DAY 52

INTERVIEW BIT PROBLEMS :

1. Anagrams

Given an array of strings, **return all groups of strings that are anagrams.** Represent a group by a list of integers representing the index in the original list. Look at the sample case for clarification.

Anagram : a word, phrase, or name **formed by rearranging the letters of another**, such as 'spar', formed from 'rasp'

Note: All inputs will be in lower-case.

Example :
Input : cat dog god tca
Output : [[1, 4], [2, 3]]
cat and tca are anagrams which correspond to index 1 and 4.
dog and god are another set of anagrams which correspond to index 2 and 3.
The indices are 1 based (the first element has index 1 instead of index 0).

Ordering of the result : You should not change the relative ordering of the words / phrases within the group. Within a group containing A[i] and A[j], A[i] comes before A[j] if i < j.

CODE :

PYTHON

```
class Solution:
    # @param A : tuple of strings
    # @return a list of list of integers
    def anagrams(self, A):
        my_dict,my_list = [{i : j.count(i) for i in set(j)} for j in A], []
        for i in my_dict:
            if i not in my_list:
                my_list.append(i)
        return [[i + 1 for i, x in enumerate(my_dict) if x == j] for j in my_list]
```