49. Group Anagrams

<https://leetcode.com/problems/group-anagrams/>

1. **Listen**

**Problem Statement:**

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.

**Input:**

Array of strings **strs**

**Goal:**

Group the anagrams together from **strs**.

**Return:**

A **list of lists**, where each list is a group of anagrams.

1. **Examples**

**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.

**Test Cases:**

* strs has only one element that is
  + a string that is a single letter
  + an empty string
  + a string with length <= 100 but > 0
* strs has multiple elements
  + all strings are empty
  + all strings are not empty
  + strs has a mix of non-empty strings and empty strings
* test for a mix of
  + anagram group of 1
  + anagram group > 1

Questions:

* Is the string ASCII or Unicode?

1. **Brute Force**

Solution 1:

We can iterate over the array.

For each string, we can turn it into a char array, and sort each character in the char array by integer value. Now, any permutation of the sorted string will end up being the same once sorted.

We convert the char array back to a string to use a key for a hashmap.

Then, we can see if there is a key matching the sorted string in the hashmap

if there is no matching key, create a new list with the key

if there is a matching key, add the associated string (before sorting) to the key’s list

We iterate over each string in the array, which takes O(N) time

Inside the loop, we sort a character array

1. **Optimize**
2. **Walkthrough**
3. **Implement**
4. **Test**