Linear Search

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

```
public int linear search(int[] a, int key) {
        for(int i=0; i < a.length; i++){
                if(a[i]== key) return i;
        }
        return -1;
}
```

Linear Search

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Ex. Linear search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Linear Search (Average Case)

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Ex. Linear search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

i

Linear Search

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Ex. Linear search for 33.

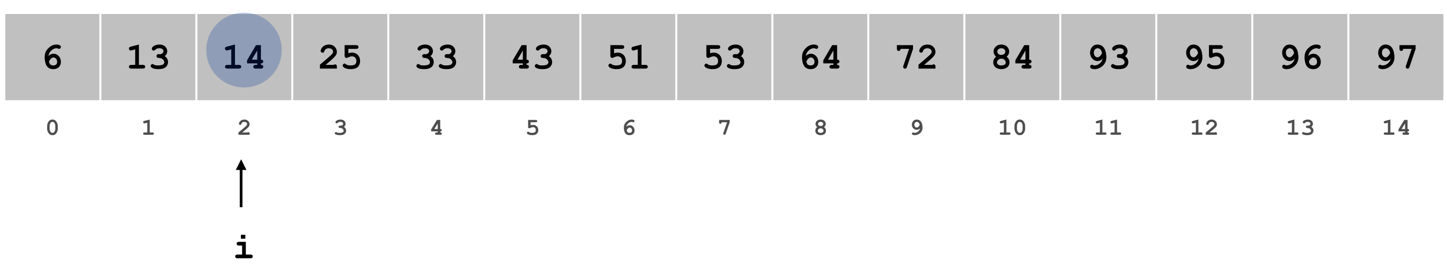
6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

i

Linear Search

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Ex. Linear search for 33.



Linear Search

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Ex. Linear search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

i

Linear Search

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Ex. Linear search for 33.

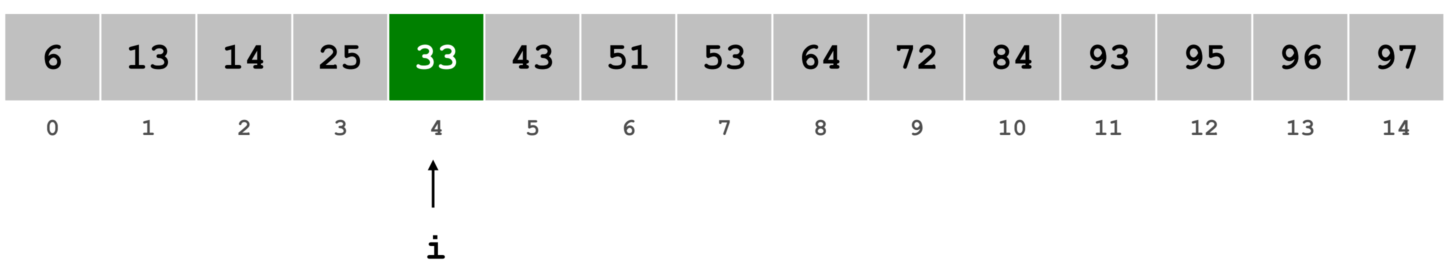
6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

i

Linear Search

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Ex. Linear search for 33.



Linear Search: Best Case

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Ex. Linear search for 6.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Linear Search

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Ex. Linear search for 6.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

↑

i

Linear Search

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Ex. Linear search for 6.

6	**13**	**14**	**25**	**33**	**43**	**51**	**53**	**64**	**72**	**84**	**93**	**95**	**96**	**97**
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
↑ **i**														

Linear Search: Worst Case

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Ex. Linear search for 97 .

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Linear Search: Worst Case

Linear search. Given `value` and an array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Ex. Linear search for 101 .

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Binary Search

Binary search. Given `value` and **sorted** array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

```
public int binary search(int[] a, int key) {
       int lo = 0;
       int hi = a.length - 1;
       while (lo <= hi) {
           // Key is in a[lo..hi] or not present.
           int mid = lo + (hi - lo) / 2;
           if      (key < a[mid]) hi = mid - 1;
           else if (key > a[mid]) lo = mid + 1;
           else return mid;
       }
       return -1;
   }
```

Binary Search: Average case

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Ex. Binary search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

lo **hi**

Binary Search

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Invariant. Algorithm maintains `a[lo]` ≤ `value` ≤ `a[hi]`.

Ex. Binary search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
↑ lo							↑ mid							↑ hi

Binary Search

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Invariant. Algorithm maintains `a[lo]` ≤ `value` ≤ `a[hi]`.

Ex. Binary search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
lo						hi								

Binary Search

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Invariant. Algorithm maintains `a[lo]` ≤ `value` ≤ `a[hi]`.

Ex. Binary search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
↑			↑			↑								
lo			**mid**			**hi**								

Binary Search

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Invariant. Algorithm maintains `a[lo]` ≤ `value` ≤ `a[hi]`.

Ex. Binary search for 33.

6	13	14	25	**33**	**43**	**51**	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
				↑ **lo**		↑ **hi**								

Binary Search

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Invariant. Algorithm maintains `a[lo]` ≤ `value` ≤ `a[hi]`.

Ex. Binary search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
				↑ lo	↑ mid	↑ hi								

Binary Search

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Invariant. Algorithm maintains `a[lo]` ≤ `value` ≤ `a[hi]`.

Ex. Binary search for 33.

6	13	14	25	**33**	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

lo
hi

Binary Search

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Invariant. Algorithm maintains `a[lo]` ≤ `value` ≤ `a[hi]`.

Ex. Binary search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

lo
hi
mid

Binary Search

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Invariant. Algorithm maintains `a[lo]` $\leq$ `value` $\leq$ `a[hi]`.

Ex. Binary search for 33.

6	13	14	25	**33**	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

lo
hi
mid

Binary Search: Best case

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Ex. Binary search for 53.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

lo **hi**

Binary Search

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Invariant. Algorithm maintains `a[lo]` ≤ `value` ≤ `a[hi]`.

Ex. Binary search for 53.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
↑							↑							↑
lo							**mid**							**hi**

Binary Search

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Invariant. Algorithm maintains `a[lo]` ≤ `value` ≤ `a[hi]`.

Ex. Binary search for 53.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
lo							mid							hi

Binary Search: Worst case

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

Ex. Binary search for 4.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
↑ lo														↑ hi

Binary Search: Worst case

Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i] = value`, or report that no such index exists.

Ex. Binary search for 101.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

↑ `lo` (index 0) ↑ `hi` (index 14)

Finding time complexity of Linear Search

```
public int linear search(int[] a, int key) {
        for(int i=0; i < a.length; i++){
                if(a[i]== key) return i;
        }
        return -1;
}
```

Approximately 3 operations per iteration
$f(n) = 3n + 2$

Finding time complexity of Linear Search: Average case

$f(n) = 3n + 2$

n = no of iterations/ location of key

$$f(1) = 3.1 + 2$$
$$f(2) = 3.2 + 2$$
$$f(3) = 3.3 + 2$$
$$\vdots$$
$$\vdots$$
$$\vdots$$
$$f(n) = 3.n + 2$$

$$f(1)+ f(2)+\ldots + f(n) = 3.(1+2+\ldots + n) + (2+ 2+ \ldots + 2)$$

$$= 3\ n(n+1)\ /2 + 2.n$$

$$= \tfrac{1}{2}\ (3n^2 + 7n)$$

No of cases = n

Average case time complexity = $\frac{1}{2}\ (3n^2 + 7n)\ /n$

$$= \tfrac{1}{2}\ (3n + 7)$$

Big O notation: $O(n)$

Finding time complexity of Binary Search: Average case

```
public int binary search(int[] a, int key) {
        int lo = 0;
        int hi = a.length - 1;
        while (lo <= hi) {
            // Key is in a[lo..hi] or not present.
            int mid = lo + (hi - lo) / 2;
            if      (key < a[mid]) hi = mid - 1;
            else if (key > a[mid]) lo = mid + 1;
            else return mid;
        }
        return -1;
    }
```

Approximately 3 comparisons per iteration

Finding time complexity of Binary Search: Average case

List size $n = 2^k$

$k = \log_2 n$

List size	No of comparisons

$$
\begin{aligned}
2^k &= 3 \\
2^{k-1} &= 3 \\
2^{k-2} &= 3 \\
&\vdots \\
&\vdots \\
&\vdots \\
1 &= 3
\end{aligned}
$$

$$= (3 + 3 + \ldots + 3)$$

$$= 3(k + 1)$$

$$= 3(\log_2 n + 1)$$

Big O notation: $O(\log_2 n)$