

lecture - 1

11/12/2020.

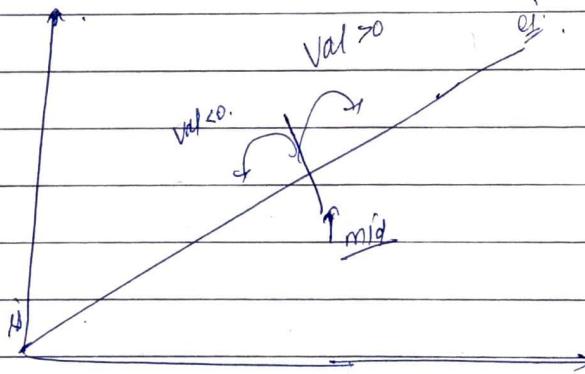
Agenda:

- ① Binary Search.
- ② Binary search first Index.
- ③ Binary search last Index.
- ④ B-S nearest Index / ceil and floor of any element.
- ⑤ 35 - search insert position.
- ⑥ ~~34~~ find first and last position of element in sorted array

Ques. Binary Search -

- ① array should be sorted for B-S to apply.

-10	-8	-7	-1	1	4	5	10	22	68	72	85	96	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13



Initially search area = N.

$$\frac{N}{2} \rightarrow = \frac{\frac{N}{2}}{2 \times 2} = \frac{N}{4}$$

$$= \frac{N}{4 \times 2} = \frac{N}{8}$$

$$N, \frac{N}{2}, \frac{N}{4}, \frac{N}{8}, \dots \dots \dots .$$

$$\frac{1}{2} \quad a_n = a_1, 2^{n-1} \quad \frac{1}{2}$$

$$\log_2 N = 1 + \log_2 (2)^{f-1}.$$

$$\log_2 N = (f-1) \log_2 2 = 1,$$

$$\boxed{f = \log_2 n}.$$

$$\boxed{f = O(\log n)}.$$

No 4 : This is very slow if $n = 10^{100}$.

$$\text{the } \log_{10} 10^{100} = 100. \text{ KCC} \approx 10^{100}$$

dry run:

return:

22 > 5	$\boxed{22 = 22}$
\downarrow	\uparrow
-10 -8 -7 -1 1 4 5 10 22 68 72 85 96 90	$\cancel{22 \neq 72}$
0 1 2 3 4 5 6 7 8 9 10 11 12 13	
	$\uparrow \quad \uparrow$

find 22

$$\frac{0+13}{2} = 6.$$

$$\frac{7+13}{2} = 10$$

$$\frac{7+10}{2} = 8.$$

// code -

```

public static int binarySearch(int arr[], int data){
    int si = 0, ei = arr.length - 1;
    while (si <= ei) {
        int mid = (si + ei) / 2;
        if (arr[mid] == data) return mid;
        else if (arr[mid] < data) ei = mid + 1;
        else si = mid - 1;
    }
    return -1;
}

```

(a) Binary Search first index of any value = val

2	4	5	5	5	5	6	6	6	7	8	8	8	8
0	1	2	3	4	5	6	7	8	9	10	11	12	13

$\uparrow f.i$ $\downarrow l.i$

val = 6

How - ?

simply run binary search and if $data = val$
then

shift $ei = mid - 1$

because we have to find first index

shift $si = mid + 1$ (if have to find last index)

// Code

```
public static int binarySearchEI(int[] arr, int data){  
    int si = 0, ei = arr.length - 1;  
    while (si <= ei) {  
        int mid = (si + ei) / 2;  
        if (arr[mid] == data) {  
            if (mid - 1 >= 0 && arr[mid - 1] == data)  
                ei = mid - 1;  
            else return mid;  
        }  
        else if (arr[mid] < data) si = mid + 1;  
        else ei = mid - 1;  
    }  
    return -1;  
}
```

// Code

for L.F

```
public static int binarySearchLFI(int[] arr, int data){  
    if (arr[mid] == data) {  
        if (mid + 1 < arr.length && arr[mid + 1]  
            == data) si = mid + 1;  
        else return mid;  
    }  
}
```

Q) Search first index and last index (34)

what.

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

tar = 8

$$f\cdot I = 3, \ L\cdot I = 4$$

[3, 4]

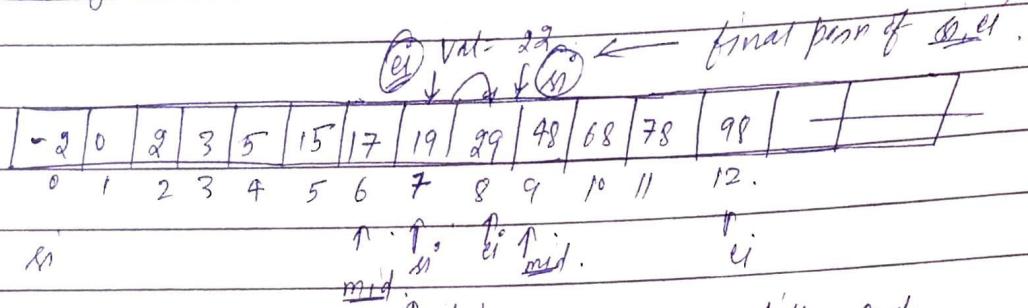
How :-

make arr of size 2 and call B.S.F.I & L.I.
Solved before

II code.

```
Public int[] searchRange( int[] arr, int data)
{
    int[] ans = new int[2];
    ans[0] = binarySearchFirstIndex(arr, data);
    ans[1] = lastIndex(arr, data);
    return ans;
}
```

④ Binary search . nearest Index (self an of (nearest)).



How :- Simply run ^{odd} Binary Search till end.
if data find then get return.

otherwise ei and si will be at

$$\begin{aligned} ei &= \text{floor } ? \\ si &= \text{ceil } . \end{aligned} \quad \text{automatically.}$$

ans will be $\min(\lceil \text{data} - \text{arr}[si] \rceil, \lceil \text{data} - \text{arr}[ei] \rceil)$

for nearest value

$$\text{mid} = \frac{0+12}{2} = 6.$$

$$\frac{7+12}{2} = 9.$$

$$\frac{7+8}{2} = 7.$$

Here Have to check some conditions.

- if ($\text{data} \leq \text{arr}[0]$) || ($\text{data} \geq \text{arr}[n-1]$).
then nearest will be
0, n-1 respectively.

val 70

1	2	3	4	5
---	---	---	---	---

ei

si

not possible so
check ceil floor

-10

1	2	3	4	5	6	7
---	---	---	---	---	---	---

si

si

not possible so check

ceil

i

j

floor data

II Code

```

public static int binarySearchNearestIndex( int[] arr, int data)
{
    if( arr.length == 0) return -1;
    int n = arr.length;
    if( data <= arr[0] || data >= arr[n-1])
        return data <= arr[0] ? 0 : n-1;
    int si = 0, ei = n-1;
    while( si <= ei)
    {
        int mid = (si+ei)/2;
        if( arr[mid] == data) return mid;
        else if( arr[mid] < data) si = mid+1;
        else si = mid-1;
    }
}

```

3.

if ei = floor, si = ceil

return ((data - arr[ei]) < arr[si] - data) ? ei : si;

9.

Ques. 35 - search sorted pumtion leet code

what given sorted array return posn of target if found
else return posn where it would be if inserted in order.

num = [1, 3, 5, 6] tar = 2

How

Here our answer index ranges from 0 to n.

so, mid can be our possible answers. so

we don't shift $ei \rightarrow mid-1$ instead

do $ei = \underline{mid}$.

→ at last si and ei will be at same position.

if $\textcircled{3}$

-10	-8	-2	0	10	20	30	40	59	66	78	88	123	258
0	1	2	3	4	5	6	7	8	9	10	11	12	13. 14
↑													↑

mid = perfect.

→ why include $\textcircled{14}$ because for 800
its position is $\textcircled{14}$ not $\textcircled{13}$

→ In above qn after $\textcircled{13}$ $\textcircled{14}$ but this should be
proper concept.

→ Using this we don't need extra condition to
check.

// Code

```
int searchInsert(vector<int>& arr, int target) {
    int si = 0, ei = arr.size();
    while (si < ei) {
        int mid = (ei + si) / 2;
        if (arr[mid] < target) si = mid + 1;
        else ei = mid;
    }
    return ei;
}
```

(lect-2.)Agenda:

- ① 74 - search in 2d matrix.
- ② Count inversion in an array (gfg)
- ③ 775 - global and local inversion
- ④ 658 - find k closest integers to b'

① Due Search in 2d matrix.What?

Write algorithm that searches for value in an $m \times n$ matrix. (sorted)

eg

1	3	5	7
10	11	16	20
23	30	34	60

target ③.

How? ① assume two d matrix as one d and to run binary search.

- ② Check valued at mid by converting linear value into 2-d by dividing columns
- ③ perform ① and ② till we get value

$$t: \Theta(\log mn) = \underline{\Theta(\log m + \log n)}$$

II. Code :-

	1	3	5	7	
.	10	11	16	20	<u>target = 80</u>
.	23	30	34	60	3x4

$$\rightarrow si = 0, ei = 12 - 1$$

$$mid = \frac{0+11}{2} = 5$$

$$\frac{5}{3} = 1.82$$

$$\rightarrow si = 0, ei = mid - 1 = 4$$

$$mid = \frac{0+4}{2} = 2 \Rightarrow 0.12$$

$$\rightarrow si = 0, ei = 1$$

$$\frac{0+1}{2} = 0.5$$

$$\rightarrow si = 1, ei = 1$$

$$mid = \frac{1+1}{2} = ①$$

and which is equal to target.

11. code - Public boolean searchMatrix (int [][] matrix, int target) {
if (matrix.length == 0 || matrix[0].length == 0)
return false;

int n = matrix.length, m = matrix[0].length;

int si = 0, ei = m * n - 1;

while (si <= ei) {

int mid = (si + ei) / 2

int val = matrix[mid / m][mid % m];

if (val == target) return true;

else if (val < target) si = mid + 1;

else ei = mid - 1;

}

return false;

}

Ques : Inversion Count in an array

what

Inversion count for an array. indicates how far the array is from being sorted.

e.g. if $a[i] > a[j]$ & $i \leq j$ then there will be an inversion.

How :-

- ① fact to fn is that left part will give sorted array and its inversion count
- ② same right part will do.
- ③ my task is to merge both part and count total inversion.
- ④ return sorted array and its total inversion count.

dry run

6	9	100	70	42	23	57	400	45	69	72	100
0	1	2	3	4	5	6	7	8	9	10	

(12)

$$\frac{6+11}{2} = 5$$

(6, 9, 23, 42, 100)

(5) 45, 51, 69, 72, 400, 708)

6	9	100	70	42	23
---	---	-----	----	----	----

51	400	45	69	72	708
----	-----	----	----	----	-----

X

6	9	100
---	---	-----

42	23
----	----

45	51	400
----	----	-----

69	72	708
----	----	-----

6	9
	100

42	23
----	----

51	400
	45

69	72
	708

(6, 9, 23, 42, 100)

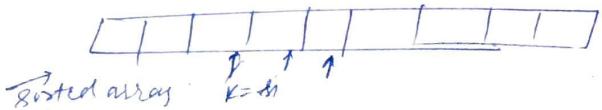
↑

(45, 51, 69, 72, 400, 708)

↓

(6, 9, 23, 42, 45, 51, 69, 72, 100, 400, 708).

total inversion count = 12 An



APCO

Date: / /

① static long totalInversionCount (long arr[], long[] sortedArr, int i, int mid, int ei) {

long count = 0;

int i = i, j = mid, k = ei; → k remains on
sorted array at
point to sorted element

while (i < mid && j <= ei) {

if (arr[i] <= arr[j]) {

sortedArr[k++] = arr[i];

} else {

sortedArr[k++] = arr[j++];

count += mid - i;

}

[while (i < mid) {

sortedArr[k++] = arr[i++];

}]

[while (j <= ei) {

sortedArr[k++] = arr[j++];

}]

while (i <= ei) {

arr[s] = sortedArr[s++];

return count;

};

②

static .

③

static

};

(2)

```
static long inversionCount( long[] arr, long[] sortedArr, int si, int ei ) {  
    if ( si > ei ) {
```

```
        return 0;
```

```
    int mid = ( si + ei ) / 2;
```

```
    long count = 0;
```

```
    count += inversionCount( arr, sortedArr, si, mid );
```

```
    count += inversionCount( arr, sortedArr, mid + 1, ei );
```

```
    count += totalInversionCount( arr, sortedArr, si, mid + 1, ei );
```

```
    return count;
```

```
}
```

(3)

```
static long inversionCount( long[] arr, long N ) {
```

```
    if ( N == 0 ) return 0;
```

```
    int n = (int) N;
```

```
    long[] sortedArr = new long[ n ];
```

```
    return inversionCount( arr, sortedArr, 0, n - 1 );
```

```
}
```

③ 775 Leet code.

global and local inversion.

what: if $A[i] > A[j]$ global inversion.

$A[i] > A[i+1] \Rightarrow$ local inversion.

Return true if count of local and global inversion are same else false.

How:- from given info. we see that local inversion will be global but global not will be local so. whenever global inversion occurs return false otherwise true.

dry = run:

[2, 0, 3, 1].

Maintain counter which ensure that value occurred at i th index or before should be less than $i+1$ th index.

If become greater any time then return false.

else return true.

// code -

```
public boolean isIdealPermutation(int[] arr) {  
    int cmax = 0;  
    for (int i = 0; i < A.length - 2; i++) {  
        cmax = max(cmax, arr[i]);  
        if (cmax > arr[i + 2]) return false;  
    }  
    return true;  
}
```

Ques 658 - find k -closest integer to x .

What

Given sorted array arr, two integers 'k' and 'x' return k closest element in array nearest to 'x'.

eg. $\text{arr} = [1, 2, 3, 4, 5]$. $k = 4$, $x = 3$.
 $\text{o/p} = [1, 2, 3, 4]$.

How :-

- ① find element in array using binary search.
- ② after finding set left pointer $\max(0, \text{id}_x - k)$.
and right pointer $\min(n-1, \text{id}_x + k)$.
- ③ now we have $2k+1$ size window
- ④ move left and right pointer by taking diff. with 'x' who ever diff is more
while (window is of $k+1$) size.

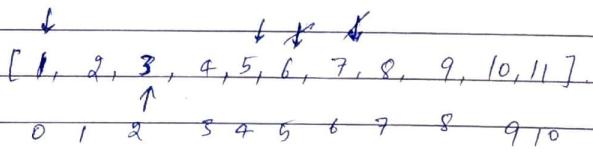
Dry Run

$$\text{arr} = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].$$

$$K = 4$$

$$x = 3$$

If



ans:

base cases - if $x = arr[0]$

then ' k ' closest will be from 0 to k .

if $x = arr[n-1]$

then ' k ' closest will be.

$(n-k, n)$:

// code:

```
public List<Integer> findClosestElements(int[] A, int k,
                                         int x) {
```

```
    List<Integer> arr = new ArrayList<>();
    for (int ele : A) arr.add(ele);
```

```
    int n = A.length;
```

```
    if ( $x \leq A[0]$ ) return arr.sublist(0, k);
    else if ( $x \geq A[n-1]$ ) return arr.sublist(n-k, n);
    else {
```

```
        int idx = binarySearch(A, x); // where we
```

```
// suppose to find the  $x$  element
```

```
        int si = max(0, idx-k);
```

```
        int ei = min(n-1, idx+k);
```

```
        while (ei - si > k) {
```

```
            if ( $x - A[si] > (A[ei] - x)$ ) si++;
            else ei--;
        }
```

```
}
```

```
        return arr.sublist(si, ei+1);
```

```
}
```

Set-3-

20/12/2020

Agenda -

- (1) 300 - Longest increasing subsequence.
- (2) 875 - koko eating banana.
- (3) cake distribution / man area serving task - leetcode.

Ques longest increasing subsequence.

what?

given an integer array 'nums'. return the length of longest strictly increasing subsequence.

$$I/P = [10, 9, 2, 5, 3, 7, 101, 18].$$

$$O/P = 4.$$

How :- Here we create an arraylist and store all subsequnces.

- This subseq didn't give correct subsequence order but give correct length.
- if perfect index of any element comes = size of list then add into list.
- otherwise replace this element into list at perfect index.

dry run

0	8	4	12	2	10	6	14	1	9	5	13	3	11	7	15
P	P	X	P	X	X	X	P	P	P	P	P	P	P	P	P

$$\ell = 3$$

list	10	0	8	12	14	13
	X	10	9			
	X	X	8			
	1	3				

boim by doing so we get largest length at last

// code

① Public int binarySearch(ArrayList < Integer > arr, int data) {
 int si = 0, ei = arr.length;
 while (si <= ei) {

int mid = (si + ei) / 2;

if (data > arr.get(mid)) si = mid + 1;
 else ei = mid;

}

return si;

}

Public int lengthOfLIS(int[] arr) {

if (arr.length <= 1) {

return arr.length;

int n = arr.length;

```
ArrayList<Integer> list = new ArrayList<>();
```

```
for (int ele : arr) {
```

```
    int idx = Collections.binarySearch(list, ele);
```

```
    if (idx < 0) idx = -idx - 1;
```

In default b.s of java. if. element not found
then that return. -(idx+1). ∴ so we

do first +ve to index and then subtract 1.
from it.

```
int idx = binarySearch(list, ele);
```

```
if (idx == list.size()) {
```

```
    list.add(ele);
```

```
list.set(idx, ele);
```

g.

```
return list.size();
```

3.

Ques 875 - koko eating banana.

what. There are n piles of banana. i^{th} pile has $p[i]$ of banana.

koko have ' H ' hours.

return min. integer ' K ' such that koko can eat all bananas in H hours.

Note: if pile has less than ' K ' bananas. She eat all of them and do not eat more than this during that hour.

How :-

since we have to find eating speed and it varies from.

$$\min = 1 \quad \text{to} \quad \max = arr[i].$$

so turn binary search on eating speed range and get min speed.

② → if at mid speed koko is able to eat then decrease speed for better min.
shift $ei = mid$. (not $mid-1$ X because answer last).

if not able to eat then.

$$si = mid + 1$$

// code :-

bool ispossible to Eat (vector<int>& piles, int eatingSpeed, int H) {
 int hours = 0;

for (int i = piles.size() - 1; i >= 0; i--) {

hours += ceil(piles[i] / (eatingSpeed * 10));

to convert into.

if (hours > H) return false; integer value, decimal value

}

return true; -

};

→ Import ~~sort~~ ^{if} binary search fn or it run b.s.
 with help of this fn.

int minEatingSpeed (vector<int>& piles, int H) {

// sort(piles.begin(), piles.end());

int n = piles.size();

int minSpeed = 1, maxSpeed = (int)1e9;

while (minSpeed < maxSpeed) {

int eatingSpeed = minSpeed + (maxSpeed - minSpeed) / 2;

if (ispossible to Eat(piles, eatingSpeed, H))

maxSpeed = eatingSpeed;

else minSpeed = eatingSpeed + 1;

};

return maxSpeed;

};

dryRun:

$\text{arr} =$	$\boxed{3 \mid 6 \mid 7 \mid 11}$	$H = 8$
	$\begin{matrix} 3 & 6 & 7 & 11 \\ 6 & 1 & 2 & 2 \end{matrix}$	$B = 8$
$\text{min speed} = 1$	$\begin{matrix} 3 & 6 & 7 & 11 \\ 1 & 2 & 2 & 3 \end{matrix}$	$\Rightarrow 10 \geq 8$ $\Rightarrow 8 \leq 8$ answer

$\text{max} = 11$

$\rightarrow \text{mid} = \frac{1+11}{2} = 6.$

S:

$\text{min speed} = 1, \text{ max} = \text{mid}$

$\text{mid} = \frac{1+6}{2} = 3 \quad \text{takes more than 8 hours.}$

$\rightarrow \text{So, } \text{min speed} = 3+1 = 4.$
 $\text{max speed} = 6.$

$\text{mid} = \frac{4+6}{2} = 5.$

\rightarrow since this is valid answer so.
 $\text{max} = \underline{\text{mid}}.$

and validate till it fails.

(B) Ques Max area serving cake.

What?

Array containing area of circular cake. Determine largest piece that can cut from cake such that every guest get a piece of cake with same area.
It is not possible to have a single piece having some part from one cake and some from another.

I/P :- [4, 3, 3]. guest = 3.
find area using $\pi \times^2$.

$$\text{Area} = [50.24, 28.26, 28.26].$$

$$\text{o/p} = 28.2643.$$

How:- We will use binary search on area.
min = 0.0.
max = max of all areas.

→ As we have accuracy here of precision of 80.
we can't increment / decrement our answer by 1
instead do with 10^{-4}

→ and make condition for jumping by 1 more power.
i.e. 10^{-5} .

50.24, 28.26, 28.26.

$$\text{mid} = \frac{0 + 50.24}{2} = 25.12$$

It's not capable & have to find max - so. take it to mid.
 $lo = \underline{\text{mid}}$.

$$\begin{aligned}\text{mid} &= \frac{25.12 + 50.24}{2} \\ &= \frac{75.36}{2} = \underline{37.68}\end{aligned}$$

not cuttable.

So, $hi = \underline{\text{mid}}$.

$hi = \underline{37.68}$.

Similarly do till find answer.

$$N = \frac{t + r + (r - v)e}{e}$$

$$= \frac{t + r(1 - e)}{e}$$

$$= \frac{t}{e} + r$$

$$\boxed{T: O(\log N)}$$

APCO

Date: / /

1) Code

① Public static void isPossibleToServe(double r[], double area, int k) {

 int count = 0;

 for (double ele : area) {

 count += ele / area;

 if (count >= k) return true;

}

return false;

2.

② Public static void getCoveringAreaCake(int r[], int k) {

 int n = radii.length;

 double r[] area = new double[n];

 double lo = 0.0, hi = 0.0, mid = 0.0;

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

 area[i] = radii[i] * radii[i] * Math.PI;

 hi = Math.max(hi, area[i]);

}

 while (Math.abs(hi - lo) > 1e-5) {

 mid = lo + (hi - lo) / 2.0;

 to make it double value

 if (isPossibleToServe(area, mid, k)) lo = mid;

 else hi = mid;

}

 return mid;

3.

lect - 4

21/12/2020

- ① 877 - ~~848~~ ^{Hintcast} Min-max distance to gas station.
- ② 33 - Search in sorted and rotated array.

Ques what

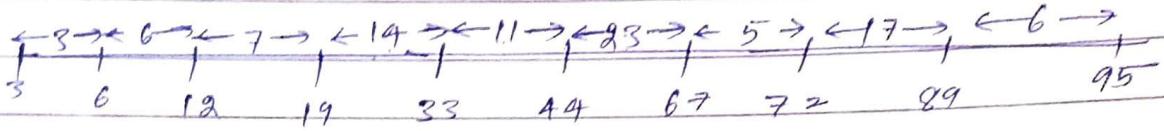
On a horizontal line we have gas station at positions. $\text{station}[0], \text{station}[1], \dots, \text{station}[N-1]$
 $N = \text{station length}$.

Now we can add ' k ' more gas station so that D ,
max distance b/w. adjacent gas station is minimized.
return smallest 'possible' value of D .

How :- Algorithm

- { ① do same as done in cake problem. if
we are able to set up more than k petrol pump.
then need to ↑ distance so.
set low = mid.
- ② if ~~it's~~ k petrol pump is equal or less
than k then decrease the distance to
bring and minimize max distance.

dry run



$$\rightarrow l_0 = 0.0, \quad k = 2.$$

$$h_i = 23.$$

$$mid = \frac{0 + 23}{2} = 11.50.$$

$$k = 3 > 2.$$

80 minimize k 80 maximize distance.

→ 2nd iteration

$$l_0 = 11.50.$$

$$h_i = 23.$$

$$mid = \frac{11.50 + 23}{2} = 17.25$$

$$k = 1 < 2.$$

→ 80, minimize dist.

$$l_0 = 11.25$$

$$h_i = 17.25$$

$$mid = \frac{11.25 + 17.25}{2} = 14.37$$

$$k(2 = 2)$$

80, now minimize dist till 0.00000 digit decimal.
by shifting $h_i = \underline{mid}$.

$$l_0 = 11.25$$

$$h_i = 14.375$$

// code.

①

```
public boolean checkIsValid(int[] arr, double mid, int k){  
    int count = 0;  
    for(int i=1; i<arr.length; i++){  
        count += (int)(arr[i]-arr[i-1])/mid);  
        if(count > k) return true;  
    }  
    return false;  
}
```

②

```
public double minmaxBiasDist(int[] arr, int k){  
    double lo = 0.0, hi = 1e9, mid = 0.0;  
    while((hi-lo) > 1e-5){  
        mid = (lo + hi)/2.0;  
        if(checkIsValid(arr, mid, k)) low = mid + 1e-6;  
        else hi = mid;  
    }  
    return hi;  
}
```

Q. 2.

33/ Search in Rotated sorted array.

what you are given a integer sorted array, and it is rotated at some unknown pivot point. given a target find target in rotated array.

eg. $\text{num8} = [4, 5, 6, 7, 0, 1, 2]$. target = 0
 $o/p = 4$.

How :- ① $O(n)$ solution get submitted. but

M-② $O(\log n)$.

→ first of all check whether region is sorted or not between (lo and mid).

if sorted check data lies in given region.
 if lies then $A[i] = \text{mid}$.
 else $\text{low} = \text{mid} + 1$.

else

if ($\text{data}^{\text{mid}} < \text{data} < h_i$) $\text{low} = \text{mid} + 1$;
 else $h_i = \text{mid} - 1$;

Dry Run

4	5	6	7	-2	-1	0	1	2	3	3.5
0	1	2	3	4	5	6	7	8	9	10

①

$l = 0$

$hi = 10$

$mid = \frac{0+10}{2} = 5$

$target = -2$

②

$l = 0$

$hi = 4$

$mid = \frac{0+4}{2} = 2$

$lo = 4$

$hi = 4$

$mid = 4$

and $arr(mid) = data$
return data - Index.

③

$lo = 3$

$hi = 4$

$mid = \frac{3+4}{2} = 3$

II Code:

```

public int search(int[] arr, int data) {
    int lo = 0, hi = arr.length - 1;
    while (lo <= hi) {
        int mid = (lo + hi) / 2;
        if (arr[mid] == data) return mid;
        else if (arr[lo] <= arr[mid]) {
            if (arr[lo] <= data && data <= arr[mid]) {
                hi = mid - 1;
            } else lo = mid + 1;
        } else {
            if (arr[mid] < data && data <= arr[hi]) {
                lo = mid + 1;
            } else hi = mid - 1;
        }
    }
}

```

lecture-524/12/2020-

Agenda:-

- ① 81 | search in sorted array 2.] binary search
- ② 153 | find in sorted rotated array.]
- ③ 001 | Two sum.
- ④ 167 | Two sum 2. } (normal array question)
- ⑤ 15 | 3 sum.
- ⑥ 18 | 4 sum.

① LeetCode | Search in Rotated sorted array II.

what? Suppose an array sorted in ascending order is rotated at some pivot unknown to us. You are given a target value to search. Return true if found else return false.

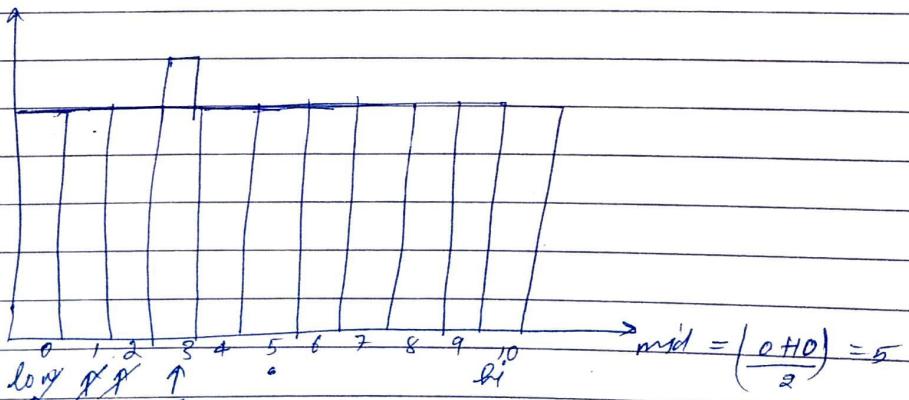
Note here duplicacy of data is allowed.

e.g. I/P = 2 5 6 0 0 1 2 . target = 0.

How? → do here the same thing as done previously always go into sorted region and shift mid by, and low accordingly.

② since here duplcy is allowed so whenever not able to decide region then either $i/(i+e)$ or $(+ei)$ by one and then take decision for si, ei and mid.

e.g.



can be able to take decision.

// Code

```
Public boolean search( int[] arr, int data) {
    int lo = 0, hi = arr.length - 1;
    while( lo <= hi) {
        int mid = (hi+lo)/2;
        if( arr[mid] == data || arr[lo] == data) return true;
        else if( arr[lo] < arr[mid]) {
            if( arr[lo] <= data && data < arr[mid])
                hi = mid-1;
            else
                lo = mid+1;
        }
        else if( arr[mid] < arr[hi]) {
            if( arr[mid] < data && data <= arr[hi])
                lo = mid+1;
            else
                hi = mid-1;
        }
        else
            lo++;
    }
    return false;
}
```

Q. Qn

153 (Find Minimum in rotated sorted Array).

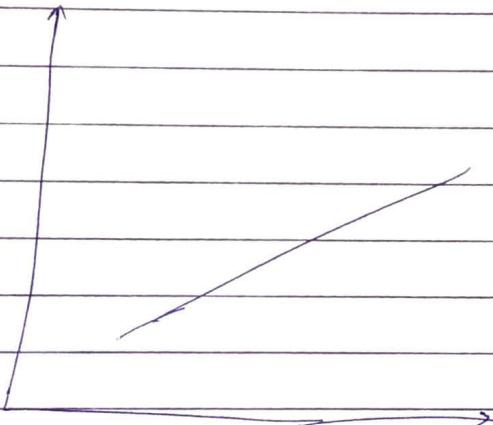
eg.

Nums = [3, 4, 5, 1, 2].

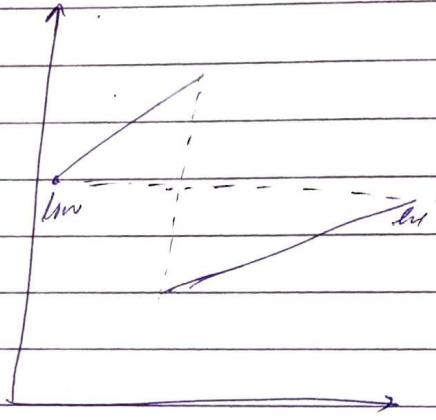
O/P = 1.

how:

First of all define region where ans can be found and then filter out area where not possibility to get answer.



Initially array like this.



after rotation.

- ① here mid can lies in rotated part or same part depending on mid our area will shrink and we can reach to our location.

11 Code

```
Public int findMin(int[] arr) {  
    int lo = 0, hi = arr.length - 1;  
    if (arr[lo] <= arr[hi]) return arr[lo];  
  
    while (lo < hi) {  
        int mid = (lo + hi) / 2;  
        if (arr[mid] < arr[hi]) = hi = mid;  
        else if (arr[lo] <= arr[mid]) lo = mid + 1;  
    }  
    return lo;  
}
```

M-2

we can also do by making condition.

if (arr[mid] > arr[hi]) low = mid + 1

else if (arr[lo] > arr[mid]) hi = mid;

else too hi--

does the same as prev but it check cross condition
so pointers never leave the answer element.

⑥ One / Two Sum -

what given array of integers and target return indices of two numbers such that they add up to target.

eg nums = [2, 7, 11, 15] target = 9
o/p = [0, 1].

How :-

As we have to find index so (index will be value in hm) and element will be key.

like:-

(
 |
 |
 |
 |
key value
2 0
7 1
11 2
15 3)

arr[target - el]

check always

map contains(target - el)
or not

11. Code

```
public int twoSum(int[] nums, int target){  
    HashMap<Integer, Integer> map = new HashMap<>();  
    for (int i = 0; i < nums.length; i++) {  
        int ele = nums[i];  
        if (map.containsKey(target - ele)) return  
            new int[] { map.get(target - ele), i };  
        map.put(ele, i);  
    }  
    return new int[] {-1, -1};  
}
```

Ques / Two sum 2.

What? we are given array of integer that is already sorted in ascending order.

find indices of two element such that they add to give target.

How:-

- ① keep two pointers $s_i = 0$ and $e_i = \text{length} - 1$
- ② if ($\text{sum} = \text{data}$). return s_i, e_i
else if ($\text{sum} < \text{data}$) s_i++ ;
else e_i-- ;

eg

9	7	11	15
0	1	2	3
↑	X	X	↓

$$s_i = 0, \quad \bullet$$

$$e_i = 3.$$

$$\text{target} = 9$$

$$\text{sum} = 17 > 9.$$

Code.

```
public int[] twoSum(int[] arr, int data) {
    int s_i = 0, e_i = arr.length - 1;
    while (s_i < e_i) {
        int sum = arr[s_i] + arr[e_i];
        if (sum == data) return new int[]{s_i, e_i};
        else if (sum < data) s_i++;
        else e_i--;
    }
    return new int[]{-1, -1};
}
```

(5) Que. 15/3 sum -

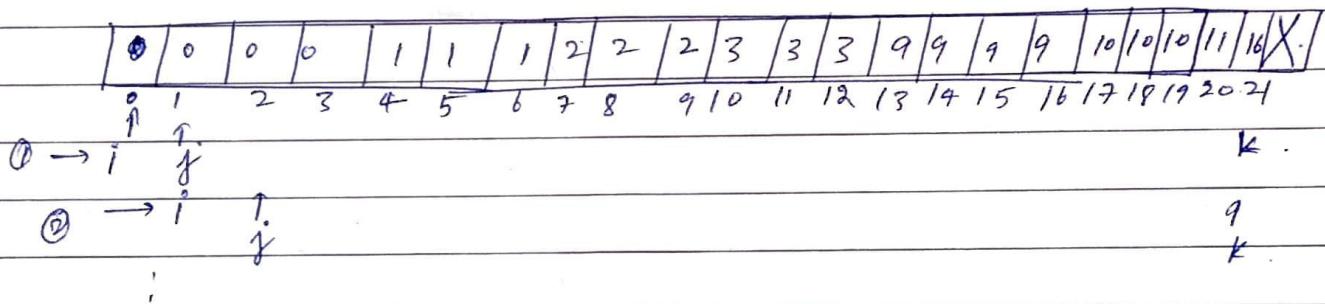
What? given array of int sums' and have to find triplet $a+b+c = 0$.
Find all unique triplet.

How? ?

- fix a index and apply two pointer form ($i+1$ and $len-1$).
- with some modification.

Dry Run

t:



80, on + $n \log n$ sorting complexity
 $t: O(n^2)$ }
 $s: O(1)$ }

Code

```
public List<List<Integer>> threeSum (int[] arr) {
```

```
    Arrays.sort (arr);
```

```
    int n = arr.length;
```

```
    int data = 0;
```

```
    List<List<Integer>> res = new ArrayList();
```

```
    for (int i = 0; i < n; i++) {
```

(skip
duplicates)

```
        → { while (i + 1 == 0 && i < n && arr[i] == arr[i + 1]) i++; }
```

```
        int j = i + 1, k = n - 1;
```

```
        while (j < k) {
```

```
            int sum = arr[i] + arr[j] + arr[k];
```

```
            if (sum == data) {
```

```
                // List<T> ar = new ArrayList();
```

```
                // ar.add(arr[i]);
```

```
                // ar.add(arr[j]);
```

```
                // ar.add(arr[k]);
```

```
                if (res.add(ar));
```

```
            or res.add(Arrays.asList (arr[i], arr[j], arr[k]));
```

```
            j++;
```

```
            k--;
```

(to skip duplicates)

```
        } while (j < k && arr[j] == arr[j + 1]) j++;
```

```
        } while (j < k && arr[k] == arr[k + 1]) k--;
```

}

```
        else if (sum < data) {
```

```
            j++;
```

```
        } else k--;
```

}

} // for loop

}

return res;

Agenda

lect - 6

26/12/2020

- ① 18 | 4 sum
- ② 454 | 4 sum count → using, HM & array.
- ③ 240 | search Matrix (from bottom left & top right).
- ④ binary search element (if -1 is inserted twice in b/w two elements)

Ques

4 sum :

what ?

given ^{an} array of ~~nunm~~ numbers of integer and an integer 'target'. Are there elements a, b, c, d in num8 such that they add to get target.
 $a+b+c+d = \underline{\text{target}}$