DATE

ARRAYS | LEETCODE 33 |

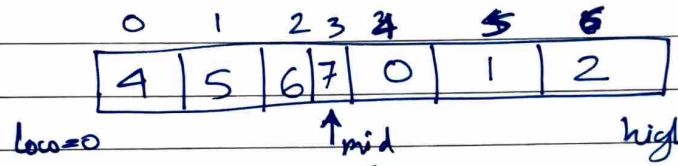
Search in Rotated Sorted Array

Input: nums = [4, 5, 6, 7, 0, 1, 2]

target : 0

1) Sorted → We can use Binary Search
2) But it is rotated, so one direction will be sorted
3) Find out which part of mid element is sorted

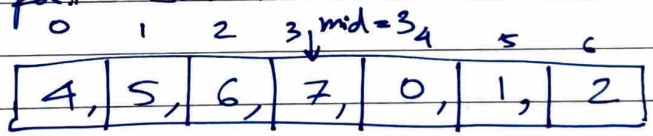
int low = 0, int high = nums.Length - 1 = 7 - 1 = 6



mid = 0 + (6-0)/2 = 3

Case 1: target == nums[mid] → return mid;

Check which part is sorted



nums[mid] < nums[low] → right part is sorted

Case 2: Left Part Sorted

nums[mid] > nums[low] - left part is sorted

Satisfies ↓

```
if ( target ≥ nums[low] && target ≤ nums[mid] )
{
    high = mid - 1;
}
else {
    low = mid + 1;
}
```

eliminate one part

DATE

Case 3: Right Part is sorted

```
if (target ≥ nums[mid] && target ≤ nums[high]){
        low = mid+1;
}
else{
        high = mid-1;
}
```

Case 4: If nothing is satisfied for (low ≤ high) condition above, return -1;