All Letter Combinations

Question: Given a digit string, return all possible letter combinations that the number could represent. A mapping of digit to letters is given below.

2 => a,b,c; 3 => d,e,f; 4 => g,h,I; 5 => j,k,l; 6 => m,n,o; 7 => p,q,r,s;

8 => t,u,v; 9 => w,x,y,z

0,1 will not produce anything

Ex: Input: Digit string "23"

Output: ["ad", "ae", "af", "bd", "be", "bf", "cd", "ce", "cf"].

Solutions:

```
class Solution:
   # @return a list of strings, [s1, s2]
   def letterCombinations(self, digits):
      def dfs(num, string, res):
         if num == length:
            res.append(string)
            return
         for letter in dict[digits[num]]:
               dfs(num+1, string+letter, res)

      dict = {'2':['a','b','c'],
            '3':['d','e','f'],
            '4':['g','h','i'],
            '5':['j','k','l'],
            '6':['m','n','o'],
            '7':['p','q','r','s'],
```

```
            '8':['t','u','v'],
            '9':['w','x','y','z']
            }
        res = []
        length = len(digits)
        dfs(0, '', res)
        return res
Solution().letterCombinations("43556")
```