

E Equivalence Categorization

21101048 尹李

题意：

给定 n 个仅包含小写英文字母的字符串 S_1, S_2, \dots, S_n 。

将 S_1, S_2, \dots, S_n 划分为**多个集合**，使得一个集合内的字符串两两相同，并且不同集合之间的字符串两两不同。

求出划分得到的**集合数量**，以及**集合中字符串数量的最大值**。

- 正整数 $n(1 \leq n \leq 10^5)$, $S_i(1 \leq |S_i| \leq 10^5)$, 保证有 $\sum |S_i| \leq 5 \times 10^5$ 。

思路：

1. 哈希

- 每个字符串对应一个哈希值，哈希值相同则字符串相同（没有冲突的话）
- 实际自己写字符串哈希可能解决不了冲突，可以用 **c++的STL** 中的 `map`
- `unordered_map` 比 `map` 快，这里不需要排序

代码

```
1 unordered_map<string, int> mp;
2 for (int i = 1; i ≤ n; i++) {
3     cin >> s;
4     if (mp.count(s)) {
5         mp[s]++;
6         maxCnt = max(maxCnt, mp[s]);
7     } else {
8         mp[s] = 1;
9     }
10 }
```

2. 字典树

- 每个字符串根节点遍历向下，每个结点代表一个字母
- 在单词结束的结点记录单词数，并更新集合数和最大值

代码

```
1 struct Trie{
2     int ch[maxn][maxc];
3     int cnt[maxn * maxc];
4     int tot, numOfWords, maxNum;
5     void init(){
6         tot = numOfWords = maxNum = 0;
7         memset(cnt, 0, sizeof(cnt));
8         memset(ch, -1, sizeof(ch));
9     }
10    void insert(string str){
11        int p = 0;
12        for(int i = 0; str[i]; ++i){
13            if (ch[p][str[i] - 'a'] == -1){
14                ch[p][str[i] - 'a'] = ++tot;
15            }
16            p = ch[p][str[i] - 'a'];
17        }
18        if (cnt[p] == 0) {
19            numOfWords++;
20        }
21        cnt[p]++;
22        maxNum = max(maxNum, cnt[p]);
23    }
24 }
```