

Our Solution(s)

Run Code

Your Solutions

Run Code

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
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 chain
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 string
14    # (as we iterate through them from left to right), we can be sure that we
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 remaining
25     # 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):
```

Solution 1

Solution 2

Solution 3

```
1 def longestStringChain(strings):
2     # Write your code here.
3     pass
4
```

Our Tests

Custom Output

Submit Code

```
10 expected = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
11 actual = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
12
13 # Test case 2
14 expected = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
15 actual = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
16
17 # Test case 3
18 expected = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
19 actual = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
20
21 # Test case 4
22 expected = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
23 actual = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
```

Run or submit code when you're ready.