CSCI 1112 Algorithms and Data Structures

Lab 5 – Loops vs. Recursion

Part 1: Searching (10 points)

1. Write a Java method called *sequentialSearch* that takes an array of integers and an *int* as arguments. The method should perform a sequential search by looping through the elements of the array, then return the index of the element if found, -1 otherwise.

2. Create a Java method called *binarySearch* that takes an array of integers and an *int* as arguments. Assuming the array is **sorted**, the method should perform binary search, and return the index of the element if found, -1 otherwise.

3. Add an *int* field in both search methods to count the number of element comparisons during the search—whenever you compare an element in the array with the requested value, increment the count variable.

4. Test the above methods using the following randomly generated array. Sort the array using insertion sort. Search for the numbers 1, 25, 50, 75, and 100 using both *sequentialSearch* and *binarySearch*. Report the number of element comparisons in each case.

```
Random generator = new Random(25);
int[] testData100 = new int[100];
//generate random data
for ( int i=0; I < testData100.length; i++){
    int randomNumber = (int) (generator.nextDouble() * 100) + 1;
    testData100[i] = randomNumber;
}
```

Part 2: Find the maximum (5 points)

5. Implement a recursive method for finding the maximum value in an array of integers

Part 3: Insertion sort (5 points)

6. Implement a recursive version of insertion sort.