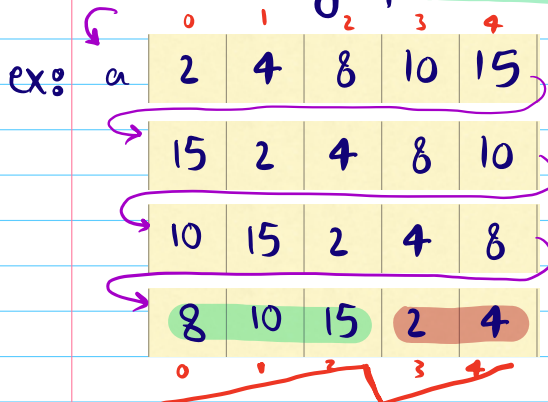


- Topics:
- 1- Search in rotated sorted array
  - 2- Median of two sorted array
  - 3- Sqrt of an int

P1 Find the index of a given element in a rotated

sorted array of distinct elements.

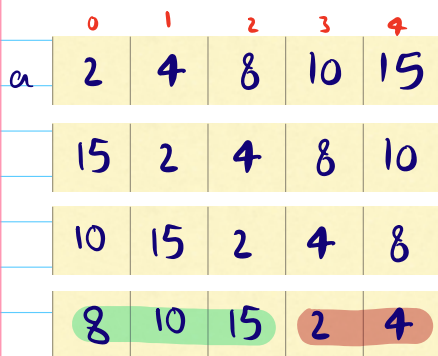
what is rotated sorted array?



⊕ # of rotation is not given

⊕ index in the rotated array not the original sorted array

how do we know if the input rotated?



$$O(1): a[0] \leq a[n-1]$$

largest element is a rotated sorted array can be found with  $O(\log_2 n) \rightarrow$  last session

ideas

Find the largest element  $\rightarrow \log_2 n$

10

4

10	15	2	4	8
----	----	---	---	---

edge



binSearch

$$O(\log_2 n)$$

Solution 1

Pseudo code

Find largest element

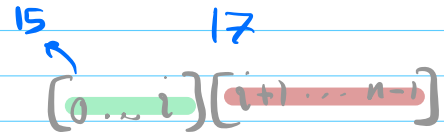
```

l = 0  r = n - 1
while (l <= r) {
    mid = (l + r) / 2
    if (a[mid] > a[mid + 1]) ret mid
    if (a[mid] < a[0]) r = mid - 1
    else l = mid + 1
}

```

edge check this with a short circuit → assignment

find index of largest element. this is not the final target.



get max element index

if (target < a[0]) → target is in red part. binSearch in red part

TC:  $O(\log(n))$   
SC:  $O(1)$

else → target is in green part. binSearch in green part

Solution 2

example

a[]	0	1	2	3	4	5	6	7	8
	50	60	100	3	9	10	25	30	35

l = 0 r = n - 1

```

while (l <= r) {
    mid = (l + r) / 2
    if (a[mid] == target) ret mid

```

25

60? ← target



TC:  $O(\log(n))$

SC:  $O(1)$

```

if (target < a[0]) {
    if (a[mid] >= a[0]) l = mid + 1
    else if (a[mid] < target) l = mid + 1
    else r = mid - 1
}

```

50, 60, 100, 200, 3, 9, 10

} else {

```

if (a[mid] < a[0]) r = mid - 1

```

```

else if (a[mid] < target) l = mid + 1

```

```

else r = mid - 1
}
}

```

P2 There are two sorted arrays, distinct.

A w/ size  $m$  and B w/ size  $n$ . find the median of two sorted arrays (i.e. the median of merge A & B sorted)

what is median of a array?

center index  
element

0	1	2	3	4
3	7	11	12	20

0	1	2	3
11	14	15	20

$$\frac{14+15}{2} = 14.5$$

ex1 A { 3, 7, 10 }

B { 1, 2, 4 }

{ 1, 2, 3, 4, 7, 10 }

ans = 3.5

ex2 A { 6, 7 }

B { 1, 2, 10 }

{ 1, 2, 6, 7, 10 }

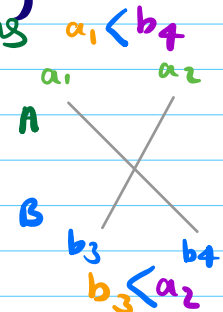
6 = ans

$O(m+n)$

merge

$O(m+n)$  : TC  
 $O(m+n)$  : SC

	0	1	2	3	4	m
A	3	7	11	12	20	
	0	1	2	3	4	5
B	1	2	4	6	8	13

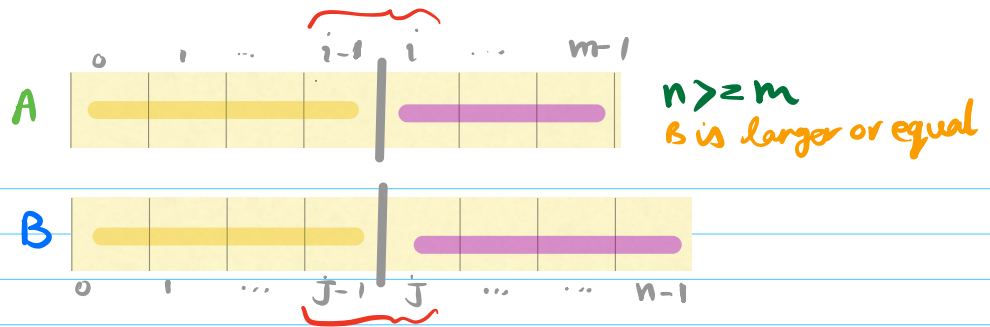


A, B merge and sorted	0	1	2	3	4	5	6	7	8	9	10
	1	2	3	4	6	7	8	11	12	13	20

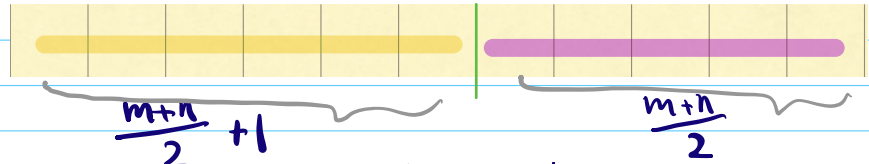
$$\# \text{ yellow } \leq \# \text{ purple } + 1$$

focus on  
odd size  
 $m+n$

$m+n$  is odd



A, B merge and sorted



$$i + j = \frac{m+n}{2} + 1$$

$$j = \frac{m+n}{2} + 1 - i$$

$$\frac{5+6}{2} + 1 = 2$$

$$l = 0 \quad r = m-1$$

$$a[0] \dots a[i-1] \mid a[i] \dots a[m-1]$$

$$b[0] \dots b[j-1] \mid b[j] \dots b[n-1]$$

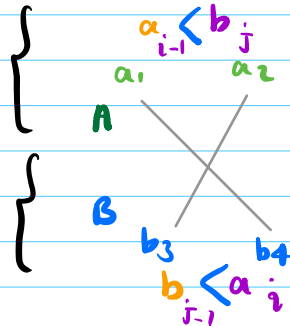


Diagram illustrating two sets of data, A and B, and their corresponding indices.

**Left Set:**

	0	1	2	3	4	m	
A	3	7	11	12	20		
	0	1	2	3	4	5	n
B	1	2	4	6	8	13	

**Right Set:**

	0	1	2	3	4	m	
A	3	5	6	7	20		
	0	1	2	3	4	5	n
B	1	8	9	10	11	13	

**Comparison:**

1	3	5	6	7	8	9	10	11	13	20
0	1	2	3	4	5	6	7	8	9	10

TC:  $O(\log n)$   
 SC:  $O(1)$

$l=0$   $r=m-1$

while ( $l \leq r$ )  
 $mid = (l+r)/2$

$i = mid$ ;  $j = (m+n)/2 + 1 - i$

if ( $(j \geq 0) \parallel (i \geq m) \parallel b[j-1] > a[i]$ )  
 $l = i + 1$

else if ( $(i \geq 0) \parallel (j \geq n) \parallel a[i-1] > b[j]$ )  
 $r = i - 1$

else if ( $((j \geq 0) \parallel (i \geq m) \parallel a[i] \geq b[j-1]) \&\&$   
 $- ((i \geq 0) \parallel (j \geq n) \parallel b[j] \geq a[i-1])$ ) {

if ( $(m+n)/2 \neq 0$  // odd size

if ( $i \geq 0$ ) ret  $b[j-1]$

if ( $j \geq 0$ ) ret  $a[i-1]$

ret  $\max(a[i-1], b[j-1])$

else // even size

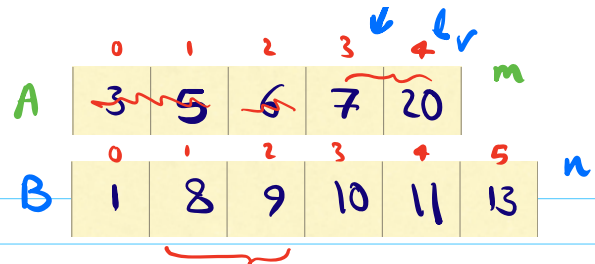
if  $i \geq 0$  ret  $(b[j-1] + \min(a[i], b[j])) / 2.0$

if  $j \geq 0$  ret  $(\max(a[i-1], b[j-1]) + \min(a[i], b[j])) / 2.0$

if  $i \geq m$  ret  $(\max(a[i-1], b[j-1]) + b[j]) / 2.0$

if  $j \geq n$  ret  $(\max(a[i-1], b[j-1]) + a[i]) / 2.0$

ret  $(\max(a[i-1], b[j-1]) + \min(a[i], b[j])) / 2.0$



P3 Find <sup>floor</sup> square root of an int without built-in functions.

$n=10$        $3^2 < 10 < 4^2$       ans = 3      *don't use math.Sqrt(...) Sqrt(...)*  
 $n=30$        $5^2 < 30 < 6^2$       ans = 5  
 $n=49$        $7^2 < 49 < 8^2$       ans = 7

$i \leq \sqrt{n}$

$5 \times 5 < 10$        $6 \times 6 > 10$

TC:  $O(\sqrt{n})$   
 for( $i=0$ ;  $i \leq n$ ;  $i++$ ) {  
     if( $i*i == n$ ) ret i  
     if( $i*i > n$ ) ret i-1  
 }

$O(\log n)$

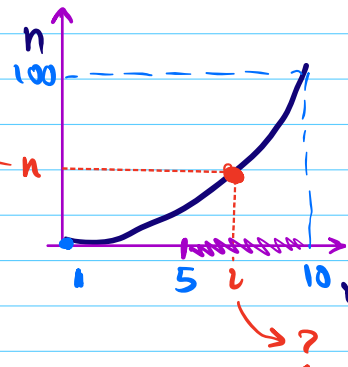
$n=10$   
 $n \geq 1$

$$1 \leq \sqrt{n} \leq 10$$

known

$i \leq \sqrt{n}$

$$n \geq i^2$$



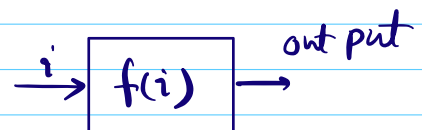
$l=1$        $r=n$   
 while( $l \leq r$ ) {

$mid = (l+r)/2$

    if(  $mid * mid \leq n$  &&  $(mid+1)*(mid+1) > n$  )  
         ret mid

    if(  $mid * mid > n$  ) → go left

    else → go right



bin search answer