Assignment # 4

(Handwritten)

CS 302 – Design and Analysis of Algorithm

Spring 2021

Deadline: 1st June 2021 (11:55PM)
Soft Copy Slate Deadline: 2nd June 2021 (11:55PM)

(This assignment is for All Sections)

This is a group Assignment.
Cross Section group strictly NOT allowed.
Total Member allowed = Max: 3
Work Division is your responsibility in any case.

---

1. A single violation of guideline will lead to Zero mark in your assignment.
2. Print the document and solve it in your own handwriting. Get it scanned (pdf) or take picture (jpg) and submit on Slate. Try to use White A4 Pages.
3. Slate address:
   http://slate.nu.edu.pk/portal
   OR
   http://203.124.42.218:8080/portal
4. You have been given whole **1 day for only submission** purpose so **NO email submission** will be accepted.
5. You are encouraged to take help from Internet and books but clearly mention the source.
6. Support your solution/proves with some examples.
7. Try to give detailed answer. Shortcuts will lead to partially negative marks.
8. This is a group Assignment, make groups on your own responsibility. Excuses will not be entertained like a member didn't attempt his/her part.
9. Cross Section group strictly NOT allowed.
10. **Total Member allowed = 3, Assignment will be checked as a whole work, not individual work will be marked as full work, so work division is your responsibility.**
11. Work Division is your responsibility if members are less than 3.
12. Only one member must Submit all members work in a zip file contains this report and your handwritten solved work.
13. Zip file must be named as:
    Rollnumber1_Rollnumber2_Rollnumber3_Section_Assignment4
    For example: 19f0123_19f0345_19f0567_A_Assignment4

Member: 1 Performed (Q1 and Q2):
Roll No: _____

Section: _____

Name: _____

Member: 2 Performed (Q3 and Q4):

Roll No: _____

Section: _____

Name: _____

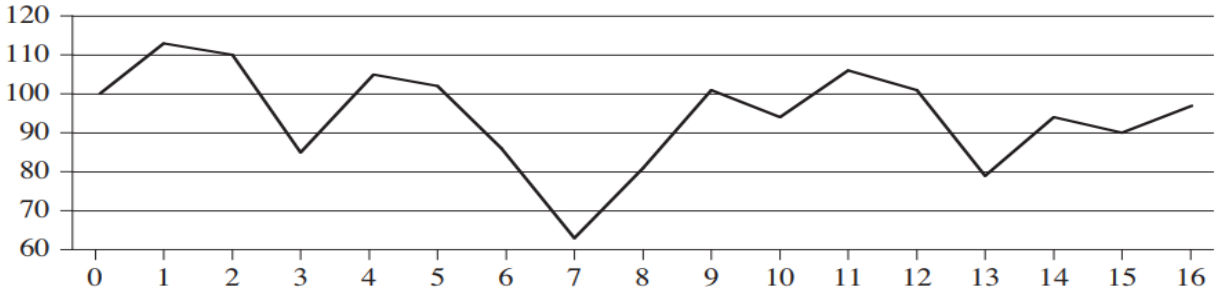
Member: 3 Performed (Q5):

Roll No: _____

Section: _____

Name: _____

Question 1.



Day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Price	100	113	110	85	105	102	86	63	81	101	94	106	101	79	94	90	97
Change		13	−3	−25	20	−3	−16	−23	18	20	−7	12	−5	−22	15	−4	7

The above Graph and Data is information about the price of stock in the Volatile Chemical Corporation after the close of trading over a period of 17 days. The horizontal axis of the chart indicates the day, and the vertical axis shows the price. The bottom row of the table gives the change in price from the previous day. Your task is to find the Sub Array that has the greatest sum of any contiguous subarray of above Volatile Chemical Corporation Stock Prices Data.

Question 2.

What does FIND-MAXIMUM-SUBARRAY return when all elements of A are negative? Support your answer with arguments.

```
FIND-MAXIMUM-SUBARRAY(A, low, high)
 1  if high == low
 2       return (low, high, A[low])          // base case: only one element
 3  else mid = ⌊(low + high)/2⌋
 4       (left-low, left-high, left-sum) =
              FIND-MAXIMUM-SUBARRAY(A, low, mid)
 5       (right-low, right-high, right-sum) =
              FIND-MAXIMUM-SUBARRAY(A, mid + 1, high)
 6       (cross-low, cross-high, cross-sum) =
              FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
 7       if left-sum ≥ right-sum and left-sum ≥ cross-sum
 8            return (left-low, left-high, left-sum)
 9       elseif right-sum ≥ left-sum and right-sum ≥ cross-sum
10            return (right-low, right-high, right-sum)
11       else return (cross-low, cross-high, cross-sum)
```

Question 3.

Write pseudocode for the brute-force method of solving the maximum-subarray problem. Your procedure should run in $O(n^2)$ time.

Question 4.

Suppose we change the definition of the maximum-subarray problem to allow the result to be an empty subarray, where the sum of the values of an empty subarray is 0. How would you change any of the algorithms that do not allow empty subarrays to permit an empty subarray to be the result?

Question 5.

Use the following ideas to develop a nonrecursive, linear-time algorithm for the maximum-subarray problem. Start at the left end of the array, and progress toward the right, keeping track of the maximum subarray seen so far. Knowing a maximum subarray of $A[1 \,..\, j]$, extend the answer to find a maximum subarray ending at index $j+1$ by using the following observation: a maximum subarray of $A[1 \,..\, j+1]$ is either a maximum subarray of $A[1 \,..\, j]$ or a subarray $A[i \,..\, j+1]$, for some $1 \le i \le j+1$. Determine a maximum subarray of the form $A[i \,..\, j+1]$ in constant time based on knowing a maximum subarray ending at index j.