Work Integrated Learning Programmes Division

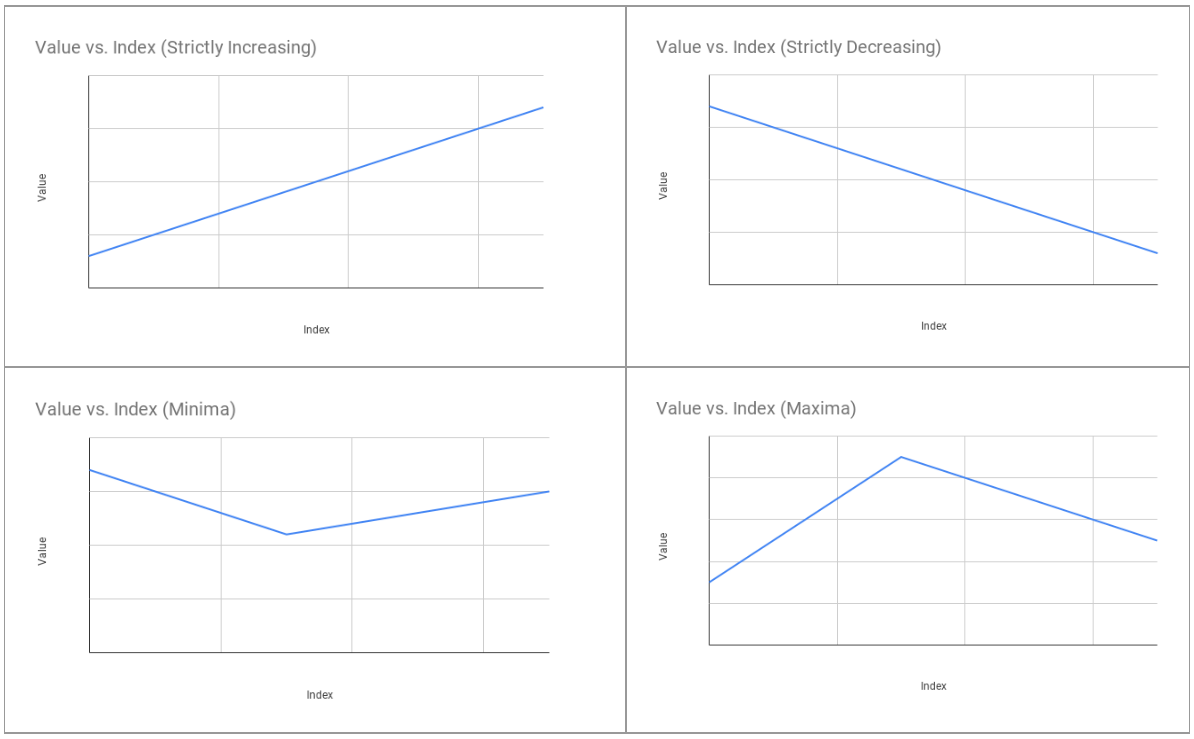
(Data Structures and Algorithms Design)
Academic Year 2021-2022

Assignment 2 – PS1 – [Maxima Minima] - [Weightage 13%]

1. Problem Statement

During data collection for an experiment, a scientist collects a stream of data that is stored in an array.

The data collected is in the form of one of the four representations:



Your task is to find the maxima OR minima in the array. If the values are strictly increasing or strictly decreasing then return the minimum value.

Note: You can be sure that the data will be from one of the above mentioned four distributions.

3	5	7	9	11	13	15	17
Strictly Increasing: return 3 (minimum value)							

17	15	13	11	9	7	5	3
Strictly Decreasing: return 3 (minimum value)							

17	15	13	11	12	13	14	15
Minima: return 11 (the minima)							

3	5	7	9	8	7	6	5
Maxima: return 9 (the maxima)							

Requirements:

1. Formulate and efficient algorithm using Divide and Conquer to first find the type of representation out of the four and then identify the output as shown in the table above which will be a single integer.
2. Analyse the time complexity of your algorithm.
3. Implement the above problem statement using Python 3.7.

Sample Input:

Input should be taken in through a file called "**inputPS1.txt**" where each line a space-separated array of integers:

Ex:

```
3 5 7 9 11 13 15 17
17 15 13 11 9 7 5 3
3 5 7 9 8 7 6 5
17 15 13 11 12 13 14 15
```

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

Sample Output:

Output the type of representation followed by a space-separated answer.

```
decreasing 3
increasing 3
maxima 9
minima 11
```

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

Display the output in **outputPS1.txt**.

2. Deliverables

1. Word document **designPS1_<group id>.docx** detailing your design and time complexity of the algorithm.

2. **[Group id]_Contribution.xlsx** mentioning the contribution of each student in terms of percentage of work done. Download the Contribution.xlsx template from the link shared in the Assignment Announcement.
3. **inputPS1.txt** file used for testing
4. **outputPS1.txt** file generated while testing
5. **.py file** containing the python code. Create a single *.py file for code. Do not fragment your code into multiple files

Zip all of the above files including the design document and contribution file in a folder with the name:

[Group id]_A2_PS1_MaximaMinima.zip and submit the zipped file.

Group Id should be given as **Gxxx** where xxx is your group number. For example, if your group is 26, then you will enter G026 as your group id.

3. Instructions

1. It is compulsory to make use of the data structure(s) / algorithms mentioned in the problem statement.
2. Ensure that all data structure insert and delete operations throw appropriate messages when their capacity is empty or full. Also ensure basic error handling is implemented.
3. 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.
4. Make sure that your read, understand, and follow all the instructions
5. Ensure that the input, prompt and output file guidelines are adhered to. Deviations from the mentioned formats will not be entertained.
6. The input, prompt and output samples shown here are only a representation of the syntax to be used. Actual files used to evaluate the submissions will be different. Hence, do not hard code any values into the code.
7. Run time analysis is to be provided in asymptotic notations and not timestamp based runtimes in sec or milliseconds.

Instructions for use of Python:

1. Implement the above problem statement using Python 3.7.
2. Use only native data types like lists and tuples in Python, do not use dictionaries provided in Python. Use of external libraries like graph, numpy, pandas library etc. is not allowed. The purpose of the assignment is for you to learn how these data structures are constructed and how they work internally.
3. Create a single *.py file for code. Do not fragment your code into multiple files.
4. Do not submit a Jupyter Notebook (no *.ipynb). These submissions will not be evaluated.

5. Read the input file and create the output file in the root folder itself along with your .py file. Do not create separate folders for input and output files.

4. How to submit

1. This is a group assignment.
2. Each group has to make one submission (only one, no resubmission) of solutions.
3. Each group should zip all the deliverables in one zip file and name the zipped file as mentioned above.
4. Assignments should be submitted via E-Learn > Assignment section. Assignment submitted via other means like email etc. will not be graded.

5. Evaluation

1. The assignment carries 13 Marks.
2. Grading will depend on
    a. Fully executable code with all functionality working as expected
    b. Well-structured and commented code
    c. Accuracy of the run time analysis and design document.
3. Every bug in the functionality will have negative marking.
4. Marks will be deducted if your program fails to read the input file used for evaluation due to change / deviation from the required syntax.
5. Use of only native data types and avoiding libraries like numpy, graph and pandas will get additional marks.
6. Plagiarism will not be tolerated. If two different groups submit the same code, both teams will get zero marks.
7. Source code files which contain compilation errors will get at most 25% of the value of that question.

6. Readings

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