**3)** (10 pts) DSN (Tries)

In many word games, the player is given some tiles with letters and must form a word with those tiles. Given a trie that stores a dictionary of valid words and a frequency array storing information of the tiles a player has, determine the number of unique words she can form with those tiles. Complete the function shown below to solve the given problem. Note: the entry in freq[i] represents the number of tiles with the letter 'a' + i. **(Hint: recursing down the trie is exactly like placing a tile down, which means updating the freq array. When you have finished "trying a tile" you have to put it back into your pool, which means editing the freq array again.)**

```
typedef struct TrieNode {
    struct TrieNode *children[26];
    int flag; // 1 if the string is in the trie, 0 otherwise
} TrieNode;

int countWords(TrieNode* root, int freq[]) {

    int res = root->flag ;                                    // 1 pt

    int i;
    for (i=0; i<26; i++) {


        if ( freq[i] == 0 || root->children[i] == NULL )      // 4 pts
            continue;


        freq[i]-- ;                                           // 1 pt


        res += countWords(root->children[i], freq) ;          // 3 pts


        freq[i]++ ;                                           // 1 pt
    }

    return res;
}
```