26/6/2023

Searching 2

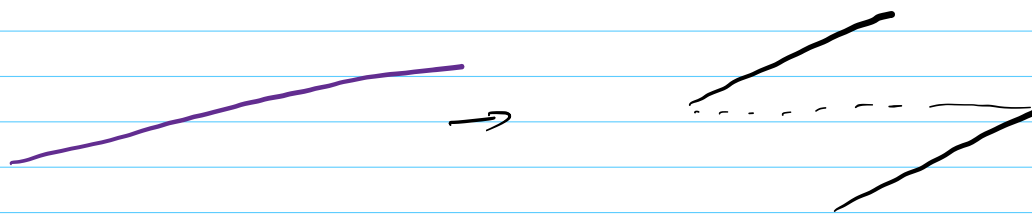
Q Find the index of an element x in a rotated sorted array with distinct elements (-1 if not present)

microsoft

A = [50 60 100 3 9 10 25 30 35]
(indices 0 1 2 3 4 5 6 7 8)

x = 25, Ans = 6
x = 27, Ans = -1

A = [5 7 10 12 1 3], x = 3, Ans = 5
(indices 0 1 2 3 4 5)



Rotated sorted array → 1> Combination of 2 sorted arrays.

2> elements in first part > element in second part.

How to check if array is not rotated?

if (A[0] < A[N-1]) ⇒ sorted

if (A[0] > A[N-1]) ⇒ rotated sorted array.

```
// define search space.

L = 0 ,   R = N-1
while (L <= R) {
    mid = (L+R)/2 ;

    // check if mid is my answer.

    if (A[mid] == X) return mid.

    // decide whether to go left or right

    if (X >= A[0]) {  // X is in first part.

        if (A[mid] < A[0])   R = mid-1   // mid in 2nd.
        else {
            if (X > A[mid]) L = mid+1
            else   R = mid-1
        }

    else {  // X in second part
        if (A[mid] >= A[0])   L = mid+1;  // mid in 1st.

        else {
            if (X > A[mid]) L = mid+1
            else   R = mid-1
        }

}
return -1;
```

A = [50 60 100 3 9 10 25 30 35]
(0 1 2 3 4 5 6 7 8)

target = 100

L	R	mid	A[mid]
0	8	4	9
0	3	1	60
2	3	2	100

target = 50

0	8	4	9	TC: O(lg N)
0	3	1	60	SC: O(1)
0	0	0	50	

Binary search on Answer

Q Find floor($\sqrt{N}$) without using internal function.

Eg, N = 10, Ans = floor(3.....) → 3
N = 50, Ans = 7
N = 25, Ans = 5
floor($\sqrt{7}$) = 1

$1 \le \sqrt{N} \le N$

// Define search space.
L = 1, R = N
while (L <= R) { overflow.
mid = (L+R)/2 → L + (R-L)/2

// check if mid is answer.
if (mid*mid <= N && (mid+1)*(mid+1) > N)
return mid;
// decide whether to go left or right.
if (mid*mid > N) R = mid-1
else L = mid+1
}

TC: O(lg N)
SC: O(1)

$1^2 \le 50$
$2^2 \le 50$
$3^2 \le 10$
$4^2 > 10$
$2^2 \le 10$ → mid
$3^2 \le 10$ → mid+1

$5^2 > 10$
↓
$2^2 < 10$

Mut at 8:23 am.

N = 10	L	R	Mid	Mid cond
1	1	10	5	25
3	1	4	2	4
	3	4	3	9

Q Given 3 integers N, X & Y.
Find N^{th} no which is divisible by X or Y or both.

Eg N = 4, X = 2, Y = 3 → 2 3 4 6

N = 5, X = 4, Y = 8 → 4, 8, 12, 16, 20

N = 8, X = 4, Y = 3 → 3 4 6 8 9 12 15 16

N = 3, X = 10000, Y = 50000 → 30000

Brute force → If $n >= 1$, check if it is divisible by X or Y or both.

Soln 1

$(n \% X == 0 \;||\; n \% Y == 0)$

Keep track of N^{th} no to see the ans.

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

Soln 2 Check all the multiples of X & Y.

N=7

X=8 → 8 16 24 32 40 48 56 } N multiples of X & Y.

Y=3 → 3 6 9 12 15 18 21

ans = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ 7

3 6 8 9 12 15 16

TC: O(N)

SC: O(N+N) ≃ O(N) ↓ O(1)

```
a=0, b=0, last=0
for i->1 to N {
    if ((a+X) < (b+Y)) { a+=X , last=a }
    else if ((a+X) > (b+Y)) { b+=Y, last=b }
    else {
        a+=X, b+=Y, last = a or b }
}
return last
```

TC: O(N)
SC: O(1)

a	b	last
0	0	0
2	0	2
2	3	3
4	3	4
6	6	6
8	6	8
8	9	9 (Ans)

N=6, X=2, Y=3

→ # elements $<= K$ that are divisible by X = K/X

$K=50$, $X=7$ Ans=7 { 7, 14, 21, 28, 35, 42, 49 }

$K=20$, $X=6$ Ans=3 { 6, 12, 18 }

$K=15$, $X=5$, Ans = 3 { 5, 10, 15 }

→ # elements $<= K$ that are divisible by X or Y or both.

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

$K=20$

$X=2 \rightarrow 20/2 \rightarrow 10$ → 2 4 6 8 10 12 14 16 18 20

$Y=3 \rightarrow 20/3 \rightarrow 6$ 3 6 9 12 15 18

$\frac{-20}{6} \rightarrow \frac{-9}{18}$

L	R	mid	cnt	N
2	10	6	4	5
7	10	8	5	5

$K=50$

$X=6 \rightarrow 50/6 \Rightarrow 8$

$Y=10 \rightarrow 50/10 \rightarrow 5$

$50/lcm(6,10)$

$50/30 \rightarrow 1$

$8+5-1 \Rightarrow 12$ (ans)

$$lcm(a,b) = \frac{a*b}{gcd(a,b)}$$

50 ← 60

// Define search space.

L = min(x,y) R = min(x,y) * N

while (L <= R) {
  mid = L + (R-L)/2
  // check if mid is answer.

  cnt = mid/x + mid/y - mid/lcm

  if (cnt == N && (mid % x == 0 || mid % y == 0))
    return mid;

  // decide whether to go left or right.
  if (cnt < N) L = mid + 1
  else R = mid - 1   // cnt >= N
}

N = 7

x = 8 → 8 16 24 32 40 48 56

y = 3 → 3 6 9 12 15 18 21

L	R	mid	cnt
3	21	12	5
13	21	17	7
13	16	14	5
15	16	15	6
16	16	16	7

TC: O(log(N * min(x,y)))

SC: O(1)

$mn(x,y) \rightarrow Z$

$Z \rightarrow N \times Z$

$N \times Z - Z \rightarrow Z \times (N-1) \approx Z \times N$

$O(NZ)$

$O(N * mn(x,y))$

$N * mn(x,y) + TC(grad)$ → 1 time