German University in Cairo
Department of Computer Science
Dr. Nourhan Ehab

CSEN 703 Analysis and Design of Algorithms, *Winter Term 2025*
Practice Assignment 6

Exercise 6-1

Give examples of problems and/or algorithms where the Divide and Conquer technique is more efficient than Dynamic Programming and state your reasons.

Exercise 6-2

Consider the factorial algorithm:

i. Write a recurrence for the algorithm.

ii. Using the dynamic programming paradigm, provide an algorithm based on the given recurrence. Make sure that you use an efficient data structure to store the partial results.

Exercise 6-3

Consider the following recursive definition for the problem of calculating a given binomial coefficient $\binom{n}{k}$:

$$\binom{n}{k} = \begin{cases} \binom{n-1}{k-1} + \binom{n-1}{k} & \text{if } 0 < k < n; \\ 1 & \text{if } k = 0\, or k = n. \end{cases}$$

i. Write a divide-and-conquer algorithm to compute the binomial coefficients.

ii. Provide the recurrence relation for your answer in part (i).

iii. Write an iterative algorithm that employs the dynamic programming technique using the bottom-up approach.

iv. Write an algorithm for the same problem using the memoization technique.

v. Modify your answer in (iii) to use a single one-dimensional array indexed from 0 to k. What was your tradeoff?

Exercise 6-4

Suppose that you are given an $n \times n$ checkerboard and a checker. You must move the checker from the bottom edge of the board to the top edge of the board according to the following rule. At each step you may move the checker to one of three squares:

a) the square immediately above,

b) the square that is one up and one to the left (but only if the checker is not already in the leftmost column),

c) the square that is one up and one to the right (but only if the checker is not already in the rightmost column).

Each time you move from square x to square y, you receive $p(x, y)$ dollars. You are given $p(x, y)$ for all pairs (x, y) for which a move from x to y is legal. Do not assume that $p(x, y)$ is positive. Give an algorithm that figures out the set of moves that will move the checker from somewhere along the bottom edge to somewhere along the top edge while gathering as many dollars as possible. Your algorithm is free to pick any square along the bottom edge as a starting point and any square along the top edge as a destination in order to maximize the number of dollars gathered along the way. What is the running time of your algorithm?