**CS010C Quiz 2: Chapters 3, 4 & 5 Practice Quiz**

*Note: This quiz is much longer than the actual quiz to provide you with various problems for you to practice and understand.*

**Runtime Analysis**

*Please denote examples of the specified runtime in Big O notation and specify examples of various algorithms/operations that fall under that runtime.*

1. Constant Time

*Sample Answer:*

*O(1) -> Linked List Append/Prepend, Stack Push/Pop/Top, Queue Push/Pop/Front*

1. Logarithmic
2. Linear
3. Quadratic
4. Linear Logarithmic

*For the following question, provide a sentence describing each scenario*

1. If a tree has 6 nodes, what is the tallest, shortest, and average tree height?

Tallest:

Shortest:

Average:

1. What is a stack?
2. What is a queue?
3. What is a heap?
4. There are some situations in which a queue has O(n) operations, what is that situation and how can you solve it?
5. What is the difference between Quicksort and Mergesort? What advantages/disadvantages do Quicksort and Mergesort have?

**Coding Questions**

*Provided the given class, please code the following functions:*

* Height
* Search
* Largest
* Smallest
* Sum

struct Node {

int val = 0;

Node\* left = nullptr;

Node\* right = nullptr;

};

class BST {  
 private:

Node\* root = nullptr;

int search(const Node\*, const int) const;

int largest(const Node\*) const;

int smallest(const Node\*) const;

int sum(const Node\*) const;

public:

int height(Node\*) const;

bool search(const int key) const {  
 return search(Node\*, key);

}

int largest() const {

return largest(root);

}

int smallest() const {

return smallest(root);

}

int sum() const {

return sum(root);

}

};

1. Height of Given Node

int height(Node\*) const {

}

1. Search

bool search(Node\* node, int key) const {

}

1. Largest

int largest(Node\* node) const {

}

1. Smallest

int smallest(Node\* node) const {

}

1. Sum

int sum(Node\* node) const {

}

**Visual Diagrams**

*The following questions 19 and 20 require you to visually draw the BST after inserting the specified element(s). DO NOT WRITE ANY CODE. ONLY DRAW/BOX THE RESULTING TREE.*

1. Insert 3 4 8 6 2 9 10 7 5 1 into a BST

1. Insert 2 8 4 5 7 6 1 9 3 10 into a BST

1. Insert 2 3 9 8 1 7 4 6 10 5 into a Heap (Visualize as a Tree)

1. Insert 5 9 7 8 3 4 2 10 1 6 into a Heap (Visualize as an Array)