

19th January 2015 [LeetCode] 179 Largest Number

Find Other LeetCode Python Answers: <https://github.com/wqr619/leetcode> [<https://github.com/wqr619/leetcode>]

## [LeetCode] 179 Largest Number

Given a list of non negative integers, arrange them such that they form the largest number.

For example, given `[3, 30, 34, 5, 9]`, the largest formed number is `9534330`.

Note: The result may be very large, so you need to return a string instead of an integer.

### Easy Explanation:

#### First Thought:

At first, I tried hard to find out the rules behind the sequence. For instance, find the number with largest first digit '9', then '5'; The problem is when numbers with same first digit like '34', '30', '3'. which number to choose? If you think more, the problem would become very complex. But while you're thinking about the rules behind finding the largest number, don't you realize the solution to the problem is very similar to sorting the numbers. The order we're looking for is [9,5,34,3,30]. But unlike regular descending sorting, 5 is ahead 34 not because  $5 > 34$  but because  $'534' > '345'$ . Therefore, it's all about **customized sorting algorithm**.

#### Algorithm -Customized Sorting:

Sort the array by different comparison function. For most programming languages, you can change the comparison function of the built-in sort function. Therefore, while sorting the array, you're choosing number  $i$  ahead of number  $j$  when  $'i'+j' > 'j'+i'$

#### Careful Edge Cases:

The requirement is to return a string not an integer, then you should be careful with integer format. Like '00' is illegal integer format. If you don't want to python built-in `int()` function to transfer integer to string. Then you should be cut unnecessary zeros your-self. However, use `int()` is also OK and won't affect your runtime.

## Python Solution:

```
class Solution:
    # @param num, a list of integers
    # @return a string
    def largestNumber(self, num):
        # Define customized compare function for sorting
        def compare(n1, n2):
            if n1+n2 > n2+n1:
                return 1
            elif n1+n2 < n2+n1:
                return -1
            else:
                return 0

        num_str = [str(n) for n in num]
        res = ""

        # Sorting according to customized function
        for n in reversed( sorted(num_str,cmp=compare) ):
            res += n

        # Remove unnecessary zeros in head
        res_list = list(res)
        i = 0
        while res_list[i] == '0' and i != len(res)-1:
            i += 1
        res_list = res_list[i:]

        return ''.join( res_list)
```

Lazy way to remove unnecessary zeros.


```
class Solution:
    # @param num, a list of integers
    # @return a string
    def largestNumber(self, num):
        # Define customized compare function for sorting
        def compare(n1, n2):
            if n1+n2 > n2+n1:
                return 1
            elif n1+n2 < n2+n1:
                return -1
            else:
                return 0

        num_str = [str(n) for n in num]
        res = ""
```

```
# Sorting according to customized function
for n in reversed( sorted(num_str,cmp=compare) ):
    res += n

# Remove unnecessary zeros in head
return str( int(res) )
```

Posted 19th January 2015 by [Anonymous](#)

 2 View comments