

165 比较版本号

Label: 字符串

给你两个版本号 `version1` 和 `version2`，请你比较它们。

版本号由一个或多个修订号组成，各修订号由一个 '.' 连接。每个修订号由 多位数字 组成，可能包含 前导零 。每个版本号至少包含一个字符。修订号从左到右编号，下标从 0 开始，最左边的修订号下标为 0，下一个修订号下标为 1，以此类推。例如，`2.5.33` 和 `0.1` 都是有效的版本号。

比较版本号时，请按从左到右的顺序依次比较它们的修订号。比较修订号时，只需比较 忽略任何前导零后的整数数值 。也就是说，修订号 1 和修订号 001 相等 。如果版本号没有指定某个下标处的修订号，则该修订号视为 0 。例如，版本 `1.0` 小于版本 `1.1`，因为它们下标为 0 的修订号相同，而下标为 1 的修订号分别为 0 和 1， $0 < 1$ 。

返回规则如下：

- 如果 `version1 > version2` 返回 1，
- 如果 `version1 < version2` 返回 -1，
- 除此之外返回 0。

- 分隔 转换

```
class Solution {
    public int compareVersion(String version1, String version2) {
        String[] array1 = version1.split("\\.");
        String[] array2 = version2.split("\\.");
        int length = array1.length > array2.length ?
array1.length:array2.length;    // 存储打的
        for (int i = 0; i < length; i++) {
            int a = i < array1.length ? Integer.parseInt(array1[i]):0;
            int b = i < array2.length ?Integer.parseInt(array2[i]):0;
            if (a > b) {
                return 1;
            }else if (a < b) {
                return -1;
            }
        }
        return 0;
    }
}
```

- 分块计算

```
class Solution {
    public int compareVersion(String version1, String version2) {

        // 两个字符串的长度
        int n = version1.length(), m = version2.length();

        int i = 0;
        int j = 0;
        while (i < n || j < m) {
            // 用v1,v2来计算每一个块中版本号的大小
            int v1 = 0;
            int v2 = 0;

            // 若当前的字符不是分隔符，则计算
            while (i < n && version1.charAt(i) != '.') {
                v1 = v1 * 10 + version1.charAt(i) - '0'; // 字符相减代表数字
                i++;
            }

            while (j < m && version2.charAt(j) != '.') {
                v2 = v2 * 10 + version2.charAt(j) - '0';
                j++;
            }

            // 判断当前块中的版本号是否一致
            if (v1 != v2) {
                if (v1 > v2) {
                    return 1;
                }
                return -1;
            }

            // 跳过分隔符
            i++;
            j++;
        }

        // 全部比较完了，没有不等的则返回0
        return 0;
    }
}
```