Our Solution(s)   Run Code

Solution 1

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

import java.util.*;

class Program {

  public static class stringChain {
    String nextString;
    Integer maxChainLength;

    public stringChain(String nextString, Integer maxChainLength) {
      this.nextString = nextString;
      this.maxChainLength = maxChainLength;
    }
  }

  // O(n * m^2 + nlog(n)) time | O(nm) space - where n is the number o
  // and m is the length of the longest string
  public static List<String> longestStringChain(List<String> 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
    // chains.
    Map<String, stringChain> stringChains = new HashMap<String, string
    for (String string : strings) {
      stringChains.put(string, new stringChain("", 1));
    }

    // Sort the strings based on their length so that whenever we visi
    // string (as we iterate through them from left to right), we can
    // already have computed the longest string chains of any smaller
    List<String> sortedStrings = new ArrayList<String>(strings);
    sortedStrings.sort((a, b) -> a.length() - b.length());
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```java
import java.util.*;

class Program {
  public static List<String> longestStringChain(List<String> strings)
    // Write your code here.
    return null;
  }
}
```

Our Tests

Custom Output   Submit Code

Run or submit code when you're ready.