

49 字母异位词分组

Label: Hash

给定一个字符串数组，将字母异位词组合在一起。字母异位词指字母相同，但排列不同的字符串。

输入: ["eat", "tea", "tan", "ate", "nat", "bat"]

输出:

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

所有输入均为小写字母。

不考虑答案输出的顺序。

- HashMap 排序

```
class Solution {
    public List<List<String>> groupAnagrams(String[] strs) {
        HashMap<String,ArrayList<String>> map = new HashMap<>();

        for(String s : strs){
            char[] ch = s.toCharArray();
            Arrays.sort(ch);

            String key = String.valueOf(ch); // 从缓存池里面取key,key为sort之后的值
            if(!map.containsKey(key))
                map.put(key,new ArrayList<>());

            map.get(key).add(s);
        }

        return new ArrayList(map.values());
    }
}
```

- Stream 简化

```
class Solution {
    public List<List<String>> groupAnagrams(String[] strs) {
        return new ArrayList<>(Arrays.stream(strs).
            collect(Collectors.groupingBy(str ->
                str.chars().sorted().collect(StringBuilder::new,
                    StringBuilder::appendCodePoint,
                    StringBuilder::append).toString())
                .values()))
    }
}
```

// 简化 2

```
new ArrayList<>(Arrays.stream(strs).collect(Collectors.groupingBy(str ->
Stream.of(str.split("")).sorted().collect(Collectors.joining()))).values()); //
```

切割, 排序, 连接->形成key->根据key分组

- 计数

```
class Solution {
    public List<List<String>> groupAnagrams(String[] strs) {
        Map<String, List<String>> map = new HashMap<String, List<String>>();
        for (String str : strs) {
            int[] counts = new int[26];
            int length = str.length();
            for (int i = 0; i < length; i++) {
                counts[str.charAt(i) - 'a']++;
            }
            // 将每个出现次数大于 0 的字母和出现次数按顺序拼接成字符串, 作为哈希表的键
            StringBuffer sb = new StringBuffer();
            for (int i = 0; i < 26; i++) {
                if (counts[i] != 0) {
                    sb.append((char) ('a' + i));
                    sb.append(counts[i]);
                }
            }
            String key = sb.toString();
            List<String> list = map.getOrDefault(key, new ArrayList<String>());
            list.add(str);
            map.put(key, list);
        }
        return new ArrayList<List<String>>(map.values());
    }
}
```