Compare Version Numbers

Compare two version numbers *version1* and *version2*.
If *version1* > *version2* return 1, if *version1* < *version2* return -1, otherwise return 0.

You may assume that the version strings are non-empty and contain only digits and the `.` character.
The `.` character does not represent a decimal point and is used to separate number sequences.
For instance, `2.5` is not "two and a half" or "half way to version three", it is the fifth second-level revision of the second first-level revision.

Here is an example of version numbers ordering:

```
0.1 < 1.1 < 1.2 < 13.37
```

Credits:
Special thanks to @ts for adding this problem and creating all test cases.

Solution 1

This code assumes that next level is zero if no mo levels in shorter version number. And than compare levels.

```
public int compareVersion(String version1, String version2) {
    String[] levels1 = version1.split("\\.");
    String[] levels2 = version2.split("\\.");

    int length = Math.max(levels1.length, levels2.length);
    for (int i=0; i<length; i++) {
        Integer v1 = i < levels1.length ? Integer.parseInt(levels1[i]) : 0;
        Integer v2 = i < levels2.length ? Integer.parseInt(levels2[i]) : 0;
        int compare = v1.compareTo(v2);
        if (compare != 0) {
            return compare;
        }
    }

    return 0;
}
```

written by pavel-shlyk original link here

Solution 2

```
int compareVersion(string version1, string version2) {
    int i = 0;
    int j = 0;
    int n1 = version1.size();
    int n2 = version2.size();

    int num1 = 0;
    int num2 = 0;
    while(i<n1 || j<n2)
    {
        while(i<n1 && version1[i]!='.'){
            num1 = num1*10+(version1[i]-'0');
            i++;
        }

        while(j<n2 && version2[j]!='.'){
            num2 = num2*10+(version2[j]-'0');;
            j++;
        }

        if(num1>num2) return 1;
        else if(num1 < num2) return -1;

        num1 = 0;
        num2 = 0;
        i++;
        j++;
    }

    return 0;
}
```

written by XUYAN3 original link here

Solution 3

I checked other Java solution and the basic idea is the same. In addition, I simply the logic by making the two version number same length. For example, if version1 = "1.0.2", and version2 = "1.0", the I will convert the version2 to "1.0.0".

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

    String[] v1 = version1.split("\\.");
    String[] v2 = version2.split("\\.");

    for ( int i = 0; i < Math.max(v1.length, v2.length); i++ ) {
        int num1 = i < v1.length ? Integer.parseInt( v1[i] ) : 0;
        int num2 = i < v2.length ? Integer.parseInt( v2[i] ) : 0;
        if ( num1 < num2 ) {
            return -1;
        } else if ( num1 > num2 ) {
            return +1;
        }
    }

    return 0;
}
```

written by benjamin19890721 original link here