# Daily Coding Problem #162

## Problem

This problem was asked by Square.

Given a list of words, return the shortest unique prefix of each word. For example, given the list:

- dog
- cat
- apple
- apricot
- fish

Return the list:

- d
- c
- app

- apr
- f

## Solution

We can do this by comparing each word to each other and finding the shortest unique prefix for each word. However, this would take O($n^2$) time. We can speed up the process by using a trie and keeping track of each character's count. Then, after adding each word, we can look for the word in the trie by character until we hit a count of 1 while building the prefix.

```python
class Node:
    def __init__(self, char=None):
        self.char = char
        self.children = {}
        self.finished = False
        self.count = 0


class Trie:
    def __init__(self):
        self.root = Node()

    def insert(self, word):
        node = self.root
        for char in word:
            if char not in node.children:
                node.children[char] = Node(char)
            node.count += 1
            node = node.children[char]
        node.finished = True

    def unique_prefix(self, word):
        node = self.root
        prefix = ''
        for char in word:
            if node.count == 1:
                return prefix
            node = node.children[char]
            prefix += char
        return prefix


def shortest_unique_prefix(lst):
    trie = Trie()
    for word in lst:
        trie.insert(word)

    return [trie.unique_prefix(word) for word in lst]
```

This takes O(n) time.