E-JUST — Egypt - Japan University of Science and Technology — الجامعة المصرية اليابانية للعلوم والتكنولوجيا — エジプト日本科学技術大学	**Assignment 7: Spring 2020**
Department	Computer Science and Engineering
Submission due	Tue, Apr 21st, 2020 by 9:30am
Date	12/04/2020
Course Title	CSE 326: Analysis and Design of Algorithms
Instructor	Prof. Walid Gomaa
Allowed Equipment	None

Answer all of the following questions

1. Assume an $n \times n$ matrix A of distinct integer numbers. Assume A is sorted such that $A[i, j] < A[i, j']$ whenever $j < j'$ and $A[i, j] < A[i', j]$ whenever $i < i'$. Give an efficient algorithm to search for an element in A. What are the time and space complexity of your algorithm? Can you derive a lower bound on the algorithm?

2. Consider the recurrence equation:

$$\begin{aligned} h(0) &= 1 \\ h(n) &= h(n-1) + h(\lfloor \frac{n}{2} \rfloor) \end{aligned} \tag{1}$$

   An obvious algorithm to compute h is a recursive one, based directly on the defining equations. Is that algorithm efficient? Why or why not? If not, how can the algorithm be improved? Justify your answer.

3. In dynamic programming, we derive a *recurrence relation* for the solution to one subproblem in terms of solutions to other subproblems. To turn this relation into a bottom up dynamic programming algorithm, we need an order to fill in the solution cells in a table, such that all needed subproblems are solved before solving a bigger subproblem. For each of the following relations, give such a valid traversal order, or if no traversal order is possible for the given relation, briefly justify why.

$$\begin{aligned} A(i, j) &= F(A(i, j-1), A(i-1, j-1), A(i-1, j+1)) \\ A(i, j) &= F(A(\min\{i, j\} - 1, \min\{i, j\} - 1), A(\max\{i, j\} - 1, \max\{i, j\} - 1)) \\ A(i, j) &= F(A(i-2, j-2), A(i+2, j+2)) \end{aligned} \tag{2}$$

4. The woodcutter cuts a given log of wood, at any place you choose, for a price equal to the length of the given log. Suppose you have a log of length L, marked to be cut in n different locations labeled $1, 2, \ldots, n$. For simplicity, let indices 0 and $n+1$ denote the left and right endpoints of the original log of length L. Let d_i denote the distance of mark i from the left end of the log, and assume that $0 = d_0 < d_1 < d_2 < \cdots < d_n < d_{n+1} = L$. The wood-cutting problem is the problem of determining the sequence of cuts to the log that will cut the log at all the marked places and minimize your total payment. Give an efficient algorithm to solve this problem.

5. Consider two vertices, s and t, in some directed acyclic graph $G = (V, E)$. Give an efficient algorithm to determine whether the number of paths in G from s to t is odd or even. Analyze its running time in terms of $|V|$ and $|E|$.