Algorithms Report2 (Due date: 5 PM, Nov. 5)

I) Problem solving manually

(Keys are considered from left to right for the insertion, deletion and etc.)
(Must show all the intermediate process leading to a solution.)

1. Consider inserting the keys 12, 28, 31, 7, 28, 15, 17, 66, 59, 21, 3, 1 into a hash table of length m = 5 using separate chaining where $h(k) = k \bmod m$. Illustrate the result of inserting these keys.

2. Consider inserting the keys 2, 18, 9, 7, 12, 10, 21, 11, 8 into a hash table of length **m = 11** using open addressing with the auxiliary hash function $h'(k) = k \bmod m$. Draw the hash tables after inserting these keys

a) using linear probing whith $h(k,i) = (h'(k) + i) \bmod m$

b) using quadratic probing with $h(k,i) = (h'(k) + c_1 + c_2 i^2) \bmod m$,
where $c_1 = 1$ and $c_2 = 3$

c) using double hashing with $h(k,i) = (h'(k) + i h_2(k)) \bmod m$,
where $h_2(k) = 1 + (k \bmod (m-1))$,

for i=0,1,...,m-1. (Must show the all the hash calculations.)

3. Using Figure 7.1 as a model, illustrate the operation of PARTITION on the array A = <11, 8, 13, 9, 12, 5, 16, 1, 15>. Pivot is the first element of the array A.

4. Answer the following questions for the keys
11, 2, 17, 5, 3, 10, 6, 7, 12, 20

a. Draw the final structure of binary search tree T when above keys are inserted from left to right.

b. Draw the tree that results after successively executing the following functions.
TREE-DELETE(T,12), TREE-DELETE(T,17), TREE-DELETE(T,5).

5. Write the pseudo for MAX(T) in a tree. The MAX(T) finds a node with the maximum key value in T.

6. Draw the red-black tree that results after TREE-INSERT is called on the tree in Figure 13.1(c) with key 36. If the inserted node is colored red, is the resulting tree a red-black tree? What if it is colored black?
Answer without TREE-INSERT-FIXUP execution.

7. Draw the red-black trees that result after successively inserting the keys in the order 7, 5, 9, 2, 8, 3, 13 into an initially empty red-black tree.
Also, for each insertion, count the number of color changes, left rotations, and right rotations and the sum for each of these three operations.

8. Draw the red-black trees that result from the successive deletion of the keys in the order 2, 7, 9, 5, 3, 13, 8 on the tree generated in exercise 7.
Also, for each deletion, count the number of color changes, left rotations, and right rotations and the sum for each of these three operations.

II) Programming

1. Hash Table (Separate Chaining)

● Construct the hash table with separate chaining according to the following direction.

1) Hash functions (Construct three hash tables.)

a) h(k) = k mod 7
b) h(k) = k mod 13
c) h(k) = k mod 17

2) Insert the 50 keys to above three tables.
The keys are generated by rand()%1000. Execute srand(time(NULL)) first.
(Duplicate keys should be avoided.)
Implement a function that inserts the randomly generated keys into the hash table.

3) Print the three tables.
(Different chains in a table should be printed in different lines.)

- Implement a print function for a hash table printing.
- Input to the function is a hash table.
- The print function should print the shortest, longest, and average length of the chains for the input hash table.

2. Construct the open address hash table according to the following description.

- m = 37
- Hash functions (not exactly same as the previous open address hash functions)
  linear probing: $h(k, i) = (h'(k) + i) \bmod m$
  where, $h'(k) = k \bmod m$
  quadratic probing: $h(k, i) = (h'(k) + c_1 i + c_2 i^2) \bmod m$
  where, $h'(k) = k \bmod m$, $c_1 = 1$, $c_2 = 3$.
  double hashing: $h(k, i) = (h_1(k) + i h_2(k)) \bmod m$,
  where, $h_1(k) = k \bmod m$, $h_2(k) = 1 + (k \bmod (m - 1))$.

- **30 keys** that are randomly generated.
  The keys are generated by rand()%1000. Execute srand(time(NULL)) first.
  (Duplicate keys should be avoided.)

1) Insert the randomly generated 30 keys to the three hash tables.
(note that you need a function for the 90 insertions, 30 for each table)

2) Print the contents of the hash table for above three different hash functions.
(Must show the correct positions of the inserted keys in the table).
The print function also should print the average number of probes and primary (largest) cluster of the hash table.

3. RBT (Red-Black Tree)

1) Program the following functions.

a. RB-INSERT(T, z) /* Do not attach z node if it is already in T (RBT) */
b. RB-DELETE(T, z)
c. PRINT-BST(T)

2) Using RB-INSERT(T, z) construct a RBT with the keys in A[20] and print the T.

Fill in A[20] by rand()%50. Execute srand(time(NULL)) first.
(Duplicate keys should be avoided.)

3) Execute RB-INSERT(T, 2), RB-INSERT(T, 22), RB-INSERT(T, 13), RB-INSERT(T, 47), RB-INSERT(T, 36), sequentially.
Execute PRINT-BST(T), each time after the insertion is performed.
(If a key is already in T the insertion is ignored.)

4) Execute RB-DELETE(T, 6), RB-DELETE(T, 17), RB-DELETE(T, 21), RB-DELETE(T, 7), RB-DELETE(T, 45) sequentially.
Execute PRINT-BST(T) each time after the deletion is performed.
(It is possible that a key to be deleted may not be in the tree.)

How to submit the report.

▶ Need to upload in the i-campus a zip file for the report2.

Refer to the manual file for uploading in the i-campus.

▶ The zip file contains the following three files.

1) Document file (.hwp, photo, or scan): Problem solving manually part.
2) C program file(s): Programming part.
3) Test result file(s): Contains all the screen copy of the test results.

▶ The zip file should be named as shown below,

report2_id_name.zip

example) report2_2020123456_HongGilDong.zip

The zip file contains above 1), 2), and 3).