Midterm Exam #2 –
Recursion, Binary Trees, Binary Search Trees and Hash Tables

60min, 100 points, open book, open notes.

1. (15 points) – Recursion
   Simple recursive algorithms – read and explain what it does (output) Given a recursive algorithm, find the error(s)

2. (20 points) – Binary Trees
   Terminology
   Breadth-First Traversal
   Depth-First Traversals (all 6)

3. (20 points) – Binary Search Trees
   Definition
   Basic BST operations, such as insert, delete, search
   Build a BST

4. (15 points) – Hash Tables
   Definition
   Hash Methods
   Hash Functions
   Collision Resolution Methods
   Basic hash table operations (insert, delete, search)

5. (30 points)
Write a C++ function or pseudocode to solve a problem with BSTs or Hash Tables.

1. (15 Points) RECURSION

A.

```
Algorithm test( n )

   if (n > 0)
       test(n / 2)
       print (n % 2)
   end if
end test
```

What is printed by the call **`test(17)`**?

B.

```
Algorithm test( x, y, sum )
```
Input Parameter: x > 0, y > 0, integers
Output Parameter: sum

```
   if ( x != 0 )
       if ( x is odd )
           sum += y
       end if
       test( x / 2, 2 * y, sum)
   end if
end test
```

What value does sum have after the call **`test(41, 20, sum)`**?
// assume that before the call sum = 0

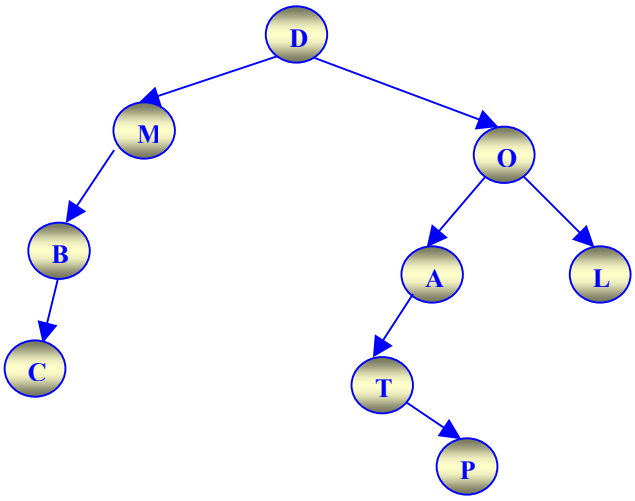
C. Describe the problem with the following recursive algorithm.

```
Algorithm test( n )

    if (n == 1)
        return n;
    return 2*test(n-1) + 5*test(n-2)
end test
```

2. (20 Points) BINARY TREES

A. Given the following tree:



1. What nodes are leaves?

2. How many levels does the tree have?

3. What node is the root of the tree?

4. What is the height of the tree?

5. How many nodes would you have to add to get a full tree of the same height?

B. Given the following tree:

1. What is the in-order traversal?

2. What is the post-order traversal?

3. What is the pre-order traversal?

4. What is the breadth-first traversal?

C. Give the postfix form of the following expression using an expression tree:

```
(A + B) * (C - D) / (E % 2)
```

D. Given the in-order traversal of a binary tree and the pre-order traversal of the same tree, draw the tree:

In: **P T R M U Q V**

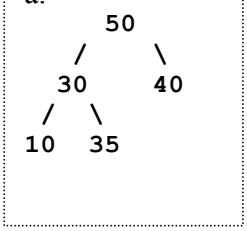
Pre: **M P R T Q U V**

3. (20 Points) BINARY SEARCH TREES

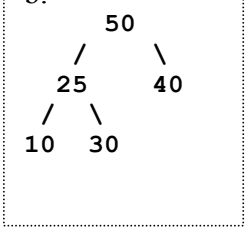
A. What binary search tree is formed when you insert the following values in the given order: **Mack, Zach, Andy, Paul, John, Bob**

B. Which of the following binary trees are not search trees? Explain why.

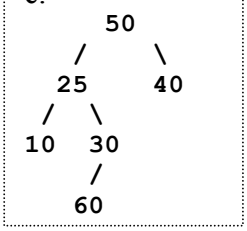
a.



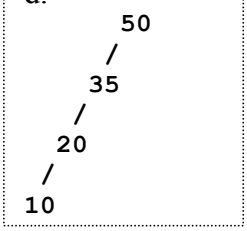
b.



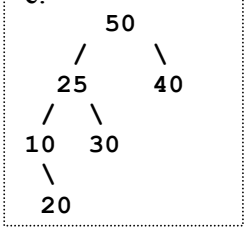
c.



d.



e.



C. Create a BST using the following data (in this order):

40 30 50 10 35 45 60 32 39 55 38

Show how the delete algorithm works to remove the root of the tree (give two solutions).

D. What is the purpose of the following BST algorithm?

```
algorithm guess( root)

    if (root->left is NULL)
        return root
    end if
    return guess(root->left)
end guess
```

E. Using the BST in exercise 3.C. show the output provided by the following function.

```
algorithm test( root)

    if (root is not  NULL)
        print( root )
        test( root->right)
        test( root->left)
    else
        print("*")
    end if
end test
```

4. (15 Points) HASH TABLES

A. Using the "modulo division" method and linked list collision resolution, store the keys shown below in an array of 5 elements. Each element has two fields: **`data`** and **`link`**. What is the load factor?

48, 79, 49, 28, 39, 58

B. Using the "modulo division" method and linear probe, store the keys shown below in an array of 10 elements. What is the load factor? What is the longest collision path?

48, 79, 49, 28, 39, 58

C. A simple hash function is given below? Is it good? Defend your answer.

```
Algorithm hash( string key, int size )

    sum = 0, i = 0;
    loop(not end of key)
        sum += key[i];
        i++;
    end loop

    return sum;
 end hash
```

5. (30 Points) BINARY SEARCH TREES

Write pseudo-code or a C++ function named **`getPath`** that fills a queue with the values of nodes in the path from the largest to the root of a binary search tree, in this order. The function returns the queue. List any assumptions used in the design.