2191. Sort the Jumbled Numbers

Solved

Medium | Topics | Companies | Hint

You are given a **0-indexed** integer array `mapping` which represents the mapping rule of a shuffled decimal system. `mapping[i] = j` means digit `i` should be mapped to digit `j` in this system.

The **mapped value** of an integer is the new integer obtained by replacing each occurrence of digit `i` in the integer with `mapping[i]` for all `0 <= i <= 9`.

You are also given another integer array `nums`. Return *the array* `nums` *sorted in* ***non-decreasing*** *order based on the* ***mapped values*** *of its elements.*

Notes:

- Elements with the same mapped values should appear in the **same relative order** as in the input.
- The elements of `nums` should only be sorted based on their mapped values and **not be replaced** by them.

```
class Solution {
    func sortJumbled(_ mapping: [Int], _ nums: [Int]) -> [Int] {
        var mappingVal:[Int] = []

        for i in nums {

            var temp = i
            var j:[Int] = []

            if(temp == 0) {
                j.append(0)
            } else {
                while(temp > 0) {
                    j.append(temp%10)
                    temp = temp / 10
                }
            }

            temp = 0
            for k in j.reversed() {
                // var t = j % 10
                temp = temp * 10 + mapping[k]
```

```
                // j = j / 10
            }

            mappingVal.append(temp)
        }

        print(mappingVal)
        let z = zip(nums,mappingVal).sorted(by:{$0.1 < $1.1})

        return z.map({$0.0})
    }
}
```