# Q1. [2 points] What does dynamic programming have in common with divide-and-conquer? What is the principal difference between them?

Text, letter

Description automatically generated

# Q2. [5 points] Consider a modification of the rod-cutting problem in which, in addition to a price pi for each rod, each cut incurs a fixed cost of c. The revenue associated with a solution is now the sum of the prices of the pieces minus the costs of making the cuts. Give a dynamic-programming algorithm to solve this modified problem. (Ex. 15.1-3)

Text, letter

Description automatically generated

A picture containing text, whiteboard

Description automatically generated

# Q3. [5 points] We say that a problem exhibits the optimal substructure property when optimal solutions to a problem incorporate optimal solutions to related subproblems, which we may solve independently. Suppose that in the rod-cutting problem, we also had limit li on the number of pieces of length I that we are allowed to produce, for i = 1, 2, . . ., n. Show that the optimal-substructure property described no longer holds. (Ex. 15.3-5)

Text, letter

Description automatically generated

# Q4. [5 points] Give pseudocode to reconstruct an LCS from the completed c table and the original sequences X = (x1, x2, ; xm) and Y = (y1, y2, ; yn) in O(m+n) time, without using the b table. (Ex. 15.4-2) (Hint: Try to benefit from the PRINT-LCS procedure)

We can also use the below pseudo code:

A picture containing background pattern

Description automatically generated

# Q4. [5 points] Give pseudocode to reconstruct an LCS from the completed c table and the original sequences X = (x1, x2, ; xm) and Y = (y1, y2, ; yn) in O(m+n) time, without using the b table. (Ex. 15.4-2)

Text, letter

Description automatically generated

# Q5. Consider two teams, A and B, playing a series of games until one of the teams wins n games. Assume that the probability of A winning a game is the same for each game and equal to p and the probability of A losing a game is g = 1 -p. (Hence, there are no ties.) Let P(i,j) be the probability of A winning the series if A needs i more games to win the series and B needs j more games to win the series. a) Set up a recurrence relation for P(i,j) that can be used by a dynamic programming algorithm. [2 points]

Text, letter

Description automatically generated

b) Find the probability of team A winning a seven-game series if the probability of it   
winning a game is 0.4. [2 points]

Text, letter

Description automatically generated

# c) Write the pseudocode of the dynamic programming algorithm for solving this problem and analyze its running time. [4 points]

Text, letter

Description automatically generated