```
public class SortAndSearch <T extends Comparable<T>>{

   public SortAndSearch() {} // Constructora: no hace nada
        public void bubbleSort(T[] tabla)
        public void selectionSort(T[] tabla)
        public void insertionSort(T[] tabla)
        public void mergeSort(T[] laTabla)
        public void quickSort(T[] laTabla)
}
```

```
public class Stopwatch { // Sedgewick & Wayne

    private final long start;
   /**  Create a stopwatch object. */
    /*  For additional documentation, see
    *  Section 3.2 of Introduction to Programming in Java: An
    *  Interdisciplinary Approach
    *  by Robert Sedgewick and Kevin Wayne. */

    public Stopwatch() {        start = System.currentTimeMillis(); }

   // Return elapsed time (in seconds) since this object was created.
    public double elapsedTime() {
        long now = System.currentTimeMillis();
        return (now - start) / 1000.0;
    }
}
```

```
import java.util.Random;

public class ArrayCreator {

    static int MAX = 1000000;

    public static Integer[] createArray(int N, int range) {
       Random randomGenerator = new Random();

      Integer[] a = new Integer[N];
      for (int i = 0; i < N; i++) a[i] = randomGenerator.nextInt(MAX);
      return a;
    }

    public static Integer[] createArray(int N) {
        return createArray(N, MAX);
    }
}
```

```
public class DoublingTest { // Sedgewick & Wayne

 public static double timeTrial(int N) {
   SortAndSearch<Integer> sortingAlgorithm = new SortAndSearch<Integer>();
   Integer[] a = ArrayCreator.createArray(N);

   Stopwatch timer = new Stopwatch();
   sortingAlgorithm.selectionSort(a);
   return timer.elapsedTime();
 }

 public static void main(String[] args) {
       int increment = 10000;
        for (int N = 250; true; N += increment) {
            double time = timeTrial(N);
            System.out.println(N + " " + time);
        }
 }
}
```