Unit code: ICTPRG418

Teacher Ahn Nguyen

Task: Algorithms task/question

Student Name: Muhammed Jabari

ID: 101668629

```
Console.WriteLine("Insertion Sorting");
Stopwatch stopwatch2 = new Stopwatch();
stopwatch2.Start();
stopwatch2.Stop();
TimeSpan timeSpan2 = stopwatch2.Elapsed;
Console.WriteLine($"Insertion Sort took: {timeSpan2.TotalMilliseconds} Milliseconds");
Console.WriteLine();
Console.ReadLine();

Console.WriteLine("Linear Searching");
Stopwatch stopwatch3 = new Stopwatch();
stopwatch3.Start();
LinearSearch(csv);
stopwatch3.Stop();
TimeSpan timeSpan3 = stopwatch3.Elapsed;
Console.WriteLine($"Linear Search took: {timeSpan3.TotalMilliseconds} Milliseconds");
Console.ReadLine();



Console.WriteLine("Binary Searching");
Stopwatch stopwatch4 = new Stopwatch();
stopwatch4.Start();
BinarySearch(csv);
TimeSpan timeSpan4 = stopwatch4.Elapsed;
Console.WriteLine($"Binary Search took: {timeSpan4.TotalMilliseconds} Milliseconds");
Console.ReadLine();
```

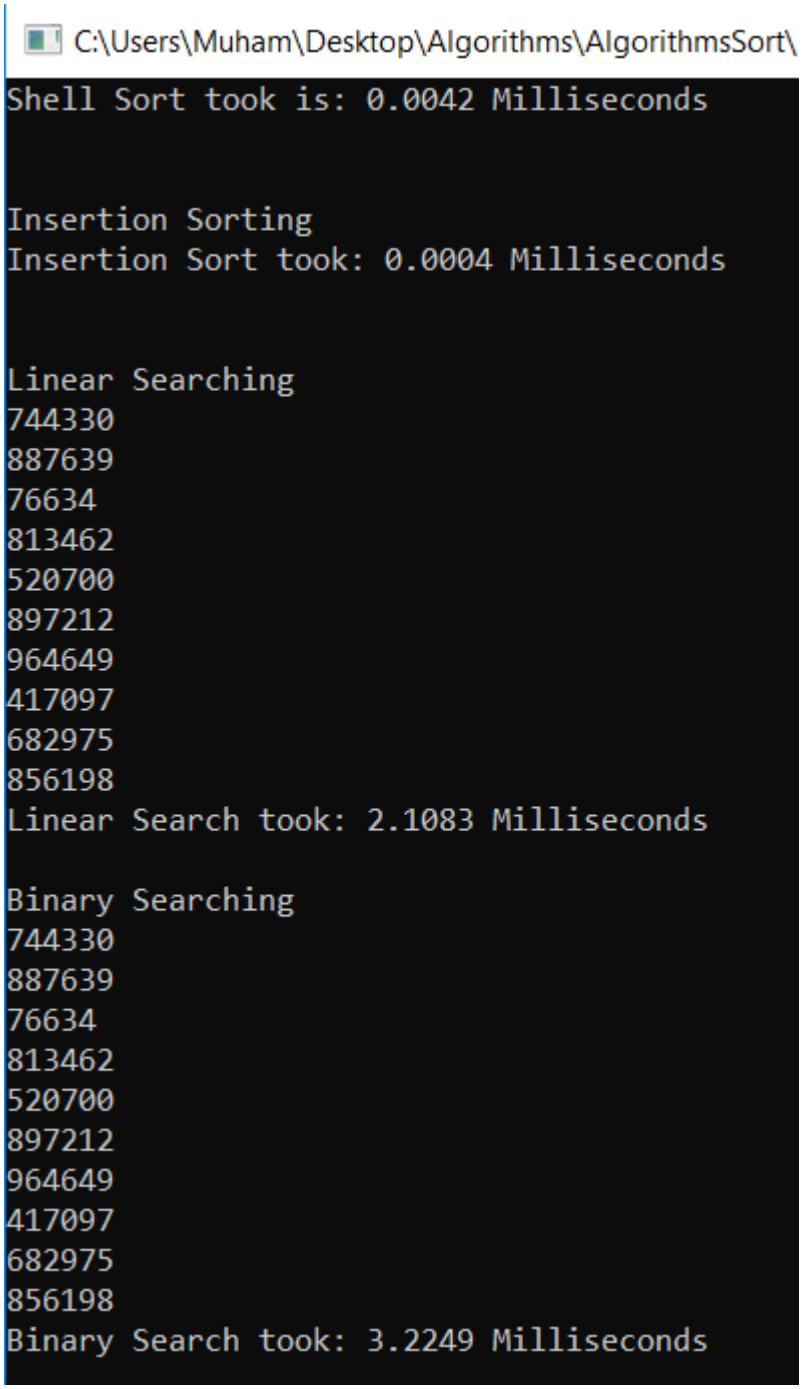
```
1 reference
public static int LinearSearch(int[] arr)
{
    int Max = arr.Length + 1;

    for (int i = 1; i < Max; i++)
    {
        if (i % 1500 == 0)
        {
            Console.WriteLine(arr[i - 1]);
        }

    }
    return -1;
}
```

```
public static int[] InsertionSort(int[] arr)
{
    //int[] inputarray = ReadFile();
    //inputarray.DoInsertionSort();

    //foreach (int inputarrays in inputarray)
    //    Console.WriteLine(inputarrays + " ");
    for (int i = 0; i < arr.Length - 1; i++)
    {
        for (int j = i + 1; j > 0; j--)
        {
            if (arr[j - 1] > arr[j])
            {
                int temp = arr[j - 1];
                arr[j - 1] = arr[j];
                arr[j] = temp;
                Console.WriteLine(arr[j]);
            }
        }
    }
    return arr;
}
```



1. A variation of Insertion Sort is Shell Sort. How is it better than Insertion?
   Insertion sort takes all numbers above the newly inserted number and needs it to be moved up one position. Which means that you'd have to relocate all the number before inserting/getting a new number, while on the other-hand shell sort isn't limited by this factor as is can relocate any number into any location by swapping places with it.

2. Could Merge Sort be run as a multi-threaded application?
   YES
3. Would there be likely to be a performance gain in doing so? Why/Why not?