CIS 360 Lab #4: Implement Algorithms / Divide and Conquer Design

1. **Implement** Algorithm for Chapter 1, Exercise 6, page 44 (Q2 in Homework 1),
   - Write a program that finds both the smallest and largest numbers in a list of n numbers. Your method does at most 1.5n comparisons of array items.
   - Show the running results for 100 randomly generate numbers.

   Algorithm submitted separately. Java file -> CIS360Lab4_Task1.java

   Analyze your algorithm and give the results using order notation. T(n) = O(3n/2)

2. Chapter 2, Exercise 6, page 89. **Design and Implement** an algorithm that searches a sorted list of *n* items by dividing it into three sublists of almost *n*/3 items. This algorithm finds the sublist that might contain the given item and divides it into three smaller sublists of almost equal size. The algorithm repeats this process until it finds the item or concludes that the item is not in the list.

```
public static int trinarySearch(int[] tempArr, int low, int high, int target) {
  //declare two middle term variables
  int mid1 = 0;
  int mid2 = 0;

  if(low <= high) {
   mid1 = low +  (high - low)/3;
   mid2 = high - (high - low)/3;
  }
  else {
   return -1;
  }

  //base cases
  if(target == tempArr[mid1]) {
   return mid1;
  }
  else if(target == tempArr[mid2]) {
   return mid2;
  }

  //cases for checking the mid-points recursively
  if(target < tempArr[mid1]) {
   high = mid1 - 1;
   return trinarySearch(tempArr,low,mid1 - 1,target);
  }
  else if(target > tempArr[mid2]) {
   low = mid2 + 1;
   return trinarySearch(tempArr,mid2 + 1,high,target);
  }
  else if(target > tempArr[mid1] && target < tempArr[mid2]) {
   low = mid1 + 1;
   high = mid2 - 1;
   return trinarySearch(tempArr,mid1 + 1,mid2 - 1,target);
  }

  return -1;
 }
```

Analyze your algorithm and give the results using order notation. T(n) = O(log3 n).