# 题目

给你一个整数数组 nums ，设计算法来打乱一个没有重复元素的数组。

实现 Solution class:

Solution(int[] nums) 使用整数数组 nums 初始化对象

int[] reset() 重设数组到它的初始状态并返回

int[] shuffle() 返回数组随机打乱后的结果

示例：

输入

["Solution", "shuffle", "reset", "shuffle"]

[[[1, 2, 3]], [], [], []]

输出

[null, [3, 1, 2], [1, 2, 3], [1, 3, 2]]

解释

Solution solution = new Solution([1, 2, 3]);

solution.shuffle(); // 打乱数组 [1,2,3] 并返回结果。任何 [1,2,3]的排列返回的概率应该相同。例如，返回 [3, 1, 2]

solution.reset(); // 重设数组到它的初始状态 [1, 2, 3] 。返回 [1, 2, 3]

solution.shuffle(); // 随机返回数组 [1, 2, 3] 打乱后的结果。例如，返回 [1, 3, 2]

提示：

1 <= nums.length <= 200

-106 <= nums[i] <= 106

nums 中的所有元素都是 唯一的

最多可以调用5 \* 104次reset和shuffle

# 分析