

面试题 10.02. 变位词组【运用哈希映射】【字母排序】【字符排序】

题目描述：

面试题 10.02. 变位词组

难度 中等

👍 67

☆

📄

🔍

🔔

💬

编写一种方法，对字符串数组进行排序，将所有变位词组合在一起。变位词是指字母相同，但排列不同的字符串。

注意： 本题相对原题稍作修改

示例：

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

复杂代码（没有充分利用C++的特性；更偏向于C风格的代码）

```
class Solution {
private:
    static bool cmp(string&a,string&b){
        if(a.size()!=b.size())return a.size()<b.size();
        vector<int>tag1(26,0),tag2(26,0);
        for(size_t i=0;i<a.size();i++){
            tag1[a[i]-'a']++;
            tag2[b[i]-'a']++;
        }
    }
};
```

```

    }
    bool flag=true;//flag为true是表示a小, false表示a大
    for(int i=0;i<26;i++){
        if(tag1[i]==tag2[i]){
            continue;
        }else if(tag1[i]>tag2[i])return true;
        else return false;
    }
    return true;
}

bool equal(string&a,string&b){
    vector<int>tag1(26,0),tag2(26,0);
    for(size_t i=0;i<a.size();i++)tag1[a[i]-'a']++;
    for(size_t i=0;i<b.size();i++)tag2[b[i]-'a']++;
    for(int i=0;i<26;i++){
        if(tag1[i]!=tag2[i]){
            return false;
        }
    }
    return true;
}

public:
    vector<vector<string>> groupAnagrams(vector<string>& strs) {
        //排序
        sort(strs.begin(),strs.end(),cmp);
        vector<vector<string>>res;
        vector<string>ans;
        string temp;
        for(size_t i=0;i<strs.size();i++){
            if(ans.empty()){
                ans.push_back(strs[i]);
                temp=strs[i];
            }else if(equal(temp,strs[i])){
                ans.push_back(strs[i]);
            }else{
                res.push_back(ans);
                ans.clear();
                ans.push_back(strs[i]);
                temp=strs[i];
            }
        }
        res.push_back(ans);
        return res;
    }
};

```

充分利用C++特性的较为简洁的代码：

```
class Solution {
public:
    vector<vector<string>> groupAnagrams(vector<string>& strs) {
        unordered_map<string, vector<string>> mp;
        for (string& str: strs) {
            string key = str;
            sort(key.begin(), key.end()); //充分利用字符也可以排序的一个特性
            mp[key].emplace_back(str);
        }
        vector<vector<string>> ans;
        for (auto it = mp.begin(); it != mp.end(); ++it) {
            ans.emplace_back(it->second);
        }
        return ans;
    }
};
```

作者：LeetCode-Solution

链接：<https://leetcode-cn.com/problems/group-anagrams-lcci/solution/bian-wei-ci-zu-by-leetcode-solution-g2a8/>

来源：力扣（LeetCode）

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。