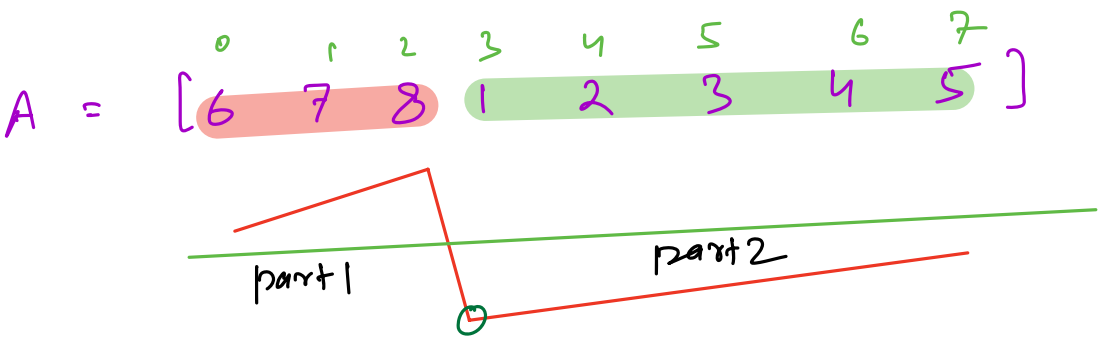
Searching 2 : Binary Search Problems

3 steps of BS

1. Define search space
2. check if mid is the answer
3. decide whether to go left or right

Question 1

Given a sorted rotated array of unique elements.
find the index of given element k.



Find smallest element ?

How to know if A[mid] is in part 1 or 2 ?

A[0] > all elements in Part 2

A[0] < all elements in Part 1

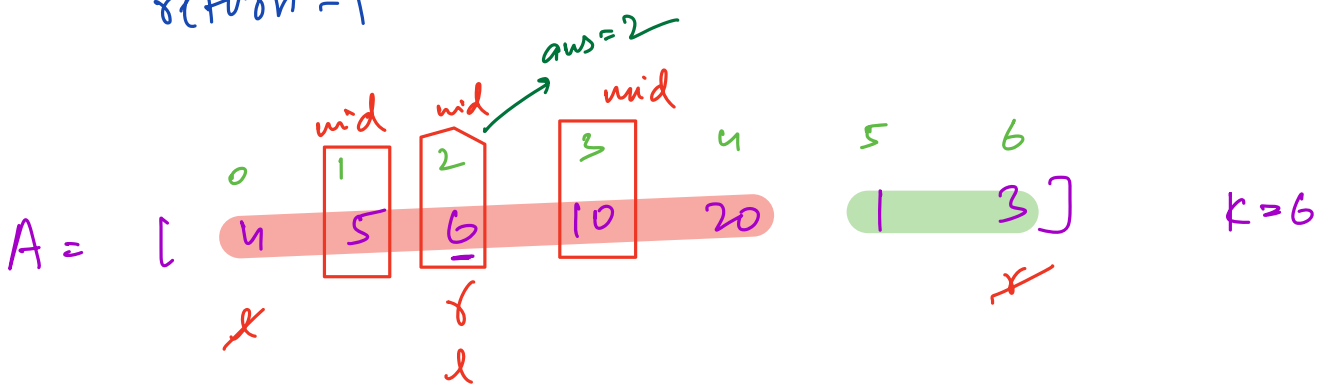
check for not rotated ? ⟶ A[0] < A[n-1]

TC = O(logN) SC = O(1)

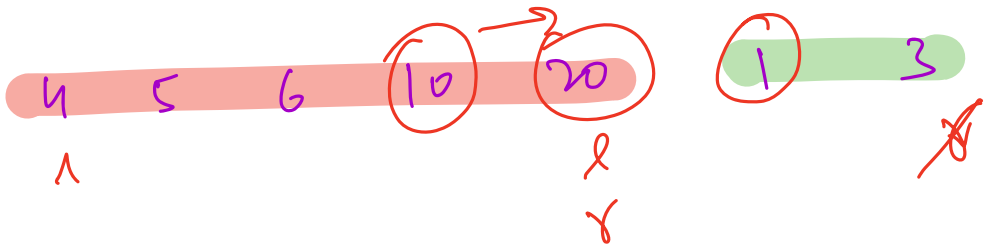
Solve in 1 binary search

```
// Define search space
l=0,   r=n-1
while (l<=r){
    //check if mid is answer
    mid = (l+r)/2
    if (A[mid] == K)   return mid
    // decide whether to go left or right
    if ( K < A[0] ) {  // K is in Part 2
        if ( A[mid] < A[0]) {  // mid is in Part 2
            if ( A[mid] < K)   l = mid+1
            else               r = mid-1
        }else{  // mid is in Part 1
            l = mid+1
        }
    }
    else {  // K is in Part 1
        if ( A[mid] > A[0]) {  // mid is in Part 1
            if ( A[mid] < K)   l = mid+1
            else               r = mid-1
        }
    }
```

```
    else{ // mid is in Part 2
        r = mid - 1
    }
}
return -1
```



TC = O(logN) SC = O(1)



Question 2

find sqrt. of a given perfect sq N. (w/o inbuilt functions)

N = 49 ans = 7

N = 25 ans = 5

N = 30 ⟵ invalid input

Bruteforce

min value = 1
max value = N
} Range

```
for ( i = 1 to N) {
    if (i*i == N)
        return i
}
```

$\sqrt{N}$ iterations

N = 25 i = 1, 2, 3, 4, 5

49 i = 1, 2, ..., 7

81 i = 1, 2, 3, ..., 9

TC = ~~O(N)~~ $O(\sqrt{N})$

SC = O(1)

```
// Define search space
l = 1, r = N
while (l <= r) {
    //check if mid is answer
    mid = l + (r-l)/2
    if (mid * mid == N)  return mid
    // decide whether to go left or right
    if( mid * mid < N)   l = mid+1     // go right
    else                 r = mid-1
}
```

TC = O(logN) SC = O(1)

N = 36

l = 1 ✗ 5

r = ~~36~~ ~~17~~ 8

mid = 18 9 5 6

$18^2 > 36$

$9^2 > 36$

$5^2 < 36$

$6^2 = 36$ ← ans

1. gcd(a,b) TC = O(log(max(a,b)))

2. lcm(a,b) → $lcm = \frac{a \times b}{gcd(a,b)}$ TC

lcm(10,15) = 30

lcm(6,10) = 30

Question 3

Find the N^{th} number which is divisible by either X or Y or both.

N = 5

X = 2

Y = 3

→ 2 3 4 6 (8) ans (5th)

N=7

x = 6 → 6 12 18 24 30 ....

y = 10 → 10 20 30 40 50 ....

6 10 12 18 20 24 (30) ans

Brute force

```
cnt=0   i=0
while (cnt < N) {
    i++
    if (i%X == 0 || i%Y == 0)  cnt++
}
return i
```

Only iterate over the multiples

N=7

x = 6 → 6 12 18 24 30 36 42

y = 10 → 10 20 30 40 50 60 70

merge 2 sorted into 1 sorted array

6 10 12 18 20 24 30

TC = O(N) SC = O(N) → HW in SC = O(1)

Optimization

# of multiples of X <= K → $\frac{K}{X}$

X = 6
K = 50
$\frac{50}{6} = 8$

multiple = 8

of multiples of X or Y or both <= K →

$$\frac{K}{X} + \frac{K}{Y} - \frac{K}{lcm(X,Y)}$$

X = 6 → 50/6 = 8
Y = 10 → 50/10 = 5
K = 50

6 12 18 24 30 36 42 48
10 20 30 40 50

lcm = X * Y / gcd(X,Y)

// define search space

l = min(X,Y) r = N * min(X,Y)

while (l <= r) {

// check if mid is answer

mid = l + (r-l)/2

$cnt = \frac{mid}{X} + \frac{mid}{Y} - \frac{mid}{lcm}$

if((mid%X==0 || mid%Y==0) && cnt==N){

return mid

}

// decide whether to go left or right

if(cnt < N) l = mid+1

else r = mid-1

}

X=6

Y=10

N=7

lcm=30

l	r	mid	cnt
~~6~~	42	24	4+2-0=6
25	~~42~~	33	5+3-1=7
~~25~~	32	28	4+2-0=6
29	32	30	5+3-1=7

ans=30

TC = O(log(N * min(X,Y))) SC=O(1)

Median in an array

middle element when array is sorted.

if odd no. of elements = $a[n/2]$

even no. of elements = $\frac{a[n/2-1] + a[n/2]}{2}$

Question 4

find median of two sorted arrays

A = [1 3 5]

B = [2 4 6]

[1 2 3 4 5 6]

$\frac{3+4}{2} = 3.5$

Idea 1 : Merge & find median

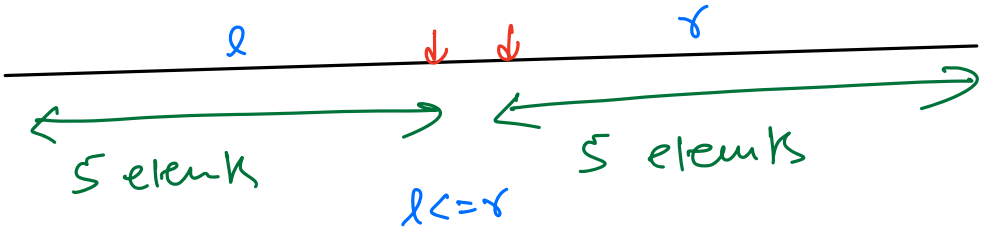
TC = O(N+M) SC = O(N+M)

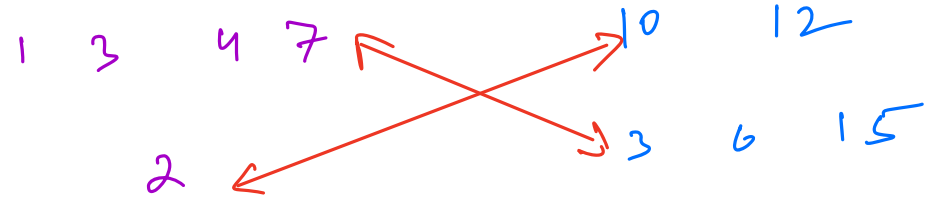
Idea 2 : Partitioning

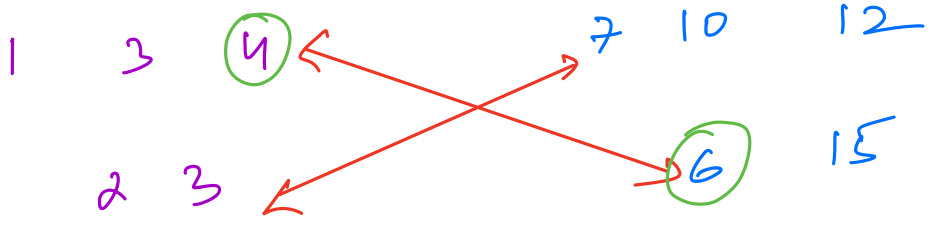
A = [1 3 4 7 10 12] # = 6

10 elements

B = [2 3 6 15] # = 4







$\frac{4+6}{2} = 5$

let's say array A is smaller than B.
↳ size N ↳ size M

define search space

l = 0, r = n - 1

while (l <= r) {

// check if mid is answer

mid = (l + r) / 2

left_count = mid + 1

complete in next class