Indian Institute of Technology, Guwahati

Department of Computer Science and Engineering

Data Structure Lab (CS210)

Assignment: 10

Date: 3rd November, 2016.

Total Marks: 20 (Lab Assignment) + 20 (Offline assignment)

Deadline of Offline Assignment Submission: 9th November, 2016.

Submission Instructions:

- Name of every file should be <Roll_No>_<Question_No>.c. eg. 150101086_1.c, 150101086_2.c and so on.
- Name of the folder you upload should be <Roll_No> eg. 154101086.
- Please do not upload files like a.out, desktop.ini,etc. If there are 'N' questions in assignment, there must be exactly 'N' ".c" or ".cpp" files in the folder. Use only zip to compress the folder to be uploaded.
- You will be awarded ZERO mark if the above rules are violated.
- Please note that you will not be able to submit once the dead line (with 12 hours' late submission) is over.

Lab Assignment:

1. Write a program that computes the number of collisions required in a random sequence of insertions using (i) Linear Probing, (ii) Quadratic Probing and (iii) Double Hashing. Consider the hash table of length m for all cases with the primary hash function h'(k) = k mod m. The table size m is user input. For quadratic probing, c1 = 1 and c2 = 3, and for double hashing, h2(k) = 1 + (k mod (m - 1)). You need to create three hash tables, one for each (i) Linear Probing, (ii) Quadratic Probing and (iii) Double Hashing. Your Insert function should handle deletion of key from hash tables. Implement Delete function as well. **(20)**
    a. For your first experiment, set m =11. Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 88, 59. For each insertion, print the index/location where it is placed in hash table and the number of collisions for each three cases (i.e., (i) Linear Probing, (ii) Quadratic Probing and (iii) Double Hashing.)
    b. In next set of experiments, you take random number of random inputs and print the index/location where each input is placed in hash table and the number of collisions for each three cases (i.e., (i) Linear Probing, (ii) Quadratic Probing and (iii) Double Hashing.). If the number of entries is more than the table size, your insertion function should errors out with overflow message.

Offline Assignment:

2. Write a function to implement the following strategy for multiplying two sparse polynomials P1 and P2 with size M and N, respectively. Each polynomial is represented as a linked list. Each cell/term consists of a coefficient, an exponent, and a next pointer. To multiply, we need to multiply each term of P1 with each term of P2. So, there are MN operation which we cannot ignore. The problem is how we combine the resultant terms with same exponent into one term. One method is to sort these terms and combine terms with exponent. But these require sorting MN records. Alternatively, we could merge terms as they are computed. Write a program to implement this alternative strategy. **(10)**
Hints: You may use a hash table to store the terms of resultant polynomial hashed by their exponents.

3. A book publisher keeps track of its books by storing the information about each book in a binary search tree (BST) ordered on the ISBN number. For each book, in addition to the ISBN number (and the left and right branches in the tree), the following information is stored: the title, author, year of publication, cost, and number of copies sold. **(20)**
    a. Create a BST containing at least 5 books. Make sure you construct your example so that the items in the tree are ordered according to the binary search tree property, on the ISBN number.
    b. Write a function increase-10-percent that consumes a binary search tree of books and produces a binary search tree the same as the original except that the cost of each book in the tree has been increased by 10 percent.
    c. Write a function copies-sold which consumes a binary search tree an ISBN number, and returns the number of copies sold for the book with the given ISBN. If a book with the given ISBN doesn't exist in the tree, the function should return -1. Your function should be written efficiently, such that it performs as few comparisons as is necessary to find the correct ISBN number in the tree.
    d. Write a function add-new-book. The function consumes a binary search tree and a book and adds the book to the binary search tree. Make sure that the tree that is produced is a binary search tree. You may assume that the ISBN number of the book to be added does not already exist in the given tree.

    Note: Your code should follow the format of ISBN number as mentioned in https://www.isbn-international.org/content/what-isbn