Group Anagrams

Given an array of strings, group anagrams together.

For example, given: `["eat", "tea", "tan", "ate", "nat", "bat"]`,
Return:

```
[
  ["ate", "eat","tea"],
  ["nat","tan"],
  ["bat"]
]
```

Note:

1. For the return value, each *inner* list's elements must follow the lexicographic order.
2. All inputs will be in lower-case.

Solution 1

What does it mean "return all groups"? But the return result is vector? How can we return all groups? I mean, for example, we have such vector ["dog","cat","god","tac"]. What should I return?

written by htzfun original link here

Solution 2

```
public class Solution {
    public List<List<String>> groupAnagrams(String[] strs) {
        if (strs == null || strs.length == 0) return new ArrayList<List<String>>(
);
        Map<String, List<String>> map = new HashMap<String, List<String>>();
        Arrays.sort(strs);
        for (String s : strs) {
            char[] ca = s.toCharArray();
            Arrays.sort(ca);
            String keyStr = String.valueOf(ca);
            if (!map.containsKey(keyStr)) map.put(keyStr, new ArrayList<String>()
);
            map.get(keyStr).add(s);
        }
        return new ArrayList<List<String>>(map.values());
    }
}
```

written by wz366 original link here

Solution 3

Why the output is list and not a list of list. What If there are multiple groups of anagrams?

written by mahdy original link here