# [字符串相乘](https://leetcode-cn.com/explore/interview/card/bytedance/242/string/1015/)

## 题目

给定两个以字符串形式表示的非负整数 num1 和 num2，返回 num1 和 num2 的乘积，它们的乘积也表示为字符串形式。

示例 1:  
  
输入: num1 = "2", num2 = "3"  
输出: "6"

* num1 和 num2 的长度小于110。
* num1 和 num2 只包含数字 0-9。
* num1 和 num2 均不以零开头，除非是数字 0 本身。
* 不能使用任何标准库的大数类型（比如 BigInteger）或直接将输入转换为整数来处理。

## 解题思路

1. 对于字符串 num2 中的每一位数与字符串 num1 相乘所得的结果，不再分开计算最后相加，而是先全部累加，最后再考虑进位的影响。
2. 对于最终结果的第i + j位数，可以由 num1 数组的第 i 位数和 num2 数组的第 j 位数组成。

public String multiply(String num1, String num2) {  
 if (num1.length() == 0 || num2.length() == 0) {  
 return ZEO;  
 }  
  
 if (num1.equals(ZEO) || num2.equals(ZEO)) {  
 return ZEO;  
 }  
  
 if (num1.equals(ONE)) {  
 return num2;  
 }  
  
 if (num2.equals(ONE)) {  
 return num1;  
 }  
  
 int[] num = new int[num1.length() + num2.length() - 1];  
 Arrays.fill(num, 0);  
 for (int i = 0; i < num1.length(); i++) {  
 for (int j = 0; j < num2.length(); j++) {  
 num[i + j] += (num1.charAt(i) - '0') \* (num2.charAt(j) - '0');  
 }  
 }  
  
 StringBuilder res = new StringBuilder();  
 int addIn = 0;  
 for (int i = num.length - 1; i >= 0; i--) {  
 int t = num[i] + addIn;  
 addIn = t / 10;  
 res.append(t % 10);  
 }  
  
 if (addIn > 0) {  
 res.append(addIn);  
 }  
 return res.reverse().toString();  
}