



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

Assignment 2 – PS4 - [ASSIGNMENT PROBLEM] - [Weightage 12%]

1. Problem Statement

As part of dissertation, the University professor has decided to ask the students about their preferences of topic before assigning the problem statements to each one of them. The topics available are:

1. Data mining
2. NLP
3. AI
4. Spatial Data Analysis
5. Image processing
6. Big-Data
7. Graph Mining
8. Machine Learning
9. E-commerce
10. Wireless Mobile communication
11. Cloud computing

There are 11 students working with the Professor. Each student can have more than 1 topic which he/she prefers. Your job is to help the professor calculate the number of unique allocation of topics to the students such that everybody gets exactly one topic of their liking and no two students get the same subject allocated.

Requirements:

1. Formulate an efficient algorithm using dynamic programming to perform the above task.
2. Analyse the time complexity of your algorithm.
3. Implement the above problem statement using Python

Input:

Input should be taken in through a file called "inputPS4.txt" which has the fixed format mentioned below using the "/" as a field separator: Student <num> / < topic 1> / <topic 2> / <topic 3>....

Ex:

S1 / DM / SDA / WMC / CC

S2 / DM / NLP / AI / SDA / IP / GM / EC

S3 / DM / SDA / GM / ML / WMC

S4 / DM / AI / SDA / IP / GM / ML / WMC / CC

S5 / NLP / AI / SDA / BD / EC / WMC / CC

S6 / DM / NLP / AI / BD

S7 / IP / GM / CC

S8 / DM / AI / SDA / CC

S9 / AI / IP / BD / WMC / CC

S10 / DM / NLP / AI / GM / EC / CC

S11 / DM / IP / BD / GM / ML

Output:

Syntax of the output should be:

The total number of allocations possible is: <number of possible combinations of allocations>

Ex: The total number of allocations possible is: 7588.

Display the output in **outputPS4.txt**.

2. Deliverables

- Word document **designPS4_<group id>.docx** detailing your algorithm design and time complexity of the algorithm.
- **Zipped AS2_PS4_AP_[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.

- **inputPS4.txt** file used for testing
- **outputPS4.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 learn 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 you 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: 5.3