

0164. 最大间距

👤 ITCharge ⌚ 大约 2 分钟

- 标签：数组、桶排序、基数排序、排序
- 难度：困难

题目链接

- [0164. 最大间距 - 力扣](#)

题目大意

描述： 给定一个无序数组 *nums*。

要求： 找出数组在排序之后，相邻元素之间最大的差值。如果数组元素个数小于 2，则返回 0。

说明：

- 所有元素都是非负整数，且数值在 32 位有符号整数范围内。
- 请尝试在线性时间复杂度和空间复杂度的条件下解决此问题。

示例：

- 示例 1:

输入：nums = [3,6,9,1]

输出：3

解释：排序后的数组是 [1,3,6,9]，其中相邻元素 (3,6) 和 (6,9) 之间都存在最大差值 3。

py

- 示例 2:

输入：nums = [10]

输出：0

解释：数组元素个数小于 2，因此返回 0。

py

解题思路

思路 1：基数排序

这道题的难点在于要求时间复杂度和空间复杂度为 $O(n)$ 。

这道题分为两步：

1. 数组排序。
2. 计算相邻元素之间的差值。

第 2 步直接遍历数组求解即可，时间复杂度为 $O(n)$ 。所以关键点在于找到一个时间复杂度和空间复杂度为 $O(n)$ 的排序算法。根据题意可知所有元素都是非负整数，且数值在 32 位有符号整数范围内。所以我们可以选择基数排序。基数排序的步骤如下：

- 遍历数组元素，获取数组最大值元素，并取得位数。
- 以个位元素为索引，对数组元素排序。
- 合并数组。
- 之后依次以十位，百位，...，直到 值元素的最高位处值为索引，进行排序，并合并数组，最终完成排序。

最后，还要注意数组元素个数小于 2 的情况需要特别判断一下。

思路 1：代码

```
class Solution:
    def radixSort(self, arr):
        size = len(str(max(arr)))

        for i in range(size):
            buckets = [[] for _ in range(10)]
            for num in arr:
                buckets[num // (10 ** i) % 10].append(num)
            arr.clear()
            for bucket in buckets:
                for num in bucket:
                    arr.append(num)
```

py

```
        return arr

def maximumGap(self, nums: List[int]) -> int:
    if len(nums) < 2:
        return 0
    arr = self.radixSort(nums)
    return max(arr[i] - arr[i - 1] for i in range(1, len(arr)))
```

思路 1：复杂度分析

- 时间复杂度： $O(n)$ 。
- 空间复杂度： $O(n)$ 。