

题目

E Equivalence Categorization

5939194	AC	263	8924
---------	----	-----	------

时间限制：1000ms 内存限制：65536kb

通过率：151/164 (92.07%) 正确率：151/681 (22.17%)

题目描述

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

定义两个字符串 A, B 相同，当且仅当 $|A| = |B|$ 并且对于任意 $1 \leq i \leq |A|$ 有 $A_i = B_i$ 。 A, B 相同记为 $A = B$ 。

现在将 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。形式化地说，你需要将 S_1, S_2, \dots, S_n 满足以下条件的多个集合 F_1, F_2, \dots, F_k ：

- $\bigcup_{i=1}^k F_i = \{S_1, S_2, \dots, S_n\}$;
- $\bigcap_{i=1}^k F_i = \emptyset$;
- 对于 $1 \leq i \leq k$, 对于任意 $S_a, S_b \in F_i$, 均有 $S_a = S_b$;
- 对于 $1 \leq i < j \leq k$, 对于任意 $S_a \in F_i, S_b \in F_j$, 均有 $S_a \neq S_b$ 。

你需要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。

输入

第一行，一个正整数 n ($1 \leq n \leq 10^5$)。

接下来 n 行，每行一个仅包含小写英文字母的字符串 S_i ($1 \leq |S_i| \leq 10^5$)。

保证有 $\sum_{i=1}^n |S_i| \leq 5 \times 10^5$ 。

输出

一行，两个整数，分别表示划分得到的集合数量，以及集合中字符串数量的最大值。

思路

本题考察字符串哈希

啥是字符串哈希

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）

不用STL(双哈希)

```
#define ll long long // 双Hash方法，不同的Base和MOD，相当于两次 单Hash
ll Base1 = 29;
ll Base2 = 131;
ll MOD1 = 1e9 + 7;
ll MOD2 = 1e9 + 9;
const int MAXN = 2e4 + 50;

class Solution {
public:
    set< pair <ll, ll> > H; // 因为是一个二元组，所以可以用 pair 容器。
    ll h1[MAXN], h2[MAXN], p1[MAXN], p2[MAXN];

    int distinctEchoSubstrings(string text) {
        int n = text.size();
        h1[0] = 0, h2[0] = 0, p1[0] = 1, p2[0] = 1;
```

```

        for(int i = 0;i < n;i++)
        {
            h1[i+1] = (h1[i]*Base1 + (text[i] - 'a' + 1)) % MOD1;
            h2[i+1] = (h2[i]*Base2 + (text[i] - 'a' + 1)) % MOD2;
        }

        for(int i = 1;i < n;i++)
        {
            p1[i] = (p1[i-1]*Base1) % MOD1;
            p2[i] = (p2[i-1]*Base2) % MOD2;
        }

        for(int len = 2; len <= n; len += 2)
        {
            for(int i = 0;i + len - 1 < n;i++)
            {
                int x1 = i, y1 = i + len/2 - 1;
                int x2 = i + len/2, y2 = i + len - 1;
                ll left1 = ((h1[y1 + 1] - h1[x1] * p1[y1 + 1 - x1]) % MOD1 + MOD1)
% MOD1;
                ll right1 = ((h1[y2 + 1] - h1[x2] * p1[y2 + 1 - x2]) % MOD1 +
MOD1) % MOD1;
                ll left2 = ((h2[y1 + 1] - h2[x1] * p2[y1 + 1 - x1]) % MOD2 + MOD2)
% MOD2;
                ll right2 = ((h2[y2 + 1] - h2[x2] * p2[y2 + 1 - x2]) % MOD2 +
MOD2) % MOD2;

                if(left1 == right1 && left2 == right2) H.insert(make_pair(left1,
left2));
            }
        }
        return H.size();
    }
};

```

STL解法

用map映射字符串

代码实现

```

# include <bits/stdc++.h>
using namespace std;

int main() {
    map<string,int>F;
    int n;
    int maxCount = 0;

```

```
    cin >> n;
    string S;
    for(int i = 0; i < n; i ++){
        cin >> S;
        F[S] ++;
        maxCount = maxCount > F[S] ? maxCount : F[S];
    }
    cout << F.size() << " " << maxCount;
    return 0;
}
```

要注意的细节

本题使用单哈希无法解决冲突，建议使用stl解法，简单好敲