

Binary Search Questions

Q Ceiling of a number

Q Given an array $\text{arr} = \{2, 3, 5, 9, 14, 16, 18\}$ 16, 18 Sorted arr \Rightarrow B.Search)
int target = 15

what is ceiling?

ceiling \rightarrow Find the smallest number element in the array which is greater or = target.

if (target = 14) \Rightarrow ceiling = 14

if (target = 15) \Rightarrow 16

target = 4 \Rightarrow 5

1. get target
2. Find all the elements greater than the target
3. Take the shortest of that element

say target = 9

$\Rightarrow [9, 14, 16, 18] \Rightarrow \text{ceiling} = 9$

\rightarrow we can apply Binary Search

My try \rightarrow if element not there in array \rightarrow don't return -1

\rightarrow Take an array list, add all elements greater than or equal to the target } wrong Kadhu! but time complexity O(n)

\rightarrow get the element

Soln
 $\text{arr} = [2, 3, 5, 9, 14, 16, 18]$ target = 15

eg s m e
14 16 17 18
target < mid

s m e
14 16 18
target < mid

s m e
14 16 18
target \geq mid

s m e
14 16 18
target $>$ mid
return start
18 \Rightarrow loop breaks

Start & end pointers

14 e 16 18

(start) → ~~contains~~ (end)

- this is the point ~~in~~

• contains is always b/w start & end

→ they are two pointers they are used to reduce the search space

→ after the loop breakdown

→ Start is becoming the answer which we need

→ as it is the next number greater than target it is the answer

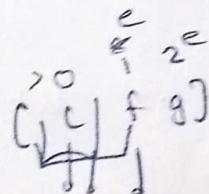
→ if we need the floor then return ~~ans~~ end //

Q2 Floor ↗

floor ⇒ Greatest no in the array that is smaller or equal to target number

→ what if not any answer will be found ie the target is greater than the greatest number

```
ie if(target > arr.length - 1) {
    return -1;
}
```



Smallest number

LeetCode 744

Given

→ char[] letters = []

→ ascending order

→ target char = "a"

→ Letters wrap around

→ return the smallest char in the letters Alphabetically

≈ similar to floor
→ ~~floor~~

Soln

letter's wrap around

If $\text{letters} = ['a', 'b']$ target = 2
the return statement will change

If no elements \Rightarrow return this

eg

$\text{arr} = [^0, ^1, ^2, ^3] \quad 'c', 'd', 'f', 'j'$, target = j

\rightarrow No char is larger than j \Rightarrow return $\text{arr}[0] = 'c'$

We can do this via modulo

Condition is violated :- start = end + 1 \Rightarrow length of array = N

return $s \% N$ or

if $s = N$:

return 0

To wrap around
use %.

\rightarrow modulo converts overflow into

circular indexing

\Rightarrow Some concept of % when

next greater
wrap around
Circular array

Q Find first and last position of element in sorted array

Leet Code 34

Given

int[] nums in Ascending order \Rightarrow B+S

algo with $O(\log(n))$

To find

starting & ending
position of a given
target value

eg $[5, 7, 7, 8, 8, 10]$ target = 8
if not found $\rightarrow [-1, -1]$
output: [3, 4]

0	1	2	3	4	5
5	7	(7)	8	8	10

target > mid

$$\Rightarrow s = \text{mid} + 1$$

3	4	5
8	(8)	10

target > m

$$\text{start} = \underline{\text{mid} + 1}$$

target < m

$$\text{end} = \underline{\text{mid} - 1}$$

target = m

$$[\underline{s}, \underline{m}]$$

target = m

$$\text{end} = \text{mid} - 1$$

8	8
---	---

Soln

$$\text{arr} = \{ 5, 7, 7, 7, 7, 8, 8, 10 \} \quad \text{target} = 7$$

$$\text{first - I} = 1$$

$$\text{L. I. - 7} = 4$$

$$\text{First Brute force} \rightarrow [5, 7, 7, 7, 7, 8, 8, 10]$$

Start Comparing from both sides \Rightarrow O(n^2)

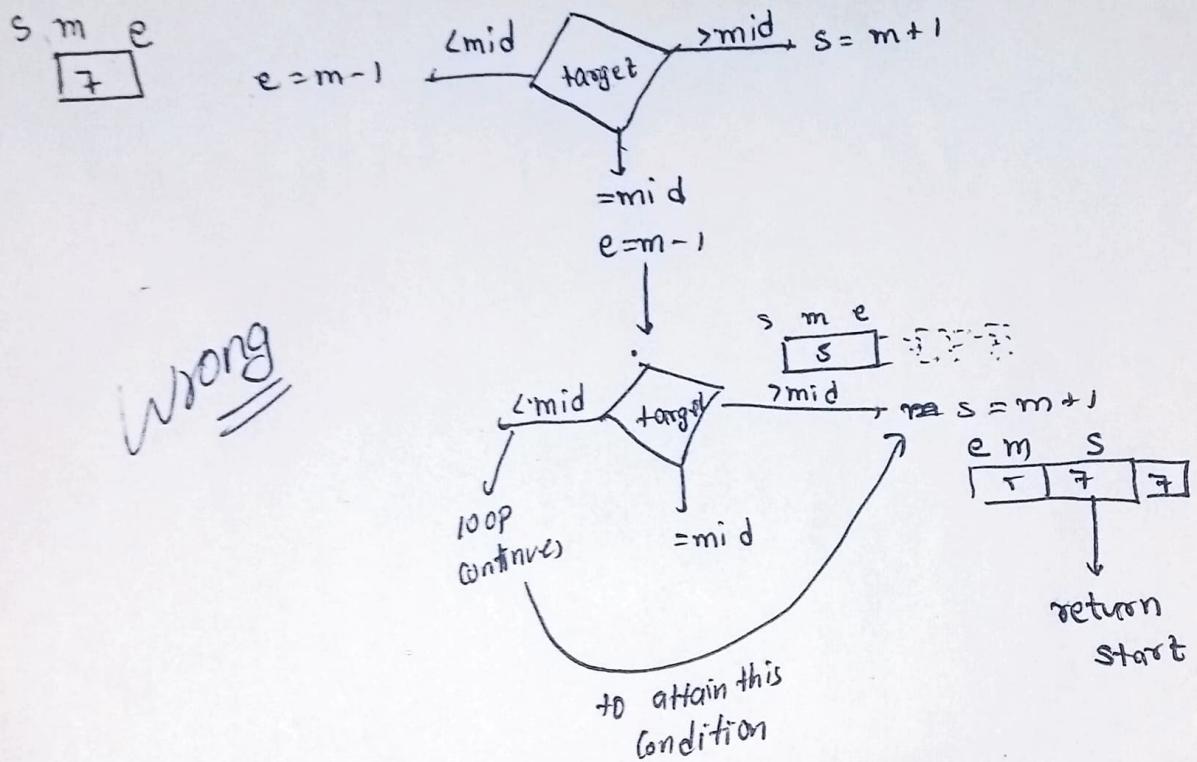
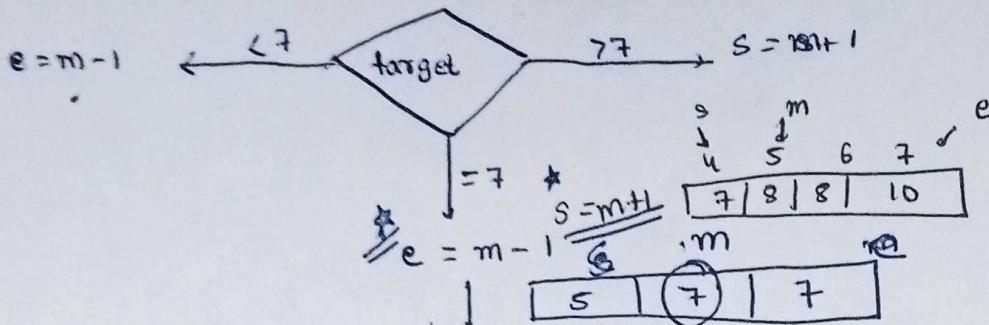
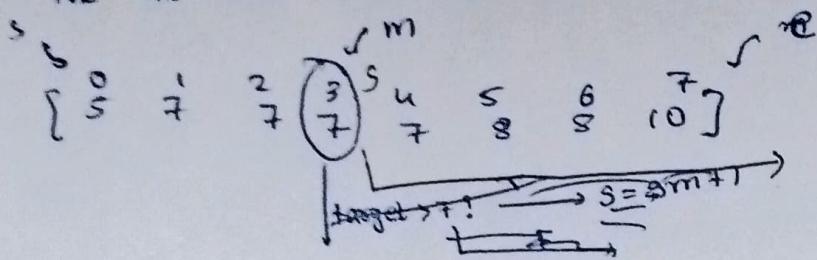
Better one:-

use Binary Search

whenever we find the target

\rightarrow do the binary search twice \rightarrow first time for first index
Second time for last index

PI of theg :- Find the first occurrence of 7 here:-

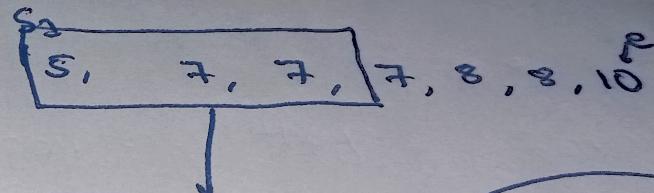


Solution:-

Find first occurrence of 7

SOL

First occurrence of $\textcircled{7} \Rightarrow m - 1 = e$



Last occurrence of $\textcircled{7} \Rightarrow m + 1 = s$