**难度: 简单**

给定两个字符串形式的非负整数 num1 和num2 ，计算它们的和。

提示：

num1 和num2 的长度都小于 5100

num1 和num2 都只包含数字 0-9

num1 和num2 都不包含任何前导零

你不能使用任何內建 BigInteger 库， 也不能直接将输入的字符串转换为整数形式

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/add-strings

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

思路:

1). 分别从两个字符串的数组的最后一位开始遍历

2). 并且定义一个变量用来表示是否需要进位

3). 遍历循环每次获取最后的(两个数值+位数)模10来表示相加的时

4). 用(两个数值+位数)除10,来确定是否需要进1

|  |
| --- |
| public String addStrings(String num1, String num2) {  int num1Index = num1.length() - 1;  int num2Index = num2.length() - 1;  StringBuilder sb = new StringBuilder();  int carry = 0; //用来表示进位  while(num1Index >= 0 || num2Index >= 0 || carry > 0) {  int x = num1Index >= 0 ? (num1.charAt(num1Index) - '0') : 0;  int y = num2Index >= 0 ? (num2.charAt(num2Index) - '0') : 0;  sb.append( (x+y+carry)%10 );  carry = (x+y+carry)/10;  num1Index--;  num2Index--;  }  //进行颠倒,因为我们每次最后1位都是放在第一位  return sb.reverse().toString();  } |