

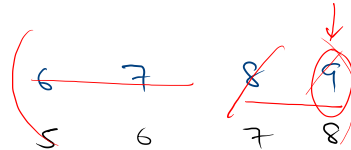
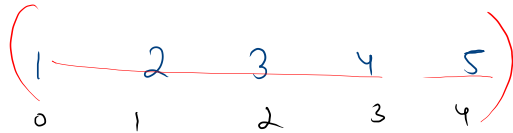
$O(n \log n)$ \rightarrow inbuilt

Binary Search

\hookrightarrow sorted array.

$\begin{cases} O(1) & O(\log n) & O(n) & O(n \log n) & O(n^2) \end{cases}$
 $\begin{cases} \text{sort} & + & \text{task} \end{cases}$
 $\rightarrow O(n \log n) + O(n) = O(n \log n)$
 $O(n^2) + O(n) = O(n^2)$
key = 70

$\log_2 n$



$$9/2 = 4$$

$$\frac{4}{2} = 2$$

$$\frac{2}{2} = 1$$

$$\frac{1}{2} = 0$$



\uparrow
m

$10^5 \rightarrow 1 \text{ lakh}$
 $10^6 \rightarrow 10 \text{ lakhs}$

$\left\{ \begin{array}{l} m=4 \\ m=6 \\ m=7 \\ m=8 \end{array} \right.$

✓
Sort



Binary Search
task ✓



TC ✓

bubble sort: $O(n^2)$

+

$O(n)$



$O(n^2)$

inbuilt sorting: $O(n \log n)$

+

$O(n)$



$O(n \log n)$ *

$O(n \log n)$

+

$O(n^2)$



$O(n^2)$

sort + linear search } TC

$O(n^2)$

$O(n)$

$$n^2 + n \rightarrow$$

$$\underline{\underline{O(n^2)}}.$$

Binary Search.

-1 \rightarrow if not present

key = 3

Sample Input 0

7 \rightarrow size
1 2 3 4 5 6 7
3

Sample Output 0

2

1	2	3	4	5	6	7
0	1	2	3	4	5	6

i j

if ($A[m]$ == key)
 $\hookrightarrow m$ is ans

else if $A[m] > \text{key}$.
 \hookrightarrow left $\rightarrow j = m - 1$

else \hookrightarrow right $\rightarrow i = m + 1$

$$m = i + j / 2$$

$$m = \underline{3}$$

$$m = 1$$

$$\underline{m} = 2$$

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5     public static int binarySearch(int [] A, int key){
6         int i = 0;
7         int j = A.length-1;
8         while(i <= j){
9             int m = (i + j)/2;
10            if(A[m] == key){
11                return m;
12            }else if(A[m] > key){ //left
13                j = m - 1;
14            }else{ // A[m] < key -> m = i+1;
15                i = m + 1;
16            }
17        }
18        return -1;
19    }
20
21    public static void main(String[] args) {
22        Scanner scn = new Scanner(System.in);
23        int n = scn.nextInt();
24        int [] A = new int[n];
25        for(int i = 0; i < n; i++){
26            A[i] = scn.nextInt();
27        }
28        int key = scn.nextInt();
29        System.out.println(binarySearch(A, key));

```

1 2 3 4 5 6 7
 0 1 2 3 4 5 6

key=3
4 2

Search Character

key = 'c'

ans = 'd'

a	b	c	d	e
0	1	2	3	4

Sample Input 0

c
⑤

a b c d e

key = c

ans = 'e'

a	b	c	e	f
0	1	2	3	4

Sample Output 0

d

key = c

a	b	g	h	z
---	---	---	---	---

key = c

④

a	b	d	f	z
---	---	---	---	---

key = c

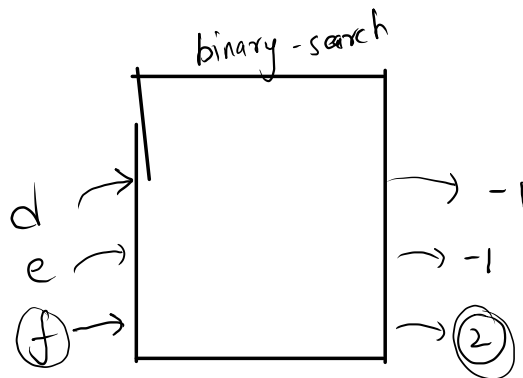
a	a	a	b	b
---	---	---	---	---

→ -1

key = c

a	b	f	g	h
0	1	2	3	4

d x
e x
f



```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5     public static int binarySearch(char [] A, char key){
6         int i = 0;
7         int j = A.length-1;
8         while(i <= j){
9             int m = (i + j)/2;
10            if(A[m] == key){
11                return m;
12            }else if(A[m] > key){
13                j = m - 1;
14            }else{
15                i = m + 1;
16            }
17        }
18        return -1;
19    }
20 }

```

key = b
ans = f

a	b	f	g	h
0	1	2	3	4

```

20
21 public static void main(String[] args) {
22     Scanner scn = new Scanner(System.in);
23     char key = scn.next().charAt(0);
24     int n = scn.nextInt();
25     char [] A = new char[n];
26     for(int i = 0; i < n; i++){
27         A[i] = scn.next().charAt(0);
28     }
29
30     key++;
31     for(char ch = key; ch <= 'z'; ch++){
32         int ans = binarySearch(A, ch);
33         if(ans != -1){
34             System.out.println(ch);
35             return;
36         }
37     }
38     System.out.println(-1);
39 }
40 }

```


Find Last Occurrence

$$m = 4 \text{ } 7$$

$$i \leq j$$

key = 3

1	2	2	2	3	3	3	3	4	5
0	1	2	3	4	5	6	7	8	9

$$\begin{matrix} j & i \\ \hline & m \end{matrix}$$

Sample Input 0

```
6
1 2 3 3 3 4
3
```

Sample Output 0

```
4
```

$$m = 4 \checkmark$$

$$m = 7 \checkmark$$

$$m = 8 \checkmark$$

$$A[m] == \text{key} \rightarrow \text{ans} = m$$

$$A[m] > \text{key} \rightarrow \text{left}$$

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5     public static int binarySearch(int [] A, int key){
6         int i = 0;
7         int j = A.length-1;
8         int ans = -1;
9         while(i <= j){
10             int m = (i + j)/2;
11             if(A[m] == key){
12                 ans = m; //update and search in right direction
13                 i = m + 1;
14             }else if(A[m] > key){ //left
15                 j = m - 1;
16             }else{ // A[m] < key -> i = m+1;
17                 i = m + 1;
18             }
19         }
20         return ans;
21     }
22 }

```

```

23 public static void main(String[] args) {
24     Scanner scn = new Scanner(System.in);
25     int n = scn.nextInt();
26     int [] A = new int[n];
27     for(int i = 0; i < n; i++){
28         A[i] = scn.nextInt();
29     }
30     int key = scn.nextInt();
31     System.out.println(binarySearch(A, key))
32
33 }
34 }

```

first occurrence.

key = 2

ans = ~~1~~ ~~2~~ ~~3~~ 1

	m					li					
1	2	2	2	2	2	2	3	3	4	5	
0	1	2	3	4	5	6	7	8	9	10	
j	i										
	fi										

A[m] == key
 ↪ ans = m
 left

m = 5
m = 2
m = 0
m = 1