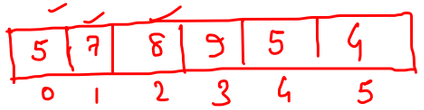
Searching

- Linear Search
- Binary search

Linear Search



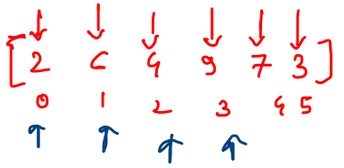
target = 9

```
for(i=0; i<6 ; i++) {
    if(a[i] == target) {
        printf("%d", i);
    }
}
```

```c
int a[10]={2,6,4,9,7,3},i;
int target = 19;
int found = 0;
for(i=0;i<6;i++){
    if(a[i]==target){
        printf("Found at index :%d\n",i);
        found = 1;
        break;

    }
}
if(found==0){
    printf("Not Found");
}
```

target = 9



n = Array size

Found at index 3

T.C → O(n)

Imil

Binary Search

↳ Array Must be Sorted.

5	4	3	7	8	2	X

2	3	9	21	32	43
0	1	2	3	4	5

lo, mid, lo, mid, hi

mid = (low + hi)/2

= 0+5/2

3+5/2

target = 32 8/2 = 4

target = 3

```
if(a[mid] == target) printf mid
if(a[mid] < target)
    lo = mid+1;
```

2	3	9	21	32	43
0	1	2	3	4	5

lo, mid, hi, mid, hi, mid

(5+0)/2 = 2

target 3 =

2 ≤ 3

1+1/2 1/2 = 0

2/2 = 1

```
if(a[mid] == target) printf mid
if(a[mid] < target)
    lo = mid+1;
```

if (a[mid] > target)

hi = mid - 1

```
int a[10]={2,3,9,17,23},lo=0,hi=4;
int target = 9;
int found =0;
while(lo<=hi){
    int mid = (lo+hi)/2;
    if(a[mid]==target){
        printf("Found at index:%d",mid);
        found=1;
        break;
    }
    if(a[mid]<target){
        lo=mid+1;
    }
    if(a[mid]>target){
        hi = mid-1;
    }
}
if(found==0){
    printf("Not Found");
}
return 0;
```

Time Complexity → O(log n)

700 ✓

1

linear

2.845

3

binary