Our Solution(s) Run Code

Solution 1

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(n * m^2 + nlog(n)) time | O(nm) space - where n is the number of
# m is the length of the longest string
def longestStringChain(strings):
    # For every string, imagine the longest string chain that starts
    # Set up every string to point to the next string in its respecti
    # string chain. Also keep track of the lengths of these longest s
    stringChains = {}
    for string in strings:
        stringChains[string] = {"nextString": "", "maxChainLength": 1}

    # Sort the strings based on their length so that whenever we visit
    # string (as we iterate through them from left to right), we can
    # already have computed the longest string chains of any smaller s
    sortedStrings = sorted(strings, key=len)
    for string in sortedStrings:
        findLongestStringChain(string, stringChains)

    return buildLongestStringChain(strings, stringChains)


def findLongestStringChain(string, stringChains):
    # Try removing every letter of the current string to see if the
    # remaining strings form a string chain.
    for i in range(len(string)):
        smallerString = getSmallerString(string, i)
        if smallerString not in stringChains:
            continue
        tryUpdateLongestStringChain(string, smallerString, stringChain


def getSmallerString(string, index):
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```python
def longestStringChain(strings):
    # Write your code here.
    pass

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.