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

(S2-18_DSECFZG519)
(Data Structures and Algorithms Design)
Academic Year 2018-2019

Assignment 2 – PS1 - [MOBILE MANUFACTURING] - [Weightage 12%]

1. Problem Statement

There are N different models of mobiles manufactured at a mobile manufacturing unit. Each mobile must go through 2 major phases: ‘parts manufacturing’ and ‘assembling’. Obviously, ‘parts manufacturing’ must happen before “assembling’. The time for ‘parts manufacturing’ and ‘assembling’ (pm_i and a_i for ith mobile) for every mobile may be different. If we have only 1 unit for ‘parts manufacturing’ and 1 unit for ‘assembling’, how should we produce n mobiles in a suitable order such that the total production time is minimized?

Requirements:

1. Write a Greedy Algorithm to select the mobile ‘parts manufacturing’ and ‘assembling’ in such a way that total production time is minimized.
2. Analyse the time complexity of your algorithm.
3. Implement the above problem statement using Python.

Input:

For example, now there are 6 different Mobiles in total. Time for each mobile ‘parts manufacturing’ and ‘assembling’ are given as shown:

Mobile i	pm_i (minutes)	a_i (minutes)
1	5	7
2	1	2
3	8	2
4	5	4
5	pm_5	a_5
6	pm_6	a_6

Input should be taken in through a file called “inputPS1.txt” which has the fixed format mentioned below using the “/” as a field separator:
<mobile i> / < pm_i (minutes)> / <a_i (minutes)>

Ex:

```
1 / 5 / 7
2 / 1 / 2
3 / 8 / 2
…
```

Output:

```
Mobiles should be produced in the order: 2, 5, 6, 1, 4, 3.
Total production time for all mobiles is: 28
Idle Time of Assembly unit: 2
```

The output should be written to the file outputPS1.txt

2. Deliverables

- Word document **designPS1_<group id>.docx** detailing your algorithm design and time complexity of the algorithm.
- **Zipped AS2_PS1_MM_[Group id].py package folder** containing all the modules classes and functions for the employee node, binary tree and the main body of the program.
- **inputPS1.txt** file used for testing
- **outputPS1.txt** file generated while testing

3. Instructions

- Do not use inbuilt data structures available in Python. The purpose of these assignments is for you to lean how these data structures and algorithms work.
- It is compulsory to use Python for implementation.
- Ensure that all data structure insert and delete operations throw appropriate messages when their capacity is empty or full.
- 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.
- Make sure that your read, understand, and follow all the instructions
- Ensure that the input and output file guidelines are adhered to. Deviations from the mentioned formats will not be entertained.
- Run time analysis is provided in asymptotic notations and not timestamp based runtimes in sec or milliseconds.

4. Deadline

- The strict deadline for submission of the assignment is **Sep 5th, 2019 EoD.**
- Late submissions won't be evaluated.

5. How to submit

- This is a group assignment.
- Each group has to make one submission (only one, no resubmission) of solutions.
- Each group should zip the deliverables and name the zipped file as below
- "ASSIGNMENT2_[BLR/HYD/DLH/PUN/CHE]_[B1/B2/…]_[G1/G2/…].zip"
- and upload in CANVAS in respective location under ASSIGNMENT Tab.
- Assignment submitted via means other than through CANVAS will not be graded.

6. Evaluation

- The assignment carries 12 Marks
- Grading will depend on
  - Efficiency of design (detailed in the design document)
    - Generic explanation copied off the internet will not be considered.
  - Every bug in the functionality will lead to negative marking.
  - Duplication of design document / code will be penalized.
  - Source code files which contain compilation errors will get at most 25% of the value of that question.
  - Fully executable code with all functionality.
- Late submissions will not be evaluated.

7. Readings

Text book: Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition). Chapters: 5.1