Program 44. Search in Rotated Sorted Array

There is an integer array nums sorted in ascending order (with distinct values).

Prior to being passed to your function, nums is possibly rotated at an unknown pivot index

k (1 <= k < nums.length) such that the resulting array is [nums[k], nums[k+1], ..., nums[n 1], nums[0], nums[1], ..., nums[k-1]] (0-indexed). For example, [0,1,2,4,5,6,7] might be

rotated at pivot index 3 and become [4,5,6,7,0,1,2].

Given the array nums after the possible rotation and an integer target, return the index of

target if it is in nums, or -1 if it is not in nums.

PROGRAM:

```
ddef search(nums, target):
    left, right = 0, len(nums) - 1
    while left <= right:
        mid = (left + right) // 2
        if nums[mid] == target:
            return mid
        if nums[left] <= nums[mid]:
            if nums[left] <= target < nums[mid]:
                right = mid - 1
            else:
                left = mid + 1
        else:
            if nums[mid] < target <= nums[right]:
                left = mid + 1
            else:
                right = mid - 1
    return -1

# Example usage
nums = [4, 5, 6, 7, 0, 1, 2]
target = 0
print(search(nums, target))  # Output: 4
```

Output::

```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA COADS.PYTHON\program 44.py"
4

Process finished with exit code 0
```

Time complexity:
O(log n)