

WIA1002/ WIB1002

Data Structure

Searching



Searching

Searching is the process of looking for a specific element in a group of items (such as in an array)

Two common searching approaches: Linear and Binary Search



Linear Search

- compares the key element, key, **sequentially** with each element in the group (such as array list).

- ✎ continues to do so until the key matches an element in the list or the list is exhausted without a match being found.
- ✎ If found, returns the index of the element in the array that matches the key.
- ✎ If no match is found, the search returns -1.



Linear Search Animation

Key

List



<https://yongdanielliang.github.io/animation/web/LinearSearchNew.html>

Linear Search Animation

Key

3

List

6	4	1	9	7	3	2	8
---	---	---	---	---	---	---	---



<https://yongdanielliang.github.io/animation/web/LinearSearchNew.html>

Linear Search Animation

Key

List

3

6	4	1	9	7	3	2	8
---	---	---	---	---	---	---	---

3

6	4	1	9	7	3	2	8
---	---	---	---	---	---	---	---



<https://yongdanielliang.github.io/animation/web/LinearSearchNew.html>

Linear Search Animation

Key

List

3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8



<https://yongdanielliang.github.io/animation/web/LinearSearchNew.html>

Linear Search Animation

Key

List

3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8



<https://yongdanielliang.github.io/animation/web/LinearSearchNew.html>

Linear Search Animation

Key

List

3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8



<https://yongdanielliang.github.io/animation/web/LinearSearchNew.html>

Linear Search Animation

Key

List

3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8
3	6	4	1	9	7	3	2	8



<https://yongdanielliang.github.io/animation/web/LinearSearchNew.html>

From Idea to Solution

```
/** The method for finding a key in the list */  
public static int linearSearch(int[] list, int key) {  
    for (int i = 0; i < list.length; i++)  
        if (key == list[i])  
            return i;  
    return -1;  
}
```

Trace the method

```
int[] list = {1, 4, 4, 2, 5, -3, 6, 2};  
int i = linearSearch(list, 4); // returns 1  
int j = linearSearch(list, -4); // returns -1  
int k = linearSearch(list, -3); // returns 5
```



Binary Search

Pre-requisite: the elements in the group must already be ordered.

e.g., 2 4 7 10 11 45 50 59 60 66 69 70 79

- first compares the key with the element in the middle of the group.



Binary Search, cont.

Consider the following three cases:

- ✦ If the key is less than the middle element, you only need to search the key in the first half of the group.
- If the key is equal to the middle element, the search ends with a match.
- If the key is greater than the middle element, you only need to search the key in the second half of the group.



Binary Search

Key

List

<https://yongdanielliang.github.io/animation/web/BinarySearchNew.html>



Binary Search

Key

List

8	1	2	3	4	6	7	8	9
---	---	---	---	---	---	---	---	---

<https://yongdanielliang.github.io/animation/web/BinarySearchNew.html>



Binary Search

Key

List

8

1	2	3	4	6	7	8	9
---	---	---	---	---	---	---	---

8

1	2	3	4	6	7	8	9
---	---	---	---	---	---	---	---



<https://yongdanielliang.github.io/animation/web/BinarySearchNew.html>

Binary Search

Key

List

8	1	2	3	4	6	7	8	9
8	1	2	3	4	6	7	8	9
8	1	2	3	4	6	7	8	9

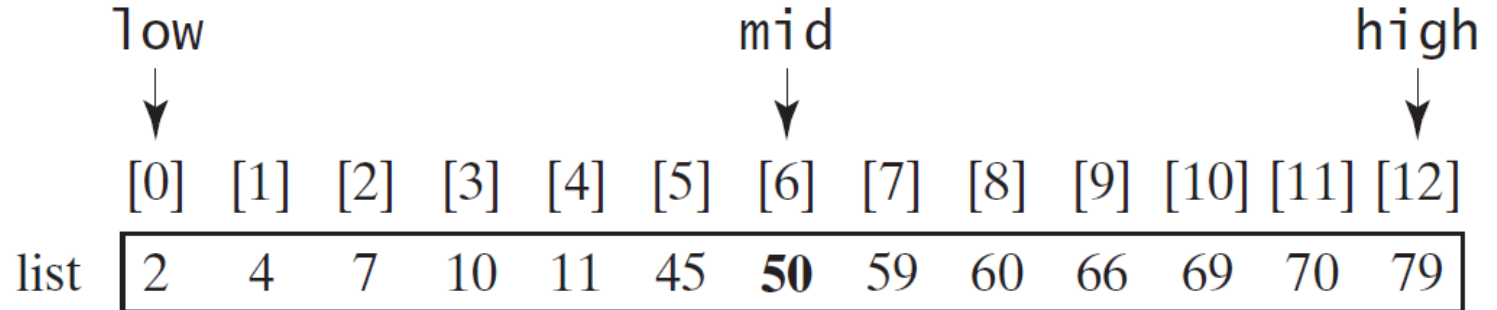
<https://yongdanielliang.github.io/animation/web/BinarySearchNew.html>



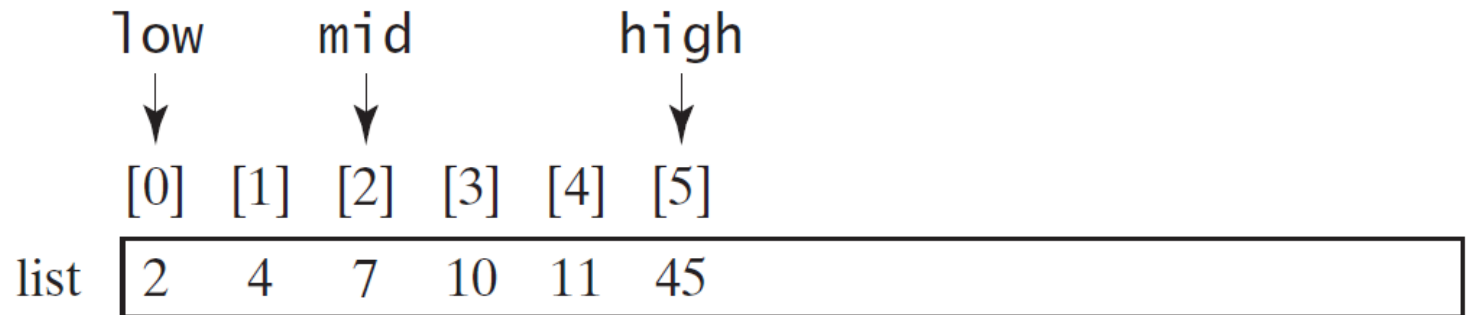
Binary Search, cont.

key is 11

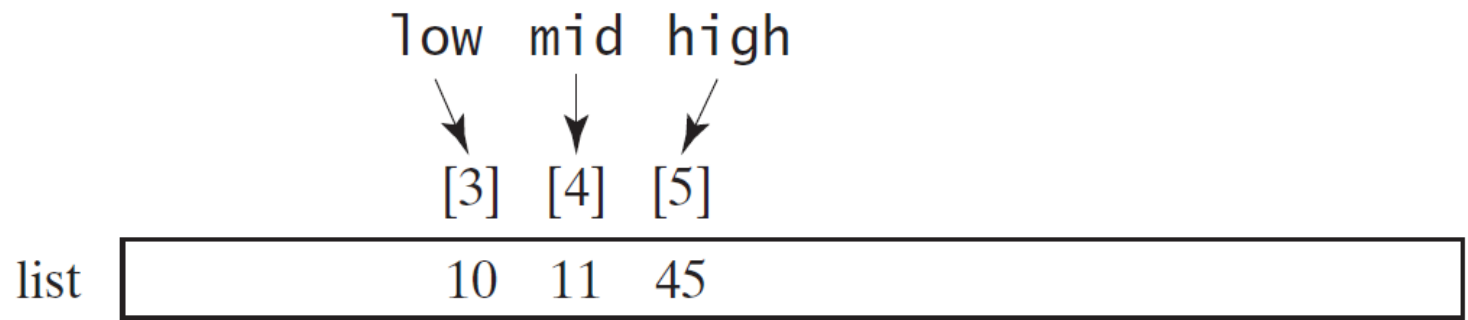
key < 50



key > 7



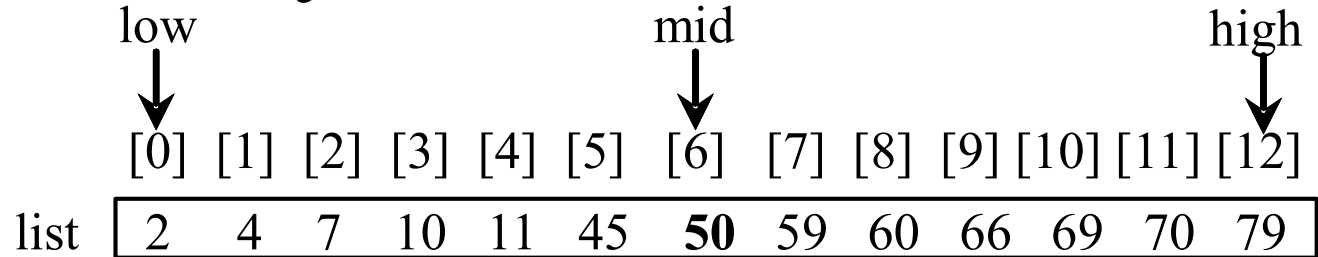
key == 11



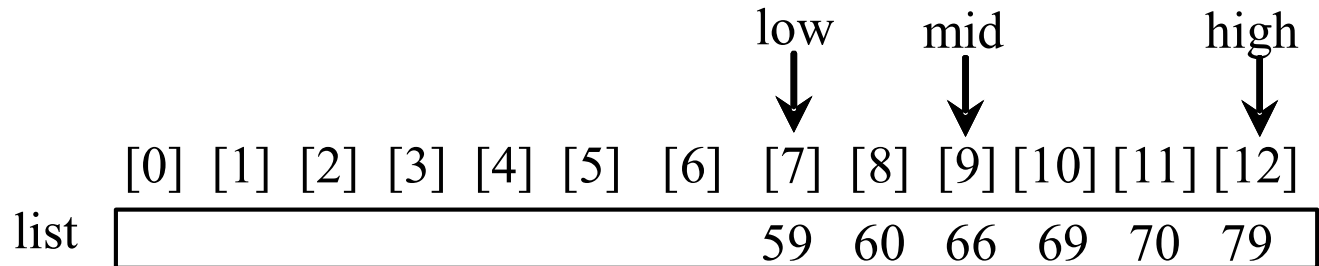
Binary Search, cont.

key is 54

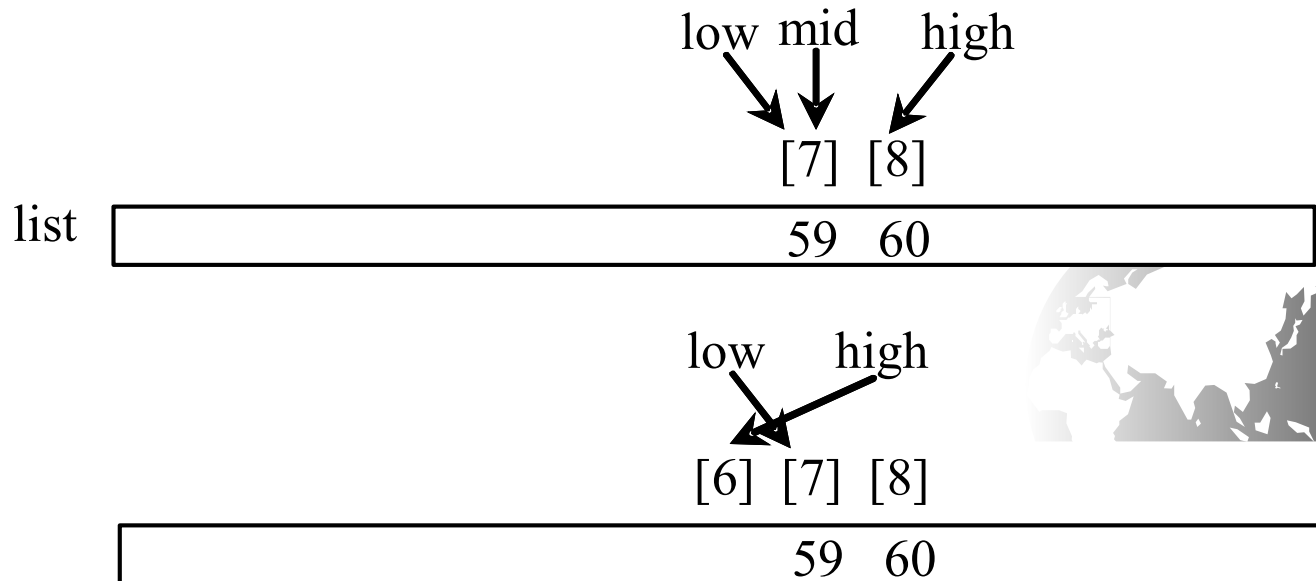
key > 50



key < 66



key < 59



Binary Search, cont.

- ✎ returns the index of the element in the list that matches the search key if it is contained in the list.
 - ✎ Otherwise, it returns :
 - insertion point - 1.
- The insertion point is the point at which the key would be inserted into the list.



From Idea to Solution

```
/** Use binary search to find the key in the list */
public static int binarySearch(int[] list, int key) {
    int low = 0;
    int high = list.length - 1;

    while (high >= low) {
        int mid = (low + high) / 2;
        if (key < list[mid])
            high = mid - 1;
        else if (key == list[mid])
            return mid;
        else
            low = mid + 1;
    }

    return -1 - low;
}
```