[**Closed book and notes.**] Show all of your work clearly in the space provided. Be sure to **read each problem carefully**. Note that the exam is double sided.

1. (10 points) Give the worst-case time complexity (big-oh notation) for the following method. Be sure to explain your reasoning.

```
public static double standardDeviation(double[] data) {
  if(data==null || data.length==0) {
    throw new IllegalArgumentException("Array cannot be null or empty");
  }
  double sum = 0;
  for(double value : data) {
    sum += value;
  }
  double average = sum/data.length;
  sum = 0;
  for(double value : data) {
    sum += (value-average)*(value-average);
  }
  return Math.sqrt(sum/data.length);
}
```

2. (10 points) What makes a binary search tree different from a binary tree?

3. (15 points) Either implement in Java or clearly explain in English the steps required for a recursive version of the contains method for a binary search tree. Assume that an empty tree has a root==null and that the recursive version is called as follows:

```
public boolean contains(E target) {
  return contains(root, target);
}
```

4. (15 points) Consider the following class which keeps track of an `int` and a String. Rewrite the class using generics so that it can keep track of an `int` and an object of a declared type. For example, it should be possible to create an object as follows: `new` Pair<Double>();

```
public class Pair {
  private int key;
  private String value;

  public Pair(int key, String value) {
    this.key = key;
    this.value = value;
  }
}
```

5. Recall the doRecursive4WaySearch method from the word search lab assignment. Assume that the recursive component was implemented as:

```
doRecursive4WaySearch(row+1, col);
doRecursive4WaySearch(row,   col+1);
doRecursive4WaySearch(row-1, col);
doRecursive4WaySearch(row,   col-1);
```

Assume the following grid of letters as input:

A	B	C
E	F	G
H	I	J

For partial credit, be sure to explain your answer.

(a) (5 points) Suppose that the method starts at **A**, i.e., doRecursive4WaySearch(0, 0); What will currentWord contain the first time it is four characters long?

(b) (5 points) Suppose that the method starts at **F**, i.e., doRecursive4WaySearch(1, 1); What will currentWord contain when it has maximum length?

(c) (5 points) Now suppose that the recursive component is changed to:

```
doRecursive4WaySearch(row,   col-1);
doRecursive4WaySearch(row-1, col);
doRecursive4WaySearch(row,   col+1);
doRecursive4WaySearch(row+1, col);
```

Use same grid:

A	B	C
E	F	G
H	I	J

and that the method starts at **F**, i.e., doRecursive4WaySearch(1, 1); What will currentWord contain the first time it is four characters long?

(d) (5 points) Now suppose that the recursive component is changed to:

```
doRecursive4WaySearch(row-1, col-1);
doRecursive4WaySearch(row-1, col+1);
doRecursive4WaySearch(row+1, col-1);
doRecursive4WaySearch(row+1, col+1);
```

and that the method starts at **F**, i.e., doRecursive4WaySearch(1, 1); What will currentWord contain when it has maximum length?

6. (15 points) Recall that the Java LinkedList class implements the Queue interface. Draw your best guess of how the data stored in memory would look after the following operations had been performed:

```
Queue<Integer> waitingIntegers = new LinkedList<Integer>();
waitingIntegers.offer(1);
waitingIntegers.remove();
waitingIntegers.offer(2);
waitingIntegers.offer(3);
waitingIntegers.peek();
waitingIntegers.offer(4);
waitingIntegers.offer(5);
waitingIntegers.remove();
waitingIntegers.remove();
waitingIntegers.offer(6);
```

7. (15 points) Create a class that implements the following PureStack interface but does not add any additional `public` methods. You may use any classes from the Java Collections Framework that you would like.

```
public interface PureStack<E> {
  E peek();
  E pop();
  void push(E item);
  int size();
  boolean isEmpty();
}
```

Additional work area for any problem. Clearly identify which problem is associated with the work on this page.