



Birla Institute of Technology & Science, Pilani

Work-Integrated Learning Programmes Division

MTech in Data Science & Engineering

S1_2024-2025, DSECLZG519- Data Structures & Algorithms Design

Assignment 2 – PS04 - [Event Management System] - [Weightage 13%]

Read through this entire document very carefully before you start!

Problem Statement

As a part of the work from home system, BITS Pilani have decided to conduct the classes and meetings virtually. It has become difficult for the faculties and they need help to manage and keep a track of the events. Your task is to develop an event management system to manage a calendar of events. Each event has a unique identifier, a start time, an end time, and an event name. To efficiently organize and search for events based on their start times, you decide to implement an AVL tree. Implement a Python class for an AVL tree specifically designed for storing events based on their start times. The Event Management System should provide the below functionalities:

1. Inserting an event
2. Deleting an event by ID
3. Searching an event by ID
4. Searching events within a given time range

To solve this problem, create an AVL tree of events along with their ID, start time, end time and event name. You'll be given a few commands in the input file.

Operations:

1. def initiateEventManagerSystem (read_input_file): This function reads the input file and creates an event management system and all associated data structures and calls the necessary functions as mentioned in the input file.

Input: Input-File name with path.

Output: None

2. def addEvent (event_ID="", start_time="", end_time="", event_name=""): This function is called by “initiateEventManagementSystem” and creates an event along with a unique event ID, start time, end time, event name and writes this event ID and event name in the output file.

Input: <Event ID> - <Start Time> - <End Time> - <Event Name>

Output: ADDED:<Event ID assigned to this event> - <Event name>

The date format should be dd/mm/yyyy hh:mm:ss (24 hour time format). This format should be maintained throughout the assignment.

3. def removeEvent (event_ID=""): This function is called by initiateEventManagementSystem and removes an event along with a unique event ID and writes this event ID and event name in the output file.

Input: Event ID

Output: REMOVED:<Unique Event ID assigned to this event.>-<Event Name>

4. def searchEvent (event_ID=""): This function is called by initiateEventManagementSystem and finds an event along with a unique event ID and writes this event details in the output file.

Input: Event ID

Output: SEARCHED:<search-ID>

<Event ID>-<Event Name> -<Event Start time>-<Event End time>

5. def searchEventByRange(Range Start Time="", Range End time=""): This function is called by initiateEventManagementSystem and finds events that are scheduled in the range provided and writes the list of events along with their start time, end time and name in the output file.

Input: <Range Start Time>-<Range End Time>

Output: SEARCHED:Events from < Range Start Time > to < Range End Time >

<Event1 ID>-<Event1 Name> -<Event1 Start time>-<Event1 End time>

<Event2 ID>-<Event2 Name> -<Event2 Start time>-<Event2 End time>

<Event3 ID>-<Event3 Name> -<Event3 Start time>-<Event3 End time>

<Event4 ID>-<Event4 Name> -<Event4 Start time>-<Event4 End time>

.

.

.

.

.

<EventN ID>-<EventN Name> -<EventN Start time>-<EventN End time>

Requirements:

1. Implement the above problem statement using **Python 3.7**
2. Read the input from a file(**inputPS04.txt**), which contains the list of actions to be performed.
3. You will output your answers to a file (**outputPS04.txt**) for each line.
4. Perform an analysis for the features above and give the running time in terms of input size: n.

Sample Input

Input will be taken from the file(inputPS04.txt).

Add Event: 1 - 01/01/2024 10:00:00 - 01/01/2024 11:00:00 - Opening Ceremony

Add Event: 2 - 01/01/2024 13:00:00 - 01/01/2024 13:30:00 - Keynote Address

Remove Event: 1

Search Event by ID: 2

Add Event: 1 - 01/01/2024 10:00:00 - 01/01/2024 11:00:00 - Opening Ceremony

Add Event: 3 - 02/01/2024 10:00:00 - 02/01/2024 11:00:00 - Panel Discussion

Add Event: 4 - 01/01/2024 16:00:00 - 01/01/2024 17:00:00 - Introduction to DSAD

Remove Event: 5

Search Event by Range: 31/12/2023 12:00:00 - 01/01/2024 13:00:00

Search Event by Range: 01/01/2024 11:00:00 - 01/01/2024 23:00:00

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

Display the output in outputPS04.txt.

ADDED: 1 - Opening Ceremony

ADDED: 2 - Keynote Address

REMOVED: 1 - Opening Ceremony

SEARCHED: 2

2 - 01/01/2024 13:00:00 - 01/01/2024 13:30:00 - Keynote Address

ADDED: 1 - Opening Ceremony

ADDED: 3 - Panel Discussion

ADDED: 4 - Introduction to DSAD

Event to be removed not found

SEARCHED: Events from 31/12/2023 12:00:00 to 01/01/2024 13:00:00

1 - 01/01/2024 10:00:00 - 01/01/2024 11:00:00 - Opening Ceremony

2 - 01/01/2024 13:00:00 - 01/01/2024 13:30:00 - Keynote Address

SEARCHED: Events from 01/01/2024 11:00:00 to 01/01/2024 23:00:00

2 - 01/01/2024 13:00:00 - 01/01/2024 13:30:00 - Keynote Address

4 - 01/01/2024 16:00:00 - 01/01/2024 17:00:00 - Introduction to DSAD

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

1. Deliverables

1. PDF document **designPS04_<group id>.docx** detailing your solution.
2. **[Group id] _Contribution.xlsx** mentioning the contribution of each student in terms of percentage of work done. Columns must be “Student Registration Number”, “Name”, “Percentage of contribution out of 100%”. If a student did not contribute at all, it will be 0%, if all contributed then 100% for all.
3. **inputPS04.txt** file used for testing
4. **outputPS04.txt** file generated while testing
5. .py file containing the python code. Create a single notebook. Do not fragment your code into multiple files
6. Zip all of the above files including the design document in a folder with the name:
 - a. **[Group id] _A2_PS04_EventManagementSystem.zip** and submit the zipped file.
 - b. 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.

2. 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 you 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 operations with the time complexity and reasons why the chosen operations are efficient for the given representation
 - c. One alternate way of modeling the problem with the cost implications.
9. Writing a good technical report and well documented code is an art. Your report cannot exceed 4 pages. Your code must be modular and quite well documented.
10. You may ask queries in the dedicated [discussion section](#). Beware that only hints will be provided and queries asked **in other channels will not be responded to**.

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. You can create in Notebook and download as .py if needed.
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.

3. Deadline

1. The strict deadline for submission of the assignment is **Sunday, March 09, 2025 11:55PM**.
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.

4. 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. Assignments submitted via other means like email etc. will not be graded.

5. 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.
 - d. Every bug in the functionality will have negative marking.
 - e. Marks will be deducted if your program fails to read the input file used for evaluation due to change / deviation from the required syntax.
 - f. Use of only native data types and avoiding libraries like numpy, graph and pandas will get additional marks.
3. We encourage students to take the upcoming assignments and examinations seriously and submit only original work. Please note that plagiarism in assignments will be taken seriously. All groups that are booked under plagiarism will be given 0 marks and no further discussion will be entertained. Please refer to the detailed policy in ELearn Files.
4. Source code files which contain compilation errors will get at most 25% of the value of that question.

6. Readings

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