AlgoExpert | Quad Layout | C# | 14px | Sublime | Monokai | 00:00:

Our Solution(s)

Run Code

Solution 1

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

using System.Text;
using System.Collections.Generic;

public class Program {
  // O(n^2 + m) time | O(n + m) space
  public static string[] PatternMatcher(string pattern, string str) {
    if (pattern.Length > str.Length) {
      return new string[] {};
    }
    char[] newPattern = getNewPattern(pattern);
    bool didSwitch = newPattern[0] != pattern[0];
    Dictionary<char, int> counts = new Dictionary<char, int>();
    counts['x'] = 0;
    counts['y'] = 0;
    int firstYPos = getCountsAndFirstYPos(newPattern, counts);
    if (counts['y'] != 0) {
      for (int lenOfX = 1; lenOfX < str.Length; lenOfX++) {
        double lenOfY =
          ((double)str.Length - (double)lenOfX *
          (double)counts['x']) /
          (double)counts['y'];
        if (lenOfY <= 0 || lenOfY % 1 != 0) {
          continue;
        }
        int yIdx = firstYPos * lenOfX;
        string x = str.Substring(0, lenOfX);
        string y = str.Substring(yIdx, (int)lenOfY);
        string potentialMatch = buildPotentialMatch(newPattern, x, y);
        if (str.Equals(potentialMatch)) {
          return didSwitch ? new string[] {y, x} : new string[] {x,
                                                                 y};
```

Your Solutions

Run Code

Solution 1 | Solution 2 | Solution 3

```csharp
public class Program {
  public static string[] PatternMatcher(string pattern, string str) {
    // Write your code here.
    return null;
  }
}

```

Our Tests

```
[illegible]
```

Custom Output

Submit Code

```
[illegible]
```

Run or submit code when you're ready.