Bahria University, Islamabad Campus

Department of Computer Science

Final Examination

Class/Section: BSCS-3(A/B)

(Spring 2021 Semester)

Paper Type: Descriptive

Course:	**Data Structure and Algorithm**	Date: **17/07/2021**
Course Code:	**CSC-221**	Time: **8:30 AM-11:00AM**
Faculty's Name:	**Momina Moetesum**	Max Marks: **50**
Time Allowed:	**2.5 Hours**	Total Pages: **3**

INSTRUCTIONS:

I. All questions are compulsory. Attempt all questions on the **"ANSWER SHEET".**
II. There are total three questions with parts.
III. The paper is closed book.
IV. The students are not allowed any helping material (books, tables, formulas, etc).
V. Use blue, black or blue-black ink only. Do NOT use lead pencil especially.

Student's Name: ______________________________ Enroll No:________________________
(USE CAPITAL LETTERS)

Question # 1: (25 Marks)

i. Consider the following ADTs for Node, TreeNode, Singly Linked List, and Binary Search Tree and attempt the following. **(3+3)**

Node and Singly Linked List ADT	TreeNode and Binary Search Tree ADT

`class Node`
`{`
`  public:`
`    int data;`
`    Node *next;`
`};`
`class List`
`{`
`  public:`
`    Node* head;`
`    List();`
`    bool isEmpty();`
`    void insert_end(int);`
`    /*All basic member functions*/`
`};`
	`class TreeNode`
	`{`
	`  public:`
	`    int data;`
	`    TreeNode *left;`
	`    TreeNode *right;`
	`};`
	`class BSTree`
	`{`
	`  public:`
	`    TreeNode* root;`
	`    BSTree();`
	`    bool isEmpty();`
	`    /*All basic member functions*/`
	`};`

a) Create a non-member function **List SortedList(TreeNode *root, List sort),** that when given the root of a BST returns a sorted singly linked list containing the data stored in the BST.

b) Create a non-member function **int SumNodes(TreeNode *root),** that when given the root of a BST returns the sum of the integers stored in the BST.

ii. Consider the given collection of words **{paint, safe, cartoon, rainbow, harsh, wise, rain, camera}** in the same order. **(2+2+2+2)**

a) Insert the given data into a Binary Search Tree (BST).
b) Insert the given data into a MaxHeap.
c) Redraw the BST in **part a** after deleting the node containing the minimum value.
d) Redraw the MaxHeap in **part b** after deleting the node containing the maximum value

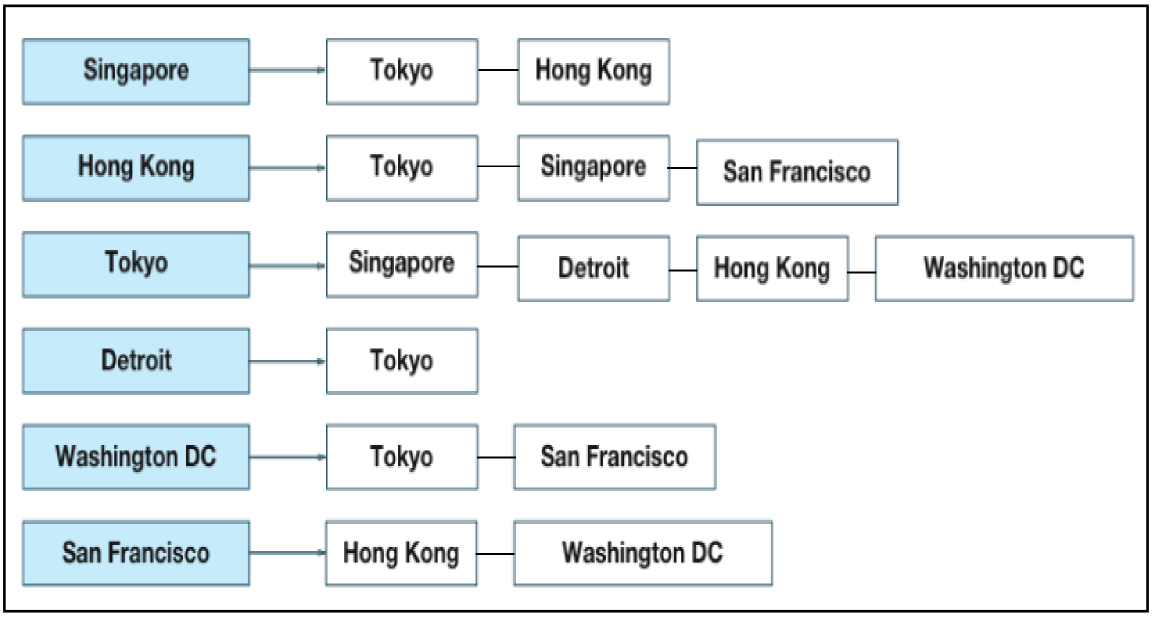
iii. Consider the given frequency table and attempt the following. **(5+2+2+2)**

Characters	C	E	I	R	S	T	X
Frequency	12	8	18	10	9	5	2

a) Draw a Huffman tree and provide codes for each character.
b) Using the constructed Huffman tree, encode the following:
  i. TEXT
  ii. EXERCISE
c) Using the constructed Huffman tree, decode the following:
  i. 1000001101001100100
  ii. 10111100001
d) Compute the average bits required to encode the string **EXERCISE** using the constructed Huffman tree.

Question # 2: **(15 Marks)**

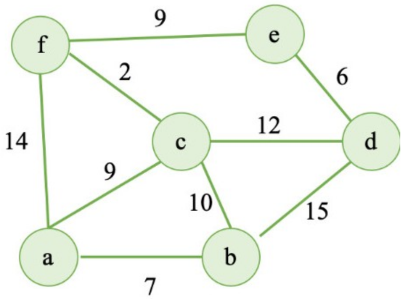
1. The flights operated by an international airline are shown by the adjacency list given below. Draw the resultant graph and also determine how many routes exist between **Tokyo** and **San Francisco** in which the plane makes a single transit (change of flight in-between). Identify those routes. **(5)**


Singapore → Tokyo — Hong Kong
Hong Kong → Tokyo — Singapore — San Francisco
Tokyo → Singapore — Detroit — Hong Kong — Washington DC
Detroit → Tokyo
Washington DC → Tokyo — San Francisco
San Francisco → Hong Kong — Washington DC

2. Consider the graph below and attempt the following. **(6+4)**

a) Find the shortest path from **Node a** to **Node e** using Djikstra's Algorithm. Your answer should contain a distance table, a list showing the order of nodes visited, and the final graph with node weights.

b) Create a Minimum Spanning Tree (MST) using Kruskal Algorithm.



Question # 3: **(10 Marks)**

1. Insert the given strings **key={energy, library, obey, pilot, look, hammer}**, in a hash table of size **7**, using the hash function **H(key)=key%tablesize**. In case of collision, use **Quadratic probing** for collision resolution. To convert non-numeric keys into numeric values, map each of the first three characters of the string to a number equivalent to its position in the alphabet (e.g. a=1, b=2, ..., z=26 etc.) and add them. For instance, numeric value of "abc" is (a=1)+(b=2)+(c=3)=6. **(4)**

2. Show the contents of the given array Arr[]={30,60,80,20,90,50,10,70,40} after **Pass 1** of Quick Sort. Consider the first element as Pivot. **(3)**

3. Compute the processing time of the following function and determine its Big-O complexity. (Hint: Be aware of the **break** statement). **(3)**

```
function(int n)
{
   if (n == 1)
     return;
   for (int i=1; i<=n; i++)
   {
      for (int j=1; j<=n; j++)
      {
         cout<< “hello”;
         break;
      }
   }
}
```

End of Question Paper