1. 题目
2. 题干

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。

1. 示例

示例 1：

输入：version1 = "1.01", version2 = "1.001"

输出：0

解释：忽略前导零，"01" 和 "001" 都表示相同的整数 "1"

示例 2：

输入：version1 = "1.0", version2 = "1.0.0"

输出：0

解释：version1 没有指定下标为 2 的修订号，即视为 "0"

示例 3：

输入：version1 = "0.1", version2 = "1.1"

输出：-1

解释：version1 中下标为 0 的修订号是 "0"，version2 中下标为 0 的修订号是 "1" 。0 < 1，所以 version1 < version2

示例 4：

输入：version1 = "1.0.1", version2 = "1"

输出：1

示例 5：

输入：version1 = "7.5.2.4", version2 = "7.5.3"

输出：-1

提示：

1 <= version1.length, version2.length <= 500

version1 和 version2 仅包含数字和 '.'

version1 和 version2 都是 有效版本号

version1 和 version2 的所有修订号都可以存储在 32 位整数 中

1. 题解
2. 思路

字符串的比较题目，思路其实很简单，按照题目的要求，通过“.”分割字符串，得到字符串数组，两个数组的长度可能是不一的，在较短的数组后面添加0，然后进行比较，当然这个添加0的操作可以直接在数组中添加，也可以在循环中通过判断该数组的长度是否是更短的来添加0。Integer的parseInt方法能够直接将字符串转换为数字，包括前面有前导零的情况。

1. 代码实现

Java：

class Solution {

    public int compareVersion(String version1, String version2) {

        //用 . 分割字符串， 对于每个String消除前导0，比较长度，长度小的添加0

        String[] v1 = version1.split("//.");

        String[] v2 = version2.split("//.");

        int n1 = v1.length;

        int n2 = v2.length;

        int maxL = Math.max(n1, n2);

        for(int i = 0; i < maxL; ++i){

            int temp1 = i < n1 ? Integer.parseInt(v1[i]) : 0;

            int temp2 = i < n2 ? Integer.parseInt(v2[i]) : 0;

            return temp1 < temp2 ? -1 : 1;

        }

        return 0;

    }

}