Birla Institute Of Technology & Science, Pilani
Work Integrated Learning Programmes Division
M.Tech (Data Science and Engineering)

(S2-21_DSECLZG519)
(Data Structures and Algorithms Design)
Academic Year 2021-2022

Assignment 1 – PS07 - [Freezer Room] - [Weightage 12%]

1. Problem Statement

An organization has a unique freezer room with easy-to-adjust temperature, lock with keys, manual defrost option, and basket or adjustable shelf etc. Employees of this organization keep moving in and out of the freezer room on work. Organization uses a unique employee ID and smart card to identify their employee's movement to the freezer room. Whenever the employee swipes his card to the freezer room employee ID is recorded. When an employee enters the room for the first time, the counter is set to 1. From then onwards, each time an employee swipes out of the room the counter is incremented and incremented again when he enters back. If the counter is odd, it means the employee is inside the freezer room and if the counter is even, it means he/she is out of the room.

The organization uses this data to perform the following analytics:

1. How many employees entered the freezer room today?
2. Number of employees that have entered the freezer room today and are currently inside?
3. Check if a specific employee is inside the freezer room or outside?
4. List of employees that have swiped (in or out) more than x number of times?
5. Which employee ids within a range of IDs entered the freezer room, the swipe counter for them, and whether they are inside or outside the freezer room?

Operations:

The basic structure of the employee node will be:

Class EmpNode:
def_init_(self, EmpId):
self.EmpID = EmpId
self.attctr = 1 self.left
= None self.right =
none

Requirements

1. Implement the above problem statement using **Python 3.7 or above using Binary Tree ADT**
2. Read the input from a file (**inputPS07.txt**), which contains the list of students and associated actions to be taken identified by relevant tags at the start of each line separated with a colon.
3. You will output your answers to a file (**outputPS07.txt**) for each line.
4. Use In order traversal to display the range of employees and their swipe count.
5. Perform an analysis for the features above and give the running time in terms of input size: n.

Sample file formats

The sample files shown below are only indicative of the structure that needs to be followed and the outputs need not be in line with the input and prompt file.

Sample Input

Each row will have one employee id indicating a single swipe for that employee followed by the set of instructions to execute the above requirements.

Sample inputPS07.txt

```
05
22
41
121
41
22
41
121

inFreezer:
checkEmp: 12
checkEmp: 22
freqVisit: 3
range: 05:125
```

Sample Output

Sample outputPS07.txt

Total number of employees recorded today: 4 2 employee(s) still inside freezer room.
Employee id 12 did not swipe today.
Employee id 22 swiped 2 times today and is currently outside freezer room
Employees that swiped more than 3 times today are:
41, 3
Range: 05 to 125 Employee swipe:
05, 1, in
41, 3, in
121, 2, out

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

2. Deliverables

1. Word document **designPS07_<group id>.docx** detailing your design and time complexity of the algorithm and a contribution table as below
2. **inputPS07.txt** file used for testing
3. **outputPS07.txt** file generated while testing
4. .py file containing the python code. Create a single python code file. Do not fragment your code into multiple files
5. Zip all of the above files including the design document in a folder with the name: **[Group id] _A1_PS07_FreezerRoom.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
    i. The data structure model you chose with justifications
    ii. Details of each operations with the time complexity and reasons why the chosen operations are efficient for the given representation
    iii. 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 Tuesday, **21st June, 2022**.
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
    i. Fully executable code with all functionality working as expected
    ii. Well-structured and commented code
    iii. 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 0 marks. 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:** 2.3