

~\OneDrive - VietNam National University - HCM INTERNATIONAL UNIVERSITY\Desktop\DSA\DSA LAB NEW\Lab 1
OOP Reviews & Arrays\ITITSB22029_DoMinhDuy_Lab1\Problem 2\Problem 2.iv\OrderedApp.java

```
1 import java.util.Random;
2
3 // OrderedArray.java
4 // demonstrates ordered array class
5
6 class OrderedArray {
7     long[] a; // ref to array a
8     private int nElems; // number of data items
9     private int comparisons; // counter for comparisons
10
11     public OrderedArray(int max) // constructor
12     {
13         a = new long[max]; // create array
14         nElems = 0;
15         comparisons = 0;
16     }
17
18     public int size() {
19         return nElems;
20     }
21
22     public int getComparisons() {
23         return comparisons;
24     }
25
26     public void resetComparisons() {
27         comparisons = 0;
28     }
29
30     public int find(long searchKey) {
31         int lowerBound = 0;
32         int upperBound = nElems - 1;
33         int curIn;
34         comparisons = 0;
35         while (true) {
36             comparisons++;
37             curIn = (lowerBound + upperBound) / 2;
38             if (a[curIn] == searchKey)
39                 return curIn; // found it
40             else if (lowerBound > upperBound)
41                 return nElems; // can't find it
42             else // divide range
43             {
44                 if (a[curIn] < searchKey)
45                     lowerBound = curIn + 1; // it's in upper half
46                 else
47                     upperBound = curIn - 1; // it's in lower half
```

```
48         } // end else divide range
49     } // end while
50 } // end find()
51
52 public void insert(long value) // put element into array
53 {
54     int j;
55     for (j = 0; j < nElems; j++) // find where it goes
56         if (a[j] > value) // (linear search)
57             break;
58     for (int k = nElems; k > j; k--) // move bigger ones up
59         a[k] = a[k - 1];
60     a[j] = value; // insert it
61     nElems++; // increment size
62 } // end insert()
63 }
64
65 public class OrderedApp {
66     public static void main(String[] args) {
67         Random rand = new Random();
68         int maxSize = 1000;
69         int trials = 100;
70
71         for (int size = 100; size <= maxSize; size += 100) {
72             OrderedArray arr = new OrderedArray(size);
73
74             // Insert random items
75             for (int i = 0; i < size; i++) {
76                 arr.insert(rand.nextInt(10000));
77             }
78
79             // Compute average comparisons over 100 trials
80             long totalComparisons = 0;
81             for (int t = 0; t < trials; t++) {
82                 long searchKey = arr.a[rand.nextInt(size)];
83                 arr.find(searchKey);
84                 totalComparisons += arr.getComparisons();
85             }
86
87             double averageComparisons = (double) totalComparisons / trials;
88             System.out.println("Average comparisons for size " + size + ": " +
89                 averageComparisons);
89         }
90     }
91 }
92
```

