



(S2-19_DSECLZG519)
(Data Structures and Algorithms Design)
Academic Year 2019-2020

Assignment 1 – PS2 - [Theme Park] - [Weightage 12%]

1. Problem Statement

The manager of a theme park is trying to improve their visitor information system and is seeking your help in developing the new system. The manager would like to use the new system to save details of visitors including their Full Name, Date of Visit, Date of Birth, Home City and Phone number. He needs this information to help him do the following:

1. **Visitor Search:** In the unfortunate instance of a child getting separated from their parent, the Manager wants to use this function to find the parent's details based on the parent's first name. The Visitor Search will then display all visitors to the park on that day with the specific first name along with their city and phone number.
2. **Visitor Count:** Find out how many visitors came on any particular day.
3. **Trending City:** Find out which city is trending with the highest number of visitors today.
4. **Birthday List:** Get a list of past visitors celebrating their birthday on a given date range. He intends to give them a special discount coupon for their birthday.

Asks:

1. **Implement the above problem statement in Python 3.7 using Hash Tables. Avoid using Python dictionaries.**
2. Design a hash table, which uses Visitor Name as the key to hash elements into the hash table. Generate necessary hash table definitions needed and provide a design document (1 page) detailing clearly the design and the details of considerations while making this design and the reasons for the specific choice of hash function.
3. Use separate chaining collision resolution technique.
4. **Perform an analysis for the features above and give the running time in terms of input size: n.**

Operations:

1. **def insertVisitor(self, key, value):** This function inserts a record for each new visitor registering or visiting the park. It reads the file **inputPS2.txt** and feeds the records in the Hash-Table. Each line in the input file contains one record. Each record has Full Name, Date of Visit, Date of Birth, Home City and Phone number separated by commas. A sample input file entry is show below:

Amar Pradhan, 12-Jan-2020, 12-Jan-2008, Delhi, 9892342134

After all visitor entries are read and recorded, the function outputs the total number of visitors in the output file **outputPS2.txt**. The output text will be:

----- insert -----

Total visitor details entered: 200

2. **def findVisitor(self, name=None):** This function is triggered when the following tag “findVisitor” is encountered in the **promptsPS2.txt** file. It takes the first name of the visitor from the prompt file.

findVisitor: Amar

The function searches for the visitors from the record(s) of today (Assume that the greatest date in the input file is the reference date for today) that match the first name, and returns the list of Full Names, Home City and Phone Number. These details are output to the **outputPS2.txt** file in the below format.

----- findVisitor: -----

3 visitors with name 'Amar' found visiting on 12-Jan-2020

Amar Pradhan, Delhi, 9892342134

Amar Singh, Agra, 9342124222

Amar Pandey, Haryana, 8234121343

3. **def visitorCount(self, visitDate=None):** This function is triggered when the following tag “visitorCount” is encountered in the **promptsPS2.txt** file. It takes the date of visit from the prompt file.

visitorCount: 12-Jan-2020

The function searches for all visitors from the record(s) who visited on the input date and returns the total count of visitors for that date. The details are output to the **outputPS2.txt** file in the below format.

----- visitorCount: -----

200 visitors found visiting on 12-Jan-2020

4. **def cityVisitor(self):** This function is triggered when the following tag “trendCity” is encountered in the **promptsPS2.txt** file.

trendCity:

The function searches for all visitors from the record(s) who are visiting today (Assume that the greatest date in the input file is the reference date for today) and finds the city with the greatest number of visitors. The details are output to the **outputPS2.txt** file in the below format.

----- trendCity: -----

45 visitors from Delhi visiting today

5. **def birthdayVisitor(self, birthDateFrom, birthDateTo):** This function is triggered when the following tag “birthdayVisitor” is encountered in the **promptsPS2.txt** file. It takes the date range from the prompt file. The from and to dates are separated by a colon ‘:’

birthdayVisitor: 12-Jan : 19-Jan

The function searches for all visitors from the record(s) who have birthdays in the given range and returns the list of visitors that match. The details are output to the **outputPS2.txt** file in the below format.

----- birthdayVisitor: -----

5 visitors have upcoming birthdays between 12-Jan and 19-Jan

Amar Pradhan, 12-Jan-2008, 9892342134

Giri Prakash, 13-Jan-2005, 9234212323

Radha R, 17-Jan-1998, 9812310112

Anju George, 13-Jan-1995, 8891231231

Prasad Murthy, 14-Jan-2002, 9123123210

6. Include all other functions that are required to support these basic mandatory functions.

2. Sample file formats

Sample inputPS2.txt

Amar Pradhan, 12-Jan-2020, 12-Jan-2008, Delhi, 9892342134

Giri Prakash, 12-Jan-2018, 13-Jan-2005, Bangalore, 9234212323

Radha R, 12-May-2019, 17-Jan-1998, Chennai, 9812310112

Anju George, 14-Sep-2018, 13-Jan-1995, Kerala, 8891231231

Prasad Murthy, 12-Feb-2019, 14-Jan-2002, Chennai, 9123123210

Amar Singh, 12-Jan-2020, 12-Apr-1995, Agra, 9342124222

Amar Pandey, 12-Jan-2020, 1-Jan-1989, Haryana, 8234121343

.....

Sample promptsPS2.txt

findVisitor: Amar

visitorCount: 12-Jan-2020

trendCity:

birthdayVisitor: 12-Jan : 19-Jan

Sample outputPS2.txt

----- insert -----

Total visitor details entered: 200

----- findVisitor: -----

3 visitors with name 'Amar' found visiting on 12-Jan-2020

Amar Pradhan, Delhi, 9892342134

Amar Singh, Agra, 9342124222

Amar Pandey, Haryana, 8234121343

```

----- visitorCount: -----
200 visitors found visiting on 12-Jan-2020
-----

----- trendCity: -----
45 visitors from Delhi visiting today
-----

----- birthdayVisitor: -----
5 visitors have upcoming birthdays between 12-Jan and 19-Jan
Amar Pradhan, 12-Jan-2008, 9892342134
Giri Prakash, 13-Jan-2005, 9234212323
Radha R, 17-Jan-1998, 9812310112
Anju George, 13-Jan-1995, 8891231231
Prasad Murthy, 14-Jan-2002, 9123123210
-----

```

2. Deliverables

- A1_PS2_TP_[Group id] package folder** containing all the modules classes and functions and the main body of the program.
- inputPS2.txt** file used for testing
- promptsPS2.txt** file used for testing
- outputPS2.txt** file generated while testing
- analysisPS2.txt** file with answers on the hash function design and running times (ask number 2 and 4)

3. Instructions

- It is compulsory to make use of the data structure/s mentioned in the problem statement.
- Do not use inbuilt data structures available in Python. The purpose of the assignment is for you to learn how these data structures are constructed and how they work internally.
- It is compulsory to use Python 3.7 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 your read, understand, and follow all the instructions

- g. Ensure that the input, prompt and output file guidelines are adhered to. Deviations from the mentioned formats will not be entertained.
- h. 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.
- i. Run time analysis is provided in asymptotic notations and not timestamp based runtimes in sec or milliseconds.

4. Deadline

- a. The strict deadline for submission of the assignment is 18th June, 2020.
- b. The deadline already accounts for extra days for the students to work on the assignment. No further extension of the deadline will be entertained.
- c. Late submissions will not be evaluated.

5. 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 **all the deliverables** into one file and name the zipped file as below “ASSIGNMENT1_[G1/G2/...].zip” and upload in CANVAS in respective location under ASSIGNMENT Tab.
- d. Assignments submitted via means other than CANVAS **will not be graded**.

6. Evaluation

- a. The assignment carries 12 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
- 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.

7. Readings

Section 2.5: Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition)