

BAHRIA UNIVERSITY (KARACHI CAMPUS)

**CSC 221 - Data Structures & Algorithms - Assignment 3**

CLO-4 Deadline: 1st February , 22

Class: BSE-3B Total Marks 10

Name : **Muhammad Junaid Saleem Qadri** E.no: **02-131202-057**

1. A database contains information about all trains leaving the Washington Union station on

Monday. Each train is assigned a departure time, a destination, and a unique 8-digit train ID number. Examine which data structures you can use to solve each of the following scenarios.

a) The schedule contains 200 trains with 52 destinations.

b) Trains with the same destination should further be sorted by departure time.

Depending on scenario, you may need to either (a) use multiple data structures or (b) modify the implementation of some data structure. Justify your choice.

**Solution :-**

**a)** One solution would be to use a dictionary where the keys are the destination, and the value is a list of corresponding chains. Any dictionary implementation would work.

Alternatively, we could modify a separate chaining hash table so it has a capacity of exactly 52 and does not perform resizing. If we then make sure each destination maps to a unique integer from 0 to 51 and hash each train based on this number, we could fill the table and simply print out the contents of each bucket.

A third solution would be to use a BST or Red-Black tree and have each train be compared first based on destination then based on their other attributes. Once we insert the trains into a tree, we can print out the trains sorted by destination by doing an in-order traversal.

**b)** Regardless of which solution we modify, we would first need to ensure that the train objects are compared first by destination, and second by departure time.

We can modify our first solution by having the dictionary use a sorted set for the value, instead of a list. (The sorted set would be implemented using a BST or a Red-Black tree).

We can modify our second solution in a similar way by using specifically a BST or a Red-Black tree as the bucket type.

Our third solution requires no modification: if the trains are now compared first by destination and second by departure time, the Red-Black and BST tree’s iterator will naturally print out the trains in the desired order.

2. You are playing a game. The game consists of 10 questions which helps to guess the name of

famous people. The rules of this game are:

* Your opponent will think of the name of a famous person
* You ask a question with Yes or No
* Your opponent gives you the answer to that question
* Based on your opponent answer, you will ask another yes-or-no question, so on
* You cannot ask more than 10 yes-or-no questions
* If you guess the name of the person, you win; otherwise, you lose.

Compare which data structure can help you to implement the above strategy.

**Solution :-**

**STRATEGY**

The data structure used for the implementation of above guessing game is **BINARY TREE** in which we have two nodes of each root.

**EXPLANATION**

* There will be ten roots which will hold each question.
* Each node will have two nodes having answers YES & NO, as binary tree only have two child nodes.
* If the answer of particular question results in YES it will not proceed further.
* But if last question will be again a NO then the game will end, as there is no further node in tree.

