Q1 Given an array which is formed by rotating a Distint sorted array by K times, Search for a given key in rotated array. You are given the rotated arr. Rotation here means bringing the last element to the front.

Ex1 arr[] =

0	1	2	3	4	5	6	7	8	9	10	11	12
-20	-14	-8	-4	1	2	4	7	11	14	19	23	27

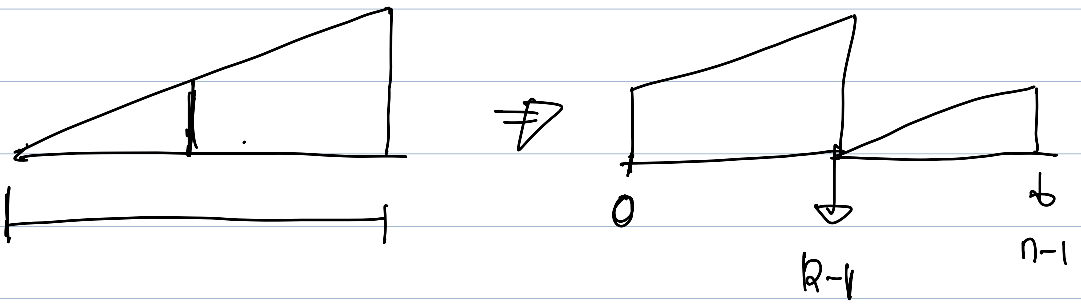
↓ K=5 { 5 times Rotation }

arr[] =

0	1	2	3	4	5	6	7	8	9	10	11	12
11	14	19	23	27	-20	-14	-8	-4	1	2	4	7

Search $\boxed{-14}$

CASE 1: K is given to you. K=5



Apply Binary Search on two array ⇒ [0, k-1] + [k, n-1]

TC: log(n)

SC: O(1)

Optimisation

Approach2

Check the key's value with the last element of rotated array.

key > arr[n-1] → search in first arr

key ≤ arr[n-1] → search in 2nd arr

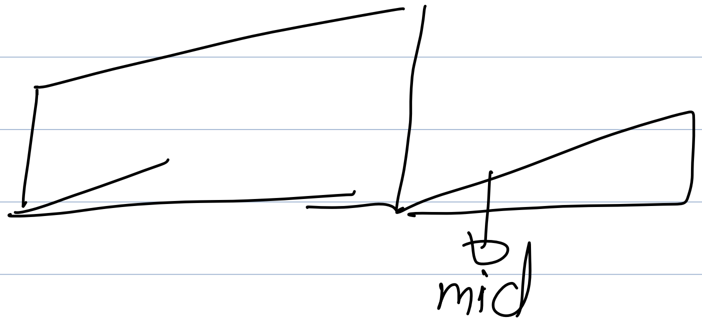
$$T_c : O(\log n)$$

Approach3 :

$$mid = \left[\left(\frac{l+h}{2}\right) + K\right] \% \, n$$

[1 2 2 2 2 2]

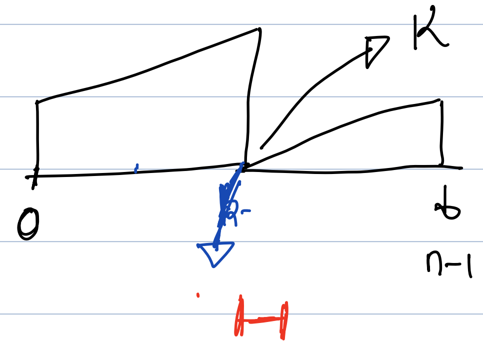
$a[mid] >= a[l]$



CASE2 : K is not given to you.

arr[] =

0	1	2	3	4	5	6	7	8	9	10	11	12
11	14	19	23	27	-20	-14	-8	-4	1	2	4	7



first element

CASE1: arr[mid] > arr[0]

⇒ go right.

CASE2: arr[mid] < arr[0]   ans ⇒ mid.

⇒ go left.   ⇒ K

arr[] =

0	1	2	3	4	5	6	7	8	9	10	11	12
11	14	19	23	27	-20	-14	-8	-4	1	2	4	7

ans ⇒ 6

1) l=0, h⇒12, mid ⇒ 6 | arr[6] ⇒ -14   h=mid-1

2) l=0, h⇒5, mid ⇒ 2 | arr[2] ⇒ 19   l ⇒ mid+1

3) l⇒3, h ⇒ 5, mid ⇒ 4) arr[4] ⇒ 27, l ⇒ mid+1

4) l ⇒ 5, h ⇒ 5, mid ⇒ 5 | arr[5] ⇒ -20, ans ⇒ 5
h ⇒ mid-1

TC: O(logn)   SC: O(1)

Q2 Every element occurs twice except for 1 element . find unique element. Duplicate are adjacent to each other. Elements are not sorted. Array is not sorted.

Ex1 arr[] =

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	3	1	1	8	8	10	10	19	6	6	2	2	4	4

Approach 1 : xor of everything : TC: O(n)
SC: O(1)

Ex1 arr[] =

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	3	1	1	8	8	10	10	$\boxed{19}$	6	6	2	2	4	4

i) First half : first occurrence of duplicate element is happening on even index

i) Second half : first occurrence of duplicate element is happening on odd index

→ [mid-1] & [mid+1]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	3	1	1	8	8	10	10	19	6	6	2	2	4	4

⇒ Check if arr [0] is the answer.

⇒ Check if arr[n-1] is the answer.

1) l ⇒ 1 , h ⇒ 13 , mid = 7

⇒ not answer ⇒ if (arr[mid-1] == arr[mid]

mid = mid-1;

⇒ mid ⇒ 6 ⇒ go right

l ⇒ mid+2

2) l ⇒ 8, h ⇒ 13, mid ⇒ 10

mid ⇒ 9 ⇒ go left.

h ⇒ mid-1

3) l ⇒ 8 , h ⇒ 8 , mid ⇒ 10

answer

Pseudo Code

```
int findUnique ( int arr[], int n ) {
    if (n == 1) return arr[0]
    if (arr[0] != arr[1])
        return arr[0]
    if (arr[n-1] != arr[n-2])
        return arr[n-1];
    l => 2 , h => n-3
    while ( l ≤ h ) {
        int mid = $\frac{(l+h)}{2}$
        if ( arr[mid-1] != arr[mid] &&
                 arr[mid] != arr[mid+1])
            return arr[mid];
        if (arr[mid] == arr[mid-1])
            mid = mid-1.
        if (mid % 2 == 0)
            l = mid+2
        else
            h = mid-1
    }
}
```

TC: O(log n)

SC: O(1)

Q3 Find the max subarray sum of len K.

Ex arr[8] =

0	1	2	3	4	5	6	7
3	2	5	4	6	3	7	2

$\boxed{K = 3}$

TC: O(n)

SC: O(1)

Q3 Given an array of +ve integers, find maximum K such that {max subarray sum of len K $\leq$ B}

$\downarrow$

given in input.

Ex arr[8] =

0	1	2	3	4	5	6	7
3	2	5	4	6	3	7	2

B $\Rightarrow$ 20

Using Sliding window for each K

K $\Rightarrow$ 1 max_Sub_sum $\Rightarrow$ 7 $\leq$ 20 ans = 1

K $\Rightarrow$ 2 max_Sub_sum $\Rightarrow$ 10 $\leq$ 20 ans $\Rightarrow$ 2

K $\Rightarrow$ 3 max_Sub_sum $\Rightarrow$ 16 $\leq$ 20 ans $\Rightarrow$ 3

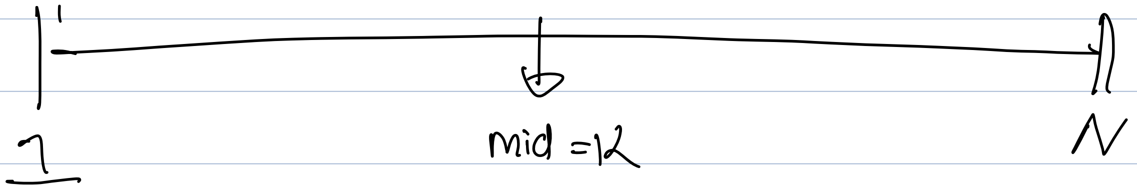
K $\Rightarrow$ 4 max_Sub_sum $\Rightarrow$ 20 $\leq$ 20 ans $\Rightarrow$ 4

K $\Rightarrow$ 5 max_Sub_sum $\Rightarrow$ 25 > 20

TC: $O(n^2)$

SC: O(1

1) Find k ⇒ target.
2) Search Space ⇒ [1, N]



1) Case 1 : max_sub_sum (mid) < B
ans ⇒ mid , l ⇒ mid+1

2) CASE 2 : max_sub_sum(mid) > B
h ⇒ mid-1

3) CASE 3 : max_sub_sum(mid) = B
return mid.

Tc: O(nlogn)
SC: O(1)

Dry Run

Ex arr[8] =

0	1	2	3	4	5	6	7
3	2	5	4	6	3	7	2

B ⇒ 21

ans ⇒ 4

l ⇒ 1, h ⇒ 8, mid ⇒ 4

max_sub_sum (4) ⇒ 20

l ⇒ mid+1

l ⇒ 5, h ⇒ 8, mid ⇒ 6

max_sub_sum (6) ⇒ 27

h ⇒ mid-1

l ⇒ 5, h ⇒ 5

max_sub_sum(5) ⇒ 25

h ⇒ mid-1

$\boxed{l \Rightarrow 5,\ h \Rightarrow 4}$

Q4 Given N, find sqrt(N). Find closest integer.

N sqrt(N) ⇒ $[1, N]$

Target = sqrt(N)

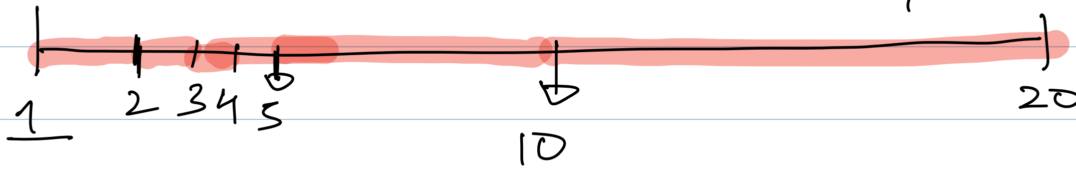
Search Space = $[1, N]$

$l = 1, h = 20$

$l = 1, h \Rightarrow 9$

$l = 1 \quad h \Rightarrow 4$

$l = 4, h \Rightarrow 4$



CASE 1 : if (mid × mid > n.)

go left

h ⇒ mid−1.

CASE 2 : ~~if~~ mid × mid < N.

go right # ans ⇒ mid

l = mid+1

CASE 3: mid × mid == N

return mid.

Q Given a string made of only $\boxed{a, b, c}$

find smallest substring which contains $\boxed{a, b, c}$

aabbccaababbc $\boxed{l=4}$

$[3, N]$ = Seq