

# Search Algorithms

## (part 1 – Linear Search)

A linear search will check all cells in an array for a match, starting at index 0.

The code below looks for an integer match.  
LinearIntSearch() will report the 1<sup>st</sup> match,  
LinearIntSearchAll() will report all matches.

```
// -----
public class LinearIntSearch {

    public static void main(String[] args) {

        // Declare and initialize an array of int data types
        int[] intArray = new int[] {48, 5, 89, 80, 81, 23, 45, 16, 2};

        // Search for a target in the array of ints.
        // Stop after locating the first instance of the target.

        // Specify the target to search for.
        int target = 45;
        boolean targetFound = false;

        int arrayIndex; // declared outside of the for loop
        int i = 0;

        for(arrayIndex = 0; arrayIndex < intArray.length; arrayIndex++) {

            if ( intArray[arrayIndex] == target ) {
                // Target found.
                targetFound = true;
                break;
                // quit the loop when the first match is found.
            }
        }
        // search terminated here.

        if ( targetFound )
            System.out.println
                ("First target found in cell " + arrayIndex );
        else
            System.out.println
                ("Target not found." );

    } // end main
} // end class
```

```
// -----
public class LinearIntSearchAll {

    public static void main(String[] args) {

        // Declare and initialize an array of integers

        int[] intArray = new int[ ] {45, 5, 89, 45, 81, 23, 45, 45, 2};

        // Search for a target in the array of integers.
        // Save all indices where a match was found in a
        // second array called foundIndex[].
        // Java initializes an empty array with 0's.
        // Use this to detect the end of valid entries in the array.

        int[] foundIndex = new int[ intArray.length ] ;

        // Specify the target to search for.
        int target = 45;
        boolean targetFound = false;

        int arrayIndex; // declared outside of the for loop
        int i = 0;

        for (arrayIndex = 0; arrayIndex < intArray.length; arrayIndex++)
        {
            if ((intArray[arrayIndex] == target )) {
                // Target found. Store its index.
                targetFound = true;
                foundIndex[i++] = arrayIndex;
            }
        }

        // Here when finished searching the integer array.
        // Verify all matches have been recorded in the foundIndex array.

        if ( !targetFound )
            System.out.println( "Target not found." );

        else {
            for ( int j = 0; j < foundIndex.length; j++ ) {

                // Bug: If the first match was found at index 0,
                // storing a 0 in the first position of foundIndex
                // caused a premature exit.
                // Fix: We can assume a match at index 0 is a valid target
                // because the targetFound flag was set true.
                // It's therefore safe to print the foundIndex array
                // before checking for its 0 terminator.

                if ( j == 0 )
                    System.out.println
                    ("Target found in cell " + foundIndex[j] );
                else {
                    if ( foundIndex[j] != 0 )
                        System.out.println
                        ("Target found in cell " + foundIndex[j] );
                }
                // end else
            }
            // end for
        }
        // end else
    }
}
```

```

        } // end main
    } // end class

```

The code below looks for a String match.  
**LinearStringSearch()** will report the 1<sup>st</sup> match,  
**LinearStringSearchAll()** will report all matches.

```

// -----
public class LinearStringSearch {

    public static void main(String[] args) {

        // Declare and initialize an array of String data types
        String[] strArray = new String[ ] {

            "Atlanta",
            "Charlie",
            "Ryan",
            "Phoenix",
            "Peoria",
            "Ryan",
            "Peoria",
            "Peoria",
            "Albany",
            "Adam",
            "David"
        };

        // Search for a target in the array of Strings.
        // Stop after locating the first instance of the target.

        // Specify the target to search for.
        String target = "Peoria";
        boolean targetFound = false;

        int arrayIndex; // declared outside of the for loop
        int i = 0;

        for ( arrayIndex = 0; arrayIndex < strArray.length; arrayIndex++ ) {

            if ( strArray[arrayIndex].equals( target ) ) {
                // Target found.
                targetFound = true;
                break;
                // quit the loop when the first match is found.
            }
        }
        // search terminated here.

        if ( targetFound )
            System.out.println("First target found in cell " + arrayIndex );
        else
            System.out.println("Target not found." );
    }
}

```

```

        } // end main

    } // end class

    // -----

    public class LinearStringSearchAll {

        public static void main(String[] args) {

            // Declare and initialize an array of String data types

            String[] strArray = new String[ ] {

                "Peoria",
                "Charlie",
                "Ryan",
                "Phoenix",
                "Peoria",
                "Ryan",
                "Peoria",
                "Peoria",
                "Albany",
                "Adam",
                "David"
            };

            // Search for a target in the array of Strings.
            // Save all indices where a match was found in a
            // second array called foundIndex[].
            // Java initializes an empty array with 0's.
            // We can use this to detect the end of valid entries in the array.

            int[] foundIndex = new int[ strArray.length ] ;

            // Specify the target to search for.
            String target = "Peoria";
            boolean targetFound = false;

            int arrayIndex; // declared outside of the for loop
            int i = 0;

            for ( arrayIndex = 0; arrayIndex < strArray.length; arrayIndex++ ) {

                if ( strArray[arrayIndex].equals( target ) ) {
                    // Target found. Store its index.
                    targetFound = true;
                    foundIndex[i++] = arrayIndex;
                }
            }

            // Here when finished searching the String array.
            // Verify all String matches have been recorded in the foundIndex array.

            if ( ! targetFound )
                System.out.println( "Target not found." );

            else {
                for ( int j = 0; j < foundIndex.length; j++ ) {

                    // Bug: If the first match was found at index 0,
                    // storing a 0 in the first position of foundIndex
                    // caused a premature exit.

```

```

// Fix: We can assume a match at index 0 is a valid target
// because the targetFound flag was set true.
// It's therefore safe to print the foundIndex array
// before checking for its 0 terminator.

        if ( j == 0 )
            System.out.println
                ("Target found in cell " + foundIndex[j] );
        else {
            if ( foundIndex[j] != 0 )
                System.out.println
                    ("Target found in cell " + foundIndex[j] );
        }
        // end else
    }
    // end for
}
// end else

} // end main

} // end class

```