

```

// Adrian Pailler//

package Lesson7;

import java.util.*;

import javax.swing.*;

public class Searching_alg {

    // array that will be created and used by all sorts/searches //
    static ArrayList<String> arr = new ArrayList<String>() ;

    // main method which will perform user required search/sort //
    public static void main(String[] args) {

        input(); // see input method (42-53) //

        int ans = choice(); // decides which method to call. see choice method (29-40) //

        if(ans==0) {

            linear(); // see linear method (60-72) //

        }

        else if(ans==1) {

            binary(); // see binary method (88-109) //

        }

        else if(ans==2){

            bubble(); // see bubble method (74-86) //

        }

        else if(ans == 3){

            selection(); // see selection method (111-125) //

        }

        else {

            JOptionPane.showMessageDialog(null, "Bye!");

        }

    }

    // 9/01/21 ; user decision //
    public static int choice() {

```

```

        JFrame choice = new JFrame();

        String[] options = new String[4];

        options[0] = new String("linear search");

        options[1] = new String("binary search");

        options[2] = new String("bubble sort");

        options[3] = new String("selection sort");

        // creates four options//

        int ans =JOptionPane.showOptionDialog(choice.getContentPane(),"What would you like to do
with your array?","Sorting and Searching",0,JOptionPane.INFORMATION_MESSAGE,null,options,null);

        return ans;

        // returns the value of ans to be used in the main method//

    }

    // 9/01/21 ; user input array //

    public static ArrayList<String> input() {

        // array that will be created and used by all sorts/searches //

        String temp;

        // Temporary value to add new elements //

        int length = Integer.parseInt(JOptionPane.showInputDialog(null,"How long do you want
your array?"));

        // allows user to give predefined length //

        for(int count = 0;length>count; count++) {

            temp = JOptionPane.showInputDialog(null,"Input your array: ");

            arr.add(temp);

        }

        return arr;

    }

    // 9//01//21 ; requests the search element //

    public static String search() {

        String element = JOptionPane.showInputDialog(null, "Which element are you looking
for?");

```

```

        return element;
    }

    // 9/01/21 ; Linear search //
    public static void linear() {
        String element = search(); // see search method (55-58) //
        boolean flag = false;
        for(int i = 0; i<arr.size();i++) {
            if(arr.get(i).compareTo(element)==0) { // compares ascii values //
                JOptionPane.showMessageDialog(null,"The index of your searched value is:
"+arr.indexOf(element)); // if true, element found //
                flag = true; // breaks loop//
            }
        }
        if(flag == false) {
            JOptionPane.showMessageDialog(null, "This value is not an element of the
array");
        }
    }

    // 10/01/21 ; bubble sort //
    public static void bubble() {
        String temp; // temporary value needed to change positions //
        for(int i = 0;i<arr.size();i++) { // outer loop//
            for(int a = 0; a<arr.size()-i-1;a++) { // inner loop //
                if(arr.get(a).compareTo(arr.get(a+1))>0) { // compares both elements
lexicographically //
                    temp = arr.get(a);    //
                    arr.set(a,arr.get(a+1)); // swaps position
                    arr.set(a+1,temp);    //
                }
            }
        }
    }

```

```

    }

    JOptionPane.showMessageDialog(null, "The sorted array is: "+arr);

}

// 11/01/21 ; binary search //
public static void binary() {
    String element = search(); // see search method (55-58) //
    bubble(); // array needs to be sorted first //
    boolean flag = false;
    int min = 0;
    int max = arr.size()-1;
    while(max>=min) {
        int mid = (min+max)/2; // gets the midpoint
        if((arr.get(mid).compareTo(element))<0) { // compares ascii //
            min = mid+1;
        } else if((arr.get(mid).compareTo(element))>0) {
            max = mid-1;
        } else {
            JOptionPane.showMessageDialog(null, "Element found at index: "+mid);
            min = arr.size()+1; // breaks while condition//
            flag = true;
        }
    }

    if(flag==false) { //if value is not an element //
        JOptionPane.showMessageDialog(null, "This value is not an element of the
array");
    }

}

// 11/01/21 ; selection sort//
public static void selection() {

```

```
String temp;

int i,e; // will be counters to compare elements //

for(i=0;i<arr.size();i++) { // outer loop//

    for(e=i+1; e<arr.size();e++) { // nested loop //

        if(arr.get(i).compareTo(arr.get(e))>0) { // conditions to compare ascii //

            temp = arr.get(i);    //

            arr.set(i, arr.get(e)); // swap values if condition is met

            arr.set(e, temp);    //

        }

    }

}

JOptionPane.showMessageDialog(null, "The sorted array is: "+arr);

}

}
```

Linear search:

Array: {g,4,6,c,e}

Search term:

Input

Which element are you looking for?

d

OK Cancel

Message

The index of your searched value is: 3

OK

Input

Which element are you looking for?

g

OK Cancel

Message

This value is not an element of the array

OK

Binary search:

Array: {g,4,6,c,e}

Search term:

Input

Which element are you looking for?

g

OK Cancel

Input

Which element are you looking for?

7

OK Cancel

Sorted array:

Message

The sorted array is: [4, 6, c, e, g]

OK

Message

The sorted array is: [4, 6, c, e, g]

OK

Message

Element found at index: 4

OK

Message

This value is not an element of the array

OK

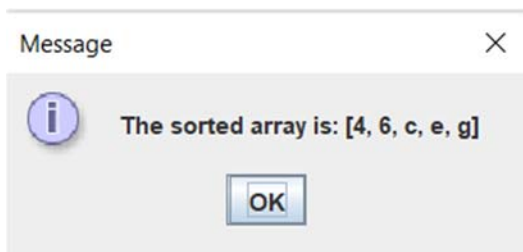
Bubble sort:

Array: {g,4,6,c,e}



Sequential sort:

Array: {g,4,6,c,e}



Array: {f,3,3,5,g,s,e,w,m,n}

