

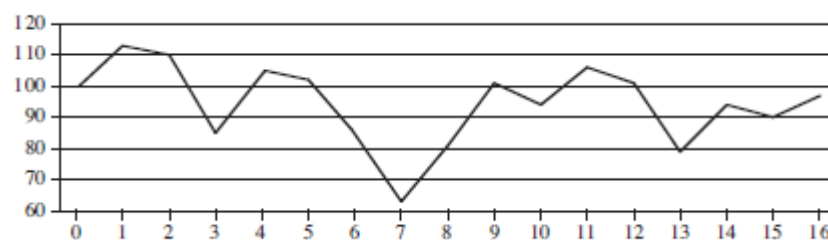


**(S1-19\_DSECLZG519)**  
**(Data Structures and Algorithms Design)**  
**Academic Year 2019-2020**

**Assignment 2 – PS5 - [Maximum Subarray] - [Weightage 13%]**

**1. Problem Statement**

Suppose that you been offered the opportunity to invest in Apple Inc. The stock price of the company is volatile. You can buy only one unit of stock at a time and then sell it at a later date, buying and selling after the close of trading for the day. To compensate for this restriction, you can learn what the price of the stock will be in the future. Your goal is to maximize your profit. The figure below shows the price of the stock over a 17-day period.



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

Information about the price of stock after the close of trading over a period of 17 days is given above. 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.

**Requirements:**

1. Formulate an efficient recursive algorithm using divide and conquer strategy to perform the above task
2. Also develop a non-recursive, linear-time algorithm for above problem.
3. Analyse the time complexity for requirement 1.
4. Implement the above problem statement using Python 3.7

**Input:**

Input should be taken in through a file called "inputPS5.txt" which has the fixed format mentioned below using the "/" as a field separator:

<day number> / <price>

Ex.

0 / 100

1 / 113

2 / 110

3 / 85

...

Note that the input data shown here is only for understanding and testing, the actual file used for evaluation will be different.

**Output:**

Maximum Profit(Divide & Conquer): 43

Day to buy: 7

Day to sell: 11

Maximum Profit(Iterative Solution): 43

Day to buy: 7

Day to sell: 11

Display the output in **outputPS5.txt**.

**2. Deliverables**

- Word document **designPS5\_<group id>.docx** detailing your algorithm design and time complexity of the algorithm.
- **Zipped AS2\_PS5\_MS\_[Group id].py package folder** containing all the modules classes and functions and the main body of the program.
- **inputPS5.txt** file used for testing
- **outputPS5.txt** file generated while testing

### **3. Instructions**

- a. It is compulsory to make use of the data structures or algorithm mentioned in the problem statement.
- b. It is compulsory to use Python 3.7 for implementation.
- c. Ensure that all data structures and functions throw appropriate messages when their capacity is empty or full.
- d. For the purposes of testing, you may implement some functions to print the data structures or other test data. But all such functions must be commented before submission.
- e. Make sure that you read, understand, and follow all the instructions
- f. Ensure that the input, prompt and output file guidelines are adhered to. Deviations from the mentioned formats will not be entertained.
- g. The input, prompt and output samples shown here are only a representation of the syntax to be used. Actual files used to test the submissions will be different. Hence, do not hard code any values into the code.
- h. Run time analysis is provided in asymptotic notations and not timestamp based runtimes in sec or milliseconds.

### **4. Deadline**

- a. The strict deadline for submission of the assignment is 16<sup>th</sup> Feb, 2020.
- b. The deadline is set for a month from the date of rollout to accommodate for the semester exams. No further extension of the deadline will be entertained.
- c. Late submissions will not be evaluated.

### **5. How to submit**

- a. This is a group assignment.
- b. Each group has to make one submission (only one, no resubmission) of solutions.
- c. Each group should zip the deliverables and name the zipped file as below  
"ASSIGNMENT2\_[BLR/HYD/DLH/PUN/CHE]\_[G1/G2/...].zip"  
and upload in CANVAS in respective location under ASSIGNMENT Tab.
- d. Assignment submitted via means other than through CANVAS will not be graded.

### **6. Evaluation**

- a. The assignment carries 13 Marks.
- b. Grading will depend on

- a. Fully executable code with all functionality
- b. Well-structured and commented code
- c. Accuracy of the run time analysis and design document
- c. Every bug in the functionality will have negative marking.
- d. Source code files which contain compilation errors will get at most 25% of the value of that question.

## **7. Readings**

Text book: Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition). Chapters: 5.2