Theory Mid Term Exam

Note: For the questions where you need to draw a figure, just draw it on a paper and then scan and attach the image in the corresponding question.

Marks

1. Write down the time complexity of Bubble sort, Insertion sort and Merge sort. **10**

2. Write two differences between array and linked-list. Why do we need a head/root node in a linked list? **10**

3. What is the basic idea behind the binary search algorithm, and how does it differ from linear search? What is the requirement for an array to be suitable for binary search? **10**

4. What is the time complexity of inserting an element at the beginning of a singly linked list? What is the time complexity of inserting an element at any index of a singly linked list? What is the time complexity of deleting an element at the beginning of a singly linked list? What is the time complexity of deleting an element at any index of a singly linked list? **10**

5. Suppose we have an array of 5 integer numbers and a singly linked list of 5 integer numbers. Here the singly linked list of 5 integer numbers takes twice as much memory compared to array. Why is that? Give a proper explanation. **10**

6. Derive the worst case and average case time complexity of Quick Sort with proper figures. **10**

7. Draw the recursion call tree with return values for Merge Sort in the array [6, 5, 1, 3, 8, 7, 10, 9] **10**

8. Write down the time complexity with proper explanation of the following code segment. **10**

```
for (int i = 1; i * i <= n; i++) {
    if (n % i == 0) {
        cout<< i <<"\n";
        cout<< (n/i) <<"\n";
    }
}
```

9. Suppose you are working in a project where you need to do many random memory accesses and binary search. Array vs Linked list which is more suitable in this case? Why? **10**
10. Alice is using a singly linked list to implement undo-redo functionality in a text editor. Bob advised Alice to use a doubly linked list in this scenario. Which approach seems more suitable to you? Give a proper explanation. **10**