**[Spring 2023] CSE 317: Design and Analysis of Algorithms** [All sections]
Homework 2 | Due: March 17, 2023

**Instructions:** You need to submit it on `Gradescope`. Details for submission will shared soon on `LMS`. Direct your all queries to the course staff (refer to syllabus for contact information and office hours).

---

1. (10 points) Consider four matrices of following respective sizes:

$$M_1 : 1 \times 5, \quad M_2 : 5 \times 2, \quad M_3 : 2 \times 3, \quad M_4 : 3 \times 9.$$

   Using dynamic programming algorithm find the minimum number of scalar multiplications required to compute the product $\prod_{j=1}^{4} M_j$ as well as an optimal parenthesization.

2. (10 points) Find the *edit distance* between the following two strings $s_1$ and $s_2$ using dynamic programming algorithm: $s_1 =$`INTENTION` and $s_2 =$`EXECUTION`.

3. (10 points) Design a *dynamic programming* algorithm to test whether a given sequence $S = \langle s_1, \ldots, s_n \rangle$ over some alphabet $\Sigma$ of length $n$ contains a *palindromic subsequence.*

   A subsequence $\langle s_{i_1}, s_{i_2}, \ldots, s_{i_k} \rangle$ of length $k$ is called palindromic if $s_{i_1} = s_{i_k}$, $s_{i_2} = s_{i_{k-1}}$, and so on. For example, $\langle 1, 2, 3, 2, 1 \rangle$ is a palindromic subsequence of $S = \langle 1, 2, 3, 4, 2, 1 \rangle$.

   Your algorithm should return the length of a longest palindromic subsequence in $S$. Also analyze time complexity of your algorithm and show that it runs in $O(n^2)$ time and use $O(n^2)$ space.

4. (10 points) Given two sequences $x = \langle x_1, \ldots, x_m \rangle$ and $y = \langle y_1, \ldots, y_n \rangle$ over some alphabet $\Sigma$ of lengths $m$ and $n$, respectively, find the length of a *shortest common supersequence* of $x$ and $y$ using a dynamic programming algorithm. Also analyze time and space complexity of your algorithm.

   For example, the shortest common supersequence of $x = \langle 1, 2, 2, 3 \rangle$ and $y = \langle 2, 3, 2 \rangle$ is the sequence $z = \langle 1, 2, 2, 3, 2 \rangle$ as both $x$ and $y$ are subsequences of $z$.

5. (10 points) Given $n$ dice each with $m$ faces, numbered from 1 to $m$, find the number of ways to get the sum $s$ i.e., the sum $s$ is the sum of values on each face when all the dice are thrown. Design a dynamic programming algorithm to solve this problem. Also analyze time and space complexity of your algorithm.

6. (10 points) Consider a sequence $S = \langle s_1, \ldots, s_n \rangle$ of $n$ numbers such that each $s_i \in \mathbb{R}$. Design a dynamic programming algorithm to find the longest increasing subsequence of $S$. Also analyze time and space complexity of your algorithm.

7. (10 points) Consider an $m \times n$ grid of numbers. Design a dynamic programming algorithm to find the maximum sum of numbers in a subgrid of size $k \times l$ where $k \leq m$ and $l \leq n$. Also analyze time and space complexity of your algorithm.

8. (15 points) Consider an array $A$ of $n$ numbers. Design a dynamic programming algorithm to find the maximum sum of a subarray of $A$ of length $k$ where $k \leq n$. Also analyze time and space complexity of your algorithm.

9. (15 points) Given a set of nonnegative integers $S$ and a target integer $t$, design a dynamic programming algorithm to find a subset of $S$ whose sum is equal to $t$. Also analyze time and space complexity of your algorithm.

---