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

(S1-19_DSECLZG519)
(Data Structures and Algorithms Design)
Academic Year 2019-2020

Assignment 2 – PS7 - [Cricket Batting Order] - [Weightage 13%]

1. Problem Statement

For the upcoming world-cup, the Indian Cricket Selection Committee has to come up with a possible batting order for their players. Instead of using the traditional approach they have decided to use computer algorithms to come up with all the possible batting orders and then decide from that. The algorithm however requires the possible batting positions for each player.

The algorithm takes a list of 11 players. Each player can have more than one position they can bat at. Your job for now is to help the selection committee calculate the total number of unique batting charts such that every player gets exactly one batting position from their list of positions and no two players are given the same batting position in one batting chart.

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 3.7

Input:

Input should be taken in through a file called "inputPS7.txt" which has the fixed format mentioned below using the "/" as a field separator: Player <num> / < position 1> / < position 2> / < position 3>….

Ex:

P1 / 1 / 2 / 3 / 4

P2 / 1 / 5 / 9 / 2 / 6 / 7 / 8

P3 / 1 / 2 / 7 / 10 / 3

P4 / 1 / 9 / 2 / 6 / 7 / 10 / 3 / 4

P5 / 5 / 9 / 2 / 8 / 3 / 4

P6 / 1 / 5 / 3 / 6

P7 / 6 / 7 / 4

P8 / 1 / 9 / 2 / 4

P9 / 9 / 6 / 11 / 3 / 4

P10 / 1 / 5 / 9 / 7 / 8 / 4

P11 / 6 / 11 / 7 / 10

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

Output:

Syntax of the output should be:

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

Ex: The total number of allocations possible is: 4646.

Display the output in **outputPS7.txt**.

2. Deliverables

- Word document **designPS7_<group id>.docx** detailing your algorithm design and time complexity of the algorithm.
- **Zipped AS2_PS7_CBO_[Group id].py package folder** containing all the modules classes and functions and the main body of the program.
- **inputPS7.txt** file used for testing
- **outputPS7.txt** file generated while testing

1. Instructions

a. It is compulsory to make use of the data structures or algorithm mentioned in the problem statement.

b. It is compulsory to use Python 3.7 for implementation.

c. Ensure that all data structures and functions throw appropriate messages when their capacity is empty or full.

d. 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.

e. Make sure that your read, understand, and follow all the instructions

f. Ensure that the input, prompt and output file guidelines are adhered to. Deviations from the mentioned formats will not be entertained.

g. The input, prompt and output samples shown here are only a representation of the syntax to be used. Actual files used to test the submissions will be different. Hence, do not hard code any values into the code.

h. Run time analysis is provided in asymptotic notations and not timestamp based runtimes in sec or milliseconds.

2. Deadline

a. The strict deadline for submission of the assignment is 16th Feb, 2020.

b. The deadline is set for a month from the date of rollout to accommodate for the semester exams. No further extension of the deadline will be entertained.

c. Late submissions will not be evaluated.

3. How to submit

a. This is a group assignment.

b. Each group has to make one submission (only one, no resubmission) of solutions.

c. Each group should zip the deliverables and name the zipped file as below

"ASSIGNMENT2_[BLR/HYD/DLH/PUN/CHE]_[G1/G2/…].zip"
and upload in CANVAS in respective location under ASSIGNMENT Tab.

d. Assignment submitted via means other than through CANVAS will not be graded.

4. Evaluation

a. The assignment carries 13 Marks.

b. Grading will depend on

   a. Fully executable code with all functionality

   b. Well-structured and commented code

   c. Accuracy of the run time analysis and design document

c. Every bug in the functionality will have negative marking.

d. Source code files which contain compilation errors will get at most 25% of the value of that question.

5. Readings

Text book: Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition). Chapters: 5.3