

## Arrays

### (Assignment Questions)

#### (EASY)

**Question 1** : Given an integer array `nums`, return true if any value appears at least twice in the array, and return false if every element is distinct. [\[link\]](#)

*Examples :*

*Input:* `nums = [1,2,3,4]`

*Output:* `false`

*Input:* `nums = [1,1,1,3,3,4,3,2,4,2]`

*Output:* `true`

#### (MEDIUM)

**Question 2** : 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  $k$  ( $1 \leq k < \text{nums.length}$ ) 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`.

You must write an algorithm with  $O(\log n)$  runtime complexity. [\[link\]](#)

*Examples :*

*Input:* `nums = [4,5,6,7,0,1,2], target = 0`*Output:*

`4`

*Input:* `nums = [4,5,6,7,0,1,2], target = 3`*Output:* -

`1`

**(MEDIUM)**

**Question 3** : Given an integer array `nums`, find a subarray that has the largest product, and return the product. The test cases are generated so that the answer will fit in a 32-bit integer. [\[link\]](#)

**Note** - This Qs might feel difficult as a beginner because it uses DP approach.

*Examples :*

*Input: `nums = [2,3,-2,4]`*

*Output: 6*

*Explanation: `[2,3]` has the largest product 6.*

*Input: `intervals =nums = [-2,0,-1]`*

*Output: 0*

*Explanation: The result cannot be 2, because `[-2,-1]` is not a subarray.*