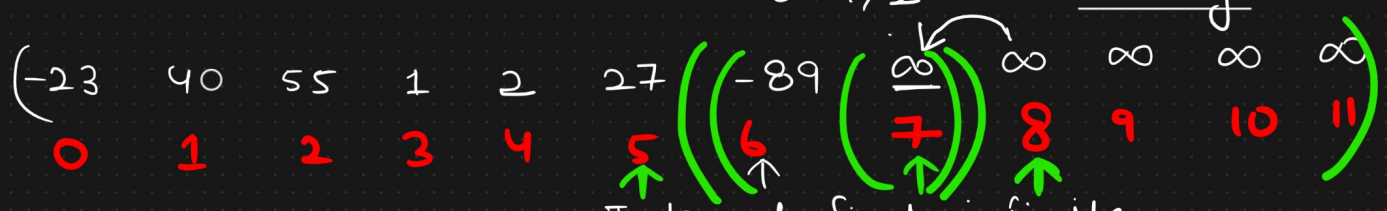
Interview Questions on Binary Search

1) facebook (Technical Round)

↳ Not sorted manner

$n \rightarrow$ size of an array



$n = 12$

output = 7

$mid = \frac{6+11}{2} = 8$

qs:1.

1) Linear search

```
for i=0 to n-1
    arr(i) == 'inf'
        return i
```

→ O(n)

2) Modified Binary Search → O(log n)

mid = 0 + (11-0)//2 = 0+5 = 5

if arr(mid) == int :

left = mid+1 ⟵ Right side

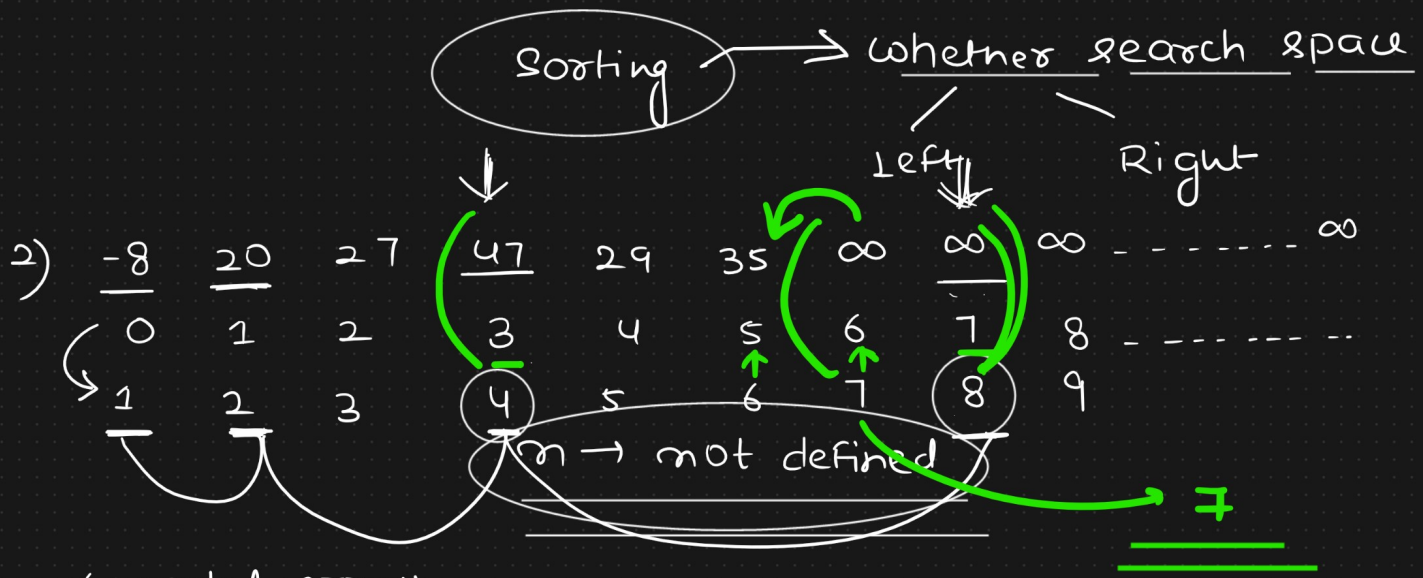
Search space

arr(mid) <= 'inf'

right = mid-1

Note → 1) Don't judge by looking whether the array is physically sorted or not.

↳ Logical sort



Unsorted array Output = 6 i = 3

Index of first infinite j = 7

$\frac{6+7}{2} = 6$ mid = 5

Note:

Infinite numbers are only available after all integers.

Exponential increase → range of numbers

+

Binary search

$1 \rightarrow 3 \rightarrow 9 \rightarrow 27$

more numbers

↓↓

increasing search space

$1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16$

Problem solving skill

array is sorted

3) 20, 40, 60, 80, 90, 120, 240

0 1 2 3 4 5 6

l r

260 > 210 → r = r-1

else → (a, b) l = l+1

a + b = 210

Output = (4,5)

sum_value

a = 20

target = 210 - 20 = 190

Approach 1

Linear search ←

↳ O(n²)

Approach 2

Binary search ← O(n log n)

O(n log n) << O(n²)

Amazon, Google

Approach 3

Two Pointer Approach

l = starting index

r = ending index

while l <= r:

if arr(l) + arr(r) == sum_value

return (l, r)

elif arr(l) + arr(r) > sum_value

r = r - 1

arr(l) + arr(r) < sum_value

else:

l = l + 1

O(n) ⟸

$O(n^2) < O(n \log n) < O(n)$

↑ Brute force approach → optimized version

→ Best time to buy & sell the stock

4) price = [7, 1, 5, 3, 6, 4] (positions 1 2 3 4 5 6)

max profit = 6 - 1 = 5

```
for i = 0 to n-1
    if price(i) < minPrice:
        minPrice = price(i)
    elif price(i) - minPrice > maxprofit
        maxProfit = price(i) - minPrice
return maxProfit
```

5) 2D Arrays → (Rows, columns)

arr ⇒

	0	1	2
0	2	4	6
1	8	10	12
2	20	27	40

3x3 → #rows, #columns

⇒ Matrix computational
⇓
- matrix multiplication,
- matrix transpose

arr[1][2] → Row number (1), column number (2)

Amazon

m x n

{ m → no of rows
{ n → no of columns

Constraints

1) Integer in each row is sorted from left to right

2) first integer of each row > last integer of previous row

10 > 7, 23 > 20

2D →

arr =

	0	1	2	3
0	1	3	5	7
1	10	11	16	20
2	23	30	34	60

3 x 4

m n

target = 3

row number → 1

Column number → 1

output ⇒ True

target = 32

Optimised Approach

output ⇒ false

target = 3

m = 3
n = 4

Virtual array

0	1	2	3	4	5	6	7	8	9	10	11
1	3	5	7	10	11	16	20	23	30	34	60

↳ Sorted array

Binary search

mid = 0+4/2 = 2

mid = 0+11/2 = 5

mid = 0+1)/2 = 0

J = 11 = m x n - 1

$n = 4$ $\quad$ $mid = 5$

$5 // 4 = 1$

$\Rightarrow$ $\begin{cases} row\_number = mid // n \\ col\_number = mid \% n \end{cases}$

$5 \% 4 = 1$

$mid = 2$

$row\_number = 2 // 4 = 0$

$Column\_number = 2 \% 4 = 2$

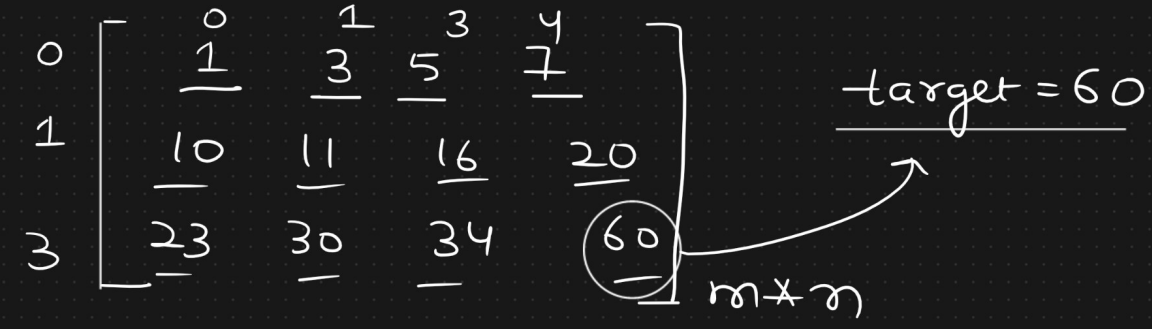
$mid = i + (j - i)/2$

$i = 0 \;,\; J = m * n - 1$

Optimized approach

Binary Search $\rightarrow$ $\log_2(m*n)$

$\Rightarrow$ $O(\log_2(m*n))$



Brute force approach $O(m*n)$

$O(m*n)$

```
for i = 0 to m-1:
    for J = 0 to n-1:
        if arr(i)(j) == target:
            return True
return false
```