COS226 Week 3 Activity

1. *Static Comparators.*

   - Given an array of N `Point2D` objects, describe a linearithmic algorithm to remove all duplicates. *Hint:* sort.

   - Write a Java code fragment that would do the sorting in your description above. Use one or more of the static comparators defined in `Point2D` (`X_ORDER`, `Y_ORDER`, `R_ORDER`)

   - Assuming that `Point2D[] pointArray` exists, what comparison method gets called when the code `Arrays.sort(pointArray)` is executed, and what ordering results?

2. *Dynamic Comparators.* Consider the following code.

```
public class Point2D
{
   public final Comparator<Point2D> POLAR_ORDER = new PolarOrder();
   private final double x, y;
   ...
   private static int ccw(Point2D a, Point2D b, Point2D c)
   { /* see lecture slides or booksite */ }

   private class PolarOrder implements Comparator<Point2D>
   {
     public int compare(Point2D q1, Point2D q2)
     {
        double dx1 = q1.x - x;
        double dy1 = q1.y - y;
        double dx2 = q2.x - x;
        double dy2 = q2.y - y;
        if (dy1 == 0 && dy2 == 0) { ... }
        else if (dy1 >= 0 && dy2 < 0) return -1;
        else if (dy2 >= 0 && dy1 < 0) return +1;
        else return -ccw(Point2D.this, q1, q2);
     }
   }
}
```

(a) What is the difference between a `Comparable` and a `Comparator`?

(b) What is the difference between a static and a dynamic `Comparator`?

(c) Given an array of points `pointArray`, and a particular point p, give an example of a call to Array.sort that would result in the pointArray being sorted in polar order relative to p.

(d) Why not just return `-ccw()`? Why all the `if-else` clauses?

(e) Why not return `-ccw(this, q1, q2)` instead of `-ccw(Point2D.this, q1, q2)`?

3. *Mergesort.*

   Show, in the style of the trace of Algorithm 2.4 on p. 273, the result of using mergesort to sort the keys:

   T   A   T   S   T   L   T   M   T   O   T