Binary Search Normal + First Index + Last Index

Tuesday, 9 November 2021 10:34 PM

3 Problems.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
array → (sorted)	10	11	13	15	15	15	16	19	20	23	23	24	29	30

↑

Normal Binary Search. data (dtf) = 13, to find

lo = 0

hi = 13

indx = -1;

first Index → if data is found, then again try to find in left side.

indx = mid;

hi = mid - 1;

→ [→ Don't break]

```
while(lo <= hi) {
    mid = (lo + hi) / 2
    if( arr[mid] == dtf){
        indx = mid;
        break;
    } else if (arr[mid] > dtf) {
        // Discard right side from mid
        hi = mid-1;
    } else {
        // Discard left side from mid
        lo = mid+1
    }
}
```

Last Index → If data is found try to search in right side.

indx = mid;

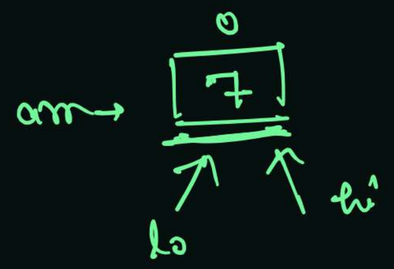
lo = mid+1

→ [Don't break]

First Transition

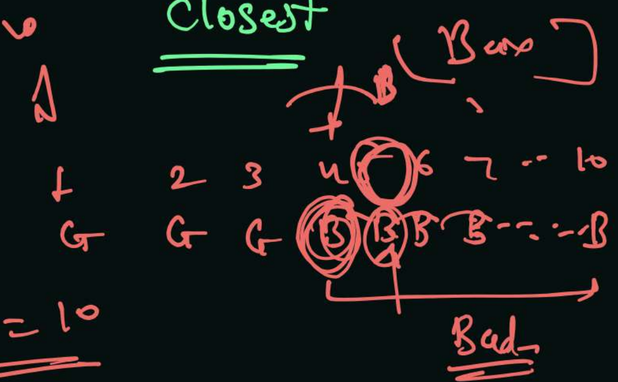
Tuesday, 9 November 2021 10:34 PM

Sorted array of 0's and 1's, find first (1).



diff = 7

mid = hi



First Bad Version } n = 10

while (lo < hi) {

}

return -1;

Why? R_1 Subtraction R_2

$$mid = lo + (hi - lo)/2$$

$$= lo + \frac{hi}{2} - \frac{lo}{2}$$

$$= \frac{2lo + hi - lo}{2}$$

$$= \frac{lo + hi}{2}$$

$R_1 + R_2 <$ Int_max

cross the Range of Integer

Leetcode

Guess the number

Saturday, 13 November 2021 8:33 PM

lower range = 1

upper range = n.

guess

Hint → +1 → pick > num.

-1 → pick < num

0 → pick == num

lo = 1

hi = n.

mid = lo + (hi - lo) / 2

```
if( guess(mid)==0){
    return mid;
} else if( guess(mid)>0){
    lo=mid+1;
} else {
    hi=mid-1;
}
```

K Closest Element

Saturday, 13 November 2021 8:41 PM

data = 45

	0	1	2	3	4	5
arr →	10	20	30	40	46	60
	lo	lo	hi	hi	hi	

lo = ~~0~~ 3 3
hi = ~~8~~ mid = ~~2~~ 4

out of loop

mid = ~~2~~ 4

go left

3

mindistance

go right

maintain

```
mid = lo + (hi-lo)/2;
// Everytime maintain gap.
if(arr[mid] == data){
    c;
}else if(arr[mid] > data){
    hi = mid-1;
}else{
    lo = mid+1
}
```

while(lo <= hi)

Step · ① Try to reach at closest point.

② put lo = mid-1.

hi = mid

k = 3

③ Sort the result

30	40	50

result

30 vs. 40 → 40

30 vs 50 → 50

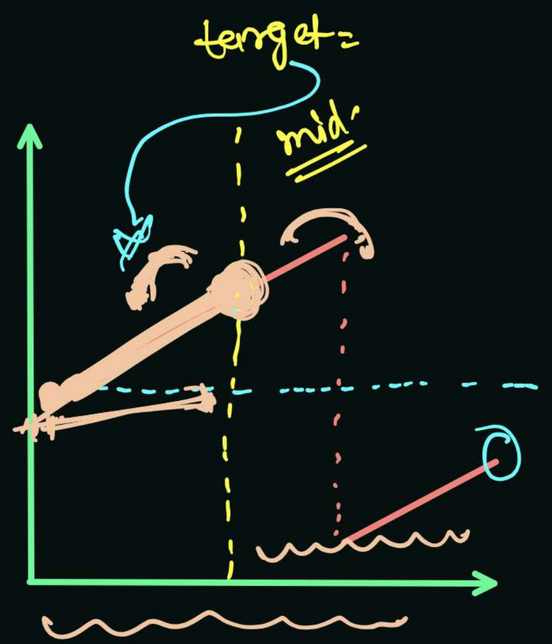
30 vs 60 → 30

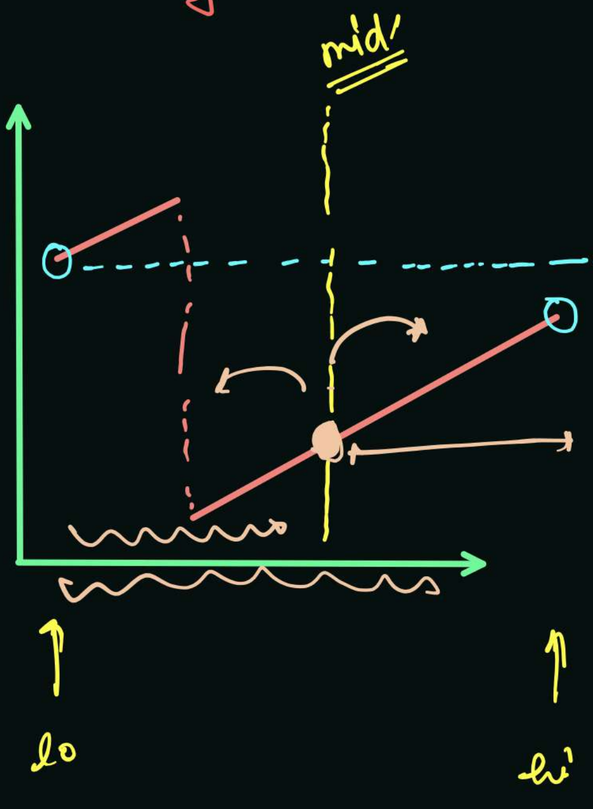
40 50 30 → sort

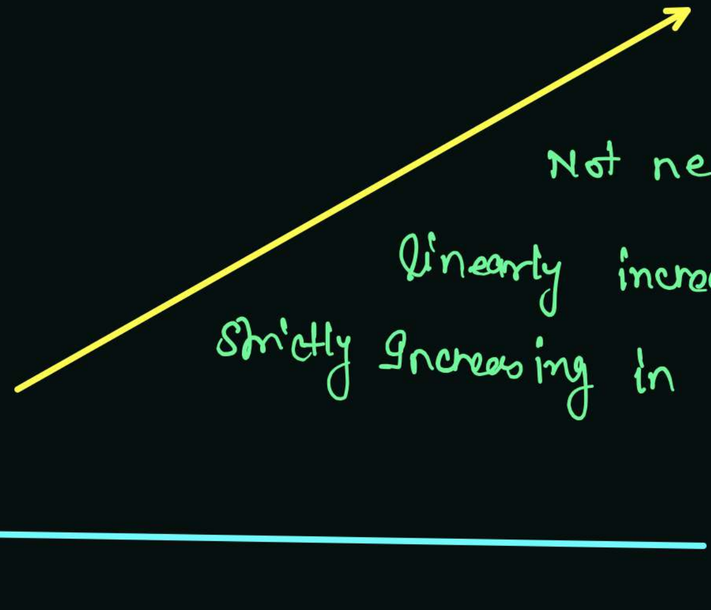
Search in rotated sorted array

Tuesday, 9 November 2021 10:34 PM

Distinct values in array

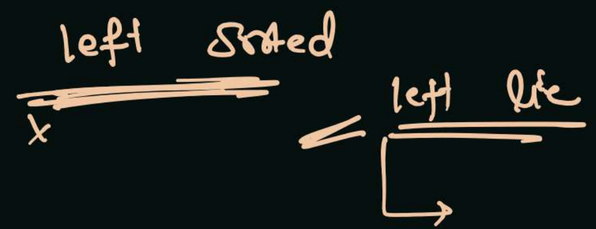


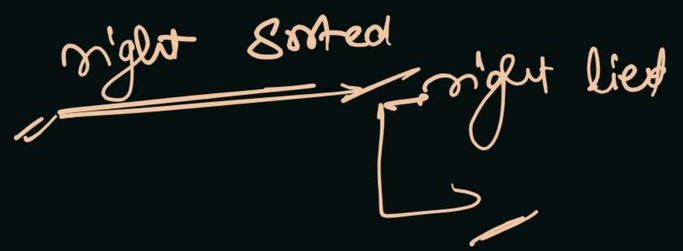




Not necessarily linearly increasing, but strictly increasing in order.

How to verify sorted part

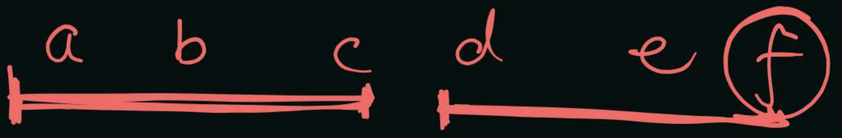


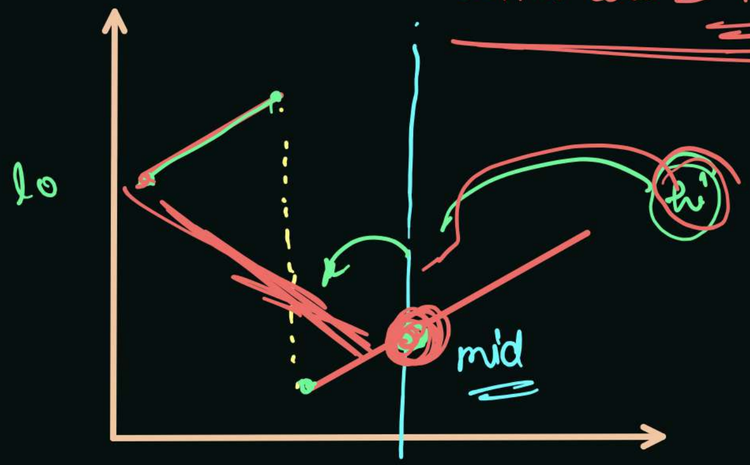


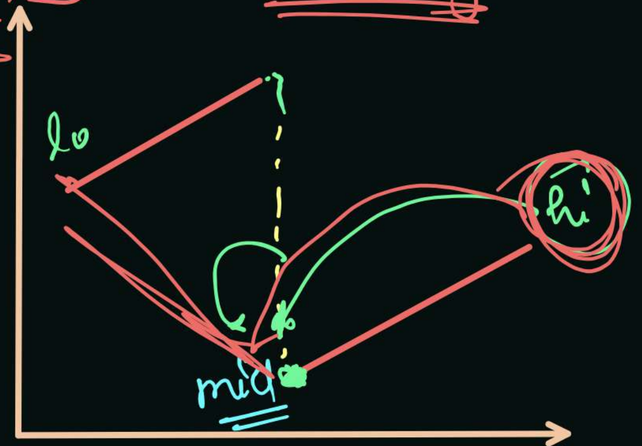
Min in Sorted rotated array

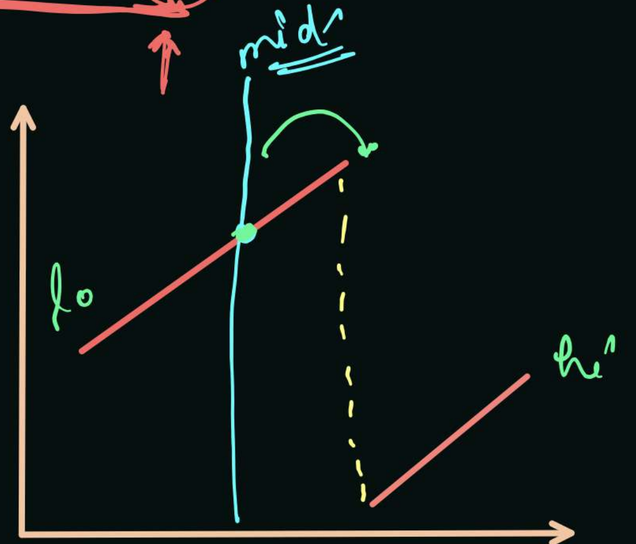
Saturday, 13 November 2021 10:58 PM

why not work with low - mid → Tuesday









case - I

arr[mid] < arr[hi]

TRUE

left side

case - II

arr[mi] < arr[hi]

TRUE

including (left side, mid)

case - III

arr[mid] < arr[hi]

FALSE

right side

```java
public static int findMinimum(int[]arr) {
    int indx = 0;
    int lo = 0;
    int hi = arr.length - 1;

    while(lo < hi) {
        int mid = lo + (hi - lo) / 2;
        if(arr[mid] < arr[hi]) {
            indx = mid;
            hi = mid;
        } else {
            lo = mid + 1;
        }
    }
    return arr[indx];
}
```

indx = 0

lo = 0 1

hi = 1

mid = 0