

Solution 1

```
1  # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  # O(n*m) | O(n) space
4  def underscorifySubstring(string, substring):
5      locations = collapse(getLocations(string, substring))
6      return underscorify(string, locations)
7
8
9  def getLocations(string, substring):
10     locations = []
11     startIdx = 0
12     while startIdx < len(string):
13         nextIdx = string.find(substring, startIdx)
14         if nextIdx != -1:
15             locations.append([nextIdx, nextIdx + len(substring)])
16             startIdx = nextIdx + 1
17         else:
18             break
19     return locations
20
21
22 def collapse(locations):
23     if not len(locations):
24         return locations
25     newLocations = [locations[0]]
26     previous = newLocations[0]
27     for i in range(1, len(locations)):
28         current = locations[i]
29         if current[0] <= previous[1]:
30             previous[1] = current[1]
31         else:
32             newLocations.append(current)
33             previous = current
34     return newLocations
35
36
37 def underscorify(string, locations):
38     locationsIdx = 0
39     stringIdx = 0
40     inBetweenUnderscores = False
41     finalChars = []
42     i = 0
43     while stringIdx < len(string) and locationsIdx < len(locations):
44         if stringIdx == locations[locationsIdx][i]:
45             finalChars.append("_")
46             inBetweenUnderscores = not inBetweenUnderscores
47             if not inBetweenUnderscores:
48                 locationsIdx += 1
49             i = 0 if i == 1 else 1
50         finalChars.append(string[stringIdx])
51         stringIdx += 1
52     if locationsIdx < len(locations):
53         finalChars.append("_")
54     elif stringIdx < len(string):
55         finalChars.append(string[stringIdx:])
56     return "".join(finalChars)
57
```