Our Solution(s) Run Code

Solution 1

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

#include <vector>
#include <numeric>
#include <algorithm>
#include <unordered_map>
#include <math.h>
using namespace std;

vector<char> getNewPattern(string pattern);
int getCountsAndFirstYPos(vector<char> pattern,
                          unordered_map<char, int> *counts);

// O(n^2 + m) time | O(n + m) space
vector<string> patternMatcher(string pattern, string str) {
  if (pattern.length() > str.length()) {
    return vector<string>{};
  }
  vector<char> newPattern = getNewPattern(pattern);
  bool didSwitch = newPattern[0] != pattern[0];
  unordered_map<char, int> counts({{'x', 0}, {'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 || fmod(lenOfY, 1) != 0) {
        continue;
      }
      int yIdx = firstYPos * lenOfX;
      string x = str.substr(0, lenOfX);
      string y = str.substr(yIdx, lenOfY);
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```cpp
#include <vector>
using namespace std;

vector<string> patternMatcher(string pattern, string str) {
  // Write your code here.
  return {};
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.