**CS263: Assignment #1**

**Task 1:**

**Algorithm for Linear Search**

1. Suppose an array contains elements in random order and x is the value to be found.
2. Using a loop, start iterating from index 0 to index size -1, to iterate over all the elements.
3. Using a n if control structure, compare whether current element during iteration is equal to x or not.
4. If there is some comparison which is true, then print element found and if iteration ends, print element not present in array or file or some data structure.

**Algorithm for Binary Search**

1. To implement binary search, requirement is for a sorted array and a value suppose x which we need to find.
2. Take this x and compare it with the middle element of array using if statement.
3. If comparison is true then return mid element and say Element present.
4. Otherwise if x is greater than middle element, start finding in the right part.
5. Else find in the left part.

**Algorithm for Insertion Sort**

1. Need an array and a value x to be found as prerequisites.
2. Start iterating from index 0 to index n - 1 in array.
3. Compare the current element to it’s predecessor.
4. If x < predecessor, compare it with the elements before.
5. Shift greater elements one position up.

**Algorithm for Selection Sort**

1. In the array, set minimum to index 0 of array.
2. Search smallest element in the array.
3. Swap index 0 element with the smallest element.
4. Now point minimum to next element.
5. Repeat the same thing again, list will become sorted.

**Algorithm for Bubble Sort**

1. Start iterating from index 0 to index size – 1
2. Compare first element with the second element. Place them as per ascending or descending order.
3. Do it for n-1 passes and array will be in sorted order.

**Time Complexity**

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Best Case | Average Case | Worst Case |
| Linear Search | O(1) | O(n) | O(n) |
| Binary Search | O(1) | O(log n) | O(log n) |
| Insertion Sort | O(n) | O(n2) | O(n2) |
| Selection Sort |  |  |  |
| Bubble Sort |  |  |  |