Minimum Product of 3

Rotated Sorted Array Search

Maximum Good Pair

1. All 3 questions → Practice Mode in less time

→ Reattempt

2. Score > 60

3. $\boxed{\text{Score} < 60}$

Reattempt 1

Tue 28 Nov Tue Midnight

| |

Thurs 30 Nov Thurs Midnight

Given an array of integers A of size N, return minimum product of any 3 nos. from A.

$3 \le N \le 10^5$

$-10^3 \le \underset{A[i]}{\cancel{x}} \le 10^3$

① 6, ③, ①, 8, 10, ②

$\boxed{\text{Ans} = 6}$

$\xrightarrow{\text{sort}}$

$\underbrace{1, 2, 3}, 6, 8, 10$

Product

↓

ans = $\underline{x \times y \times z}$

② $-1, 2, 1, 0 \xrightarrow{\text{sort}} \underbrace{-1, 0, 1}_{0?}, 2$

ans = -2 $(-1 \times 1 \times 2)$

all nos are +ve

↓

product of 3 smallest nos

sign

↓

-ve

Product

↓

sign value

+ve

-ve

$x \times y \times z$

-ve -ve +ve × → +ve

OR $\boxed{\begin{matrix} -ve & +ve & +ve \\ -ve & -ve & -ve \end{matrix}}$

$\underbrace{-2 \quad -1} \quad 0 \quad \underbrace{1 \quad 4}$

⑧

-8

$\underbrace{-8 \quad -6 \quad -5}_{\text{first 3}} \quad -4 \quad -2 \quad 1 \quad 10 \quad 30$

$-240 \leftarrow -8 \times -6 \times -5 \leftarrow$ -ve -ve -ve ①

$-2400 \leftarrow \underline{-8} \times \underline{10} \times \underline{30} \leftarrow$ -ve +ve +ve ②

-ve -ve -ve → smallest 3 nos

-ve +ve +ve → smallest -ve no × biggest +ve values

$-4 \quad \underbrace{-3 \quad -2}_{} \quad -1 \quad 0 \quad 0 \quad 0$

✓ ① -ve -ve -ve → $-4 \times -3 \times -2 = -24$

② -ve +ve +ve → $-4 \times 0 \times 0 = 0$

ans = min (product of 3 smallest nos., product of min × max × 2nd max)

Tc : O(Nlog N)

Sc : O(1)

Merge sort
↓
N log N

```
sort ( A )
if ( A[0] ≥ 0)
return        A[0] * A[1] * A[2]
else
return  min ( A[0] * A[1] * A[2],
              A[0] *  A[N-1] * A[N-2])
```

✓ Min
✓ 2nd Min
✓ 3rd Min

Max ✓
2nd Max ✓

Traverse arr
↓
update var
↓
Tc : O(N)
Sc : O(1)

Given a sorted array of integers A of size N and an integer B where A is rotated at some unknown pivot beforehand.

For eg. $(0,1,2,\underbrace{4,5,6,7})$ becomes $(\underbrace{4,5,6,7},0,1,2)$

Find target B in A and return index.

No duplicates exist in array.

$1 \le N \le 10^6$

$1 \le A[i] \le 10^9$

A = 4, 5, 6, 7, 0, 1, 2, 3

B = 4

ans = 0

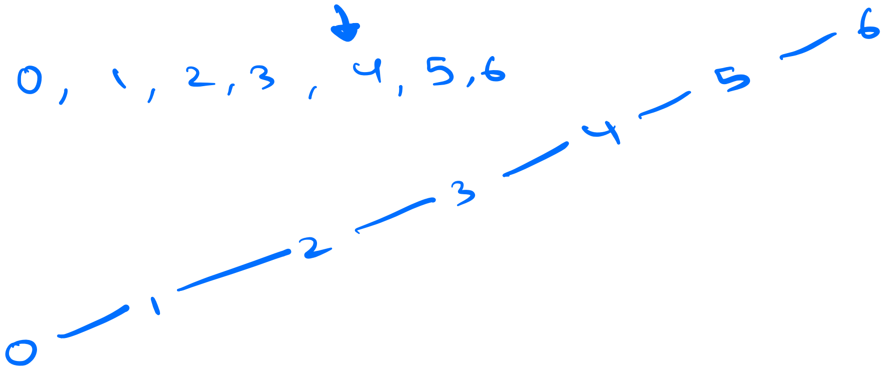
A = 9, 10, 3, 5, 6, 8

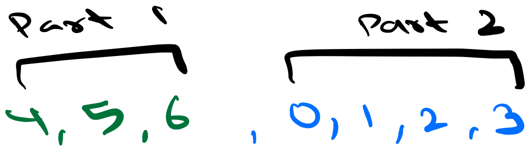
B = 5

ans = 3

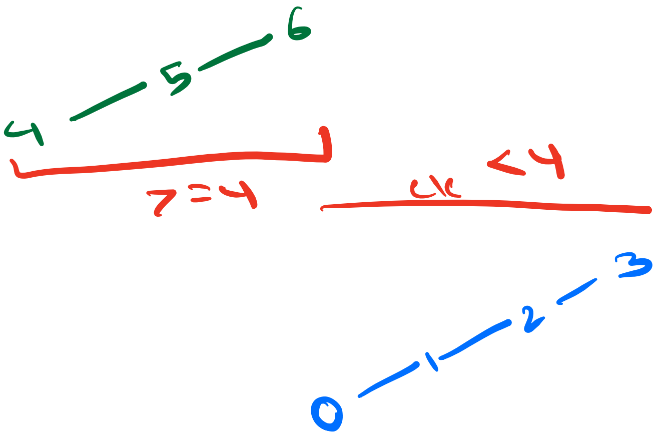
B = 11

ans = -1



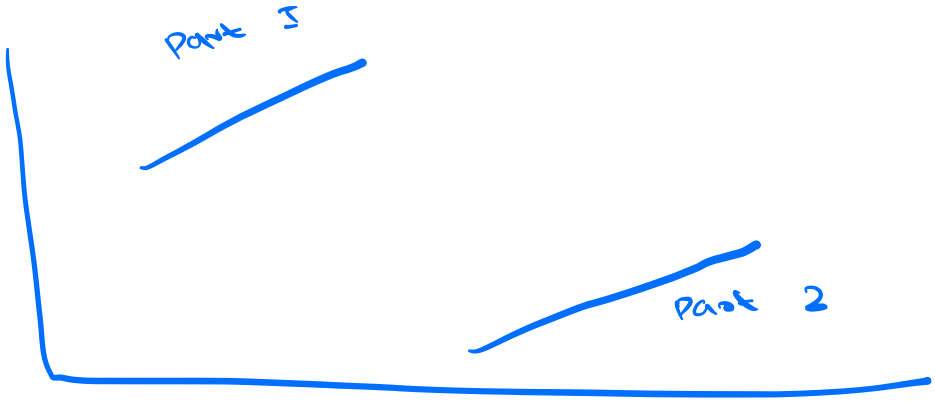


Part 1 > Part 2



A[0] =< ele → Part 1

A[0] > ele → Part 2



① Target Part I → Mid Part 2 Left
→ Mid Part 1 BS in part 1

② Target Part II → Mid Part 1 Right
→ Mid Part 2 BS

	↓											
	0	1	2	3	4	5	6	7	8	9	10	11
A =	10	20	30	1	2	3	4	5	6	7	8	9

I > II

fixed B = 10

s	e	mid	A[Mid] → Part	Target → Part	
0	11	5	2	1	Left
	↓ mid-1				
0	4	2	1	1	BS → Left
	↓ mid-1				10
0	1	0	A[mid] = 10	10	return 0

```
int s = 0, e = N-1
while ( s <= e) {
    mid = s + $\frac{(e-s)}{2}$    // $\frac{(e+s)}{2}$
    if (A[mid] == B)
        return mid
    if ( B >= A[0]) { // target Part I
        if ( A[mid] >= A[0]) { // mid Part I
            if (B < A[mid])
                e = mid -1      // left
            else
                s = mid+1      // right
        }
        else { // mid Part II
            e = mid -1      // left
        }
    }
    else { // target Part II
        if ( A[mid] < A[0]) { // mid Part II
            if (B < A[mid])
                e = mid -1      // left
            else
                s = mid+1      // right
        }
```

```
            else {      // mid  Part  I
                 s = mid+1     //  right
               }
           }
      }
  return -1
```

TC : O(log N)

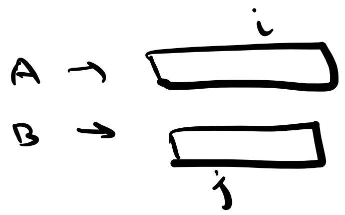
SC : O(1)

③

Given 2 strings A and B which are sorted in dictionary order.
For any pair (i, j) where i is index of A and j index of B, pair is good if i<=j and A[i] ≥ B[j]. Score is abs (i-j)
Return max possible score out of all good pairs. Return 0 if no good pair.

$i <= j$ $\quad$ $A[i] >= B[j]$

$\downarrow$ A $\quad \downarrow$ B $\quad \downarrow$ large $\quad \downarrow$ small



$\text{Score}(i, j) = |i-j|$

$1 \le N, M \le 10^5$

$$A = \begin{array}{ccc} 1 & 2 & 3 \\ b & c & c \end{array}$$

ans = 1

$$B = \begin{array}{ccc} 1 & 2 & 3 \\ a & b & c \end{array}$$

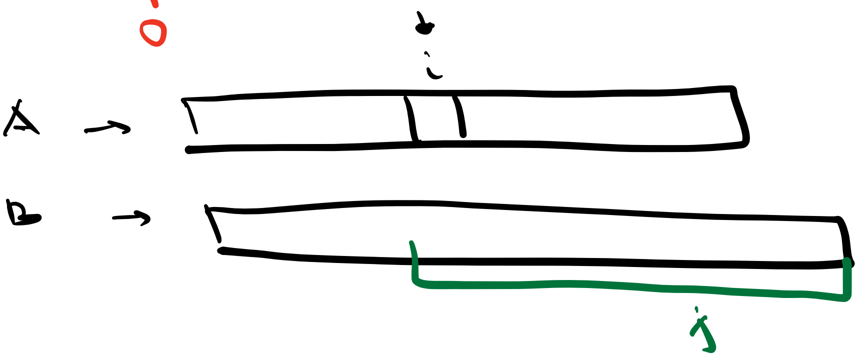
i j $\quad$ i<=j ✓ $\quad$ A char ≥ B char ✓

1 1

(b, a) → score = 0

i, j	score
2, 2	0
2, 3	1
3, 3	0

1, 2 ✓ score = 1



A[i] → smaller element in B from i till last

BF : consider all pairs (i,j)

```
int ans=0
for (i=0 ; i<M ; i++) {
    for ( j= i ; j<N ; j++) {
        if (A[i] ≥ B[j])
            ans = max (ans, j-i)
    }
}
```

i ≤ j →

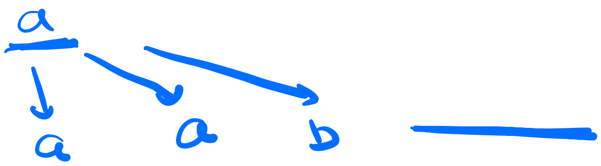
TC: O(M+N)

SC: O(1)

		i						
0	1	2	3	4	5	6	7	8
A → a	a	b	c	c	c	c	c	d

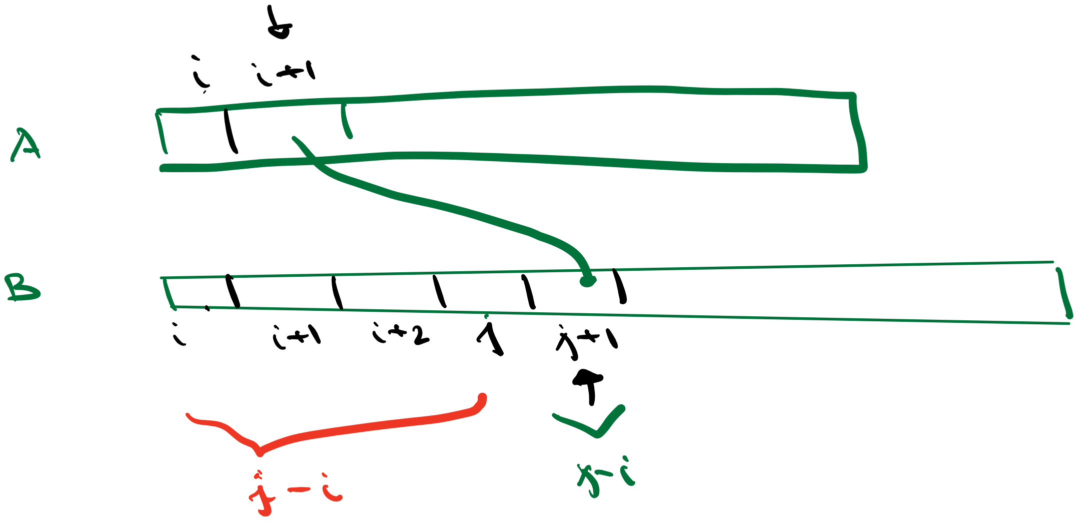
0	1	2	3	4	5	
B → a	a	b	b	b	b	
						j

res=0| 0 1 1 1 2 3



A[i] < B[j]   i++

A[i] >= B[j]   update res
j++



```
int  i=0,  j=0 , res=0
while ( i<m  &&  j<n) {
    if (A[i]  ≥ B[j]) {
        res = max (res, abs(j-i))
        j++
    }
    else
        i++
    if (i>j)
        j=i
}
```

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

return res

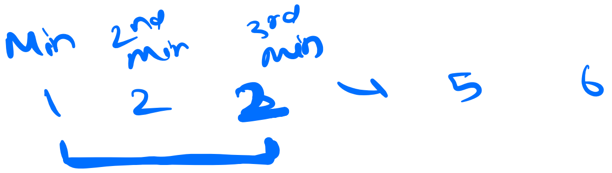
---

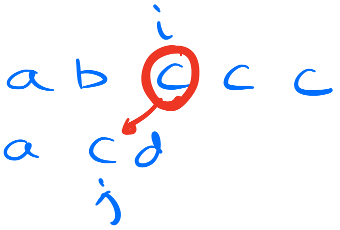
max
2nd max > 1 iter

min
2nd min > 1 iter

3rd min > 1 iter

i <= n

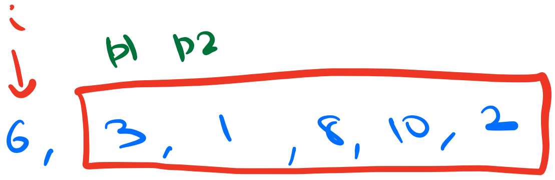




res = 0

res = 1

(i, n)



min
6 x (_ _) = min _ _ _