

AC自动机

[P3808 【模板】AC 自动机 (简单版)](<https://www.luogu.com.cn/problem/P3808>)

[参考](https://oi-wiki.org/string/ac-automaton/#_8)

```
```cpp
#include <bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
using namespace std;

const int N=1e6+7;
int n;

namespace AC {
int tr[N][26],tot;
int e[N],fail[N];

void insert(char *s) {
 int u=0;
 for(int i=1;s[i];i++){
 if(!tr[u][s[i]-'a']) tr[u][s[i]-'a']=++tot;//如果没有则插入新节点
 u=tr[u][s[i]-'a'];//搜索下一个节点
 }
 e[u]++;//尾为节点 u 的串的个数
}

queue<int> q;

void build() {
 FOR(i,0,25)
 if(tr[0][i]) q.push(tr[0][i]);
 while(q.size()){
 int u=q.front();
 q.pop();
 FOR(i,0,25){
 if(tr[u][i]){
 fail[tr[u][i]]=tr[fail[u]][i];//fail数组：同一字符可以匹配的其他位置
 q.push(tr[u][i]);
 }
 else tr[u][i]=tr[fail[u]][i];
 }
 }
}

int query(char *t){
 int u=0,res=0;
 for(int i=1;t[i];i++){
 u=tr[u][t[i]-'a'];//转移
 for(int j=u;j and e[j]!=-1;j=fail[j]){
 res+=e[j],e[j]=-1;
 }
 }
}
```

```

 }
 return res;
}
} // namespace AC

```

```
char s[N];
```

```

int main(){
 cin>>n;
 FOR(i,1,n)
 cin>>(s+1),AC::insert(s);
 cin>>(s+1);
 AC::build();
 cout<<AC::query(s);
 return 0;
}
...

```

## # KMP

[P3375 【模板】KMP字符串匹配](<https://www.luogu.com.cn/problem/solution/P3375>)

[参考](<https://fanfansann.blog.csdn.net/article/details/107555602>)

```

```cpp
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
using namespace std;

const int N=1000007;

int nex[N];
char a[N],b[N];

int main(){
    cin>>(b+1)>>(a+1); //A:模式串;B:文本串
    int lena=strlen(a+1),lenb=strlen(b+1);
    nex[1]=0;

    int j;
    FOR(i,2,lena){ //A串自己匹配
        while(j>0 and a[i]!=a[j+1]) j=nex[j];
        if(a[i]==a[j+1]) j++;
        nex[i]=j;
    }
    for(int i=1,j=0;i<=lenb;++i){ //A串向B串匹配
        while(j>0 and b[i]!=a[j+1]) j=nex[j];
        if(b[i]==a[j+1]) j++;
        if(j==lena) cout<<i-lena+1<<"\n",j=nex[j];
    }
    FOR(i,1,lena)
        cout<<nex[i]<<" ";
    return 0;
}

```

```
```
```

## # Manacher

[P3805 【模板】manacher 算法](<https://www.luogu.com.cn/problem/P3805>)

[参考](<https://12349.blog.luogu.org/manacher-zong-jie>)

```
```cpp
```

```
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
using namespace std;
```

```
const int maxn=3e7;
char s[maxn],st[maxn];
int R[maxn];
```

```
int pre_process(){
    int len=strlen(st);
    int j=2;
    s[0]='^';
    s[1]='$';
    FOR(i,0,len-1){
        s[j++]=st[i];
        s[j++]='$';
    }
    s[j]='&';
    return j;
}
```

```
int manacher(){
    int len=pre_process();
    int mid=1,RB=1,ans=-1;
    FOR(i,1,len-1){
        if(i<RB) R[i]=min(RB-i,R[mid*2-i]);
        else R[i]=1;
        while(s[i-R[i]]==s[i+R[i]]) R[i]++;
        if(RB<i+R[i]){
            mid=i;
            RB=i+R[i];
        }
        ans=max(ans,R[i]-1);
    }
    return ans;
}
```

```
int main(){
    cin>>st;
    cout<<manacher();
    return 0;
}
```
```

## # 字典树(Trie)

## [P3879 [TJOI2010] 阅读理解](https://www.luogu.com.cn/problem/P3879)

### #### 字典树写法

[参考1](https://www.luogu.com.cn/blog/\_post/31110)

[参考2](https://www.luogu.com.cn/blog/\_post/82818)

[bitset用法](https://www.cnblogs.com/zwfymqz/p/8696631.html)

[854ms/45.96MB](https://www.luogu.com.cn/record/69304560)

```
```cpp
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
using namespace std;

int nex[300007][26],n,cnt=0;
bitset<1001> b[500007]; //bool b[500007][1001];

void insert(char* s,int rol){
    int len=strlen(s+1);
    int now=0;
    FOR(i,1,len){
        int p=s[i]-'a';
        if(!nex[now][p])//如果Trie树中没有这个单词的前缀就进行编号
            nex[now][p]=++cnt;//上文中说到的编号
        now=nex[now][p]; //接着深入一层，更新现在的位置
    }
    b[now][rol]=1; //这个单词在第x行出现了
}

void check(char* s){
    int len=strlen(s+1);
    int now=0,flag=1;
    FOR(i,1,len){
        int p=s[i]-'a';
        if(!nex[now][p]){ //如果在Trie树中没有当前的字符，就可以直接break掉了
            flag=0;
            break;
        }
        now=nex[now][p]; //否则就更新位置
    }
    if(flag){
        FOR(i,1,n)//题面上说按字典序输出
            if(b[now][i]) cout<<i<<" "; //输出在哪些句子中出现过
    }
    putchar('\n');
}

int main(){
    cin>>n;
```

```

char s[25];
FOR(i,1,n){
    int l;cin>>l;
    FOR(j,1,l){//一个单词一个单词的插入Trie树里
        cin>>(s+1);
        insert(s,i);
    }
}
int m;cin>>m;
FOR(i,1,m){
    cin>>(s+1);
    check(s);
}
return 0;
}
...

```

STL: MAP + VECTOR 写法

[参考](https://www.luogu.com.cn/blog/_post/44774)

用`STL`可以不必纠结数组开多大以及爆空间，但会更费时

[2.06s/3.96MB](<https://www.luogu.com.cn/record/69303923>)

```

...cpp
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
#define mem(a) memset(a,0,sizeof(a))
using namespace std;

const int maxn = 100001;
int n,m,num,cnt[maxn];
string s;
map<string,vector<int> >a;

int main(){
    cin>>n;
    FOR(i,1,n){
        cin>>num;
        FOR(j,1,num){
            cin>>s;
            a[s].push_back(i);//每一个单词就是一个vector,存句子的编号
        }
    }
    cin>>m;
    FOR(i,1,m){
        cin>>s;
        mem(cnt);//cnt就是去重的桶。每用一次输出不同的查询要清零。
        int len=a[s].size();
        FOR(j,0,len-1)
            if(cnt[a[s][j]] == 0){
                cout<<a[s][j]<<" ";
                cnt[a[s][j]]++;//用桶去重
            }
    }
}

```

```

    }
    cout<<endl;
}
return 0;
}
...

```

STL: MAP + SET 写法

[参考](https://www.luogu.com.cn/blog/_post/76028)

用`set`可以自动去重，更方便。同时****耗时非常优秀****，仅次于手写`Trie`

[972ms/5.34MB](<https://www.luogu.com.cn/record/69305757>)

```

```cpp
#include<bits/stdc++.h>
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
using namespace std;

map<string,set<int> > m;

int main(){
 int n,p,l;
 string s;
 cin>>n;
 FOR(i,1,n){
 cin>>l;//单词个数
 FOR(j,0,l-1){
 cin>>s;//单词
 m[s].insert(i);
 }
 }
 cin>>p;
 while(p--){
 cin>>s;
 if(m.count(s)){//如果m中存在元素s
 for(auto iter=m[s].begin();iter!=m[s].end();++iter)
 cout<<*iter<<" ";
 }
 cout<<endl;
 }
 return 0;
}
...

```

#### # 字符串哈希

[P3370 【模板】字符串哈希](<https://www.luogu.com.cn/problem/P3370>)

[参考](<https://fanfansann.blog.csdn.net/article/details/107555631>)

```

```cpp
#include<bits/stdc++.h>

```

```
#define ll Long Long
#define FOR(i,a,b) for(int i=(a);i<=(b);++i)
using namespace std;
```

```
const int p=131;//13331
const int N=1e7+7;
int n,m;
ll a[N];
char str[N];
```

```
ll get_hash(char s[]){
    ll res=0;
    int len=strlen(s);
    FOR(i,0,len-1)
        res=res*p+(ll)s[i];
    return res;
}
```

```
int main(){
    cin>>n;
    FOR(i,1,n){
        cin>>str;
        a[i]=get_hash(str);
    }
    int ans=1;
    sort(a+1,a+1+n);
    FOR(i,1,n-1)
        if(a[i]!=a[i+1]) ans++;
    cout<<ans;
    return 0;
}
```