E-JUST الجامعة المصرية اليابانية للعلوم والتكنولوجيا Egypt - Japan University of Science and Technology エジプト日本科学技術大学	**Assignment 6: Spring 2020**
Department	Computer Science and Engineering
Submission due	Tue, Apr 14th, 2020 by 9:30am
Date	05/04/2020
Course Title	CSE 326: Analysis and Design of Algorithms
Instructor	Prof. Walid Gomaa
Allowed Equipment	None

Answer all of the following questions

1. (a) Let $A[1,\ldots,m]$ and $B[1,\ldots,n]$ be two arbitrary arrays. A *common subsequence* of A and B is both a subsequence of A and a subsequence of B. Give a simple recursive definition for the function $lcs(A, B)$, which gives the length of the longest common subsequence of A and B.

   (b) Let $A[1,\ldots,m]$ and $B[1,\ldots,n]$ be two arbitrary arrays. A *common supersequence* of A and B is another sequence that contains both A and B as subsequences. Give a simple recursive definition for the function $scs(A, B)$, which gives the length of the shortest common supersequence of A and B.

   (c) Call a sequence $X[1,\ldots,n]$ of numbers *bitonic* if there is an index i with $1 < i < n$, such that the prefix $X[1,\ldots,i]$ is increasing and the suffix $X[i,\ldots,n]$ is decreasing. Give a simple recursive definition for the function $lbs(X)$, which gives the length of the longest bitonic subsequence of an arbitrary array X of integers.

   (d) Call a sequence $X[1,\ldots,n]$ *oscillating* if $X[i] < X[i+1]$ for all even i,and $X[i] > X[i+1]$ for all odd i. Give a simple recursive definition for the function $los(X)$, which gives the length of the longest oscillating subsequence of an arbitrary array X of integers.

   (e) Give a simple recursive definition for the function $sos(X)$, which gives the length of the shortest oscillating supersequence of an arbitrary array X of integers.

   (f) Call a sequence $X[1,\ldots,n]$ *convex* if $2 \cdot X[i] < X[i-1] + X[i+1]$ for all i. Give a simple recursive definition for the function $lxs(X)$, which gives the length of the longest convex subsequence of an arbitrary array X of integers.

2. For each of the following problems, the input consists of two arrays $X[1,\ldots,k]$ and $Y[1,\ldots,n]$ where $k \leq n$.

   (a) Describe a recursive algorithm to determine whether X is a subsequence of Y. For example, the string PPAP is a subsequence of the string PENPINEAPPLEAPPLEPEN.

   (b) Describe a recursive algorithm to find the smallest number of symbols that can be removed from Y so that X is no longer a subsequence. Equivalently, your algorithm should find the longest subsequence of Y that is not a supersequence of X . For example, after removing two symbols from the string PENPINEAPPLEAPPLEPEN, the string PPAP is no longer a subsequence.

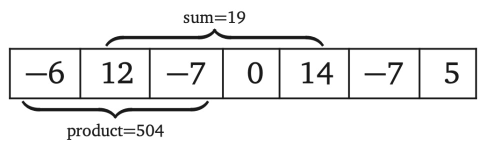
   (c) Describe a recursive algorithm to determine whether X occurs as two disjoint subsequences of Y . For example, the string PPAP appears as two disjoint subsequences in the string PENPINEAPPLEAPPLEPEN.

3. Suppose you are given an array $A[1, \ldots, n]$ of numbers, which may be positive, negative, or zero, and which are not necessarily integers.

   (a) Describe and analyze an algorithm that finds the largest sum of elements in a contiguous subarray $A[i, \ldots, j]$.

   (b) Describe and analyze an algorithm that finds the largest product of elements in a contiguous subarray $A[i, \ldots, j]$.

   For example, given the array $[-6, 12, -7, 0, 14, -7, 5]$ as input, your first algorithm should return 19, and your second algorithm should return 504.



   For the sake of analysis, assume that comparing, adding, or multiplying any pair of numbers takes $\mathcal{O}(1)$ time.