DIT181 Data Structures and Algorithms: Assignment 1

Maximum points you can get from Assignment 1 is **100 points**. Distribution of points within questions has been given in parentheses below.

- **Criteria to pass "Assignments" sub-course (3.5 HEC)**: From each assignment (i.e. Assignment #1, Assignment #2 and Assignment#3) you have to get minimum 50 points (out of 100 points for Assignments #1and #2, and out of 130 points for Assignment #3).
-

- **Information about Deliverables:** You will submit a pdf file for the answers to questions that do not require programming. For the answers consisting of implementation in Java, you will submit the java skeleton file that you worked on and completed.

- **Submission Deadline:** 11.02.2018 at 08:00 am in the morning SHARP!!

Dynamic Arrays

(Use `Array2.java` skeleton code to solve the following questions about "Dynamic Arrays")

Question 1 Implement the method `reverse()`, which should reverse the array in place (i.e., without using an additional array) **(8 points)**. Determine big-O time complexity of the method `reverse()`**(2 points)**.

Question 2 Suppose that you are going to implement removing an element of a given index from the array. What would be the time complexity of this operation? **(2 points)** Implement it as the method `remove()` **(5 points)** What would be the big-O time complexity if you were allowed to change the order of the remaining elements in the array? **(2 points)** Implement it as the method `remove2()`. **(5 points)**

Question 3 Implement the method `find(int x)`, which should return the index of the first occurrence of `x` in the array **(5 points)** and `-1` if `x` does not occur in the array **(5 points)**. Determine the big-O time complexity of the method `find(int x)` **(2 points)**.

Question 4 Implement the method `maxPalindrome()`, which should find the length of the longest palindrome that is a contiguous subsequence of the array. A palindrome is a word that is symmetric. For example, palindromes may have the form `ABCBA` (length 5) or `ABCCBA` (length 6). Come up with the best solution you can in reasonable time **(8 points)**. Determine the big-O time complexity of the method `maxPalindrome()`**(2 points)**.

Complexity

Question 5 Determine the big-O time complexity classes of the following functions:

a. $\sum_{i=1}^{n} i^2$ **(6 points)**

b. $\sum_{i=1}^{\log n} 2^i$ **(6 points)**

c. $2x^2 + 16x^3$ **(6 points)**

d. $(x - \log x^3)\left(x - 2\sqrt{x}\right) + 4x \log x^2$ **(6 points)**

Question 6 Show that $4^n \notin O(2^n)$ **(10 points)**

Question 7 Consider the problem of evaluating a single variable polynomial of degree n:

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n$$

The value of the polynomial for a particular value of x, can be computed by performing some sequence of additions and multiplications. During Problem Session #1, you were shown the pseudocode for the naïve algorithm that can perform it using $O(n^2)$ operations, where an operation is either an addition or a multiplication. With the guidance of the TAs/teacher, you were also expected to write the pseudocode for an improved algorithm where we speeded up the computation of x^n, while keeping the formula the same. The improved algorithm used $O(n)$ operations.

You are now expected to come up with another improved algorithm that uses a different formula resulting in $O(n)$ operations to calculate the value of the polynomial for a particular value of x. **(20 points)**