

LAB №04

Alyssandra Cordero, Bryan Adame, Michael Castaneda

3/283/2019

Listing 1: Lab 04 -LookForThisWords

```
1 import java.util.Scanner;
2 import java.io.*;
3 public class LookForThisWords1
4 {
5     public static void main(String[] args) throws IOException
6     {
7         /*Saving the content of the file into an array.*
8         //Variable that holds the name of the file.
9         String fileName = "dictionary[1724].txt";
10        //Open the file.
11        File file = new File(fileName);
12        Scanner inputFile = new Scanner(file);
13        String [] toSort = new String[641412];
14
15        //Store each line into an array.
16        String line = inputFile.nextLine();
17        int i = 0;
18        while(inputFile.hasNext()){
19            toSort[i] = line;
20            i++;
21            line = inputFile.nextLine();
22        }
23        toSort[i] = line;
24        //Close the file.
25        inputFile.close();
26
27        String t = "computer";
28        String t2 = "ComPuteR";
29        String t3 = "ComPuteR";
30        String t4 = "ComPuteR";
31        //for t.*
32        //sequential unsorted.
33        long start = System.nanoTime();//Start count.
34        // task 1.
35        sequential(toSort,t);
36        long end = System.nanoTime();//end count.
37        long total = end - start;//calculate total.
38        System.out.println("sequential search, unsorted array: ↵
        "+total);//print total.
```

```

39 //sorted array declaration*
40 String [] sorted = new String [toSort.length];
41 sorted = doQuicksort(toSort);
42 //after sorting
43 //binary search.
44 start = System.nanoTime();//Start count.
45 // task 2.
46 binary(sorted,t);
47 end = System.nanoTime();//end count.
48 total = end - start;//calculate total.
49 System.out.println("binary search, sorted array: "+total);//print total.
50 //sequential search.
51 start = System.nanoTime();//Start count.
52 // task 3.
53 sequential(sorted,t);
54 end = System.nanoTime();//end count.
55 total = end - start;//calculate total.
56 System.out.println("sequential search, sorted array: "+total);//print ↵
    total.
57 //another binary.
58 start = System.nanoTime();//Start count.
59 // task 4.
60 binary(sorted,t);
61 end = System.nanoTime();//end count.
62 total = end - start;//calculate total.
63 System.out.println("binary search, sorted array: "+total);//print total.
64
65
66 //for t2.*
67 //sequential unsorted.
68 start = System.nanoTime();//Start count.
69 // task 1.
70 sequential(toSort,t2);
71 end = System.nanoTime();//end count.
72 total = end - start;//calculate total.
73 System.out.println("sequential search, unsorted array: ↵
    "+total);//print total.
74 //after sorting:
75 //binary search.
76 start = System.nanoTime();//Start count.
77 // task 2.
78 binary(sorted,t2);
79 end = System.nanoTime();//end count.
80 total = end - start;//calculate total.
81 System.out.println("binary search, sorted array: "+total);//print total.
82 //sequential search.
83 start = System.nanoTime();//Start count.

```

```

84 // task 3.
85 sequential(sorted,t2);
86 end = System.nanoTime();//end count.
87 total = end - start;//calculate total.
88 System.out.println("sequential search, sorted array: "+total);//print ↵
    total.
89 //another binary.
90 start = System.nanoTime();//Start count.
91 // task 4.
92 binary(sorted,t2);
93 end = System.nanoTime();//end count.
94 total = end - start;//calculate total.
95 System.out.println("binary search, sorted array: "+total);//print total.
96
97 //for t3.*
98 //sequential unsorted.
99 start = System.nanoTime();//Start count.
100 // task 1.
101 sequential(toSort,t3);
102 end = System.nanoTime();//end count.
103 total = end - start;//calculate total.
104 System.out.println("sequential search, unsorted array: ↵
    "+total);//print total.
105 //after sorting:
106 //binary search.
107 start = System.nanoTime();//Start count.
108 // task 2.
109 binary(sorted,t3);
110 end = System.nanoTime();//end count.
111 total = end - start;//calculate total.
112 System.out.println("binary search, sorted array: "+total);//print total.
113 //sequential search.
114 start = System.nanoTime();//Start count.
115 // task 3.
116 sequential(sorted,t3);
117 end = System.nanoTime();//end count.
118 total = end - start;//calculate total.
119 System.out.println("sequential search, sorted array: "+total);//print ↵
    total.
120 //another binary.
121 start = System.nanoTime();//Start count.
122 // task 4.
123 binary(sorted,t3);
124 end = System.nanoTime();//end count.
125 total = end - start;//calculate total.
126 System.out.println("binary search, sorted array: "+total);//print total.
127

```

```

128     //for t4.*
129     //sequential unsorted.
130     start = System.nanoTime();//Start count.
131     // task 1.
132     sequential(toSort,t4);
133     end = System.nanoTime();//end count.
134     total = end - start;//calculate total.
135     System.out.println("sequential search, unsorted array: ↵
        "+total);//print total.
136 //after sorting:
137 //binary search.
138     start = System.nanoTime();//Start count.
139     // task 2.
140     binary(sorted,t4);
141     end = System.nanoTime();//end count.
142     total = end - start;//calculate total.
143     System.out.println("binary search, sorted array: "+total);//print total.
144     //sequential search.
145     start = System.nanoTime();//Start count.
146     // task 3.
147     sequential(sorted,t4);
148     end = System.nanoTime();//end count.
149     total = end - start;//calculate total.
150     System.out.println("sequential search, sorted array: "+total);//print ↵
        total.
151     //another binary.
152     start = System.nanoTime();//Start count.
153     // task 4.
154     binary(sorted,t4);
155     end = System.nanoTime();//end count.
156     total = end - start;//calculate total.
157     System.out.println("binary search, sorted array: "+total);//print total.
158
159 //    //toCheckSortingAlgorithms*
160 //    String t = "n&y";
161 //    String [] unsorted = ↵
        {"todo","nada","vale","hi","crazy","fish","girl","j","w","N&y"};
162 //    print(unsorted);
163 //    //after sorting
164 //    String [] sorted = new String [unsorted.length];
165 //    sorted = merge_sort(unsorted);
166 //    print(sorted);
167
168 //    System.out.print(binary(sorted, t));
169 }
170 public static boolean binary(String [] a, String t){//binary search.*
171     int l = 0;

```

```

172     int r = a.length-1; //-1 important!
173
174     while(l <= r){//limit
175         int m = l + (r-l) / (2);//initialize the middle.
176
177         if(a[m].equalsIgnoreCase(t)){// check if the element youre looking ↵
            for is at the middle.
178             return true;//if its at the middle, return true.
179         }
180         if(a[m].compareToIgnoreCase(t) > 0){//check if the element at the ↵
            middle is grater than t.
181             r = m-1;//if its grater then: r = m-1.
182         }
183         if(a[m].compareToIgnoreCase(t) < 0){//check if the middle is less ↵
            than t.
184             l = m+1;//if its less than t, then: l = m+1.
185         }
186     }
187     return false;
188 }
189
190 public static boolean sequential(String[] a, String t){
191     for(int i = 0; i < a.length; i++){
192         if(a[i].equalsIgnoreCase(t)){
193             return true;
194         }
195     }
196     return false;
197 }
198 public static String [] doQuicksort(String [] array){
199     return doQuicksort(array,0, array.length-1);
200 }
201 private static String[] doQuicksort(String[] array, int start, int end){
202
203     //set pivot to middle value
204     int mid = (start + end) / 2;
205     String pivot = array[mid];
206
207     int i = start;    //controls start
208     int j = end;      //controls end
209
210     //the next while loop chekcs if i is less than or equal to j
211     while(i <= j){
212         //the next while loop looks for a value larger than the pivot, on the
213         //pivot's left, sets i to that value's index
214         while((array[i].compareToIgnoreCase(pivot) )<0){
215             i++;

```

```

216     }
217     //the next while loop looks for a value smaller than the pivot, on the
218     //pivot's right, sets j to that value's index
219     while((array[j].compareToIgnoreCase(pivot) )> 0){
220         j--;
221     }
222     //if i is less than or equal to j, swap their values
223     //increment i, decrement j
224     if(i <= j){
225         swap(array, i,j);
226         i++;
227         j--;
228     }
229 }//return to while loop
230 if(start < j){                //if start is less than j
231     doQuicksort(array, start, j); //recursion on left side, sends array, ←
        start, j (as end)
232 }
233 if(end > i){                //if end is less than i
234     doQuicksort(array, i, end); //recursion on right side, sends array, ←
        i (as start), end
235 }
236 return array;
237 }//end of doQuicksort
238
239 private static void swap(String[] array, int x, int y){
240     String temp = array[x]; //set temp to the first value (x)
241     array[x] = array[y]; //set first position (x) to second value (y)
242     array[y] = temp; //set second position(y) to temp (x)
243 }
244 public static void print(String [] a){
245     for(int i =0; i<a.length;i++){
246         System.out.print(a[i]);
247     }
248     System.out.println();
249 }
250 }

```
