

CLASS - 12 → 18/09/2023

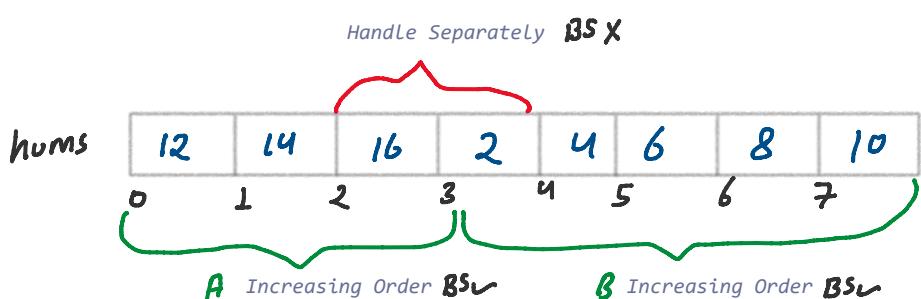
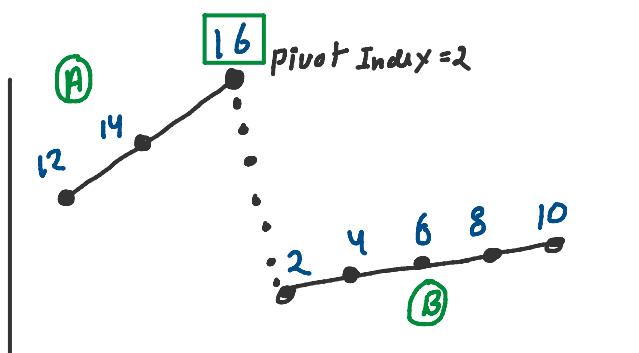
<https://www.linkedin.com/in/manojofficialmj/>

SEARCHING & SORTING LEVEL - 02

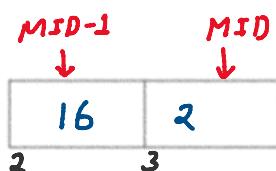
- Find pivot element index from sorted and rotated array

Example:

| | | |
|-----------------|---------------------|-----------|
| nums | 12 14 16 2 4 6 8 10 | Output: 2 |
| 0 1 2 3 4 5 6 7 | | |

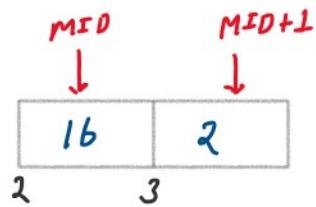


Handle Separately



if (nums[mid] < nums[mid-1])
ans = mid-1; ②

MID MID+1



$\text{if } (\text{nums}[mid] > \text{nums}[mid+1])$
 $\text{ans} = \text{mid};$ ②

Kaise pata karen ki hum kis line par exist karte hai A ya B

$\text{if } (\text{nums}[s] > \text{nums}[mid]) \rightarrow B \text{ line}$

Left $\Rightarrow e = mid - 1$

Else $\rightarrow A \text{ line}$

Right $\Rightarrow s = mid + 1$

Lets suppose ki hamare pass ek aisa array hai jo sirf 1 element store karta hai to us case me wo maximum(pivot) index khud hogा

Corner Case

Example: 02

nums . output = 0

$s = 0$ $\text{if } (s == e)$
 $e = 0$ $\hookrightarrow \text{return } s;$
 $mid = 0$

Lets suppose ki hamare pass ek aisa array hai jo sirf 2 element store karta hai to us case me hamse galti ho skati hai HOW?

Example: 03

nums . output: 1

$s = 0$
 $e = 1$
 $mid = 0$
 $size = n = 2$

$\text{if } (\text{nums}[mid] > \text{nums}[mid+1])$ X
 $\text{ans} = \text{mid};$

$\text{if } (\text{nums}[mid] < \text{nums}[mid-1])$
 $\text{ans} = \text{mid}-1;$

$$0 - 1 = -1$$

$$0 - 1 = -1$$

nums [-1]
Runtime Error

How to solve it

```
if (mid+1 < n && nums[mid] > nums[mid+1])
    ans = Mid;
```

```
if (Mid-1 >= 0 && nums[mid] < nums[Mid-1])
    ans = Mid-1;
```

```
// PROGRAM 01: Find pivot element index from sorted and rotated array
#include<iostream>
#include<vector>
using namespace std;

// Pivot index function
int findPivotIndex(vector<int> nums){
    int n=nums.size();
    int s=0;
    int e=n-1;
    int mid=s+(e-s)/2;

    while(s<=e){
        // Corner Case for single element
        if(s==e){
            return s;
        }
        // Handle separately condition
        if(mid+1 < n && nums[mid] > nums[mid+1]){
            return mid;
        }
        else if(mid-1 >= 0 && nums[mid] < nums[Mid-1]){
            return mid-1;
        }
        // Apply binary search in line B
        else if(nums[s] > nums[mid]){
            e=mid-1;
        }
        // Apply binary search in line A
        else{
            s=mid+1;
        }
        // Updated mid index
        mid=s+(e-s)/2;
    }
}

int main(){
    vector<int> nums{12,14,16,2,4,6,8,10};

    int pivotIndex = findPivotIndex(nums);
    cout<<"Maximum element at index: "<<pivotIndex<<endl;
}

/*
Example 01:
INPUT: [12,14,16,2,4,6,8,10]
OUTPUT: Maximum element at index: 2

Example 02:
INPUT: [1,3]
OUTPUT: Maximum element at index: 1

```

Example 02:
INPUT: [1, 3]
OUTPUT: Maximum element at index: 1

Example 03:
INPUT: [3]
OUTPUT: Maximum element at index: 0
*/

2. Search in a rotated and sorted array (Leetcode-33) *V-U-Imp.*

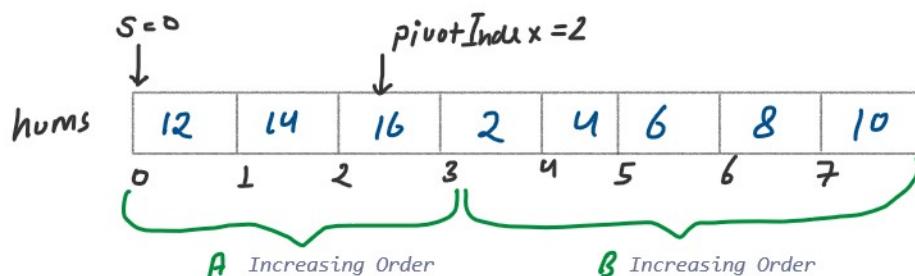
Example: 01

| | | | | | | | | |
|------|----|----|----|---|---|---|---|----|
| nums | 12 | 14 | 16 | 2 | 4 | 6 | 8 | 10 |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Target = 14
Output = 1

Approach:
Step 01: Find pivot index
Step 02: Find out ki target kis line par lie karta hai
Step 03: Target jis line par lie karta hai us line par Binary Search Laga deneg

Target = 8
Output = 6



in case of
Target = 14

Step 02:
 $nums[0] \leq Target \leq nums[pivotIndex]$ → *Ans A*

Step 03:
 $ans = binarySearch(nums, 0, pivotIndex, Target);$
 $\quad \quad \quad s=0 \quad e=2 \quad Target = 14$

in case of
Target = B

Else {
 $ans = binarySearch(nums, pivotIndex+1, n-1, Target);$
 $\quad \quad \quad s=3 \quad e=7 \quad Target = 8$

```

// PROGRAM 02: Search in a rotated and sorted array (Leetcode-33)
int search(vector<int>& nums,int target){
    // Step 01
    int pivotIndex = findPivotIndex(nums);
    int n=nums.size();
    int ans=-1;

    // Step 02: Line A
    if(nums[0]<=target && target<=nums[pivotIndex]){
        // Step 03: Apply binary search
        ans=binarySearch(nums,0,pivotIndex,target);
    }
    // Step 02: Line B
    else{
        ans=binarySearch(nums,pivotIndex+1,n-1,target);
    }

    // return target index
    return ans;
}

/*
Example 01:
INPUT: [12,14,16,2,4,6,8,10] target=14
OUTPUT: Target element at index: 1

Example 02:
INPUT: [12,14,16,2,4,6,8,10] target=8
OUTPUT: Target element at index: 6
*/

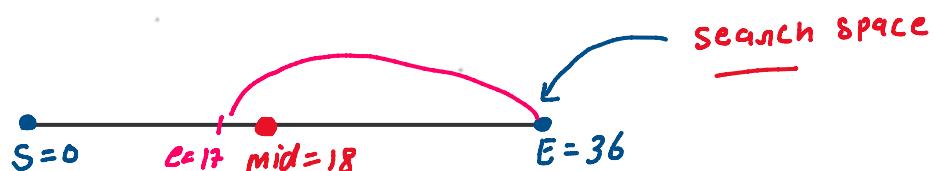
```

3. Sqrt of X (Leetcode-69)

| | | |
|---|--|--|
| <i>Example 01:</i> Input: x = 4 Output: 2 | <i>Example 02:</i> Input: x = 36 Output: 6 | <i>Example 03:</i> Input: x = 68 Output: 8 |
|---|--|--|

DRY RUN

*Example 02:
Input: x = 36*



$$mid = \frac{0+36}{2}$$

$$= 18$$

$$mid * mid > x$$

$$324 > 36$$

$$E = mid - 1;$$

$$E = 18 - 1$$

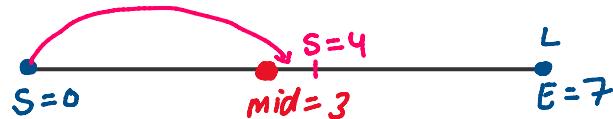
$$= 17$$





$$\begin{aligned} \text{Mid} &= \frac{0+17}{2} \\ &= 8 \end{aligned}$$

$$\begin{aligned} \text{Mid} * \text{mid} &> X \\ 64 &> 36 \\ E &= \text{mid}-1; \\ E &= 8-1 \\ &= 7 \end{aligned}$$



$$\begin{aligned} \text{Mid} &= \frac{0+7}{2} \\ &= 3 \end{aligned}$$

$$\begin{aligned} \text{Mid} * \text{mid} &< X \\ 9 &< 36 \\ \text{ans} &= \text{mid} \quad \textcircled{3} \\ S &= \text{mid}+1 \\ &= 3+1 \\ &= 4 \end{aligned}$$



$$\begin{aligned} \text{Mid} &= \frac{4+7}{2} \\ &= 5 \end{aligned}$$

$$\begin{aligned} \text{Mid} * \text{mid} &< X \\ 25 &< 36 \\ \text{ans} &= \text{mid} \quad \textcircled{5} \\ S &= \text{mid}+1 \\ &= 5+1 \\ &= 6 \end{aligned}$$



$$\begin{aligned} \text{Mid} &= \frac{6+7}{2} \\ &= 6 \end{aligned}$$

$$\begin{aligned} \text{Mid} * \text{mid} &== X \\ 36 &== 36 \\ \underline{\text{return mid}} &\quad \underline{\text{END}} \end{aligned}$$

```

● ● ●

// Program 03: Sqrt of X (Leetcode-69)
class Solution {
public:
    int mySqrt(int x) {
        int s = 0;
        int e = x;
        long long int mid = s+(e-s)/2;
        int ans = -1;

        while(s <= e){
            // kya mid hi to ans nhi hai to return mid
            if(mid*mid == x){
                return mid;
            }
            // agar mid*mid greater then hai x se to left me chale jao
            else if(mid*mid > x){
                e = mid-1;
            }
            // agar mid*mid less then hai x se to ans me mid store karlo and right me chale jao
            else{
                ans = mid;
                s = mid+1;
            }
            mid= s+(e-s)/2;
        }
        return ans;
    }
};


```

$T.C. \Rightarrow O(\log n)$

4. Binary search in 2D array (Leetcode-74)

| | 0 | 1 | 2 | 3 |
|---|----|----|----|----|
| 0 | 2 | 4 | 6 | 8 |
| 1 | 10 | 12 | 14 | 16 |
| 2 | 20 | 22 | 24 | 26 |

target = 14 output = True

Hum jante hai 2D array hamesha internally 1D array ki form me hi store hota hai

2D \longrightarrow 1D $index = C * i + j$
 ↳ Number of columns
 ↳ i = row index
 ↳ j = col index

1D \longrightarrow 2D $i = mid / col$ & $j = mid \% col$

$$\begin{aligned} \text{Row}(i) &= \text{mat.size();} \rightarrow 3 \\ \text{Col}(j) &= \text{mat[0].size();} \rightarrow 4 \\ \text{Total}(N) &= i * j; \rightarrow 12 \end{aligned}$$

Apply
Binary Search

$$\begin{aligned} s &= 0; \\ e &= n-1; \\ \text{mid} &= s + (e-s)/2; \end{aligned}$$

$$\left\{ \begin{array}{l} \text{RowIndex} = \text{mid}/\text{col}; \\ \text{ColIndex} = \text{mid} \% \text{col}; \end{array} \right.$$

DRY RUN

Iteration: 01

| | 0 | 1 | 2 | 3 |
|---|----|----|----|----|
| 0 | 2 | 4 | 6 | 8 |
| 1 | 10 | 12 | 14 | 16 |
| 2 | 20 | 22 | 24 | 26 |

$$\text{Target} = 14$$

| MAT | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 20 | 22 | 24 | 26 |
|-----|---|---|---|---|----|----|----|----|----|----|----|----|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |

$$s = 0$$

$$e = 11$$

$$\begin{aligned} \text{MID} &= \frac{0+11}{2} \\ &= 5 \end{aligned}$$

$$\begin{aligned} \text{RowIndex} &= \text{MID}/\text{col} \\ &= 5/4 \\ &= 1 \end{aligned}$$

$$\begin{aligned} \text{ColIndex} &= \text{MID} \% \text{col} \\ &= 5 \% 4 \\ &= 1 \end{aligned}$$

$$\begin{aligned} \text{CN} &= \text{mat}[1][1] \\ &= 12 \end{aligned}$$

(target > CN)

$$\rightarrow s = \text{mid} + 1$$

Iteration: 02

| MAT | 2 | 4 | 6 | 8 |
|-----|----|----|----|----|
| | 10 | 12 | 14 | 16 |
| 2 | 20 | 22 | 24 | 26 |

$$\text{Target} = 14$$

| MAT | 14 | 16 | 20 | 22 | 24 | 26 |
|-----|----|----|----|----|----|----|
| | 6 | 7 | 8 | 9 | 10 | 11 |

$$c = h \quad p = 11$$

| | | | | | | | |
|-----|---|----|----|----|----|----|------|
| MAT | 6 | 14 | 16 | 20 | 22 | 24 | 26 |
| | 6 | 7 | 8 | 9 | 10 | 11 | c=11 |

$$\text{MID} = \frac{6+11}{2}$$

$$= 8$$

$$\text{RowIndex} = \text{MID}/\text{Col}$$

$$= 8/4$$

$$= 2$$

$$\text{ColIndex} = \text{MID \% Col}$$

$$= 8 \% 4$$

$$= 0$$

$$CN = \text{mat}[2][0]$$

$$= 20$$

$(\text{target} < CN)$
 $\leftarrow e = \text{mid}-1;$

Iteration: 3

| | 0 | 1 | 2 | 3 |
|---|----|----|----|----|
| 0 | 2 | 4 | 6 | 8 |
| 1 | 10 | 12 | 14 | 16 |
| 2 | 20 | 22 | 24 | 26 |

target = 14

| | | |
|-----|----|-----|
| MAT | 14 | 16 |
| S=6 | 6 | C=7 |

$$\text{MID} = \frac{6+7}{2}$$

$$= 6$$

$$\text{RowIndex} = \text{MID}/\text{Col}$$

$$= 6/4$$

$$= 1$$

$$\text{ColIndex} = \text{MID \% Col}$$

$$= 6 \% 4$$

$$= 2$$

$$CN = \text{mat}[1][2]$$

$$= 14$$

$(\text{target} == CN)$
 $\rightarrow \text{Return True}$ ENI

```
● ○ ●

class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int rowSize = matrix.size();
        int colSize = matrix[0].size();
        int matrixSize = rowSize * colSize;

        int s = 0;
        int e = matrixSize - 1;
        int mid = s + (e - s) / 2;

        while(s <= e){
            int rowIndex = mid / colSize;
            int colIndex = mid % colSize;
            int currentNumber = matrix[rowIndex][colIndex];

            if(target==currentNumber){
                return true;
            }
            else if(target>currentNumber){
                s = mid + 1;
            }
            else{
                e = mid - 1;
            }
            mid=s+(e-s)/2;
        }
        return false;
    }
};
```