

# JAVA.UTIL.ARRAYS.BINARYSEARCH METHOD

[http://www.tutorialspoint.com/java/util/arrays\\_binarysearch\\_super\\_index.htm](http://www.tutorialspoint.com/java/util/arrays_binarysearch_super_index.htm)

Copyright © tutorialspoint.com

## Description

The **java.util.Arrays.binarySearch***T[]a, Tkey, intfromIndex, intoIndex, Comparator < ?superT > c* method searches a range of the specified array for the specified object using the binary search algorithm. The range must be sorted into ascending order according to the specified comparator before making this call. If it is not sorted, the results are undefined.

## Declaration

Following is the declaration for **java.util.Arrays.binarySearch***super, index* method

```
public static <T> int binarySearch(T[] a, T key, int fromIndex, int toIndex, Comparator<? super T> c)
```

## Parameters

- **a** -- This is the array to be searched.
- **fromIndex** -- The index of the first element *inclusive* to be searched.
- **toIndex**-- The index of the last element *exclusive* to be searched.
- **key** -- This is the value to be searched for.
- **c** -- This is the comparator by which the array is ordered. A null value indicates that the elements natural ordering should be used.

## Return Value

This method returns index of the search key, if it is contained in the array within the specified range; otherwise,  $-(\text{insertionpoint} - 1)$ . The insertion point is defined as the point at which the key would be inserted into the array: the index of the first element in the range greater than the key, or toIndex if all elements in the range are less than the specified key.

## Exception

- **ClassCastException** -- if the range contains elements that are not mutually comparable using the specified comparator, or the search key is not comparable to the elements in the range using this comparator.
- **IllegalArgumentException** -- if fromIndex > toIndex
- **ArrayIndexOutOfBoundsException** - if fromIndex < 0 or toIndex > a.length

## Example

The following example shows the usage of **java.util.Arrays.binarySearch***super, index* method.

```
package com.tutorialspoint;

import java.util.Arrays;
import java.util.Comparator;

public class ArrayDemo {

    public static void main(String[] args) {

        // initializing unsorted short array
        Short shortArr[] = new Short[]{5, 2, 15, 52, 10};
```

```

// use comparator as null, sorting as natural ordering
Comparator<Short> comp = null;

// sorting array
Arrays.sort(shortArr, comp);

// let us print all the elements available in list
System.out.println("The sorted short array is:");
for (short number : shortArr) {
    System.out.println("Number = " + number);
}

// entering the value to be searched
short searchVal = 15;

// search between index 1 and 4
int retVal = Arrays.binarySearch(shortArr, 1, 4, searchVal, comp);
System.out.println("The index of element 15 is : " + retVal);

// search between index 0 and 3, where searchVal doesn't exist
retVal = Arrays.binarySearch(shortArr, 0, 3, searchVal, comp);
System.out.println("The index of element 15 is : " + retVal);
}
}

```

Let us compile and run the above program, this will produce the following result:

```

The sorted short array is:
Number = 2
Number = 5
Number = 10
Number = 15
Number = 52
The index of element 15 is : 3
The index of element 15 is : -4

```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js