Binary Search (Algorithm)

```
int[] arr = { 1, 3, 7, 10, 11, 14, 20, 40 }
            0  1  2  3   4   5   6   7
```

target = 14

sorted

Brute force

```
for(int i=0; i<n; i++)
{
    if(arr[i] == target)
        return i;
}
```

Linear Search

TC : O(N)

SC : O(1)

what is the lowest floor from which, throwing brick will break

1000 Bricks

↳ Brute force

Optimized Approach :

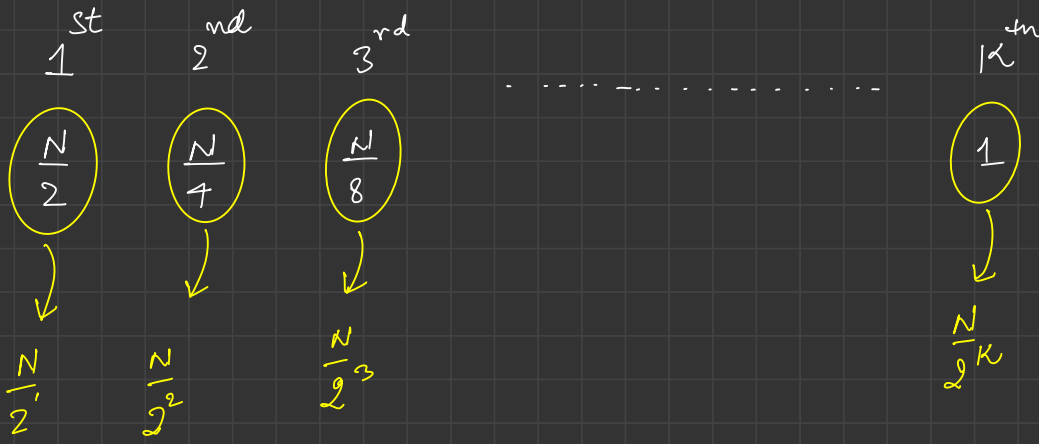
$\log_2(1000) = \log_2 10^3 = 3 \times \log_2 10$ (3.1)

$\approx$ 10 Bricks

→ By using first Brick, I eliminated 500 floors

→ By using second Brick, I eliminated 250 more floors

→ By using 3rd Brick, I eliminate 125 more floors.



$$\frac{N}{2^K} = 1$$

$$N = 2^K$$

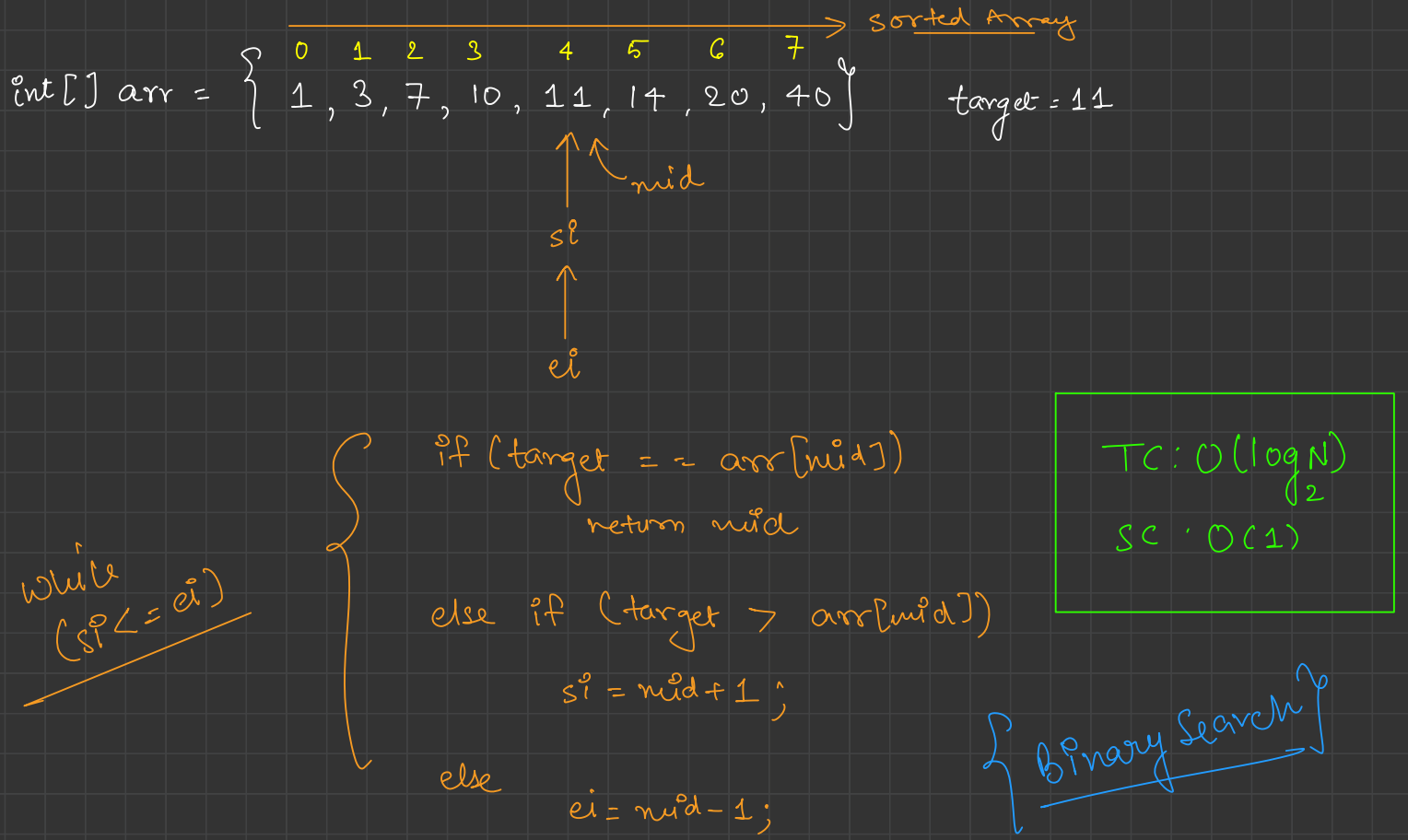
taking $\log_2$ Both Side

$$\log_2 N = \log_2 2^K$$

$$\log_2 N = K \log_2 2 \quad (\log_2 2 = 1)$$

$$\boxed{K = \log_2 N}$$

→ Bricks one required!


sorted Array
0 1 2 3 4 5 6 7
int[] arr = { 1, 3, 7, 10, 11, 14, 20, 40 }
target = 11
mid
si
ei
while (si <= ei)
if (target == arr[mid])
return mid
else if (target > arr[mid])
si = mid + 1;
else
ei = mid - 1;
TC : O(log N) base 2
SC : O(1)
Binary Search

Binary Search

↳ always implemented over a sorted region

⟶ Sorted region

⟶ TC: $O(\log_2 N)$ (Expected) } 99% ⟶ Binary Search

Search insert position. (ceil value)

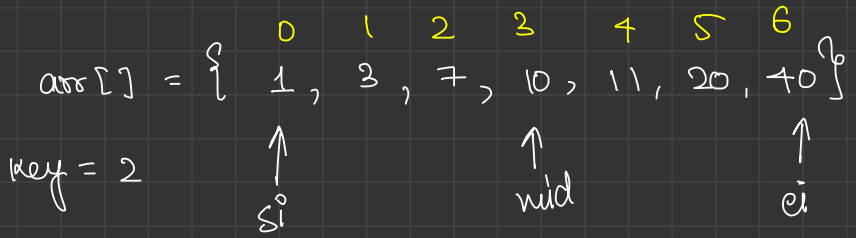
	0	1	2	3	4	5	6
arr[] = {	1,	3,	7,	10,	11,	20,	40 }

key = 2

✓ Brute force.

```
{
    if (arr[i] == key) return i;
    else if (arr[i] > key) return i;
```

TC: O(N)
SC: O(1)



Case 1 arr[mid] == key

Case 2 arr[mid] > key

ans = mid

ei = mid - 1;

Case 3 arr[mid] < key

si = mid + 1;

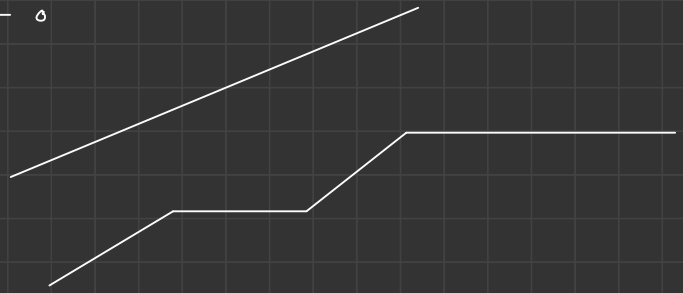
ans = arr.length

(default value)

ans ↓ ceil value

find first and last Pos. of a element.

- inc. array
- non-decreasing array



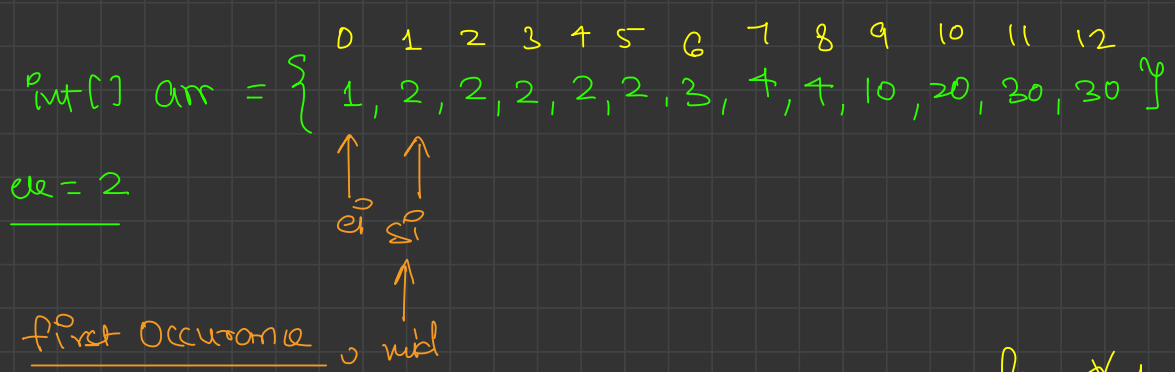
	0	1	2	3	4	5	6	7	8	9	10	11	12
int[] arr = {	1,	2,	2,	2,	2,	2,	3,	4,	4,	10,	20,	30,	30 }

ele = 2

fo = 1 lo = 5

Brute force

TC : O(N)
SC : O(1)



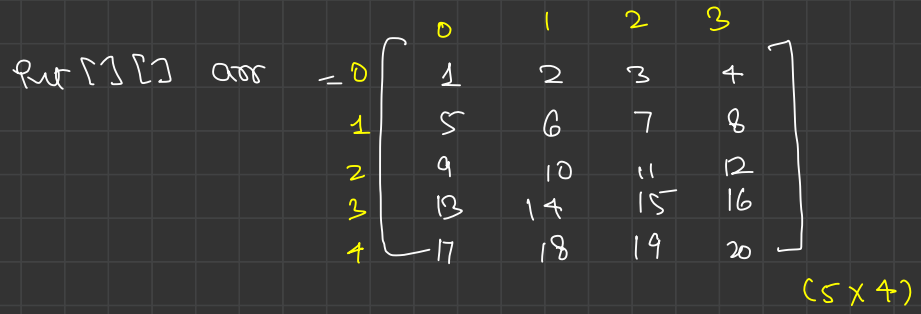
fo = -1

case 1 : arr[mid] == ele
fo = mid;
ei = mid - 1;

case 2 : arr[mid] > ele
ei = mid - 1;

case 3 : arr[mid] < ele
si = mid + 1;

Search In a 2D-Matrix (2D Sorted Matrix)



target = 14