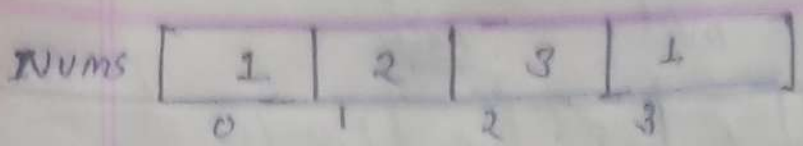
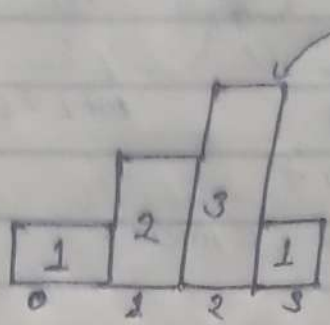
Find Peak Element (Leetcode-162)

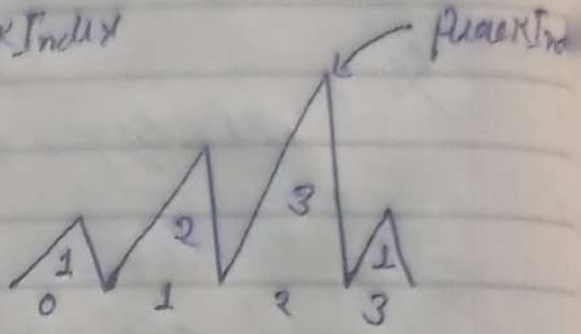
Binary SEARCH

Linkedin @manojofficialmj

[162] Find peak Element







Note 1: Peak Element is strictly greater than its neighbours.

↳ (3) > (1) && (3) > (2)

Note 2: Num[-1] = -∞

EDGE cases

Imagine

(-∞) < (1) > (2)

peak

-1 0 1

Note 3: Num[N] = -∞

N=2

(2) < (1) > (-∞)

peak

0 1 2

Note 4: N=1 ⇒ return 0

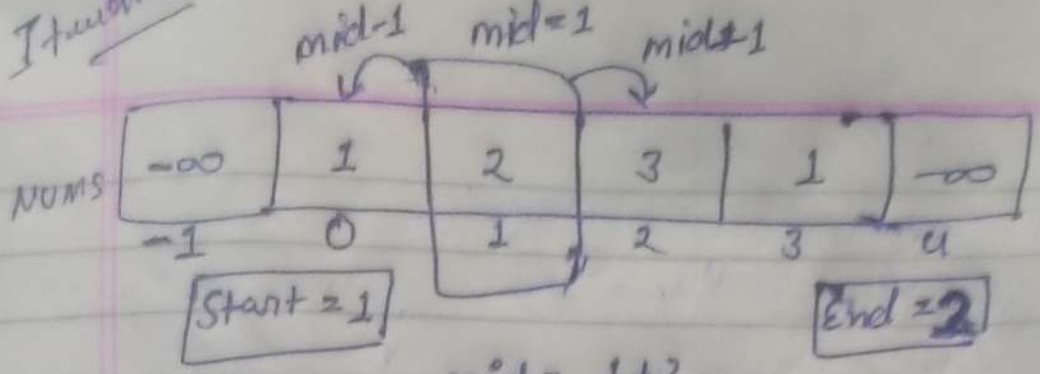
Num [1]

(-∞) < (1) > (-∞)

peak

-1 0 1

Iteration 1



$mid = \frac{1+2}{2}$

$= 1$

Nums[mid] > Nums[mid+1]

2 3 ✗

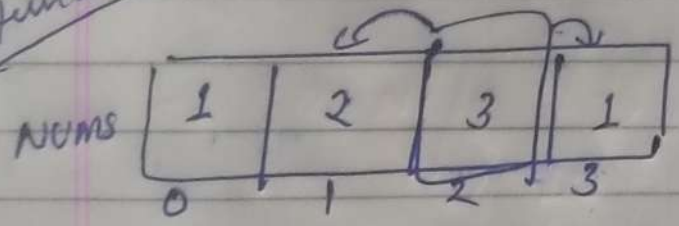
Nums[mid] > Num[mid-1]

2 1 ✓

→ Start = mid + 1

= 1 + 1 = (2)

Iteration 2



Start = 2

End = 2

mid = 2

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

T.C = O(log N)

S.C = O(1)

Nums[mid] > Nums[mid+1) ✗✗

3 2 ✓

Nums[mid] ≥ Nums[mid-1]

3 1 ✓

→ return mid

→ Condition Nums[mid] < Nums[mid-1]

End = mid - 1

```cpp
// Find Peak Element (Leetcode-162)
class Solution {
public:
    int findPeakElement(vector<int>& nums) {
        int n = nums.size();
        // Edge Case 1: Nums contains only one element
        if(n == 1){
            return 0;
        }
        // Edge Case 2: Nums[-1] = -∞
        if(nums[0] > nums[1]){
            return 0;
        }
        // Edge Case 3: Nums[n] = -∞
        if(nums[n-1] > nums[n-2]){
            return n-1;
        }

        // Now apply the Binary Search
        // Me already nums[0] and nums[n-1] ko check kar chuka hu
        int start = 1;
        int end = n - 2;
        int mid = start + (end - start) / 2;
        while(start <= end){
            if(nums[mid] > nums[mid+1] && nums[mid] > nums[mid-1]){
                // Found the peak index
                return mid;
            }
            else if(nums[mid] > nums[mid-1]){
                // Go to right side
                start = mid + 1;
            }
            else{
                // Go to left side
                end = mid - 1;
            }
            // Updating the mid
            mid = start + (end - start) / 2;
        }
        // when not found the peak index then return -1
        return -1;
    }
};
```