HOW MANY TIMES HAS ARRAY BEEN ROTATED

☆ In this problem, we are given an array of integers that was initially sorted but is now rotated. We need to find and return how many times it was rotated.

We use exactly the same logic as finding minimum in rotated sorted array but instead of the element we return the index of the minimum element.

Pseudocode :

```
minimumInRotatedSortedArray (arr, N) {
    low = 0
    high = N - 1
    ans = 0
    while (low <= high) {
        mid = (low + high)/2
        if (arr[mid] < arr[ans]) {
            ans = mid
        } else if (arr[mid] <= arr[high]) {
            ans = arr[mid] > arr[ans] ? ans : mid
            high = mid
        } else {
            ans = arr[low] > arr[ans] ? ans : low
            low = mid
        }
    }
```

```
        }
    return ans
}
```