Problem 165: Compare Version Numbers

Problem Information

Difficulty: Medium
Acceptance Rate: 45.80%
Paid Only: No
Tags: Two Pointers, String

Problem Description

Given two **version strings** , `version1` and `version2`, compare them. A version string consists of **revisions** separated by dots `'.'`. The **value of the revision** is its **integer conversion** ignoring leading zeros.

To compare version strings, compare their revision values in **left-to-right order**. If one of the version strings has fewer revisions, treat the missing revision values as `0`.

Return the following:

* If `version1 < version2`, return -1. * If `version1 > version2`, return 1. * Otherwise, return 0.

Example 1:

Input: version1 = "1.2", version2 = "1.10"

Output: -1

Explanation:

version1's second revision is "2" and version2's second revision is "10": 2 < 10, so version1 < version2.

Example 2:

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

Output: 0

Explanation:

Ignoring leading zeroes, both "01" and "001" represent the same integer "1".

Example 3:

Input: version1 = "1.0", version2 = "1.0.0.0"

Output: 0

Explanation:

version1 has less revisions, which means every missing revision are treated as "0".

Constraints:

* `1 <= version1.length, version2.length <= 500` * `version1` and `version2` only contain digits and `'.'`. * `version1` and `version2` **are valid version numbers**. * All the given revisions in `version1` and `version2` can be stored in a **32-bit integer**.

Code Snippets

C++:

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

}
};
```

Java:

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

}
```

```
}
```

Python3:

```
class Solution:
def compareVersion(self, version1: str, version2: str) -> int:
```