

# [每日 LeetCode] 893. Groups of Special-Equivalent Strings

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1561988867818>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

原文链接 <https://leetcode.com/forward?goto=https%3A%2F%2Fwww.tuhaoxin.cn%2Farticles%2F2019%2F07%2F01%2F1561988812983.html> [每日 LeetCode] 893. Groups of Special-Equivalent Strings

**Description:**

You are given an array `A` of strings.

Two strings `S` and `T` are *special-equivalent* if after any number of *moves*,  $S == T$ .

A *move* consists of choosing two indices `i` and `j` with  $i \% 2 == j \% 2$ , and swapping `S[i]` with `S[j]`.

Now, a *group* of special-equivalent strings from `A` is a non-empty subset `S` of `A` such that any string not in `S` is not special-equivalent with any string in `S`.

Return the number of groups of special-equivalent strings from `A`.

**Example 1:**

```
Input: ["a","b","c","a","c","c"]
Output: 3
Explanation: 3 groups ["a","a"], ["b"], ["c","c","c"]
```

**Example 2:**

```
Input: ["aa","bb","ab","ba"]
Output: 4
Explanation: 4 groups ["aa"], ["bb"], ["ab"], ["ba"]
```

**Example 3:**

```
Input: ["abc","acb","bac","bca","cab","cba"]
Output: 3
Explanation: 3 groups ["abc","cba"], ["acb","bca"], ["bac","cab"]
```

**Example 4:**

```
Input: ["abcd","cdab","adcb","cbad"]
Output: 1
Explanation: 1 group ["abcd","cdab","adcb","cbad"]
```

---

思路：本题题意为求字符串数组中特殊的相等字符串，只读一遍题目发现有点绕，多读几遍就好。题目要求找出字符串数组中存在的特殊相等字符串对，特殊性体现在字符串对中的每个字符串需要奇数偶数位置上的字符相同（不考虑顺序）。解决方案可以分别记录某个字符串中奇数位和偶数位上的字符串，然后对两个字符串排序并使用 pair 结构判断保证记录的唯一性。如原字符串："abcdefgggyy"，奇数串："acegy"，偶数串："bdfgy"，然后使用嵌入 map 记录符合符合条件的字符串有多少组。

---

C++ 代码

```
class Solution {
public:
```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> int numSpecial
quivGroups(vector<string>& A) {
</span></span><span class="highlight-line"><span class="highlight-cl">     map<std::
air<string,string>,int> mymap; // key:pattern pair
</span></span><span class="highlight-line"><span class="highlight-cl">     for(int i = 0 ; i
&lt; A.size() ; i++){
</span></span><span class="highlight-line"><span class="highlight-cl">         string s[2];
</span></span><span class="highlight-line"><span class="highlight-cl">         for(int j = 0
; j &lt; A[i].length() ; j++)
</span></span><span class="highlight-line"><span class="highlight-cl">             s[j%2]+
A[i][j];
</span></span><span class="highlight-line"><span class="highlight-cl">             sort(s[0].b
gin(),s[0].end()),sort(s[1].begin(),s[1].end());
</span></span><span class="highlight-line"><span class="highlight-cl">             mymap[{{s[
],s[1]}}]++;
</span></span><span class="highlight-line"><span class="highlight-cl">         }
</span></span><span class="highlight-line"><span class="highlight-cl">     return myma
.size();
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> };
</span></span></code></pre>

```

<p>运行时间: 12ms</p>

<p>运行内存: 10.1M</p>