

Debug Exercise

Date...3/10/23...

① → Search in sorted rotated array
else if (nums [0] ≤ nums [mid])

② → Find Square root of x →

last = mid - 1;

else → first = mid + 1;

③ → Find minimum in rotated sorted array

s = 0

e = n - 1 → changed this

ans = start

while (nums [s] > nums [e])

changed this ← m = s + e / 2; // m = (s + e) >> 1;

if (nums [m] ≤ nums [s] && ~~nums~~
nums [m] ≤ nums [e])
e = m;

changed this ← else if (nums [m] > nums [s] && nums [m] > nums [e])
s = m + 1;

else return nums [m];
ans = s;

}

return nums [ans];

Date... 5/10/23

④ → A peak element is an element that is strictly greater than its neighbors. Find Peak element

$n = \text{nums.size}();$

$l = 0;$

$h = n - 1;$

while ($l < h$) {

$m = (l + h) >> 1;$ → changed this

Remove equal sign ← if ($\text{nums}[m] > \text{nums}[m+1]$)
high = m;

}

else {

$l = m + 1;$ // added +1

}

return low; → Return low

~~③~~ find the length of longest strictly increasing Subsequence

int length of LIS (vector<int> & nums) {

vector<int> ans;

ans.push_back(nums[0]);

for (int i = 0; i < nums.size(); i++) {

if ($\text{nums}[i] < \text{ans.back}()$)

ans.push_back(nums[i]);

} else {

int index = lower_bound(ans.begin(), ans.end(), nums[i]) - ans.begin();

ans[index] = nums[i];

APCO

Teacher's Sign.....


```

    }
    return ans.size();
}

```

Lower bound \rightarrow Returns pos, if element is present
then returns next pos

⑥ \rightarrow Selection Sort

No change here

⑦ \rightarrow Debug the code

```

binary Search  $\rightarrow$  while (l < r) {
    m = (l+r)/2;
    if (nums[m] <= target) {
        l = m+1;
    } else {
        r = m;
    }
}

```

change this \rightarrow add +1

```

    l to l-1
    if (left >= 0 & nums[l-1] == target) {
        return l-1;
    } else {
        return -1;
    }
}

```

also can use $\text{if } r = \text{nums.size()} - 1$

Then can return l also
& $\text{nums}[l] = \text{target}$