**BINARY\_SEARCH**

**Sorted elements or COnstrainsts or Numbers think Me.**

**Q)** [**Find First and Last Position of Element in Sorted Array**](https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/)

**Soln: First find left and than right, Separate BS’s for both.**

**Desc:Individually find out first and last element in array by using Binary search.**

**2Q)** [**Search in Rotated Sorted Array**](https://leetcode.com/problems/search-in-rotated-sorted-array/)

**Soln: First check whether array is reverse by mid and first and compare with target for low and high moving.if(input[mid] >= input[lo]) {**

**if(target >= input[lo] and target <= input[mid]) {**

**hi = mid - 1;**

**} else {**

**lo = mid + 1;**

**}**

**}else {**

**if(target >= input[mid] and target <= input[hi]) {**

**lo = mid + 1;**

**} else {**

**hi = mid - 1;**

**}**

**}**

**Desc: check with mid and calculate element by moving low and right.**

**3Q)** [**Single Element in a Sorted Array**](https://leetcode.com/problems/single-element-in-a-sorted-array/)

**SolN: if((mid-1>=0 && nums[mid] != nums[mid-1])&&(mid+1<n && nums[mid] != nums[mid+1]) || (mid-1<0 && mid+1 < n && nums[mid] != nums[mid+1]) || (mid+1>=n && mid-1>=0 && nums[mid] != nums[mid-1]))return nums[mid];**

**else if((mid - 1 >= 0 && nums[mid] == nums[mid-1] && mid%2 == 0) || (mid + 1 < n && nums[mid] == nums[mid+1] && mid%2 != 0) ){**

**j = mid-1;**

**}else i = mid+1;**

**Desc: Figure out which direction to move on absed on index as Twice repeating is their in array.**