Linear Search

60

0	1	2	3	4	5
10	30	20	40	60	50

```
for(int i=0; i<arr.length; i++;) {
    if (arr[i] == val) {
        return i;
    }
}
return -1;
```

② $\underbrace{\text{Binary}}_{\substack{1\\2}}$ Search

* Array should be sorted in

55

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

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

$l = 0$

$r =$ arr.length - 1;

m

① Calculate m

② Compare with mid

③ Adjust the search area

$(l+r)/2$

$l + (r-l)/2$

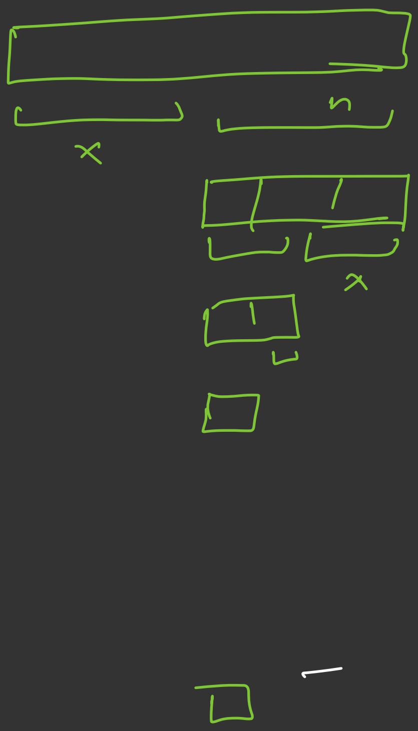
$\Rightarrow l + \frac{r}{2} - \frac{l}{2}$

$\Rightarrow \frac{l}{2} + \frac{r}{2}$

$\Rightarrow \frac{l+r}{2}$

Time Complexity

	Size	
	n	
1	$\frac{n}{2}$	$\frac{n}{2^1}$
2	$\frac{n}{4}$	$\frac{n}{2^2}$
3	$\frac{n}{8}$	$\frac{n}{2^3}$
⋮	⋮	⋮
k	1	$\frac{n}{2^k}$



k is time complexity

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

$$n = 2^k$$

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

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

$$\log_2 n = k$$

Time $\Rightarrow O(\log_2 n)$

Comparision

$O(n)$ and $O(\log_2 n)$

10^9

10^9

$10^9 \approx 2^{30}$

$2^{10} = 1024$

≈ 1000

10^3

$\log_2(10^9)$

$\log_2(2^{30})$

$\Rightarrow 30$

$$\boxed{2^{30} \approx 10^9}$$

Ques First occurrence

0	1	2	3	4	5	6
10	20	60	60	60	60	60

60

```
l=0, ans = -1;
r = arr.length;
while( l <= r) {
   m =(l + r)/2;
   if ( arr[m] == val) {
         ans =m;
         r= m - 1;
   } else if( arr[m] > val ) {
            r = m-1;
   } els {
            l = m+1;
   }
}

return ans;
```

Ques Last occurrence

0	1	2	3	4	5	6
10	20	60	60	60	60	60

60