COM2067/COM267 **01.02.22** **120 m. Resit Exam**

Name:

Student ID:

Answer 10 questions.

1. **(10 p)** Insert the following integer keys into the hash table using the following two different methods. (m = 10).
   25, 55, 13, 23, 98, 11, 34, 48, 74
   a-) Double Hashing
   h(k, i) = [h1(k) + ih2(k)] mod m (h1= k mod 10, h2= k mod 7)
   b-) Quadratic Probing
   h(k, i) = [h'(k) + c1*i + c2*i^2] mod 10 (h'= k mod 10, c1=2, c2=1)

   a-)

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

   b-)

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

2. **(10 p)** Insert the following integer keys into a red-black tree. Show the red-black tree after each insertion.
   224, 218, 243, 257, 290, 235, 260

3. **(10 p)** Insert the following integer keys into an AVL tree. Show the AVL tree after each insertion.

   119, 127, 150, 115, 112, 121, 136

   Then, delete keys **127** and **121** from the AVL tree. Show the AVL tree after each deletion.

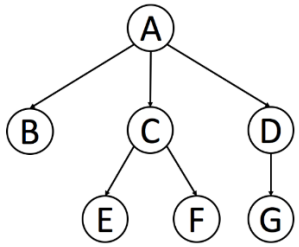
4. **(10 p)** Apply the following insert operations on B tree (m=5). Show B tree after each operation.
   Insert **224, 218, 243, 257, 290, 235, 260, 19, 127, 150, 115, 12, 121, 136**

5. **(10 p)** Construct a binary tree from the following traversal results.
   In-Order Traversal: 1 3 5 7 9 2 4 6 8
   Pre-Order Traversal: 2 3 1 7 5 9 4 8 6

6. **(10 p)** Insert the following integer keys into a binomial-heap. Show the binomial-heap after each insertion.
   110, 20, 10, 90, 60, 40, 70, 100, 120, 50
   Extract the node with the minimum key from that binomial-heap. Draw the produced binomial heap after this operation.

7. **(10 p)** Apply the following insert operations on 2-3 tree. Show 2-3 tree after each operation.
   174, 121, 116, 152, 141, 125, 147

8. **(10 p)** Convert the given general tree into a binary tree.



9. **(10 p)** Create a Huffman tree using the following uppercase letters. Assume that only those letters are used in the text. Given values represent the estimated frequencies of each letter in a paragraph. Using the created Huffman tree, specify a binary coding for each letter.

M	N	O	U	I	E	A
15	18	22	28	35	45	65

10. **(10 p)** Insert the following integer keys into max-heap. Show the max-heap after each insertion.

265, 249, 281, 227, 288, 245, 236

11. **(10 p)**

```
struct node {
  int data;   struct node * leftChild;   struct node * rightChild;   };
```

a) (5 p) Write pseudo-code of the algorithm (or C code) to calculate the total number of internal nodes in a binary search tree.

int totalInternalNodes(struct node * tree)

b) (5 p) Write a function (C code) to print all the data values of nodes in a binary search tree using post-order walk.

void printPostOrder(struct node * tree)

12. **(10 p)** Write the definition of following functions (C or C++) using the given node structure.

```
struct node {
  int data;
  struct node * parent;
  struct node * leftChild;
  struct node * rightChild;
};
```

struct node * grand_parent(struct node *n)

struct node * uncle(struct node *n)

Grandparent node (G) of a node (N) refers to the parent of N's parent (P).

Uncle node (U) of a node (N) refers to the sibling of N's parent (P).