integer elements and prints a drawing of that tree.

3. (3 points) Your task is to implement two types of sorting algorithms based on priority queues.

(a) Implement a function that, given a list and a comparator, performs the sorting scheme with a priority queue.

(b) Selection sort: provide a priority queue implementation using an unsorted list. Your implementation can be based on the code given in the textbook.

(c) Insertion sort: provide a priority queue implementation using a sorted list. Your implementation can be based on the code given in the textbook.

(d) Test both selection and insertion sort (using the function from part (a) combined with classes from (b) and (c), respectively) on a few examples.

(e) What are the best- and worst-case running times of these implementations of selection and insertion sort? Explain your answer.

4. (2 points) Your task is to implement heap sort.

(a) Provide a priority queue implementation using a heap. Your implementation can be based on the code given in the textbook.

(b) Test heap sort (using the function from part (a) of previous task combined with the class from (a) of this task) on a few examples.

(c) For input sequence {5, 6, 4, 7, 3, 8, 2, 9, 1, 10} explain the execution of your heap sort implementation step-by-step using both a tree and an array for visualisation of the heap.

(d) What are the best- and worst-case running times of this implementation of heap sort? Explain your answer.

5. (4 points) Your task is to implement in-place heap sort for arrays (see pages 388–389 and 351–352 of the Java and C++ textbooks, respectively).

(a) Provide a heap sort function implementation based on the description in the textbooks.

(b) Test this heap sort (using the function from part (a)) on a few examples.

(c) For input sequence {5, 6, 4, 7, 3, 8, 2, 9, 1, 10} explain the execution of this heap sort implementation step-by-step using a tree for heap visualisation and giving the updated array.

(d) What are the best- and worst-case running times of this implementation of heap sort? Explain your answer.

(e) What are the space requirements for the two implementations of heap sort (the previous task and this one)? Explain your answer.