COMPUTER INFORMATION SYSTEMS

COMP 251: DATA STRUCTURES AND ALGORITHMS

UNIVERSITY OF THE FRASER VALLEY

Lab 1: The Maximum Contiguous Subsequence Sum Problem

Objectives: The aim of this lab session is to give you an understanding of the time complexity of algorithms. We will be using the solutions to the maximum contiguous subsequence sum problem as the example.

In this lab you will;

- Implement three solutions to the maximum contiguous subsequence sum problem discussed in the textbook.
- Test the correctness of implementation using appropriate sample inputs.
- Evaluate the time taken by three solutions for inputs of different sizes and analyze them.

You are expected to use Java programming language for completion of this lab exercise. You should upload your completed code in separate .Java files and a report (in pdf format) explaining how the running time of your solutions vary with the input size to the assignment in the Blackboard.

Task 1: *Implementing the Three solutions to the maximum contiguous subsequence sum problem*

In the first task, you are expected implement the three solutions for the maximum contiguous subsequence sum problem in the textbook. The three solutions should be implemented in three separate java files (solution1.java, solution2.java and solution3.java). In each java file;

1. Implement a method ***public static int[] theMCSS(int [] A)*** which accepts the input and returns an array which contains the start and end position of the maximum contiguous subsequence.
   Some sample inputs and return values of the function are shown below.

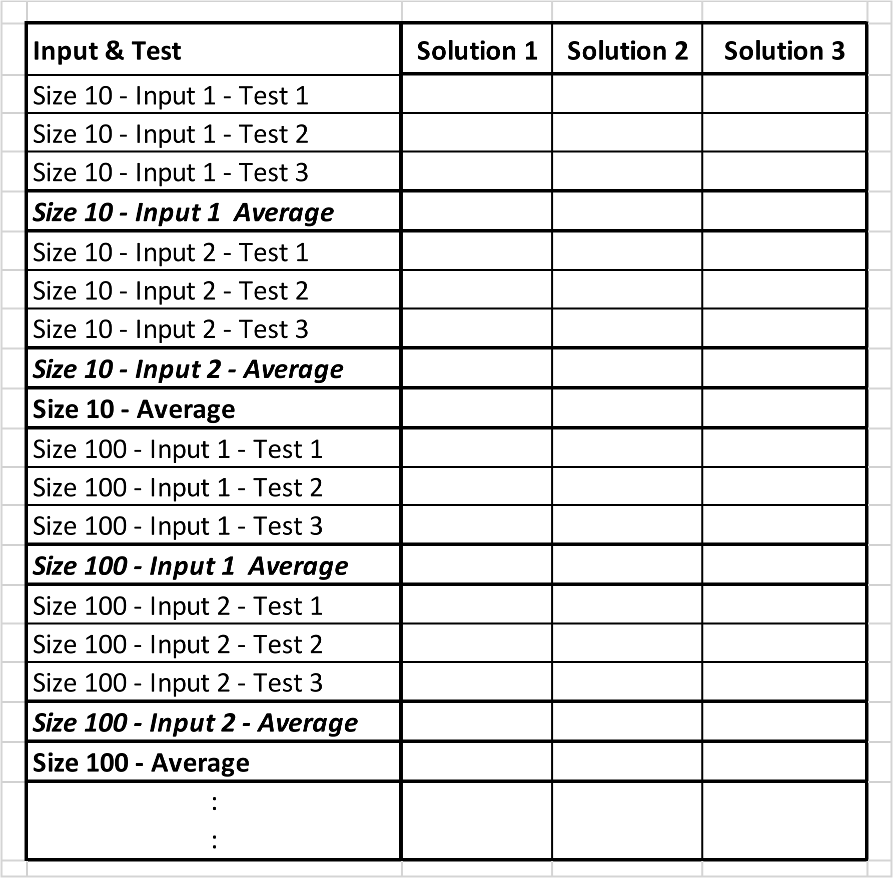
Input	Return Value
{ 1, 3, -5, 7, 4, -2}	{3, 4}
{ -2, 11, -4, 13, -5, 2} => 20	{1, 3}
{ -2, -5, -20, -5, -7}	{-1, -1}

2. Write a method ***public static void testSolution()*** to test your ***theMCSS*** method using some sample inputs and to assert that the correct output is given.

Task 2: *Analyzing the Performance*

Test and record the time takes by three solutions for different input sizes (at least for 10, 100, 1000, 10000).

- **For each input size,** generate at least two testcases (two different inputs)
- **For each testcase**, run the program for at least three times and collect the time value and take the average of all the runs.
- **Record the results in the following table (or a similar table).**



Input & Test	**Solution 1**	**Solution 2**	**Solution 3**
Size 10 - Input 1 - Test 1			
Size 10 - Input 1 - Test 2			
Size 10 - Input 1 - Test 3			
Size 10 - Input 1 Average			
Size 10 - Input 2 - Test 1			
Size 10 - Input 2 - Test 2			
Size 10 - Input 2 - Test 3			
Size 10 - Input 2 - Average			
Size 10 - Average			
Size 100 - Input 1 - Test 1			
Size 100 - Input 1 - Test 2			
Size 100 - Input 1 - Test 3			
Size 100 - Input 1 Average			
Size 100 - Input 2 - Test 1			
Size 100 - Input 2 - Test 2			
Size 100 - Input 2 - Test 3			
Size 100 - Input 2 - Average			
Size 100 - Average			
: :			

Plot the results (average execution time against the input size) for three solutions in the same graph. Comment about any observations and interesting patterns. Submit your report as a pdf document.

Submission:

Upload your work in a single zip file to the assignment in the blackboard. The files should be named as follows.

1. **Name of the zip file: <your_student_number>.zip (for example 300321321.zip)**

   Files inside the zip file: solution1.java, solution2.java, solution3.java & lab1_<your_student_number>.pdf (for example lab1_300321321.pdf)

2. In addition to submission of the files, you need to record a video and explain your answer. Submit the link to your video clip (not more than 4min) that **explains clearly your approach, run your code, and demonstrate your solution.** You could provide the link to your video (on Youtube/Vimeo/…) in the body of your submission. Please don't upload the video on BB.

Only past the link in the submission body.