

桶排序

题目：347. Top K Frequent Elements

语言: python3

英文版链接: <https://leetcode.com/problems/top-k-frequent-elements/>

中文版链接: <https://leetcode-cn.com/problems/top-k-frequent-elements/>

题目分析

给定一个非空的整数数组，返回其中出现频率前 **k** 高的元素。

示例 1:

输入: `nums = [1,1,1,2,2,3]`, `k = 2`

输出: `[1,2]`

示例 2:

输入: `nums = [1]`, `k = 1`

输出: `[1]`

本题考验的知识点是桶排序，希望大家用桶排序的思想来做。关于桶排序：

<https://blog.csdn.net/YinhJiang/article/details/80397415>

题解

我们可以设置若干个桶，每个桶存储出现频率相同的数，并且桶的下标代表桶中数出现的频率，即第 *i* 个桶中存储的数出现的频率为 *i*。

把数都放到桶之后，从后向前遍历桶，最先得到的 *k* 个数就是出现频率最多的的 *k* 个数。

```
class Solution:
    def topKFrequent(self, nums: 'List[int]', k: 'int') -> 'List[int]':
        frequent_of_number = {}
        for num in nums:
            frequent_of_number[num] = frequent_of_number.get(num, 0) + 1
        # 建桶
        buckets = [[] for i in range(len(nums) + 1)] # 如果用[]*(len(nums)+1)则是浅拷贝
        for key, value in frequent_of_number.items():
            buckets[value].append(key)
        print(buckets)
        result = []
        for x in range(len(nums), -1, -1):
            if k > 0 and buckets[x]:
                result += buckets[x]
                k -= len(buckets[x])
            if k == 0:
                return result
```