

简单思路描述

注意到，如果把 aab,aba,baa 按照字母从小到大排序，我们可以得到同一个字符串 aab。

而对于每种字母出现次数不同于 aab 的字符串，例如 abb 和 bab，排序后为 abb，不等于 aab。

所以当且仅当两个字符串排序后一样，这两个字符串才能分到同一组。

根据这一点，我们可以用哈希表来分组，把排序后的字符串当作 key，原字符串组成的列表（即答案）当作 value。

最后把所有 value 加到一个列表中返回。

代码

```
1  class Solution {
2  public:
3      vector<vector<string>> groupAnagrams(vector<string>& strs) {
4          unordered_map<string, vector<string>> m;
5          for (string& s : strs) {
6              string sorted_s = s;
7              ranges::sort(sorted_s);
8              m[sorted_s].push_back(s); // sorted_s 相同的字符串分到同
           一组
9          }
10
11         vector<vector<string>> ans;
12         ans.reserve(m.size()); // 预分配空间
13         for (auto& [_, value] : m) {
14             ans.push_back(value);
15         }
16         return ans;
17     }
18 };
```

✧ Do by me（升级版）

```
1  class Solution {
2  public:
3      vector<vector<string>> groupAnagrams(vector<string>& strs) {
4
5          vector<vector<string>> result;
6
7          // 创建哈希表
8          unordered_map <string , vector<string>> hashtable;
9          // 对每个字符串进行排序
10         for(int i=0;i<strs.size();i++){
11
12             // 每个字符串排序后的字符串
13             string temp_str= strs[i];
14
15             ranges::sort(temp_str);
16
17             hashtable[temp_str].push_back(strs[i]);
18         }
19
20         for(const auto& pair :hashtable){
21             result.push_back(pair.second);
22         }
23
24         return result;
25     }
26 };`
```

✧ 官方题解

49. 字母异位词分组 - 力扣（LeetCode）

✧ 知识补充

字符串的排序: **ranges::sort**

哈希表的创建: `unordered_map <string , vector> hashtable;`

遍历哈希表的关键字:

```
for(const auto& pair :hashtable){  
    result.push_back(pair.second);  
}
```