

Binary Search Questions

Q Ceiling of a number

Q Given an array $arr = [2, 3, 5, 9, 14, 16, 18]$
int target = 15
Sorted arr \Rightarrow B-Search

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

Say target = 9

(2) $\Rightarrow [9, 14, 16, 18] \Rightarrow$ ceiling = 9

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

\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

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

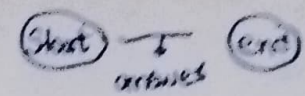
s m e
14 16
target < mid

s m e
14 16
target < mid

return start
18 \Rightarrow loop breaks

Start & end pointers

e 5
14 16 18



- This is the point imp
- answer is always b/w start & end

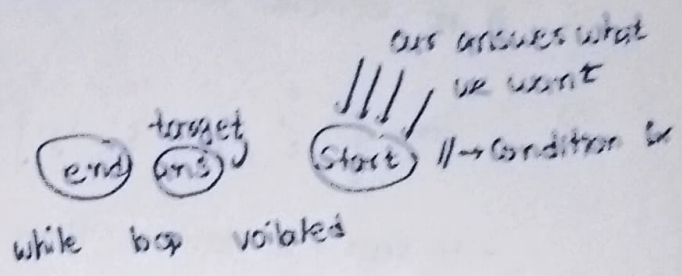
→ They are two pointers they are used to reduce the search space

→ 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

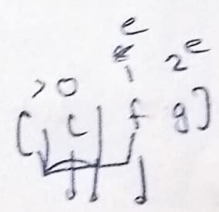


Q2 floor ↑
floor ⇒ Greatest no in the array that is smaller or equal to target number

→ What if not any answer will be found i.e. the target is greater than the greatest number

i.e. if (target > arr[arr.length - 1]) {

return -1;
}



Q3 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
→ ~~to~~

Soln

letters wrap around

if letters = ['a', 'b'] target = 2

the return statement will change

if no element return this

eg

arr = ['c', 'd', 'f', 'j'], target = j

→ No char is larger than j ⇒ return arr[0] = 'c'

We can do this via modulo

Condition is violated :→ start = end + 1 ⇒ length of array = N

return S % N or

To wrap around
Use %

if S = N:
return 0

→ modulo converts overflow into
circular indexing

⇒ Some concept of % when
next greater
wrap around
circular array

Q7 Find first and last position of element in sorted array

Leet Code 34

Given

int[] nums in ascending order ⇒ B.S

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

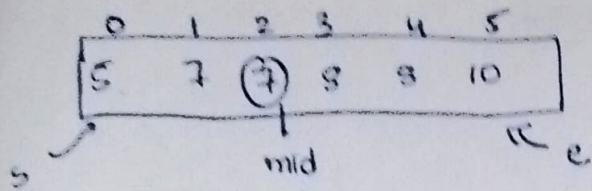
To find

starting & ending
position of a given
target value

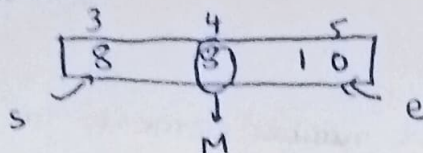
eg [3, 7, 7, 8, 8, 10] target = 8

if not found → [-1, -1]

output: [3, 4]



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



target > m
 start = mid + 1

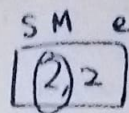
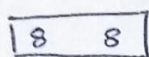
target < m
 end = mid - 1

target == m

[s, M]

target == m

end = mid - 1



target

Sam

eg

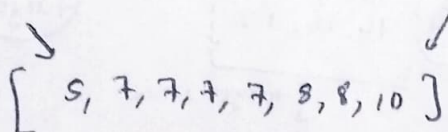
arr = [5, 7, 7, 7, 7, 8, 8, 10]

target = 7

first-I = ①

L.I. 7 = ④

First Brute force

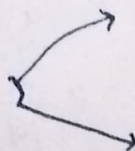


Start Comparing from both sides \Rightarrow ~~only~~ Ocn?

Better one:-

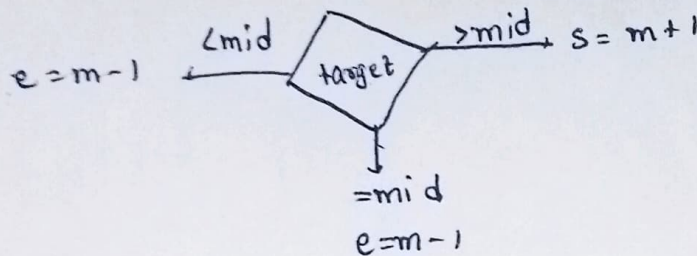
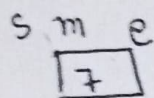
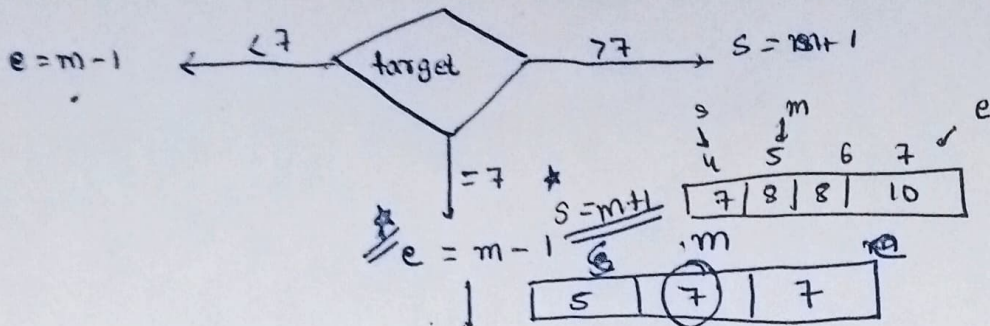
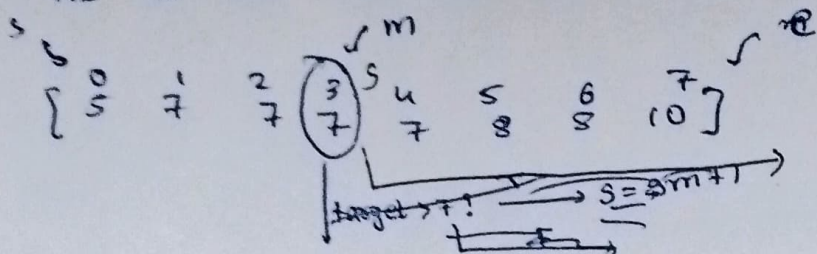
Use Binary Search

whenever we find the target

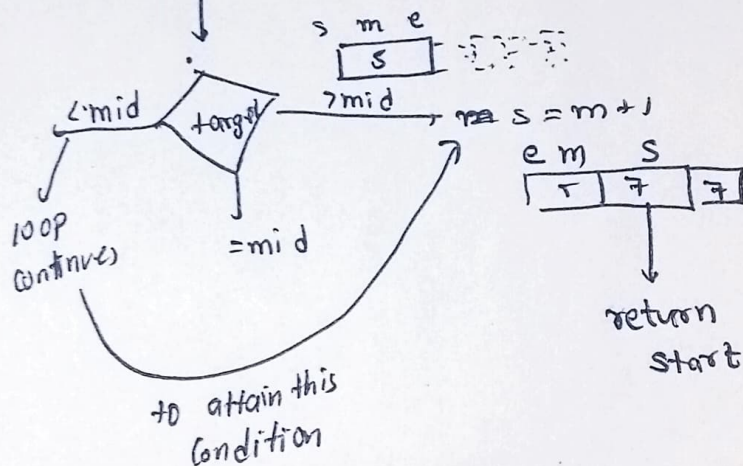


\rightarrow run the binary search twice \rightarrow First time for first index
 Second time for last index

PS of the q: Find the first occurrence of 7 here:-



Wrong



Solution:-

Find first occurrence of 7

Soln

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

s_2
5, 7, 7, 7, 8, 8, 10^e

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