

---

# CSCA48 Final Exam Part 1

- Questions on this test will use the following struct and functions, please copy them down carefully:

```
typedef struct BinaryTreeNode{                                I
    int data;
    struct BinaryTreeNode *left;
    struct BinaryTreeNode *right;
}Node;

int get_range(Node *r)
//returns the range of values in the tree rooted at r
//range is the maximum value in a tree - minimum value in a tree
```

---

```
Node* mystery(Node *root)
{
    Node *r;
    if (root->left!=NULL){
        root->left=mystery(root->left);
    }
    if (root->right!=NULL) {
        root->right=mystery(root->right);
    }
    printf("%d, %d\n", root->x, BSTRange(root));

    r=root->left;
    root->left=root->right;
    root->right=r;
    root->x = -1 * root->x;
    printf("%d, %d\n", root->x, BSTRange(root));

    return root;
}
```

## Question 1 [5 marks]

Write the function `get_range(Node *r)` where `r` is:

- a) The root of a binary search tree
- b) The root of a binary tree

## Question 2 [14 marks]

- a) Draw the BST resulting from the following operations:
1. Insert(1)
  2. Insert(3)
  3. Insert(5)
  4. Insert(2)
  5. Insert(6)
  6. Insert(4)
- b) Write the output of running mystery on the root of your tree from part a
- b) Draw the resulting tree



### Question 3 [3 marks]

Provide a better name for mystery and a brief function description (2-3 sentences max)

## Question 4 [8 marks]

Give the worst case big-O complexity of mystery being run on the following types of trees where  $n$  is the number of nodes in the tree:

- a. A binary tree
- b. A binary search tree
- c. A balanced binary tree
- d. A balanced binary search tree

You only need to give big-O complexities for each, but partial marks may be given for incorrect answers with justifications

---

## Question 1 [15 marks]

A FirstList is a linked list of nodes where each node links to the next node in the list, as well as the first node in the list that shares its value. The struct for the node is given below.

Write a function insert which takes 2 parameters: the head of a FirstList h, and a char c and deletes the first occurrence of c in h, if c does not occur in h, it should leave h unchanged

```
typedef struct FirstListNode{
    char data;
    //the next node in the FirstList
    //NULL if no such node exists
    struct FirstListNode *next;
    //the first node in the FirstList
    //with the same data value as this node
    //NULL if no such node exists
    struct FirstListNode *prev;
}Node;
```

## Question 2 [12 marks]

Write the following functions. (note: t is the root of a binary tree, not a BST). You may add helper functions and additional parameters if you wish. You may define your own return type. Your code must be recursive.

- a) `set_sibs(BTNode *t)` – set the data of all nodes which have siblings (i.e., nodes which share a parent with another node) to 'S', and the data of all nodes without siblings to 'O'
- b) `num_ancestors(BTNode *t, char c)` – return the total number of ancestors (children, children of children, etc) of all nodes in t that have a data value of c
- c) `inorder(BTNode *t, char s[MAX_NODES])` – fill s with the letters in t as they would be encountered in an inorder traversal. You may assume t has no more than MAX\_NODES total nodes.

```
typedef struct BinaryTreeNode{
    char data;
    struct BinaryTreeNode* left;
    struct BinaryTreeNode* right;
}BTNode;
```



---

### Question 3 [3 marks]

List 3 items from our “wishlist for software development”, for each item provide a 1 sentence example of software that would not satisfy this wish

