

Searching

Linear Search

0	1	2	3	4	5
10	8	7	9	12	1

$\{ \text{ } \}$ $\text{int linearSearch}(\text{int } a[], \text{int } val) \{$

$\text{for}(\text{int } i=0; i < a.length; i++) \{$

$\text{if } (a[i] == val) \{$

$\text{return } i;$

$\text{return } -1;$

$\}$

Time $\Rightarrow O(n)$

Binary Search

val 65

2

* Array should be sorted

0	1	2	3	4	5	6	7	8	9
10	20	30	40	50	60	70	80	90	100

l

r

m

① Calculate mid

② Compare with element at m

③ Adjust search area

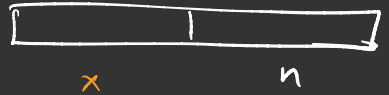
$$l = 0$$

$$r = a.length - 1$$

$$m = \frac{l + r}{2}$$

Time complexity

Iteration no	Size	
0	n	
1	$n/2$	$n/2^1$
2	$n/4$	$n/2^2$
3	$n/8$	$n/2^3$
k^{th}	1	$n/2^k$



$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$\log_2 n = \log_2 2^k$$

$$\log_2 n = k \log_2 2$$

Time complexity

$$k = \log_2 n$$

Comparison

LS

BS

$$n = 10^9$$

$$10^3 = 1000 \approx 2^{10}$$

$$10^3 \approx 2^{10}$$

$$10^9 \Rightarrow 10^3 \times 10^3 \times 10^3$$

$$\Rightarrow 2^{10} \times 2^{10} \times 2^{10}$$

$$n \Rightarrow 2^{30}$$

$$\log_2 2^{30} \Rightarrow 30$$

Ques Find first occurrence

Val = 4

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

→ 3

possible Answer = -1;

while (l <= r) {

if (arr[mid] == val) {

 p = mid;

 r = mid - 1;

}

else if {

} {

 else {

 }

}

Ques Last Occurrence

Ques Count total occurrences
arr is sorted.

last - first + 1