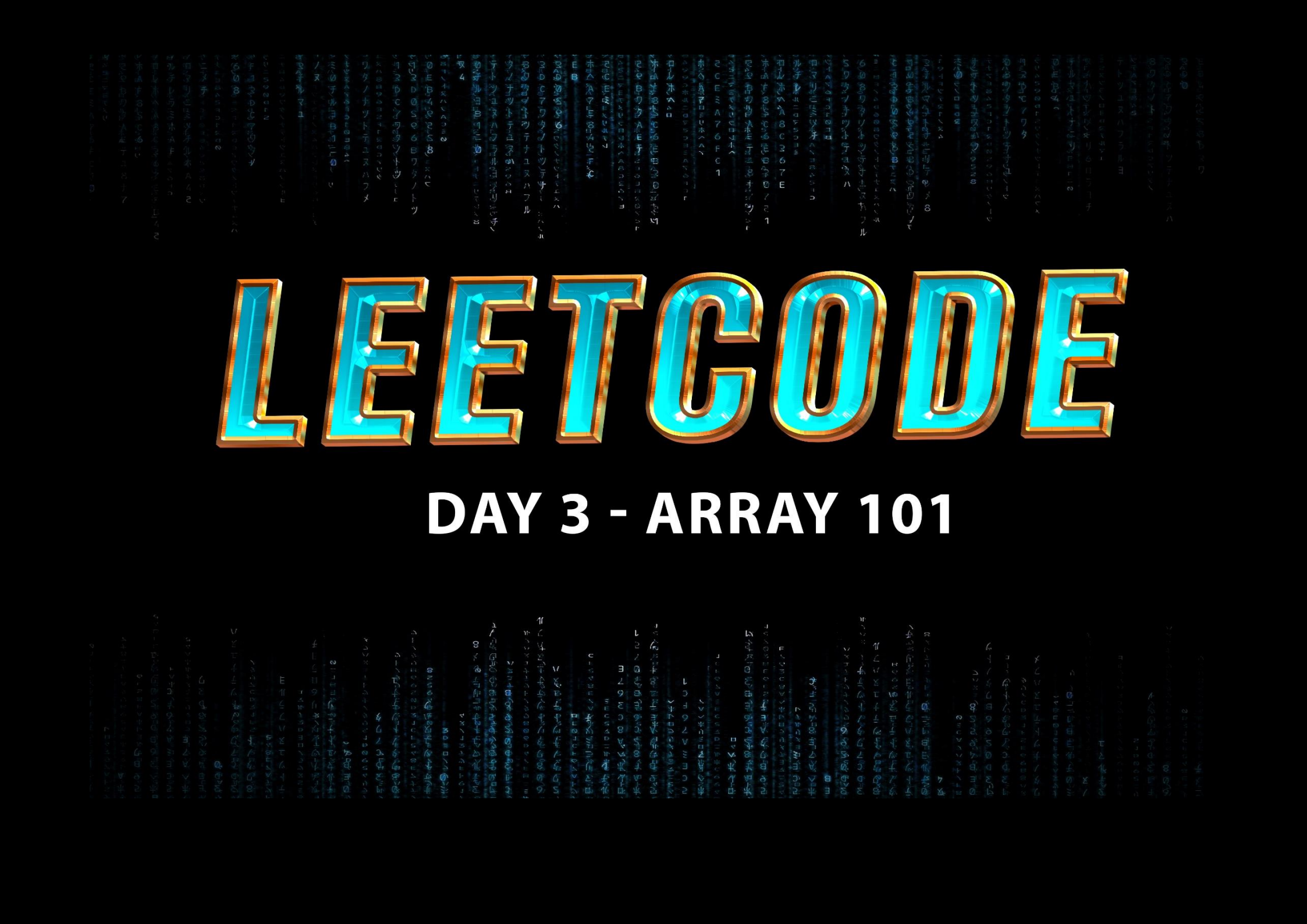
LEETCODE
DAY 3 - ARRAY 101

Problem List

1. Two Sum
26. Remove Duplicates from Sorted Array
189. Rotate Array
53. Maximum Subarray
11. Container With Most Water
54. Spiral Matrix
217. Contains Duplicate
219. Contains Duplicate II
605. Can Place Flowers
744. Find Smallest Letter Greater Than Target

Problem 1: 1. Two Sum

Given an array of integers nums and an integer target, return *indices of the two numbers such that they add up to* target.You may assume that each input would have *exactly* **one solution**, and you may not use the *same* element twice. You can return the answer in any order.

Solution:

TwoSum.py

```python
class Solution:
    def twoSum(self, nums: List[int], target: int) -> List[int]:
        ret = [0, 1]
        for i in range(len(nums)):
            for j in range(i + 1, len(nums)):
                if nums[i] + nums[j] == target:
                    ret[0] = i
                    ret[1] = j
        return ret
```

Problem 2: 26. Remove Duplicates from Sorted Array

Given an integer array nums sorted in **non-decreasing order**, remove the duplicates **in-place** such that each unique element appears only **once**. The **relative order** of the elements should be kept the **same**. Then return *the number of unique elements in* nums.

Solution:

removeDuplicates.py

```
class Solution {
  public int removeDuplicates(int[] nums) {
    int i=0;
    for(int a:nums){
      if(i<1||a>nums[i-1])
        nums[i++]=a;
    }
    return i;
  }
}
```

Problem 3: 189. Rotate Array

Given an integer array nums, rotate the array to the right by k steps, where k is non-negative.

Solution:

removeDuplicates.py

```python
class Solution:
    def rotate(self, nums, k):
        n=len(nums)
        k=k%n
        nums[:]=nums[n-k:]+nums[:n-k]
```

Problem 4: 53. Maximum Subarray

Given an integer array nums, find the subarray with the largest sum, and return its sum.

Solution:

removeDuplicates.py

```python
class Solution:
    def maxSubArray(self, nums: List[int]) -> int:
        c=g=nums[0]
        for num in nums[1:]:
            c=max(num,c+num)
            if c>g:
                g=c
        return g
```

Problem 5: 11. Container With Most Water

You are given an integer array height of length n. There are n vertical lines drawn such that the two endpoints of the ith line are (i, 0) and (i, height[i]). Find two lines that together with the x-axis form a container, such that the container contains the most water.Return *the maximum amount of water a container can store*.

Solution:

py

```py
class Solution:
    def maxArea(self, height: List[int]) -> int:
        a = float()
        for i in range(len(height)):
            for j in range(i + 1, len(height)):
                b=(j-i)*min(height[i], height[j])
                a=max(a,b)
        return a
```

Problem 6: 54. Spiral Matrix

Given an m x n matrix, return *all elements of the* matrix *in spiral order*.

Solution:

removeDuplicates.py

```python
class Solution:
    def spiralOrder(self, matrix: List[List[int]]) -> List[int]:
        m=len(matrix)
        n=len(matrix[0])
        result=[]
        if m==0:
            return result
        seen = [[False] * n for _ in range(m)]
        dr = [0,1,0,-1]
        dc = [1,0,-1,0]
        r,c=0, 0
        di=0
        for i in range(m*n):
            result.append(matrix[r][c])
            seen[r][c]=True
            newR,newC=r+dr[di],c+dc[di]
            if 0<=newR<m and 0<=newC<n and not seen[newR][newC]:
                r,c=newR, newC
            else:
                di=(di+1)%4
                r+=dr[di]
                c+=dc[di]
        return result
```

Problem 7: 217. Contains Duplicate

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.

Solution:

removeDuplicates.py

```python
class Solution:
    def containsDuplicate(self, nums: List[int]) -> bool:
        nums.sort()
        print(nums)
        for i in range(len(nums)-1):
            if(nums[i]==nums[i+1]):
                return True
        return False
```

Problem 8 : 219. Contains Duplicate II

Given an integer array nums and an integer k, return true if there are two distinct indices i and j in the array such that nums[i] == nums[j] and abs(i - j) <= k.

Solution:

removeDuplicates.py

```python
class Solution:
    def containsNearbyDuplicate(self, nums: List[int], k: int) -> bool:
        seen=set()
        for i, num in enumerate(nums):
            if num in seen:
                return True
            seen.add(num)
            if len(seen)>k:
                seen.remove(nums[i - k])
        return False

```

Problem 9: 605. Can Place Flowers

Given an integer array flowerbed containing 0's and 1's, where 0 means empty and 1 means not empty, and an integer n, return true *if* n *new flowers can be planted* in the flowerbed without violating the no-adjacent-flowers rule and false otherwise.

Solution:

removeDuplicates.py

```python
class Solution:
    def canPlaceFlowers(self, flowerbed: list[int], n: int) -> bool:
        count = 0
        flowerbed = [0] + flowerbed + [0]
        for i in range(1, len(flowerbed) - 1):
            if flowerbed[i - 1:i + 2] == [0, 0, 0]:
                count += 1
                flowerbed[i] = 1
            if count>=n:
                return True
        return False
```

Problem 10: 744. Find Smallest Letter Greater Than Target

You are given an array of characters letters that is sorted in **non-decreasing order**, and a character target. There are **at least two different** characters in letters. Return *the smallest character in* letters *that is lexicographically greater than* target. If such a character does not exist, return the first character in letters.

Solution:

removeDuplicates.py

```python
class Solution:
    def nextGreatestLetter(self, letters: List[str], target: str) -> str:
        letters.sort()
        for i in letters:
            if(ord(target)<ord(i)):
                return i
        return letters[0]
```