

DATA STRUCTURES AND ALGORITHMS

Exam: Final Term
Type of Paper: Regular
Semester: Fall

Instructor: Dr. Seemab Latif
Total Marks: 50
Time Allowed: 2.5 hours

Question No.1:

(6+3)

You are given two linked lists S and Q, which contain integers in ascending order.

- Write down the function Print(S, Q) which will print the elements in S that are in position specified by Q, e.g. if Q = {1, 3, 8} the first, third and eighth element in S are printed.
- Check for all possible conditions, e.g. length of linked list S is less than the position specified by Q.

```
void print(List q1, List q2)
{
    List temp1=q1->front; List temp2=q2->front;
    do
    {
        if(temp1==NULL)
            cout<<"there is no data in the first linklist";
        if(temp1!=NULL)
        {
            if(q2->counter < temp1->data)
                //check to see that the data of first linklist is smaller than or equal to the size of 2nd linklist
                cout<<temp1->data<<" location is larger than 2nd list size "<<endl;
            else{
                int i=1;
                do
                {
                    if(temp2->next!=NULL)
                        //CHECK THAT IF IT IS'NT TEMP'S NEXT IS EQUEL TO NULL
                        {
                            if(temp1->data!=1)
                                temp2=temp2->next;
                        }
                    i++;
                }while(i < temp1->data); //end of do-while loop
                cout<<"location from 1st LINKLIST: " <<temp1->data<<endl;
                cout<<"data from 2nd LINKLIST: " <<temp2->data<<endl;
            } //end of else
        } //end of if
        temp1=temp1->next;
        temp2=q2->front;
    }while(temp1!=NULL); //END OF WHILE
} //end of function
```

Question No.2:

(5+4+4)

- Write a recursive function 'Sum' for a Binary Search Tree that sums up the contents of the nodes in the tree.

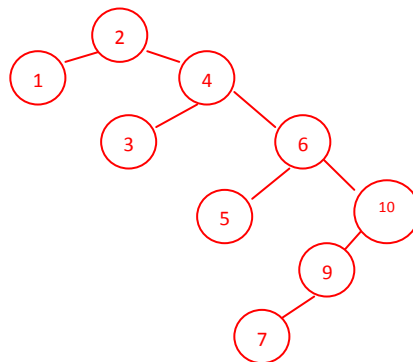
```
int Tree:: sum(TreeNode* r, int s)
{
    if(r == NULL)
        return (s);
    s = sum(r->left,s);
    s += r->data;
    s = sum(r->right,s);
    return (s);
}
```

b. Write a recursive function 'Height' to calculate the height of Binary Search Tree.

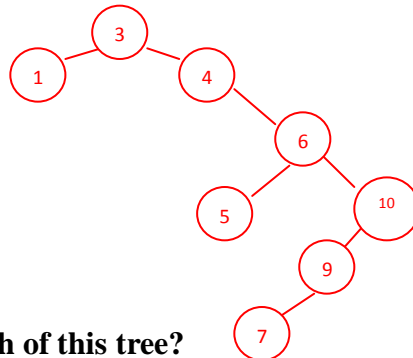
```
int Tree::height(TreeNode *r)
{
    if(r==NULL)
        return -1;
    int heightleft=height(r->left);
    int heightright=height(r->right);

    if (heightleft>heightright)
        return 1+heightleft;
    else
        return 1+heightright;
}
```

c. Show the result of inserting 2, 1, 4, 6, 10, 3, 5, 9, 7 into an initially empty binary search tree.



i. Show the result of deleting the root node.



- ii. What is the depth of this tree?
5
- iii. What is the height of this tree?
5
- iv. Write the pre-order traversal of the tree
2 1 4 3 6 5 10 9 7
- v. Write the post-order traversal of the tree
1 3 5 7 9 10 6 4 2
- vi. Write the in-order traversal of the tree
1 2 3 4 5 6 7 9 10
- vii. Write the breath-first traversal of the tree
2 1 4 3 6 5 10 9 7

Question No. 3:

(5)

Write a POPSUM function for dynamic implementation of stack which sums up all the elements in the stack while popping.

```
Void POPSUM()
{
    int sum = 0;
    While(top!= null)
    {
        sum += top->data;
        top = top->next
    }
    cout<<"POPSUM = "<<sum;
}
```

Question No.4:

(5+5)

- a. Write down a recursive function for binomial coefficient at a given column and row. The mathematical description of the formula is:

$$Binomial(c, r) = \begin{cases} c = 0 & \text{return 1} \\ r = 0 & \text{return 1} \\ Binomial(r-1, c-1) + binomial(r-1, c) & \text{otherwise} \end{cases}$$

```
long coef(int r, int c)
{
    if( ( c == 0 ) || ( r == c ) )
        return (1);

    return ( coef( (r-1), (c-1)) + coef( (r-1), c ) );
}
```

- b. Write down a recursive function for GCD (greatest common divisor) of two numbers. Greatest common divisor (gcd), of two non-zero integers, is the largest positive integer that divides the numbers without a remainder. For example, *the GCD of 8 and 12 is 4.*

```
int gcd(int x, int y)
{
    if(y%x == 0)
        return (x);

    return( gcd( (y % x), x ) );
}
```

Question No. 5:

(5)

- a. Write a function for Singly linked list to search an element in the list. The function should return true if data is found and false if data not found. The function should accept a data item which is to be searched. `bool Search (int searchdata)`

```
bool search (int searchdata)
{
    List temp = LinkedList->front;
    while(temp!=NULL)
    {
        if(temp->data== searchdata)
        {
            return (true);
        }
        temp=temp->next;
    }
    return(false);
}
```

Question No. 6:

(4+4)

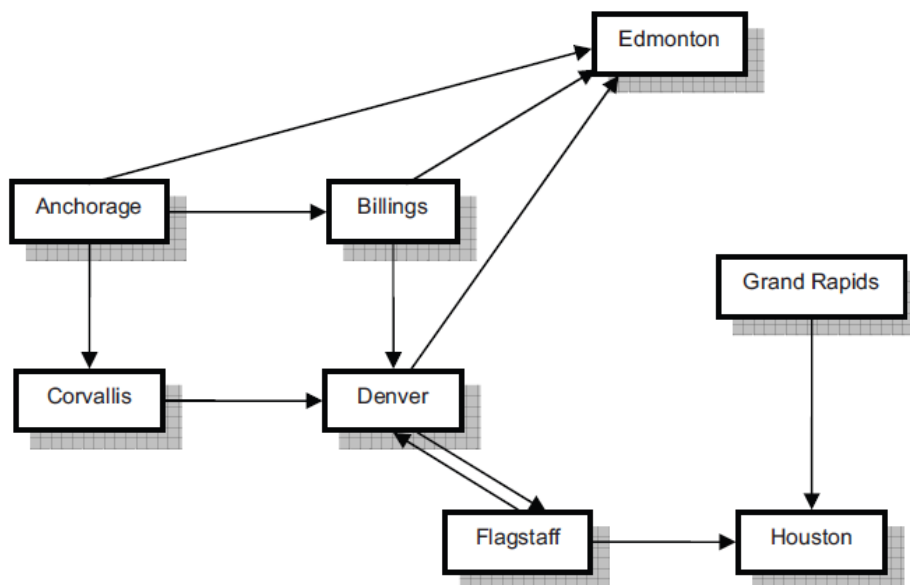
Figure 1 shows a graph in which the vertices are the names of cities in North America. The edges represent flights between these cities.

1. Traverse the graph in Breath-first manner starting from Anchorage and give the order of traversal.

Order of traversal: Anchorage, Billings, Corvallis, Edmonton, Denver, Flagstaff, Houston

2. Find the order of a depth-first traversal of the graph in the figure 1 starting at Anchorage.

Order of traversal: Anchorage, Billings, Denver, Flagstaff, Houston, Corvallis, Edmonton



Extra Credit:

(1)

FIFO stands for:

- 1. Frog In Frog Out**
- 2. First Input First Output**
- 3. First In First Out**
- 4. Fairy In Fairy Out**