BITS Pilani
Pilani | Dubai | Goa | Hyderabad

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

**(S1-20_DSECFZG519)**

**(Data Structures and Algorithms Design)**

**Academic Year 2020-2021**

# Assignment 2 – PS3 – [Project Cost Management] - [Weightage 13%]

## 1. Problem Statement

The Department Head in a Software Company has to allocate funds for the implementation of projects on new innovative topics. Each project lead has submitted the cost details and its overall potential return on investment (ROI) that can be achieved. The total budget available to the Department Head to allocate to the project is 150 crores. Here your task would be to help the department head to select as many projects as possible with the budget constraints such that the total ROI is maximized. If there are multiple solutions with same ROI, choose the one which results in maximum utilization of funds and selection of projects as well.

**Requirements:**

1. Formulate an efficient recursive algorithm using Dynamic Programming to determine how to select the projects to be funded and maximize ROI.
2. Analyse the time complexity of your algorithm.
3. Implement the above problem statement using Python 3.7

**Sample Input:**

For example, if there are 10 different projects in total and their budget requirements and values are given as shown:

| Projects (i) | Cost $C_i$(crores) | ROI $R_i$ |
|---|---|---|
| 1 | 30 | 60 |
| 2 | 17 | 35 |
| 3 | 20 | 55 |
| 4 | 10 | 40 |
| 5 | 13 | 66 |
| 6 | 10 | 20 |
| 7 | 16 | 35 |
| 8 | 25 | 50 |
| 9 | 15 | 70 |
| 10 | 18 | 10 |

Input should be taken in through a file called "**inputPS3.txt**" which has the fixed format mentioned below using the "/" as a field separator:

<Project name i> / < Cost Ci(crores)> / < ROI Ri>

Ex:

1 / 30 / 60
2 / 17 / 35
3 / 20 / 55
…

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

**Sample Output:**

The projects that should be funded: 1,2,3,4,5,7,8,9
Total ROI: 411
Fund remaining: 28

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

Display the output in **outputPS3.txt**.

## 2. Deliverables

1. Word document **designPS3_<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. **inputPS3.txt** file used for testing
4. **outputPS3.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]_A2_PS3_ProjectCostManagement.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.

**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 **28th Feb, 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 13 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. If two different groups submit the same code, both teams will get zero marks.
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:** 5.3