New **Y**ork **U**niversity
Computer Science Department
Data Structures
Dr. Anasse Bari

Homework Two: Recursion, Searching, and Sorting Algorithms

Deadline: *See NYUclasses for the deadline, 15% off per day after the deadline (4 days maximum).*

Learning Objectives:

- Learning how to design recursive algorithms
- Learning how to derive the running time of algorithms
- Learning major sorting and recursive sorting
- Analyzing sorting' running time

Read the guidelines bellow carefully to avoid receiving a zero grade on the HW:

- Attach the Java source files and include them into HW's zip file. The file name should be **YourLastName_HW2.zip**
- Make an archive (zip file or compressed file) with all the **java files (the .java files NOT the .class files)** and post it on NYU Classes
- You must comment you code (basic comments explaining the role of a class, a method or variables used in your submission)
- Compile and run the program before you submit.
- It is your responsibility to make sure if the Zip files has your actual latest files. You may send the file to yourself by email to double check that is the actual file before you upload on NYU classes.
- **If the graders cannot open the file, you will receive a grade of zero.**
- **If you send the .class files instead of the .java files (source files) you will receive a zero.**
- An act of cheating will be severely addressed with an immediate zero on the homework and a report to the academic advisor and the administration.
- You will automatically lose 50% of the points for an exercise if the program does not compile and run correctly.
- **Plagiarized assignments will get a ZERO grade**. You cannot change the variable names of other student's solution and submit it as yours. The program structure of other students must not match yours. Every student must come up with his/her own solution. Any cheating (e.g. copying from internet without citing sources) is a serious violation of the University student code.
- **Homeworks sent by *email* to the instructor or to the graders will NOT be reviewed and will not be accepted.**

Maximum Number of Points: 120pts

Instructions: Submit your code (*.java code not the *.class files) and include a word document that has your answers to the exercises that do not require programming. Please organize your answers very clearly.

I. Arrays (20pts)

1. Exercise One (10pts) Design and implement and algorithm that compare two strings and check if they are reversed. For instance, if the two strings are google and elgoog, then the method returns 1. If the two strings are "data", "ata" then the method returns 0. The algorithm should ignore white spaces, lower/upper cases.
   Specify the running time of your algorithm in terms of Big-O notation.
   Attach Exercise One.doc that explains the running time.

2. Exercise Two (10pts) Design and implement an algorithm that takes a set of strings. Then, for every three consecutive strings (assuming the list of strings is larger than 3 and the number of strings in your sentence is a multiple of 3) leave only the shortest of the three. In case two string among the three are of the same length then you leave the first one.

   For instance, consider the following set of strings: "Other entries include a historic district **in Charlottesville Virginia** cut-flower greenhouse complex" → "Other a in complex"

   Specify the running time of your algorithm in terms of Big-O notation.
   Attach Exercise Two.doc that explains the running time.

II. Recursion (50pts)

1. (10pts) Design and implement a *binary recursive algorithm* that finds the maximum number in array of size n.

2. (20pts) Design and implement a *recursive* algorithm that returns the number of zeroes in the binary representation of N.
   In this exercise the input to the algorithm must be an integer N. Your algorithm should not calculate the binary representation of N but recursively finds the number of zeroes in the binary representation of N. You are NOT allowed to use arrays.

3. (20pts)Implement a recursive algorithm that checks if word is A palindrome.
   A palindrom is a word or number that is identical when read both forwards and backwards.
   Please ignore all punctuation.
   Some simple palindromes include
   kayak
   radar
   dad

4	77	98	30	20	50	77	22	49	2

III. Sorting (60pts)

3.1. Provide your own implementation of the four algorithms mentioned above. You must provide your own code. Code copied form the internet or modified will be result into a Zero grade for the entire homework.

Provide a table (attached in a word document) where you compare the running time of the sorting algorithms mentioned bellow in terms of best case and worst case running time.

1. Bubble Sort Non-Recursive (5pts)
2. Bubble Sort Recursive (5pts)
3. Selection Sort (Non-recursive) (5pts)
4. Insertion Sort (Non-recursive) (5pts)
5. Merge Sort (Recursive) (5pts)
6. Quick Sort (Recursive) (5pts)

3.2 Consider the array mentioned above, show a step by step illustration (either by hand or written in a word document) of applying the sorting algorithms mentioned to the array.
Include all your answers for this HW into a zip file.