

Solution 1

```
1  # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  # O(n^2 + m) time | O(n + m) space
4  def patternMatcher(pattern, string):
5      if len(pattern) > len(string):
6          return []
7      newPattern = getNewPattern(pattern)
8      didSwitch = newPattern[0] != pattern[0]
9      counts = {"x": 0, "y": 0}
10     firstYPos = getCountsAndFirstYPos(newPattern, counts)
11     if counts["y"] != 0:
12         for lenOfX in range(1, len(string)):
13             lenOfY = (len(string) - lenOfX * counts["x"]) / counts["y"]
14             if lenOfY <= 0 or lenOfY % 1 != 0:
15                 continue
16             lenOfY = int(lenOfY)
17             yIdx = firstYPos * lenOfX
18             x = string[:lenOfX]
19             y = string[yIdx : yIdx + lenOfY]
20             potentialMatch = map(lambda char: x if char == "x" else y, newPattern)
21             if string == "".join(potentialMatch):
22                 return [x, y] if not didSwitch else [y, x]
23     else:
24         lenOfX = len(string) / counts["x"]
25         if lenOfX % 1 == 0:
26             lenOfX = int(lenOfX)
27             x = string[:lenOfX]
28             potentialMatch = map(lambda char: x, newPattern)
29             if string == "".join(potentialMatch):
30                 return [x, ""] if not didSwitch else ["", x]
31     return []
32
33
34 def getNewPattern(pattern):
35     patternLetters = list(pattern)
36     if pattern[0] == "x":
37         return patternLetters
38     else:
39         return list(map(lambda char: "x" if char == "y" else "y", patternLetters))
40
41
42 def getCountsAndFirstYPos(pattern, counts):
43     firstYPos = None
44     for i, char in enumerate(pattern):
45         counts[char] += 1
46         if char == "y" and firstYPos is None:
47             firstYPos = i
48     return firstYPos
49
```