

Strings

7. Reverse Integer

Reverse String

解法一：一重For 循环反向重组

451. Sort Characters By Frequency

HashMap; Bucket Sort

Given a string, sort it in decreasing order based on the frequency of characters.

解法一：Bucket Sort + Ordered Map

因为仅有256个字符，可以创建一个长度为256的数组来存储每个字符的出现频率。然后使用一个`<int, vector<char>>map`来存储出现频率为i的字符有哪些。最后使用`reverse_iterator`输出map即可。

```
class Solution {
public:
    string frequencySort(string s) {
        int charMap[256];
        memset(charMap, 0, sizeof(charMap));
        map<int, vector<char>> freqMap;
        for (int i = 0; i < s.length(); ++i)
            charMap[(int) s[i]]++;
        for (int i = 0; i < 255; ++i)
        {
            if (charMap[i] != 0)
                freqMap[charMap[i]].push_back((char) i);
        }
        stringstream ss;
        for (map<int, vector<char>>::reverse_iterator itr =
freqMap.rbegin();
            itr != freqMap.rend(); ++itr)
        {
            for (int i = 0; i < itr->second.size(); ++i)
```

```
        ss << string(itr->first, itr->second[i]);
    }
    return ss.str();
};
```

解法二：Bucket Sort + 循环确定频率

可以不使用Map而直接使用二重循环来将频率降序排列的，从而节省Map占用的空间。

476. Number Complement

Reverse String; Binary Arithmetic

Given a positive integer, output its complement number. The complement strategy is to flip the bits of its binary representation.

解法一：进制转换 + Reverse

十进制转二进制

对于一个十进制正整数

1. 对其 mod 2 取余数
2. 将其 div 2 之后的数（如果不为0）继续重复第一步
3. 重复以上步骤直到得到0为止
4. 将所有余数反向重组成为这个数的二进制表示，可以调用C++ Reverse 函数

二进制转十进制

调用C++ stoi 函数，设定进制为2

解法二：二进制减法取反

参考

一个数的二进制互补数是大于该数的最小11...11减去这个数，例如： 94 = 1011110 -> 1111111 - 1011110 = 100001 = 33 94 -> 33

```
class Solution {  
    public int findComplement(int num) {  
        int i = 0;  
        int j = 0;  
        while (i < num)  
        {  
            i += Math.pow(2, j);  
            j++;  
        }  
        return i - num;  
    }  
}
```