Work Integrated Learning Programmes Division
M.Tech (Data Science and Engineering)

(S2-20_DSECLZG519)
(Data Structures and Algorithms Design)
Academic Year 2020-2021

Assignment 1 – PS8 - [Maximiser] - [Weightage 12%]

1. Problem Statement

Sherlock and Watson are playing a game named Maximize the Score. In this game, there are n balls placed on the table. Each ball has a value written on it.

The game starts with a coin toss. If the coin falls head side up, then Sherlock starts the game, else Watson starts the game. The game is played in the form of rounds. Hence, if Sherlock wins the toss, the first round will be played by Sherlock, second round will be played by Watson, third round will be played by Sherlock and so on. In each round, the player is allowed to take at max k turns. In each turn, the player picks one ball from the table.

Note:- Each round is entirely played by one player only so if it is Sherlock's round, then he will play x turns(x <= k) and the next round will be played by Watson.

Watson is crazy about sum of digits and so he will only pick the ball whose sum of digits of the value is maximum (for e.g. if 4 and 11 are present on the table, he will pick 4 as the sum of digits of 4 is greater than that of 11). If more than one ball has maximum sum of digits, then he can pick any one of them. Sherlock doesn't care about sum of digits so he can pick any ball from table. They both want to maximize their score and so both of them will play optimally. Print the score Sherlock and Watson will achieve if they both play optimally.

Note:- Score of a player is the sum of values of all the balls taken by the player.

Note:- Use of fast I/O is recommended

Requirements:

1. Implement the above problem statement using Priority Queues and Python 3.7
2. Read the input from a file(**inputPS8.txt**), Print the score Sherlock and Watson will achieve if they both play optimally.
3. You will output your answers to a file (**outputPS8.txt**) for each line.
4. Perform an analysis for the features above and give the running time in terms of input size: n.

Sample file formats

Sample Input:

First line contains an integer T, denoting the number of test-cases.

Next 3T lines contain the test-cases as shown below:-

First line contains two integers n and k, representing the number of balls present on the table and the maximum number of turns allowed per round.

Second line contains n space separated integers(a1, a2, ... an). Here, ai indicates the score written on ith ball.

Third line contains result of the toss. Here, "HEADS" indicates Sherlock starts the game, while "TAILS" indicates Watson starts the game.

Input will be taken from the file **inputPS8.txt**

```
2
3 2
1000 99 98
TAILS
2 1
5 6
HEADS
```

Sample Output:

For each testcase, print scores achieved by Sherlock and Watson, if they both play optimally. Display the output in **outputPS8.txt.**

```
1000 197
6 5
```

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

Explanation

In first testcase, the winner of toss is Watson and so Watson plays first round. In this round, he can have 2 turns and so he will take 2 turns. In first turn, he has no choice but to pick 99 as it has largest sum of digits. Similarly, in second turn he has to pick 98. Now, Sherlock plays his round. As there is only one ball left, Sherlock can only take one turn and pick that ball. So, final score of Sherlock = 1000 and final score of Watson = 99 + 98 = 197.

In second testcase, the winner of toss is Sherlock and so Sherlock plays the first round. In this round, he will ofcourse pick 6 as he wants to maximize the score. Then Watson plays second round and picks 5. So, final score of Sherlock = 6 and final score of Watson = 5.

2. Deliverables

1. Word document **designPS8_<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. **inputPS8.txt** file used for testing
4. **outputPS8.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]_A1_PS8_Maximizer.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.
8. Please note that the design document must include
    a. The data structure model you chose with justifications
    b. Details of each operations with the time complexity and reasons why the chosen operations are efficient for the given representation
    c. One alternate way of modelling the problem with the cost implications.
9. Writing good technical report and well document code is an art. Your report cannot exceed 4 pages. Your code must be modular and quite well documented.

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. Deadline

1. The strict deadline for submission of the assignment is **Monday, 21th Jun, 2021.**
2. The deadline has been set considering extra days from the regular duration in order to accommodate any challenges you might face. No further extensions will be entertained.
3. Late submissions will not be evaluated.

5. 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 Canvas > Assignment section. Assignment submitted via other means like email etc. will not be graded.

6. Evaluation

1. The assignment carries 12 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. Copy / Paste's from web resources / or your friends' submission will attract severe penalty to the extent of awarding negative 10 percent. We will not measure the extent of such blatant copy pastes and details of who copied from whom and such details while awarding the penalties. It's the responsibility of the team to solve and protect your original work.**
7. 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:** 2.4