Strangely Compatible

locked

by isat1729

Editorial by isat1729

The question asks to find the number of pairs of strings which differs in exactly one position.

My approach is that for each string S of length m, I added m hash values in a map MP. In each of these strings, the hashes are calculated by replacing one of the letters with a character (I used ASCII value = 123) not found in any of the strings. Also, I added the hash of the original string in a separate map $Same$ to be used for calculation in order to not overcount because of cases where both the strings are same.

Now, for each hash value H in MP, I calculated the total number of pairs of strings having this hash value. Also, for each hash value HS in $Same$, I subtracted the total number of pairs of strings having this hash value from the final answer.

If you are unfamiliar with string hashing, there is a very nice tutorial on this topic in this link.

Statistics

Difficulty: Medium
Time Complexity: O(n*m log(n*m)) Required
Knowledge: String Hashing, Map
Publish Date: Mar 16 2019

Set by isat1729

Problem Setter's code :

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

typedef unsigned long long ll;
const int MX = 1e6 + 5;
const ll p = 31;
map<ll, ll> MP, Same;
ll H[MX], p_pow[MX];
char S[MX];
int n, m;

void init(){
    p_pow[0] = 1LL;
    for(int i = 1; i <= m; i++){
        p_pow[i] = p_pow[i - 1] * p;
    }
}

void process(){
    for(int i = 0; i < m; i++){
        H[i] = (S[i] - 'a' +  1) * p_pow[i];
        if(i) H[i] += H[i - 1];
    }

    Same[H[m - 1]]++;
```

```
    for(int i = 0; i < m; i++){
        ll h = 0;
        if(i == 0){
            h = H[m - 1] - H[i];
            h += 27;
        }else if(i == m - 1){
            h = H[i - 1];
            h += 27 * p_pow[i];
        }else{
            h = H[i - 1];
            h += H[m - 1] - H[i];
            h += 27 * p_pow[i];
        }
        MP[h]++;
    }
}

int main() {
    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);

    scanf("%d %d", &n, &m);
    init();
    //assert(n >= 1 && n <= 100000);
    //assert(m >= 1 && m <= 100000);
    //assert(n * m <= 1000000);
    int a = 0;
    for(int i = 0; i < n; i++){
        scanf("%s", S);
        int val = strlen(S);
        //assert(val == m);
        a++;
        process();
    }

    //assert(a == n);

    ll Ans = 0LL, cnt;
    map<ll, ll>:: iterator it;
    for(it = MP.begin(); it != MP.end(); it++){
        cnt = it->second;
        Ans += (cnt * (cnt - 1LL))/2LL;
    }

    map<ll, ll>:: iterator itr;
    for(itr = Same.begin(); itr != Same.end(); itr++){
        cnt = itr->second;
        ll val = (cnt * (cnt - 1LL))/2LL;
        val *= (ll)m;
        Ans -= val;
    }
    printf("%llu\n", Ans);
    return 0;
}
```