MINIMUM IN ROTATED SORTED ARRAY

☆ In this problem, we are supposed to return the minimum element from an array of integers which was initially sorted but then rotated.

Brute force solution is to use linear search but we have to use binary search. Very similar to our previous two questions, we will store an ans variable. We check if the middle element is lesser than our stored minimum if it is, it is the new minimum. We now figure out which half of the array is sorted and compare its minimum. Then we continue our search in the other half of the array.

Pseudocode :

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

```
            high = mid
        } else {
            ans = min(ans, arr[low])
            low = mid
        }
    }
    return ans
}
```