**[Group Anagrams](https://leetcode.com/problems/group-anagrams/)**

Given an array of strings strs, group **the anagrams** together. You can return the answer in **any order**.

An **Anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

**Example 1:**

**Input:** strs = ["eat","tea","tan","ate","nat","bat"]

**Output:** [["bat"],["nat","tan"],["ate","eat","tea"]]

**Example 2:**

**Input:** strs = [""]

**Output:** [[""]]

**Example 3:**

**Input:** strs = ["a"]

**Output:** [["a"]]

**Constraints:**

* 1 <= strs.length <= 104
* 0 <= strs[i].length <= 100
* strs[i] consists of lowercase English letters.

CODE :

class Solution {

public:

    string getSignature(const string& s) {

        vector<int> count(26, 0);

        for (char c : s) {

            count[c - 'a']++;

        }

        stringstream ss;

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

            if (count[i] != 0) {

                ss << (char)('a' + i) << count[i];

            }

        }

        return ss.str();

    }

    vector<vector<string>> groupAnagrams(vector<string>& strs) {

        vector<vector<string>> result;

        unordered\_map<string, vector<string>> groups;

        for (const string& s : strs) {

            groups[getSignature(s)].push\_back(s);

        }

        for (const auto& entry : groups) {

            result.push\_back(entry.second);

        }

        return result;

    }

};

Link : <https://leetcode.com/problems/group-anagrams/?envType=daily-question&envId=2024-02-06>