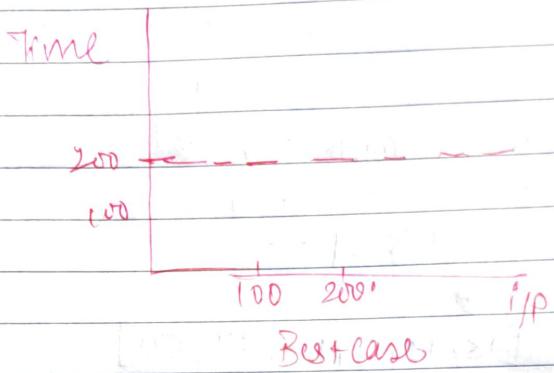
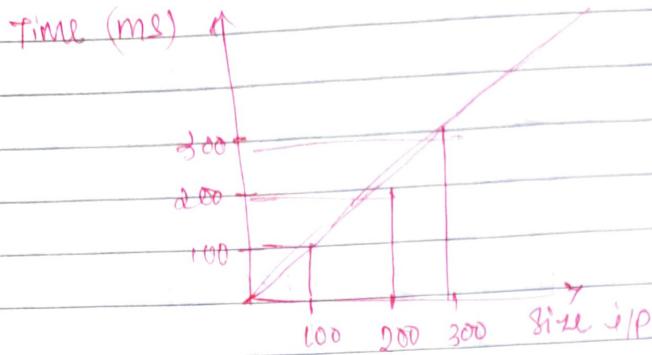


HOW TIME GROWS AS YOUR I/P GROWS.

CLASS
Date _____
Page _____



Upper Bound $O(n) \rightarrow \text{Big O}$

INTEGER.MAX-VALUE
+
const.

public class Main {

public static void main (String [] args) {

Scanner sc = new Scanner (System.in)

If you make a new object of

Scanner class
and store to
variable sc

at the same time calling constructor
of the class with parameters

int arr[] = {2, 3, 4, 5, 6, 7, 8, 9, 10}

int target = 5

int ans = linear search (arr, target)

System.out.println (ans)

O/p \rightarrow (3)

method and function
when function defined inside a class
it's called a method.

classmate
Date _____
Page _____

static int linearSearch(int[] arr, int target) {

if (arr.length == 0)
return -1;

for (int element : arr)
if (element == target)

int i=0; if (arr.length ; i++)

if (arr[i] == target)

for (datatype item : arr) {
if (item == target)
return i;

4

return -1;

max = 24

min = 1

max = 24

length \Rightarrow variable \Rightarrow size of array

length \Rightarrow no. of elements in a string

String's array of characters \Rightarrow

in function return

INT, MIN-VALUE

new int[10] \Rightarrow now, 104.

minimum value
that an integer
can hold

enhanced for loop &

you can't traverse in
reverse order

min = MAX VALUE

if element < min

min = element

max > element

element > max

max = element -

element != null

min = element

max = INTMIN

11469

length of number even?

- ① count number of digits.
 - ② convert to string and string.length

rows of code.

while (nyo)

count + 1

num / = 10;

between no of digits 0/02 = 0

* not adding static in leetcode

~~88~~ ~~73~~ ~~12~~
~~1500000~~ ~~11~~

return int (math.log10 (num))

deoxy
 $\text{g} \rightarrow \text{octa}$

$2 \rightarrow$ binary

1 August \rightarrow 365 days
↓

Resume

Top no 5th 3^m → certificates
self → of projects

Hackathon (31)

→ Clark Sheldow →
← 3 years 1 year Iowa, USA, 100 self development?
(JS) 9+3 5th Aug

from → ↗ +
back ↘ ↙ frame

24th August → 30th August.
30+

Portmion → Target → 15th Binary Searches
Interview Questions

After dinner → session

9:30 AM breakfast → 3 →
10 → 1 PM. 2 → 5 PM
10th Class → 2 → 9 →
3 → Shift → 2 hours → 6 hours
→ Total 6 hours daily.
12th Class → (2-3) hrs
charge → -2 hrs → 8 hours

OOPS

return $f(\text{int})$ (Math. $\log_{10}(\text{num})$)

↓
number of digits.

→ (if is even?)

+1

② → 9

10 → 10

(log n)

$$\log_{10} 10 = 1$$

$$\log_{10} = \log_2$$

$$\log_2 2 = 1$$

$$\log_{10} 10 = 2.3025857$$

$$10^0 = 1 \quad \log(10^0) = 0$$

$$10^1 = 10 \quad \log(10^1) = 1 \times \log_{10}(10) = 1$$

$$10^2 = 100 \quad \log_{10} 10^2 = 2 \log_{10} 10 \Rightarrow$$

$$10^4 = 10000 \quad \log_{10}(10000) = 4$$

Logarithms

$$\log_a b = n$$

$$a^n = b$$

$$\log_{10} 10 =$$

$$10^n = 10$$

$$(n=1)$$

$$(10)$$

Properties

$$\log_a a = 1$$

$$\log_a (1) = 0$$

$$\log_a (a^n) = n$$

$$\log_a b = n$$

$$a^n = b$$

BINARY SEARCH →

- ① Array should be sorted ascending
- ② Get the middle element. Search (S+C)
- ③ if target \rightarrow mid == middle element return index
- ④ if target $>$ mid then
 - Search in the right side
 - discard left portion
- else
 - search in left side
 - discard right portion
- ⑤ if middle == target
 - return middle

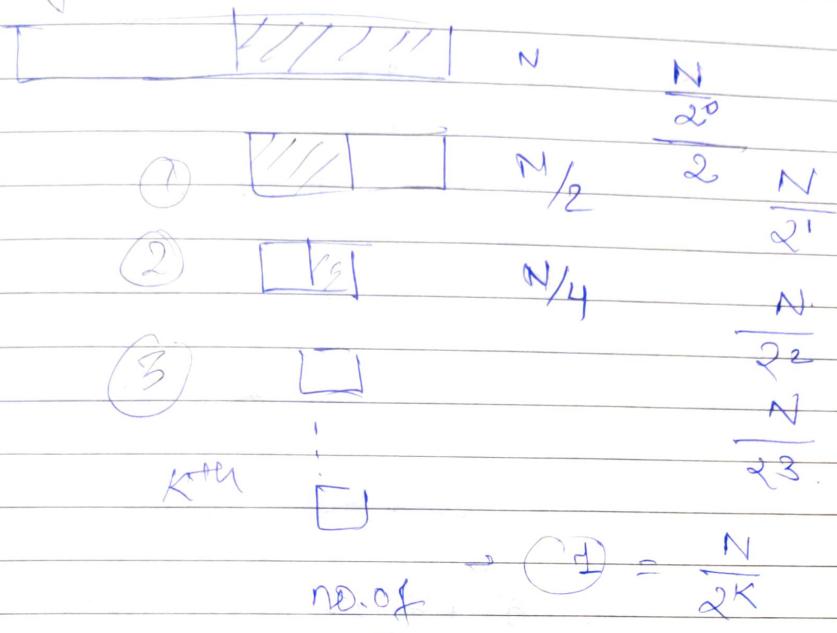
We are not creating a new array
we are changing pointers

Time



size.

Why Binary Search.



$$\text{no. of element} \rightarrow C_1 = \frac{N}{2^K}$$

element

left.

$$(N = 2^K)$$

To find total num of element

$$\log N = \log 2^K$$

$$\log N = K(\log 2) \rightarrow ①$$

$$K = \log N$$

1,000,100 → comp

LS → 1 min

$$\log_2(10000)$$

$$\log_2(1 \text{ million})$$

$$(19)$$

almost

difference is (20)
night/day

Code: Binary Search

for large value of start and end

this may exceed the int range

!!

$$* \quad m = \frac{s+e}{2}$$

$$m = s + \frac{(e-s)}{2}$$

$$= \frac{2s+e-s}{2}$$

$$= \frac{(s+e)}{2}$$

2, 58, 78, 454, 748, 799, 856

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
0 @ m \rightarrow e \rightarrow n

10

① ②

⑤

$$s \quad e$$
$$s^2 \quad m^2 \quad e^2$$
$$s^2 \quad (m^2 + 1) \quad e^2$$

classmate
Date _____
Page _____

if sorted
but
you don't know sorted in asc/
desc order

ORDER AGNOSTIC BINARY SEARCH

arr = [90, 75, 18, 12, 6, 4, 3, 1]

target < middle

→ right hand side

$m = s + 1$

target > middle

$m = e - 1$

** Hint → take any two element and compare
not figure out how.

3, 3, 3, ..., 3, 3

↑

if last

3

if arr[start] < arr[last]

first check then asc ord

else

else ord

if arr[start] = arr[last]

then whole array

change in algorithm

has
same elements.

bookish asc = arr[start] < arr[last]

when (start <= end)

if target = mid

return mid;

if ($!isAsc$)

target > mid

$m = s + 1$

else

$e = m - 1$

else

target < mid

$s = m + 1$

only this sign

changed $e =$

else $e = m - 1$

4 HOURS

BINARY SEARCH $\frac{1}{2}$ HOURS?

- if problem statement is sorted.
- if we have the following sequence
 - don't hesitate

Q: ceiling of a number.

ceiling \rightarrow smallest element in the array

greater than $y = \text{target}$

array = $\{2, 3, 5, 9, 14, 16, 18\}$

ceiling (array, target = 14) = 14.

ceiling (array, target = 15) = 16

target = 14 - (5)

4 | 4
2 | 4

$\Rightarrow (2, 3, 5, 9, 14, 16, 18)$
1 2 3 4 5 6

3 | 3
3 3
3 3

target 14

5 | 5
5 5

target 10

5 6 5 7
5 7

ceiling 14

5 8

target 18

9 | 9

ceiling \neq 18

PPP approach

int start = 0;

int end = array.length - 1;

while (start \neq end)

if int mid = start + $\frac{(end - start)}{2}$

if array[mid] == target

return target;

else if (target > mid)

start = mid + 1

else

end = mid - 1

classmate

target, mid (array)
max = min - int
element < max
max > element

9. $\text{for } (\text{int } i=0; \text{ int } i < \text{array.length}; i++)$
int ans

if target > arr[i],

2, 3, 5, 9.

while (target < arr[i])

i++;

15 56 57 58 59 59 70 71 45.

for (element : arr)

if (target > element)

sort. element.

2, 3, 5, 9.

(4)

if (target > element)

sort (element)

Ceiling Concept.

return (8) instead

the idea is the S. (ans) e of -1

WHEN loop breaks. ie (start <= end) (sync)

e (ans) (5)

denominator

is the next smallest element greater than the target.

FLOOR CONCEPT -

Floor - greatest element \leq target after loop break

e (ans) s

when loop breaks

s = m + 1

(2) \neq (3) ✓

return end; instead of -1

AFTER LOOP BREAKS

(e) target \neq (s)

(greatest number \leq target) floor.

ceiling (smallest number \geq target)

ceiling in an array \rightarrow

for the smallest number $x = \text{target}$

can be done by returning the start
pointing after the while loop breaks

while (start \leq end)

⑧ [15, 23, 48, 5] \rightarrow BREAKS
target = 5
when target \geq arr.length
return -1;
the array (index)

floor of a number in an array \rightarrow

for the greatest number $x = \text{target}$

return end pointing after the while loop
breaks.

while (start \leq end)

end = mid - 1
 \uparrow
0 - 1
 \uparrow
(-1)
(medium) \rightarrow (index)
 \rightarrow numerical
c \rightarrow s
 \uparrow
floor
 \uparrow
(root)
8 15 23

LEETCODE

744. ceiling of characters

exact algo for the ceiling of number.
ignore target(=)

S / N

5, 6, 8, 18, 21, 17

S, M, E, D, C, B, A

S, N

S, M, E, D, C, B, A

J, G, W, A, M, Y, T, E

J, G, W, A, M, Y, T, E

By, next will problem.

(Starting ending position of target)

two pointer method $\rightarrow O(n)$
 \rightarrow \leftarrow
 F, , F

find first occurrence of F:-

it might be possible that you might have
more targets either in the left
half or right half after
finding the target
element.

for this you have to apply
again to find max elements

so that we get starting and ending
index of target.

Wrt: 107 mins

Q4. array of nums find the smallest and largest end of target element

arr = [5, 7, 7, 7, 7, 8, 9, 10]
0 1 2 3 4 5 6 7

ans = [1, 7]

try come with bruteforce \rightarrow worst possible soln.

start search from first and from last
two pointer method

when you find \oplus

2 possibility

seen binary search twice

① 1st time to find start of \oplus
② 2nd time to find last instance of \oplus

$\log n + \log n$

~~$\log n$~~ ✓

\rightarrow find first occurrence of \oplus
simple binary search

5 7 7 7 7 7 8 9 10
s m e
0 0 0 0 0 0 0 0 0
1 2 3 4 5 6 7 8 9
↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

5, 7, 7, 7, 7, 7, 8, 9, 10

s m e

possible ans index = 3

1 2 3 4 5 6 7 8 9

when you find ~~when~~ there are more possibility than you might find behind it

WHEN ($\text{TARGET} == \text{A}[mid].\text{Num}[\text{mid}]$)

APPLY BINARY SEARCH

first occurrence on left side $s = \text{mn} + 1$

last occurrence on right side.

ans (3)

ans (2)

$e = m - 1$

Wows hoor, found about

\leftarrow found found

b found b

potential ans found.

start (true)
end (false)

function returns index value of target

int min, target, boolean

find start index

ans = mid. potential
(ans)

if (find start index)

start = not L

else

end = mid - 1

in LC we don't use static
but functions

class Solution {
 public int searchRange(int[] nums, int target) {
 // we have to consider what we are supposed to return
 // in this case (-1, -1) if element not found

→ int start = -1, -1;

int start = searchStart(int[] nums, int target, true);

int end = searchEnd(int[] nums, target, false);

int[]

ans[0] = start // since we only want start

ans[1] = end

return ans;

}

// function to search start and end indices
// of target element

int searchIndices(int[] nums, int target, bool isStart)

// check what is return structure

int possibleIndex = -1 // return -1

// now apply binary search

int start = 0

int end = nums.length - 1; // last index

int mid;

while (start <= end)

if (nums[mid] == target) {

possibleIndex = mid;

if (isStart) {

start = mid + 1;

} else {

end = mid - 1;

}

F: not found
T: found

} else if (nums[mid] > target) {

else if (target > numbers)

if $target > mid$ will lie in the second half of the array

$$start = mid + 1;$$

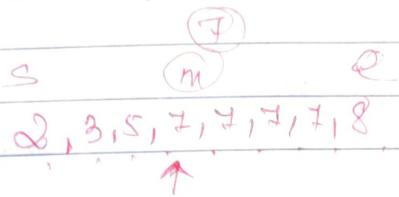
else

$$end = mid - 1;$$

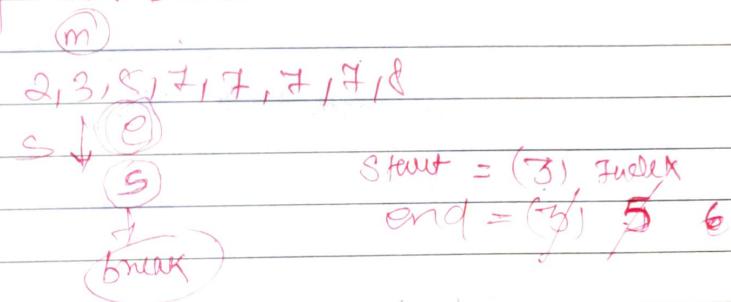
q

(2, 3, 5, 7, 7, 7, 7, 8.)

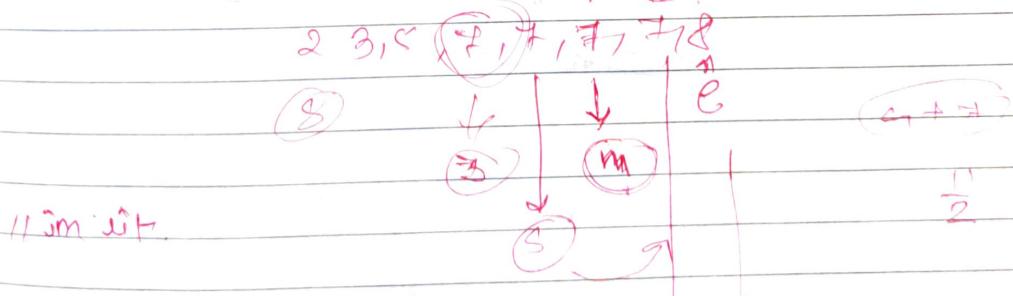
any run #1



when for function runs. //start.



2nd function runs. //end. start



//exit.

$$start = 3$$

$$end = 6$$

$$arr \rightarrow 2, 3, 6 \uparrow$$

(s)

(m)

(7)

(8)

start > end

while break

11/11/18 Today

We are not finding the amount of time we are finding a relation $\frac{1}{100} \rightarrow$ the size and time

$$65000 - 112000 \\ 0 \qquad \qquad 100.$$

$$\frac{72}{2}$$

$$250/10 = 65 \quad \frac{58}{4} \\ \frac{1}{4} \times 65 = 1600 \quad 3750 \quad 3000 \\ \text{cup} \quad \text{stack} \quad \text{bulb} \\ 6w = 15w = 45w$$

consonant

funny

stud y \rightarrow lies

lies

moves

PI@ y \rightarrow S

S

(going steady)

vowel

two fast actions,

(dating)

action

happened first

(NOT)

past perfect

you are cute (NOT)

simp

(duh) \rightarrow obvious

stupid its raining

Learn

phrases

bona fide \rightarrow genuine / truly
gentleman

vice versa \rightarrow

switch front/back

i can go to

you

harvard

\rightarrow alma mater \rightarrow college you

graduate.

Curbie dictu \rightarrow seize the day

status quo \rightarrow norm in society.
regular

common English Idioms -
about to

by the way \rightarrow transition, relate
in a way \rightarrow two sides -ive, +ive
on the other hand \rightarrow share two
different side

as a matter of fact = actually

continue

Phylos +

Dark Fantasy

Amazon Interview Experience.

Funny Pie.

search element in infinite array.

Deliciously Left

Great

bruteforce approach.

remove all elements from the start

How to say we can apply binary search.

If the array is sorted we can directly apply
binary search without hesitation

we use array.length

as we don't know the size start
and end.

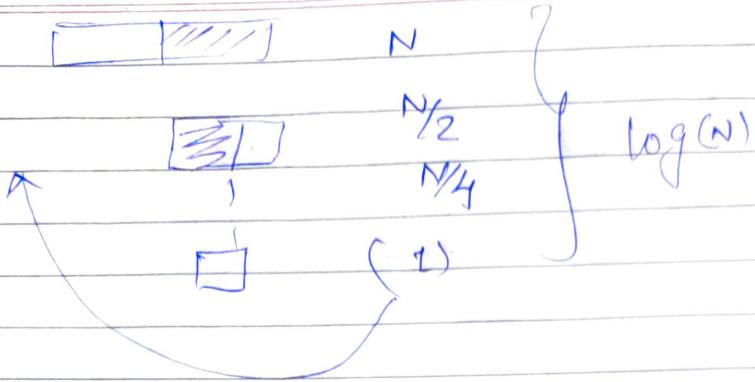
So what does the question becomes?

find start and end of target

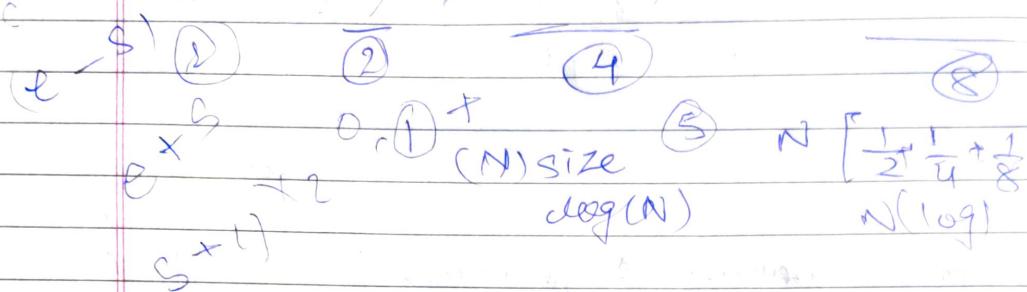
$S = S + 6$ $e = e + 6$

target lies b/w? now

\oplus length



h [2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13]
* 2 5 8 e s - - - - -



e¹ infinite Array.

while

(target > end)

downsizing

end + end

newStart = $\frac{\text{end} * 2}{2}$
= end + 1

new end = end + 2 * end

what?

box size

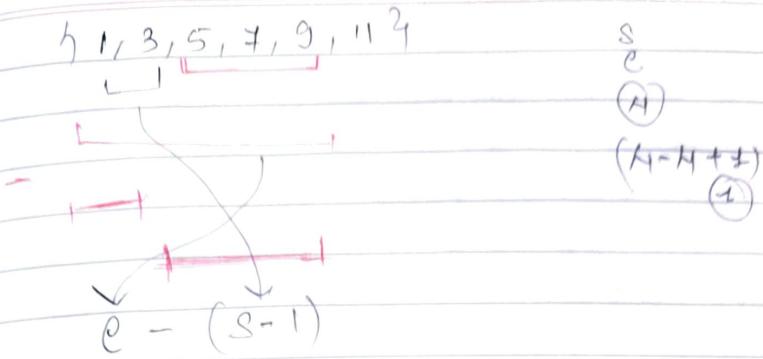
5 4, 13, 18, 16, 100, 200 }

0 1 2 3 4 5

↓ ↓

5 5 length box

c -



Q. FIND ELEMENT IN INFINITE ARRAY —

int element (int[] nums, int target, int start, int end)
 int ans = -1;

int index (int ans, f (int[] nums, int target))

int start = 0

int end = 1

while (target > end)

start =

int temp = start

end = end + (end - start + 1)

start = end + 1

return binarySearch (int[] nums, int target, int start, int end)

while (start <= end)

int mid = start + (end - start) / 2

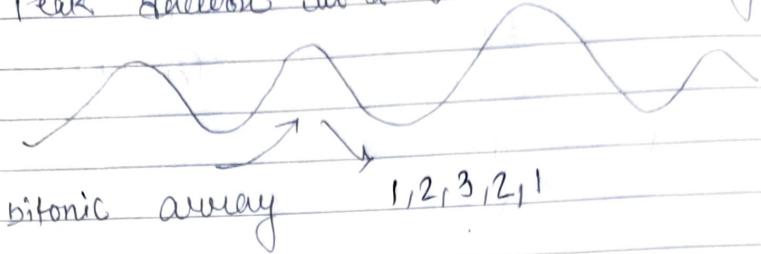
if mid + target == arr[mid]

else if (target > arr[mid]) return mid;

else
 $s = e + 1$;
 $e = mid - 1$;

852 (C)

Peak element in a mountain array



inc ↑ then decreases

→ 1, 2, 3, 4, 5, ↑ 6, 3

↓ 1, 2, 3, 5, ↑ 6, 3, 2, 4

↑

5
3
2
1
6
3
2

Q: Find peak in mountain array.

0, 2, ↑ 10

← →

how to do

linear search → Brute force

optimized Binary Search.

don't have duplicate

2, 5

if mid

by convention middle element should be smaller than mid + 1

if ie $(\text{num}[\text{mid}] > \text{num}[\text{mid} + 1])$

we can if $(\text{max} > \text{num}[\text{mid} + 1])$

$\text{max} = \text{num}[\text{mid} + 1]$

① if $e[\text{mid}] > e[\text{mid}+1]$
you are decreasing pair of array.

$$e = \text{mid}$$

$$s \boxed{m} e$$

ans lies in 1st half.

we don't have a target.

② $e[\text{mid}] < e[\text{mid}+1]$

ascending pair of array

greater num lies

2nd half of array.

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

for next element
nice work.

$$s \ m \ m+1 \ e$$

S

③ When will loop break

say num

1, 2, 3, 5, 6, 4, 3, 2

s \textcircled{m} $\textcircled{m+1}$ e

\textcircled{m} $\textcircled{m+1}$ e

\textcircled{m} $\textcircled{m+1}$

\textcircled{m} $\textcircled{m+1}$

while ($s == e$)

s & e both will point

to largest element

when
($s == e$)
return
num \boxed{SI}

EMail 1-15-15

CLASSMATE
Date _____
Page _____

Email Tricks

see:

1. Schedule Q → ~~cancel~~
2. confidential P → pass code
3. snooze → remainder
4. undo email → ⚡ bottom left corner

Unsolved

☒

↑

client
cancel

→ file to

300

5. signature → corporate
set it
automatically

6. vacation Remindley. →

7. labels → multiple responsibility
marketing
office
self.

8. CC / Bcc -

↓ ↓

copy blind carbon copy
copy

9. Text Formatting - Professional

A

10. (→ link) alternate - text
link - actual

11.



classmate

Date _____

Page _____



1. Canned response

→ Template

→ Advanced →

2. Canned reply, → check spelling grammar

and - 1st

form of

wing

(:)

canned reply

same draft

as template

3. Mail Merge →

1 email → (name) excel sheet)

50 people array. ↓

HR

candidate

Yet another mail merge

go to excel sheet → add on

google (salesforce

hh FirstName44

1

excel sheet

forget ticket.

hh PNR. No. 44

→ template

click or bounce

meetcode.io

150 questions



↓

Bind.

24/7/22

Interactive Problem

OOPS needed

classmate
Date _____
Page _____

YOU NEED INTUTION -

Q: $\text{arr} = [1, 2, 3, 4, 5, 3, 1]$

A: find peak element. (Element = 5)
index = 4

- search in one array

In Ternary

Leetcode hard 1092. additional use
intervall

find peak element

search in asc array
(0-h)

if not found binary search
in (l-h)

desc array

order agnostic binary search,

int search

int Peak

int Firsttry =

if (Firsttry != -1)

return Firsttry

else

search in second array

SEARCH IN ROTATED ARRAY

← (sorted) → find target.

distinct value.

2 methods

1) Elaborate

2) C/cisp. one line.

Explanation

$[2, 4, 5, 7, 8, 9, 10, 12]$

of by one notation (↓)

$[12, 2, 4, 5, \dots, 10]$

$[10, 12, 2, 4, \dots, 9]$

(1) FIND THE PIVOT IN THE ARRAY.



(largest element)

from where next number is
(else)

$\rightarrow [3, 4, 5, 6, 7, 0, 12, 3]$

thought process

every rotated array will be pivot.

★ find pivot

★ search in first half (start, pivot)

★ search in second half (pivot +, end)

Pivot - Central point where something turns or balances.

Now question becomes how to find (pivot)

0 1 2 3 4 5 6 7
[3, 4, 5, 6, 7, 0, 1, 2]
ASC ASC
pivot = ? desc

ANS WHEN?

when $a[i] > a[i+1]$

CASE 1:

if ($a[mid] > a[mid+1]$)
mid if pivot = $a[mid]$

CASE 2: what other way?

not minimized if $a[mid-1] < a[mid]$
try and now (only 1 call) pivot point
pivot. = $a[mid]$

CASE 3: if start element = mid element

In this case

all elements
from mid to i

be smaller than

start, we can ignore ~~smaller~~ smaller others

them by
($e = m - 1$)

so we can ignore these elements

how to ignore

end = $[mid] - 1$

CASE 4: start element < mid element

if $m = p$
then max element
would have been by case 1, 2
sorted

[3, 4, 5, 6, 1, 2]
m e
s = m

in this case
 $s = m + 1$

Hence found that bigger has lie ahead.

Hence, ignore mid & put $i = m + 1$

inc p inc

* worse case for duplicate elements in the array.

Playing with cases:

~~int findpivot(int arr)~~

~~int start = 0~~

~~int end = arr.length - 1;~~

~~while (start <= end)~~

~~mid = (start + end) / 2 // 4 cases.~~

~~if (arr[mid] > arr[mid + 1])~~

~~pivot = arr[mid] // return arr[mid]~~

~~if (arr[mid] < arr[mid + 1])~~

~~pivot = arr[mid]~~

~~if (arr[mid] < arr[mid - 1])~~

~~pivot = arr[mid - 1] //~~

~~return~~

~~return~~

int findPivot (int[] nums)

int start = 0;

int end = nums.length - 1;

while (start <= end) {

mid = (start + end) / 2;

MODIFY THIS

if (nums[mid] > nums[0])
return nums[0];

** As for mid+end
mid+1
will give underbound
error.

else if (nums[mid] < nums[mid-1])
return nums[mid-1];

so add a and if condition

else if (mid < end) if nums[mid] >
nums[mid+1]
return nums[mid];

condition a if condition b

if this is

false all statement won't be checked even
ans = false

$$\frac{N}{2}$$

for (int i = 0; i <= 0; i++)

$$\frac{N}{2^0}$$

$$\frac{N}{4}$$

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

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

$$\frac{N}{8}$$

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

$$\log_2 n = k$$

$$\frac{N}{2^2}$$

$$\frac{N}{2^k}$$

$$\frac{N}{2^k} \log_2 n$$

$$\frac{N}{2^3}$$

$$\log_2 n \cdot \log_2 n$$

$$\textcircled{1} \quad \boxed{1}$$

$$\boxed{1} \quad 1$$

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

$$N = 2^k$$

if $\text{array}[\text{mid}] \leq \text{array}[\text{start}]$

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

else

$\text{start} = \text{end} + 1;$

case 1

4, 5, 6, 7, 10, 1, 2
S P E

now to find ans using pivot

3 cases (array is sorted, chunk out, loop)

case 2 (pivot = target)

case 2. $\text{target} > \text{start}$

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

all numbers smaller than start

target lies in ~~first~~ left side of pivot

else if $\text{target} < \text{start}$

situation - thought

we know ~~all numbers~~ from the start to pivot

each will be greater than target

so

search space \rightarrow start = pivot + 1

(pivot)

7 8 2 3 4 5 S < mid

↑ -

P.

$\text{mid} \rightarrow \text{mid} + 1$

else

~~mid~~ $\rightarrow \text{mid} - 1$

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