Assignment 3

Timothy Zwart Cmpt\_435\_11\_18F

**Date Assigned: 09/14/2018**

**Due: Midnight 09/21/2018 on iLearn**

**Please read turn-in checklist at the end of this document before you start doing exercises.**

**Section 1: Pen-and-paper Exercises**

1. Arrange the following functions in ascending order of growth rate (7 points):



You are NOT required to justify your ordering.



Note:

In this problem, you are asked to identify if f1(n) < f2(n) for a “sufficiently large” input size n. However, for small values of n this is not always true.

1. For each of the following tasks, find the complexity of the algorithm using big O notation. You must justify your answer with 1-2 lines of explanation.
2. Insert a new element into an unsorted array.

O(n)+O(1)= O(n) looking trough array so O(n) then inserting into an array is O(1);

1. Insert a new element into a sorted array.

O(n)+O(1)= O(n) looking trough array so O(n) then inserting into an array is O(1)

1. Remove the minimum element in an unsorted array.

O(n)+O(n)=O(n) look for array to get the minimum element so O(n), then when it is removed all of the elements must be shifted so O(n);

1. Remove the minimum element in a sorted array.

O(1)+O(n)=O(n) Do not have to search for the element because already sorted to one end of the array. Removing is O(1), but then to shift over all elements you must perform O(n).

1. In this course, we have studied the following data structures: an unsorted array, linked list, stack, queue, and hashtable. For each of the following applications, indicate which of these data structures would be most suitable and give a brief justification for your choice.
2. Customers at the airport check-in counter 🡪 Queue 🡪 first in first out, first person at counter is served and then they get to leave

Note:

(1 point) data structure.

(1 point) justification for your choice.

1. A list of items, random access. 🡪 unsorted array 🡪 unsorted array has no set order just placed based upon a contiguous block of memory
2. A list of items, output items in opposite order. 🡪 Stack 🡪 first in last out. The stack places the next element on top of the other element. So as you take elements out it the order of how they were put in is reversed as the out put.
3. A list of items, constant time search and update. 🡪 Hash table 🡪 acts with multiple linked lists and connects each element with the elements related to it.

**Section 2: Java Implementation**

1. Create a linked list and remove the minimum element in the list in Java.

Note:

Find a file called LinkedList.java in assignment 1 folder.

Complete the method of Removemin().

Test your method in the main method provided following the comments.

**Important: In all of the assignments of this course, when you are asked to implement an algorithm for a problem, your code will be evaluated based on:**

**5 points - Execution**

**Each file must run without error or warning on valid input described in the main method provided.**

**5 points - Within Code Documentation**

**Is the code documented for obvious understanding of the use, preconditions, and postconditions of each function?**

**20 points - Correctness**

**Is the algorithm implemented correctly? Does your method pass the test?**

**TURN-IN CHECKLIST:**

1. **Answers to Section 1 (.doc/.txt/.pdf), and to Section 2 (all your source Code (.java files)). Remember to include your name, the date, and the course number in comments near the beginning of your code/report.**
2. **Create a folder and name it 'FirstName\_LastName\_assignment\_1'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and push it to iLearn.**