National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	**Design and Analysis of Algorithms**	**Course Code:**	**CS302**
	Degree Program:	**BSCS**	**Semester:**	**Fall 2021**
	Exam Duration:	**60 Minutes**	**Total Marks:**	**26**
	Paper Date:	**2 Dec 2021**	**Weight**	**13**
	Section:	**ALL**	**Page(s):**	**4**
	Exam Type:	**Midterm-II**		

Student : Name:______________________________ Roll No.________________ Section:_______

Instruction/Notes: Attempt the examination on the question paper and write concise answers. You can use extra sheet for rough work. Do not attach extra sheets used for rough with the question paper. Do not fill the table titled Questions/Marks.

Question	1	2	3	Total
Marks	/ 6	/10	/10	/26

Q1) [6 Marks]Dry run the dynamic programming algorithm to find the Max Sub-array Sum of the following input array, A:

-9	40	50	-160	25	90	-10	-8	90	-8

You have to create two arrays for the solution. In array S, fill into S[i] : the optimal sum of a subarray ending at A[i]. In array P, fill into P[i]: the starting index of the optimal array ending at A[i]

Solution:

S[]

-9	40	90	-70	25	115	105	97	187	179

P[]

1	2	2	2	5	5	5	5	5	5

Q2) [10 Marks] Given two arrays A and B of equal size n, you have to design an efficient algorithm that minimizes the sum A[1] x B[1] + A[2] x B[2] + … + A[n] x B[n]. You are allowed to shuffle the elements of each array, A and B.

Example:
N = 3
A = {3, 1, 1}, B = (6, 5, 4)
Minimum sum = 1 x 6 + 3 x 4 + 1 x 5 = 23
There are other possible ways of taking the sum like 3 x 6+ 1 x4+1 x5 = 27, but the minimum sum is 23.
Hint: Use greedy algorithm

Solution

Sort one array in increasing order and other array in decreasing order.

```
For (i = 1 to n)

    MinSum += A[i]*B[i]
```

Q3) [10 Marks] Given a matrix M * N of integers where each cell has a cost associated with it, the cost can also be negative. Find the minimum cost to reach the last cell (M-1, N-1) of the matrix from its first cell (0,0). We can only move one unit right (Column No + 1), one unit down (Row No + 1), and one unit in bottom diagonal (Row No + 1, Column No + 1). For example from index (i , j) you can move to (i , j+1), (i+1 , j), and (i+1, j+1) where i = row no and j = column no.

Example

4	7	8	6	4
-6	7	3	9	2
3	8	1	-2	4
7	1	7	3	7
2	9	8	9	3

4	7	8	6	4
-6	**7**	3	9	2
3	8	**1**	**-2**	4
7	1	7	**3**	7
2	9	8	9	**3**

Path with minimum cost = 4 -> -6 -> 7 -> 1 -> -2-> 3 -> 3 = 10

Provide a Dynamic Programming solution for it.

a) Provide recurrence for sub-problem

Solution

C (i , j) = ***Min*** (**C**[i][j - 1] , **C**[i - 1][j] , C[i-1][j-1]) + **A**[i][j]

b) Provide pseudo code for DP solution

c) Provide time complexity of DP solution