

Solution 1

```
1  # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  # O(n * m^2 + nlog(n)) time | O(nm) space - where n is the number of strings and
4  # m is the length of the longest string
5  def longestStringChain(strings):
6      # For every string, imagine the longest string chain that starts with it.
7      # Set up every string to point to the next string in its respective longest
8      # string chain. Also keep track of the lengths of these longest string chains.
9      stringChains = {}
10     for string in strings:
11         stringChains[string] = {"nextString": "", "maxChainLength": 1}
12
13     # Sort the strings based on their length so that whenever we visit a
14     # string (as we iterate through them from left to right), we can
15     # already have computed the longest string chains of any smaller strings.
16     sortedStrings = sorted(strings, key=len)
17     for string in sortedStrings:
18         findLongestStringChain(string, stringChains)
19
20     return buildLongestStringChain(strings, stringChains)
21
22
23 def findLongestStringChain(string, stringChains):
24     # Try removing every letter of the current string to see if the
25     # remaining strings form a string chain.
26     for i in range(len(string)):
27         smallerString = getSmallerString(string, i)
28         if smallerString not in stringChains:
29             continue
30         tryUpdateLongestStringChain(string, smallerString, stringChains)
31
32
33 def getSmallerString(string, index):
34     return string[0:index] + string[index + 1 :]
35
36
37 def tryUpdateLongestStringChain(currentString, smallerString, stringChains):
38     smallerStringChainLength = stringChains[smallerString]["maxChainLength"]
39     currentStringChainLength = stringChains[currentString]["maxChainLength"]
40     # Update the string chain of the current string only if the smaller string leads
41     # to a longer string chain.
42     if smallerStringChainLength + 1 > currentStringChainLength:
43         stringChains[currentString]["maxChainLength"] = smallerStringChainLength + 1
44         stringChains[currentString]["nextString"] = smallerString
45
46
47 def buildLongestStringChain(strings, stringChains):
48     # Find the string that starts the longest string chain.
49     maxChainLength = 0
50     chainStartingString = ""
51     for string in strings:
52         if stringChains[string]["maxChainLength"] > maxChainLength:
53             maxChainLength = stringChains[string]["maxChainLength"]
54             chainStartingString = string
55
56     # Starting at the string found above, build the longest string chain.
57     ourLongestStringChain = []
58     currentString = chainStartingString
59     while currentString != "":
60         ourLongestStringChain.append(currentString)
61         currentString = stringChains[currentString]["nextString"]
62
63     return [] if len(ourLongestStringChain) == 1 else ourLongestStringChain
64
```

