Ques Search in a rotated sorted array.

distinct elements.

→ 2, 4, 8, 10, 15

k = 1, 15, 2, 4, 8, 10

k = 2, 10, 15, 2, 4, 8 , x = 2 .

rotated

→ 2, 4, 8, 10, 15

unrotated .

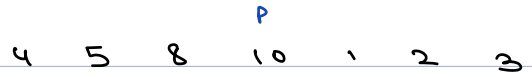
if array is rotated

if (A[0] > A[n-1])

yes

else {

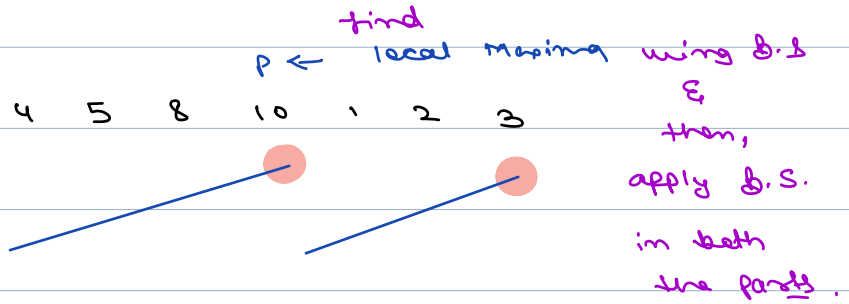
Binary search, ——→ 1 no



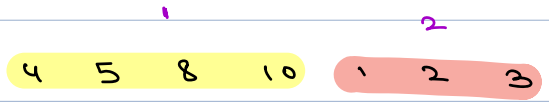
if largest element idx is given,

apply B.S. (0, p) & (p+1 to n-1).

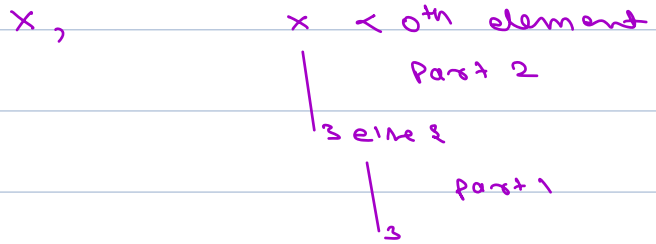
Twist:- if largest element idx is not given.



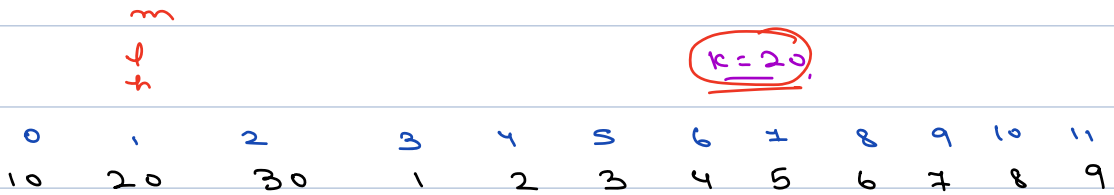
Twist:- Do it in 1 B.S.



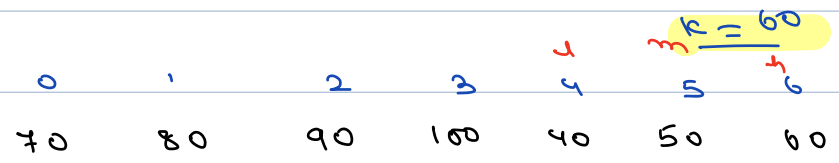
Part 1 > Part 2



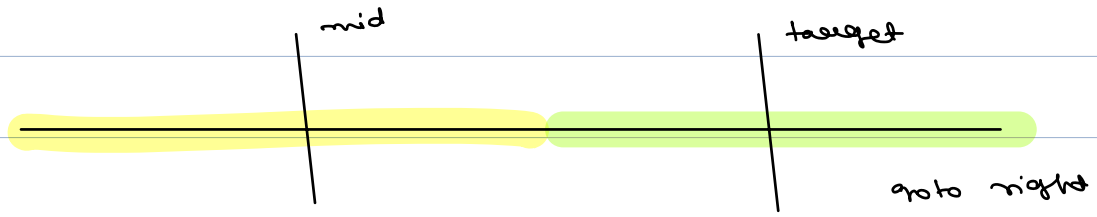
Get mid find in which part our middle is and in which part our target, is if both are in different parts, will move mid towards the target else apply normal binary research

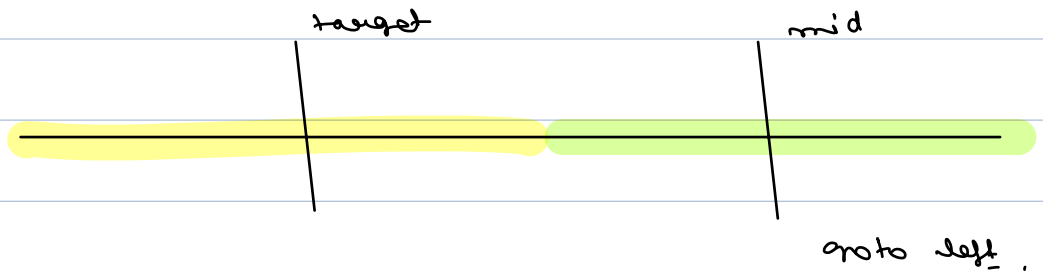


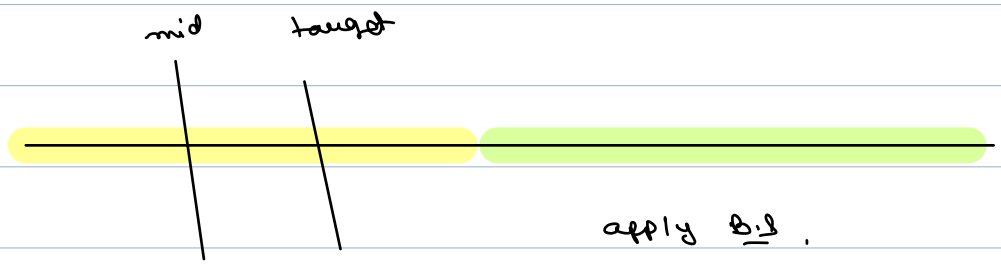
lo	hi	mid	mid Area	Target Area	
0	11	5	2	1	go to left
0	4	2	1	1	go to left.
0	1	0	1	1	go right
1	1	1	return mid.		



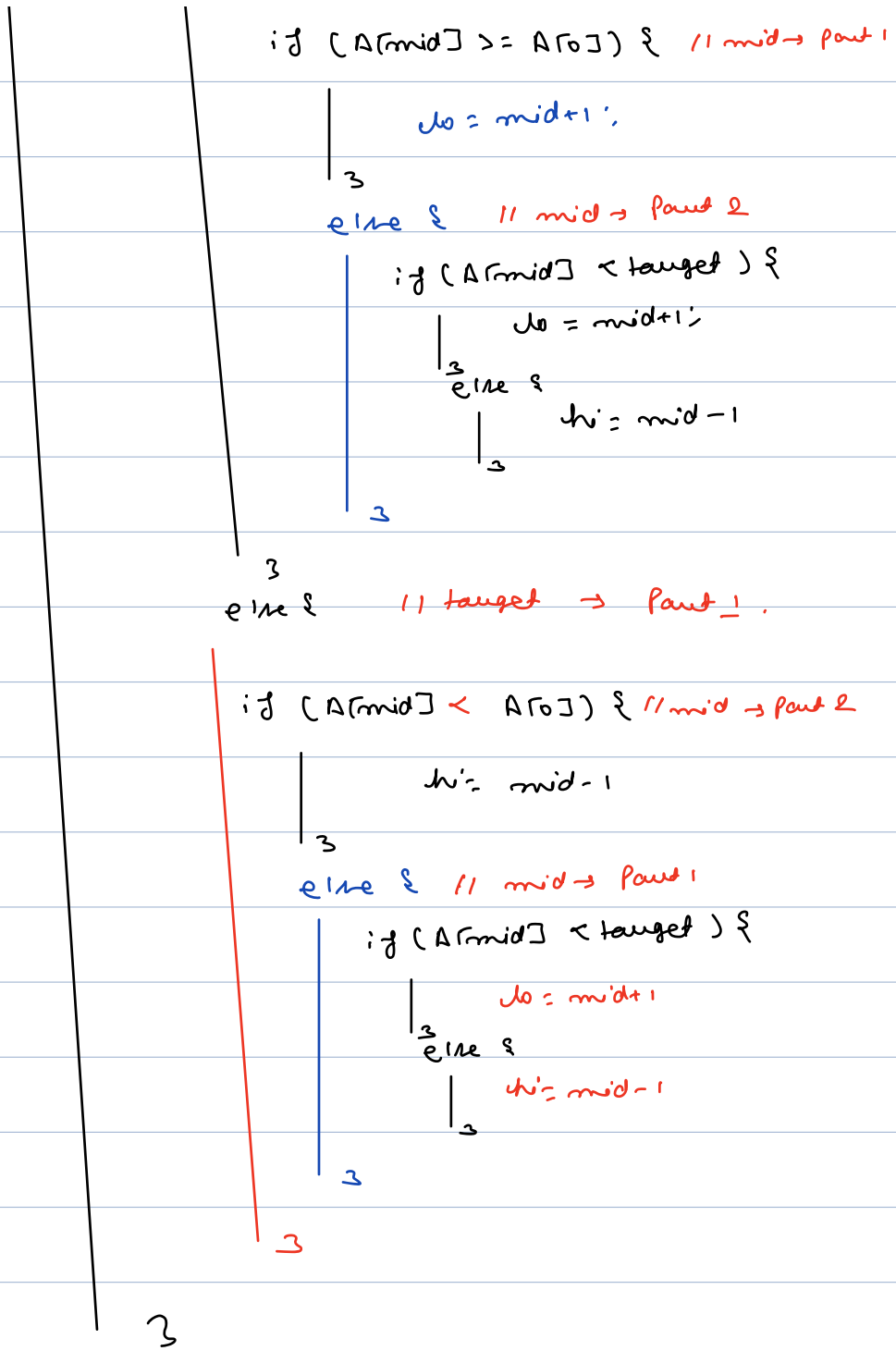
lo	hi	mid	mid Area	Target Area	
0	6	3	1	2	go to right
4	6	5	2	2	move right
6	6	found.			








```
l = 0,    r = n-1
while ( l <= r ) {
    m = l + (r-l)/2
    if( A[m] == target ) {
        return m;
    }
    if (target < A[0] ) {   // target -> Part 2
```



T.C → O(log n)

S.C → O(1)

Ques Given +ve N, find sqrt(N)

int part

floor(sqrt(N))

sqrt(25) → 5

sqrt(20) → 4

sqrt(10) → 3

```
i = 1 ;  (1)
while (i*i <= N) {
    ans = i
    i++
}  3
```

N = 20

T.C → O(√N)

S.C → O(1)

BS :-

Target → floor(sqrt(N))

Search Space → 1 to N

Case-1

mid * mid = N

return mid

Case-2 mid * mid > N

goto left

Case-3 mid * mid < N

ans = mid

goto right

N = 50

ans = 7

j	hi	m		
1	50	25	25 × 25 > 50	move left
1	24	12	12 × 12 > 50	goto left
1	11	6	6 × 6 < 50	ans = 6 goto right
7	11	9	9 * 9 > 50	goto left
7	8	7	7 * 7 < 50	ans = 7 goto right
8	8	8	8 * 8 > 50	move left.
8	7	Break		

T.C → O (log n)

S.C → O(1)

A^th Magical Number

multiples of 3 [1 100]

$\rightarrow \frac{100}{3} = 33$

4 from [1 100]

$\rightarrow \frac{100}{4} = 25$

6 from [1 100]

$\rightarrow \frac{100}{6} = 16$

4 or 6 from [1 100]

$\frac{100}{4} + \frac{100}{6} - \frac{100}{12}$ → LCM

4 → 4 8 12 16 20 24 . .

6 → 6 12 18 24 30 . .

> no. of multiples of A or B from 1 to x → $\frac{x}{A} + \frac{x}{B} - \frac{x}{LCM(A,B)}$

Ques A, B & C → find A^{th} Magical no.

↳ A no' is said to be magical if it is divisible by B or C.

B	C	A
2	3	8

2, 3, 4, 6, 8, 9, 10, 12

→

B	C	D
4	6	5

4, 6, 8, 12, 16

1) Brute force.

```
count = 0;
for (i=1; ____; i++) {        // min(b,c)*A.
    if (i%B==0 || i%C==0) {
        count++;
    }
    if (count==A) {
        return i;
    }
}
```

1) Search Space

↓

[min(b,c) , min(b,c) * A]

2) Target → A^{th} Magical number.

B	C	A
4	6	10

is 36 my 10^{th} magical no?

$\frac{36}{4}^{9} + \frac{36}{6}^{6} - \frac{36}{12}^{3} = 12.$

go left

36 37 38

is 24 my 10^{th} magical no?

$\frac{24}{4} + \frac{24}{6} - \frac{24}{12} = 8$

go right.

Tracing, d, C, A
5 7 4

lo	hi	mid		
5	20	12	$\frac{12}{5} + \frac{12}{7} - \frac{12}{35} = 9$	move right,
13	20	16	$\frac{16}{5} + \frac{16}{7} - \frac{16}{35} = 5$	move left,
13	15	14	$\frac{14}{5} + \frac{14}{7} - \frac{14}{35} = 4$	ans = 4 move left.
13	13	13	$\frac{13}{5} + \frac{13}{7} - \frac{13}{3} = 9$	goto right
14	13	break		

B = 5, C = 7, A = 3 ↓

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

A^th Magical No (A, B, C) {

```
lo = min(b, c);
hi = A * min(b, c);
lcm = lcm(b, c);
while (lo <= hi) {
    m = lo + (hi - lo)/2
    int co = m/B + m/C - m/lcm
    if (co < A) {
        lo = m+1;
    } else if (co > A) {
        hi = m-1;
    } else {
        ans = m;
        hi = m-1;
    }
}
return ans;
}
```

T.C → O(log(min(b, c) * a))

S.C → O(1)

Ques Median of two Sorted Arrays.

→ Most Important

1) Just after the class → add recording.

2) Next day → doubt session.