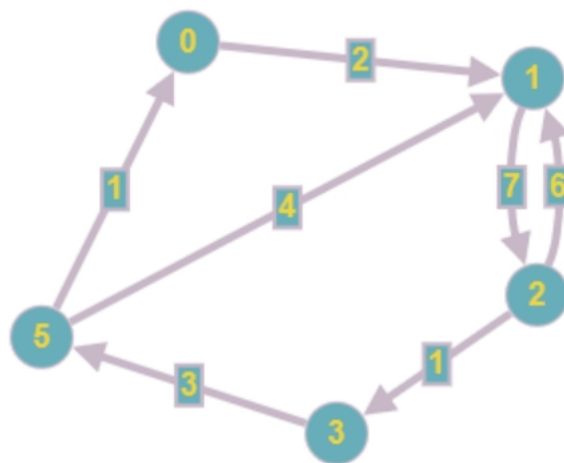


(S1-21_DSECLZG519)
(Data Structures and Algorithms Design)
Academic Year 2020-2021

Assignment 2 – PS14 - [State Roadmap] - [Weightage 13%]

1. Problem Statement

There are dedicated teams in the state roadways department, which handles the task of preparation of tables indicating distances between all pairs of major cities and towns in road maps of states. Assume that you are part of such a team for State X. Your task is to find the shortest distances between all pairs of major cities and towns within the state X. The aim should be to find the highly efficient method for obtaining these shortest paths. The following graph contains five nodes, and various directed and weighted edges. Consider this as the representation of the cities and the paths between them.



Requirements:

1. Formulate an efficient algorithm to perform the above task using Dynamic programming
2. Provide a description about the design strategy used
3. Analyse the time complexity of the algorithm and show that it is an “efficient” one.
4. Implement the above problem statement using Python 3.7

Sample Output:

The shortest distances between all pairs of major cities and towns within the state X in matrix format is

0, 2, 9, 10, 13,

12, 0, 7, 8, 11,

5, 6, 0, 1, 4,

4, 6, 13, 0, 3,

1, 3, 10, 11, 0,

Explanation:

First row of output is the shortest distance from state 0 to all other states in sequence and so on.

State	0	1	2	3	5
0	0	2	9	10	13
1	12	0	7	8	11
2	5	6	0	1	4
3	4	6	13	0	3
5	1	3	10	11	0

Display the output in **outputPS14.txt**.

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 **designPS14_<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. **inputPS14.txt** file used for testing
4. **promptsPS14.txt** file used for testing (If applicable)
5. **outputPS14.txt** file generated while testing
6. **.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_PS14_StateRoadmap.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
 - a. The data structure model you chose with justifications
 - b. Details of each operation with the time complexity and reasons why the chosen operations are efficient for the given representation
 - c. 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 **Saturday, 12th Mar, 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 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. Copy / Paste's from web resources / or your friends' submission will attract severe penalty to the extent of awarding negative 10 percent. 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.

