ASSUMPTION UNIVERSITY

Vincent Mary School of Science and Technology
Department of Computer Science
Department of Information Technology

Midterm Examination
Semester 2/2021

Subject Code :	CS3201/CSX3009
Subject Title :	Algorithm Design
Date :	January 18, 2022
Time :	**12.00 – 14.00 (2 hours)**
Instructors :	Asst. Prof. Dr. Thitipong Tanprasert (Full-Time)

Instructions:

1. Read the questions carefully and answer each question completely, legibly, and concisely.
2. You must type/write your answers in your computer only. Writing answers in a paper and taking a photo using camera are not allowed.
3. An answer to a question is either a text file, a .docx file, or a Python 3 program.
4. To submit your answers, compressing (zip) all of your answers into one file and upload the zipped file to the Midterm Examination created as an assignment for the course. no later than 17:00 the latest. Any late submission received after 17:01 will not be graded.
5. This is an opened book examination; you can use any materials as references, including online search. However, any form of communication with anyone regarding the exam, directly or indirectly, will be considered "CHEATING".
6. You MUST turn on your camera and microphone, share your working screen (the entire screen of your PC), and record the video in MS Team for the whole examination period. The answered file will NOT BE GRADED IF THERE IS NO COMPLETELY RECORDED VIDEO CLIP.
7. This examination paper and recorded video are an intellectual property of Assumption University; you are NOT allowed to duplicate, share, or publicize it.
8. If you cheat or contribute to cheating at the exam, you will get zero score and will be considered to get the grade 'F' for this course.

Marking Scale:

Essay and/or Programming	5 questions	50 marks

Total 50 marks

Total 3 pages (including this page)

1) [10 marks] The following list of numbers; 31, -41, 59, 26, -53, 58, 87, -93, -88, 84, is input to the Kadane's maximum subsequence algorithm. List out the current sum and current maximum subsequence's sum become when the specified number *has just been considered*.

number	current sum	current maximum subsequence's sum
59		
58		
84		

2) [10 marks] Given a matrix A, consisting of m rows and n columns. The element at row i and column j is A[i][j].

a) [5 marks] Write a Python code segment that transforms matrix A into the accumulation matrix B, in which the element B[r][c] is the sum of all elements in the rectangle defined by rows in the closed interval [0, r] and columns in the closed interval [0, c].
The code must have the running time of O(m x n).

Example.

A	B
0 -2 -7 0 9 2 -6 2 -4 1 -4 1 -1 8 0 -2	0 -2 -9 -9 9 9 -4 -2 5 6 -11 -8 4 13 -4 -3

Explanation: B[1][1] = A[0][0]+A[0][1]+A[1][0]+A[1][1] = 0-2+9+2 = 9

b) [5 marks] Write a Python code segment that calculates the sum of all elements of matrix A in the rectangle scoped by rows in the closed interval [r1, r2] and columns in the closed interval [c1, c2].

As an example, for the matrix A in question a),

sumA(r1=2, r2=3, c1=1, c2=2) = 1-4+8+0 = 5

The code must have a *constant running time* i.e. independent of the values of r1, r2, c1, c2

Hint: Make use of matrix B

3) [10 marks] Given that C = [C[0], C[1], C[2], … , C[m-1]], where C[i] is a distinct integer; i = 0, 1, … , m-1. Write a Python program that prints out *the total number of different ways* to sum integers to V, taking only the integers in C. Each integer can be used as many times as preferred.

For example, suppose that C = [1,2,5,9] and V = 10, both 1+9 and 5+5 are valid ways, as well as a few other possible ways (e.g. 1+2+2+5)

However, 1+9 and 9+1 are considered as the same way, as both consist of one 1 and one 9.

4) [10 marks] The following program solves a problem with a recursive brute-force technique.

```
n = int(input())
x = list(map(int, input().split()))
half = sum(x)//2

def f(i, A):
    if i == n or A > half:
        return 0
    else:
        iB = x[i] + f(i+1, A)
        iA = 1000000000
        if A+x[i] <= half:
            iA = -x[i] + f(i+1, A+x[i])
        return min(iA, iB)

print(f(0,0))
```

Given that $0 \leq n \leq 1000$ and $0 \leq A \leq 1500$, transform the program into a memoization version so that it handles every possible test case within 1 second.

5) [10 marks] Follow up on the problem 4 above, if $0 \leq n \leq 15000$, while and $0 \leq A \leq 100$. The recursive Python program will reach its recursion depth limit and cannot handle a large value of n correctly.

   Transform the program from question 4 into a dynamic programming version in order to cope with the new limitation on the argument values.