Homework 1 Part 2
CSE 246 Analysis of Algorithms, Spring 2018
Due 03.04.2018, Tuesday (Midterm Exam)

1. (a) An array A[0..n-2] contains n-1 integers from 1 to n in increasing order. (Thus one integer in the range is missing.) Design the most efficient algorithm you can to find the missing integer and indicate its time efficiency.

(b) Repeat (a) for an unsorted array (instead of an array in increasing order).

(c) Now consider A[0..n-3] which contains n-2 integers from 1 to n in increasing order. (Thus two integer in the range are missing.) Design the most efficient algorithm you can to find the missing integers and indicate its time efficiency.

2. For each of the following problems design **a divide-and-conquer** algorithm. Code your algorithm in a programming language of your choice.
For each of the problem give the following:

i. Step-by-step description of your algorithm.
ii. Code (No need to give in a seperate file, just a printed copy inside the HW report)
iii. At least three sample input/output.

a) Problem 1: Consider an integer list A[0..n-1] that includes negative and positive integers. Among all subsequences in this list, find the sum of the subsequence that has largest sum.
Example: For the input: [-2 -5 6 -2 -3 1 5 -6]; Output is: 7.
Note: In this example, subsequence with the maximum sum is [6 -2 -3 1 5].

b) Problem 2: Consider an integer list A[0..n-1] that includes negative and positive integers. Find the longest all-negative subsequence.
Example: For the input: [2, 5, 0, -3, -5, 0, -1, -2, -1, 2]; Output is: [-1, -2, -1]

c) Problem 3: Consider a binary list B[0..n-1] that includes 0's and1's. Find the length of the longest alternating subsequence 010101......
Example: For the input: [0, 1, 0, 1, 0, 0, **0, 1, 0, 1, 0, 1, 0, 1**, 1]; Output is: 8.
Another example: For the input: [0, 1, 0, 1, 0, 0, **0, 1, 0, 1, 0, 1, 0,** 0, 1]; Output is: 7.
Note: Longest alternating subsequences are shown in bold.

3. Let A = { a_1 , ... a_n } and B = { b_1 , ... b_m } be two sets of numbers and $m=n^2$. Consider the problem of finding their intersection, i.e., the set C of all the numbers that are in both A and B. Design an efficient algorithm for solving this problem and determine its **efficiency class in terms of n**.

Note: Please do not give pseudocode. Only give a step-by-step description. **Your algorithm should perform better than brute-force**.

Note: You are supposed to answer all the questions but selected questions will be graded.