

179. 最大数

给定一组非负整数
nums，重新排列它们每个数字
的顺序（**每个数字不可拆分**）
使之组成一个最大的整数。

排序？什么排序可以做？
冒泡？快排？归并……

每个数字不可拆分！！

直接拼接数字，可能导致数值溢出，这是一个隐形的大数问题，需要把数字转换成字符串。然后需要定义一种比较两个数的规则，即把数字m和n拼接为mn，nm，只需要按照字符串大小的比较。

基本思想：
整数变为字符串，排序

示例 1:

输入: nums = [10,2]
输出: "210"

示例 2:

输入: nums = [3,30,34,5,9]
输出: "9534330"

示例 3:

输入: nums = [1]
输出: "1"

示例 4:

输入: nums = [10]
输出: "10"

其他办法???

递归:

首先比较两者前缀的字典序大小
若前缀不相等，则字典序大的应该放前面
若前缀相等，递归比较长字符串的后缀和短字符串

链表:

分解: 将每个数字分解成一个链表。
比较: 比较两个数字如何组合最大，返回在前面的链表。
排序: 基于比较函数，对链表列表进行排序。
结果: 基于有序数组生成结果

Python3

智能模式

```
1 class Solution:
2     def largestNumber(self, nums: List[int]) -> str:
3
4         # 拼接数字: 把数字转换为字符串, 比较两个数xy和yx的大小
5         if not nums:
6             return ''
7         nums = map(str, nums)
8         key = cmp_to_key(lambda x, y: int(y + x) - int(x + y))
9
10        # lstrip() 方法: 截掉字符串左边的空格或指定字符 0010->10
11
12        res = ''.join(sorted(nums, key=key)).lstrip('0')
13        # 000->' '
14        return res or '0'
```

函数解释:

map()函数 会根据提供的函数对指定序列做映射。

sorted()函数 对所有可迭代的对象进行排序操作。

join()方法 用于将序列中的元素以指定的字符连接生成一个新的字符串。

复杂度分析
时间复杂度: $O(n \lg n)$
空间复杂度: $O(n)$