

字符串

1.KMP

```
int kmp(string text,string paattern) {
    int n=text.size(),m=pattern.size();
    if(m==0) {
        return 0;
    }
    vector<int>next(m);
    for(int i=1,j=0; i<m; i++) {
        while(j>0&&pattern[i]!=pattern[j]) {
            j=next[j-1];
        }
        if(pattern[i]==pattern[j]) {
            j++;
        }
        next[i]=j;
    }
    for(int i=0,j=0; i<n; i++) {
        while(j>0&&text[i]!=pattern[j]) {
            j=next[j-1];
        }
        if(text[i]==pattern[j]) {
            j++;
        }
        if(j==m) {
            return i-m+1;
        }
    }
    return -1;
}
```

2.trie树

```
struct trie {
    int nex[100000][26], cnt;
    bool exist[100000]; // 该结点结尾的字符串是否存在

    void insert(char *s, int l) { // 插入字符串
        int p = 0;
        for (int i = 0; i < l; i++) {
            int c = s[i] - 'a';
            if (!nex[p][c]) nex[p][c] = ++cnt; // 如果没有，就添
```

加结点

```

        p = nex[p][c];
    }
    exist[p] = 1;
}

bool find(char *s, int l) { // 查找字符串
    int p = 0;
    for (int i = 0; i < l; i++) {
        int c = s[i] - 'a';
        if (!nex[p][c]) return 0;
        p = nex[p][c];
    }
    return exist[p];
}
};

```

3.AC自动机

- 给你一个文本串 S 和 n 个模式串 $T_1 \sim T_n$, 请你分别求出每个模式串 T_i 在 S 中出现的次数。

```

#include<bits/stdc++.h>
#define maxn 2000001
using namespace std;
char s[maxn],T[maxn];
int n,cnt,vis[200051],ans,in[maxn],Map[maxn];
struct kkk{
    int son[26],fail,flag,ans;
    void clear(){memset(son,0,sizeof(son)),fail=flag=ans=0;}
}trie[maxn];
queue<int>q;
void insert(char* s,int num){
    int u=1,len=strlen(s);
    for(int i=0;i<len;i++){
        int v=s[i]-'a';
        if(!trie[u].son[v])trie[u].son[v]=++cnt;
        u=trie[u].son[v];
    }
    if(!trie[u].flag)trie[u].flag=num;
    Map[num]=trie[u].flag;
}
void getFail(){
    for(int i=0;i<26;i++)trie[0].son[i]=1;
    q.push(1);
    while(!q.empty()){
        int u=q.front();q.pop();
        int Fail=trie[u].fail;
        for(int i=0;i<26;i++){

```

```

        int v=trie[u].son[i];
        if(!v){trie[u].son[i]=trie[Fail].son[i];continue;}
        trie[v].fail=trie[Fail].son[i]; in[trie[v].fail]++;
        q.push(v);
    }
}
}
void topu(){
    for(int i=1;i<=cnt;i++)
        if(in[i]==0)q.push(i);
    while(!q.empty()){
        int u=q.front();q.pop();vis[trie[u].flag]=trie[u].ans;
        int v=trie[u].fail;in[v]--;
        trie[v].ans+=trie[u].ans;
        if(in[v]==0)q.push(v);
    }
}
void query(char* s){
    int u=1,len=strlen(s);
    for(int i=0;i<len;i++)
        u=trie[u].son[s[i]-'a'],trie[u].ans++;
}
int main(){
    scanf("%d",&n); cnt=1;
    for(int i=1;i<=n;i++){
        scanf("%s",s);
        insert(s,i);
    }getFail();scanf("%s",T);
    query(T);topu();
    for(int i=1;i<=n;i++)printf("%d\n",vis[Map[i]]);
}

```

4.字符串Hash

自然溢出哈希

```

#include <cstdio>
#include <cstring>
#include <algorithm>
using namespace std;
typedef unsigned long long ull;
ull base=131;
ull a[10010];
char s[10010];
int n,ans=1;
ull hashs(char s[])
{
    int len=strlen(s);

```

```

    ull ans=0;
    for (int i=0;i<len;i++)
        ans=ans*base+(ull)s[i];
    return ans&0x7fffffff;
}
main()
{
    scanf("%d",&n);
    for (int i=1;i<=n;i++)
    {
        scanf("%s",s);
        a[i]=hashs(s);
    }
    sort(a+1,a+n+1);
    for (int i=2;i<=n;i++)
        if (a[i]!=a[i-1])
            ans++;
    printf("%d\n",ans);
}

```

双哈希

```

#include <stdio>
#include <cstring>
#include <algorithm>
using namespace std;
typedef unsigned long long ull;
ull base=131;
struct data
{
    ull x,y;
}a[10010];
char s[10010];
int n,ans=1;
ull mod1=19260817;
ull mod2=19660813;
ull hash1(char s[])
{
    int len=strlen(s);
    ull ans=0;
    for (int i=0;i<len;i++)
        ans=(ans*base+(ull)s[i])%mod1;
    return ans;
}
ull hash2(char s[])
{
    int len=strlen(s);
    ull ans=0;

```

```

    for (int i=0;i<len;i++)
        ans=(ans*base+(ull)s[i])%mod2;
    return ans;
}
bool comp(data a,data b)
{
    return a.x<b.x;
}
main()
{
    scanf("%d",&n);
    for (int i=1;i≤n;i++)
    {
        scanf("%s",s);
        a[i].x=hash1(s);
        a[i].y=hash2(s);
    }
    sort(a+1,a+n+1,comp);
    for (int i=2;i≤n;i++)
        if (a[i].x≠a[i-1].x || a[i-1].y≠a[i].y)
            ans++;
    printf("%d\n",ans);
}

```