

# AC自动机

P3808 【模板】AC 自动机（简单版）

[参考](#)

```
1  #include <bits/stdc++.h>
2  #define FOR(i,a,b) for(int i=(a);i<=(b);++i)
3  using namespace std;
4
5  const int N=1e6+7;
6  int n;
7
8  namespace AC {
9  int tr[N][26],tot;
10 int e[N],fail[N];
11
12 void insert(char *s) {
13     int u=0;
14     for(int i=1;s[i];i++){
15         if(!tr[u][s[i]-'a']) tr[u][s[i]-'a']=++tot; //如果没有则插入新节点
16         u=tr[u][s[i]-'a']; //搜索下一个节点
17     }
18     e[u]++; //尾为节点 u 的串的个数
19 }
20
21 queue<int> q;
22
23 void build() {
24     FOR(i,0,25)
25         if(tr[0][i]) q.push(tr[0][i]);
26     while(q.size()){
27         int u=q.front();
28         q.pop();
29         FOR(i,0,25){
30             if(tr[u][i]){
31                 fail[tr[u][i]]=tr[fail[u]][i]; //fail数组：同一字符可以匹配的其他位置
32                 q.push(tr[u][i]);
33             }
34             else tr[u][i]=tr[fail[u]][i];
35         }
36     }
37 }
38
39 int query(char *t){
40     int u=0,res=0;
41     for(int i=1;t[i];i++){
42         u=tr[u][t[i]-'a']; //转移
```

```

43         for(int j=u;j and e[j]!=-1;j=fail[j]){
44             res+=e[j],e[j]=-1;
45         }
46     }
47     return res;
48 }
49 }//namespace AC
50
51 char s[N];
52
53 int main(){
54     cin>>n;
55     FOR(i,1,n)
56         cin>>(s+1),AC::insert(s);
57     cin>>(s+1);
58     AC::build();
59     cout<<AC::query(s);
60     return 0;
61 }

```

# KMP

[P3375 【模板】KMP字符串匹配](#)

[参考](#)

```

1  #include<bits/stdc++.h>
2  #define FOR(i,a,b) for(int i=(a);i<=(b);++i)
3  using namespace std;
4
5  const int N=1000007;
6
7  int nex[N];
8  char a[N],b[N];
9
10 int main(){
11     cin>>(b+1)>>(a+1);//A:模式串;B:文本串
12     int lena=strlen(a+1),lenb=strlen(b+1);
13     nex[1]=0;
14
15     int j;
16     FOR(i,2,lena){//A串自己匹配
17         while(j>0 and a[i]!=a[j+1]) j=nex[j];
18         if(a[i]==a[j+1]) j++;
19         nex[i]=j;
20     }
21     for(int i=1,j=0;i<=lenb;++i){//A串向B串匹配
22         while(j>0 and b[i]!=a[j+1]) j=nex[j];
23         if(b[i]==a[j+1]) j++;

```

```

24         if(j==lena) cout<<i-lena+1<<"\n",j=nex[j];
25     }
26     FOR(i,1,lena)
27         cout<<nex[i]<<" ";
28     return 0;
29 }

```

# Manacher

P3805 [【模板】manacher 算法](#)

[参考](#)

```

1  #include<bits/stdc++.h>
2  #define FOR(i,a,b) for(int i=(a);i<=(b);++i)
3  using namespace std;
4
5  const int maxn=3e7;
6  char s[maxn],st[maxn];
7  int R[maxn];
8
9  int pre_process(){
10     int len=strlen(st);
11     int j=2;
12     s[0]='^';
13     s[1]='$';
14     FOR(i,0,len-1){
15         s[j++]=st[i];
16         s[j++]='$';
17     }
18     s[j]='&';
19     return j;
20 }
21
22 int manacher(){
23     int len=pre_process();
24     int mid=1,RB=1,ans=-1;
25     FOR(i,1,len-1){
26         if(i<RB) R[i]=min(RB-i,R[mid*2-i]);
27         else R[i]=1;
28         while(s[i-R[i]]==s[i+R[i]]) R[i]++;
29         if(RB<i+R[i]){
30             mid=i;
31             RB=i+R[i];
32         }
33         ans=max(ans,R[i]-1);
34     }
35     return ans;
36 }

```

```

37
38  int main(){
39      cin>>st;
40      cout<<manacher();
41      return 0;
42  }

```

# 字典树(Trie)

## P3879 [TJOI2010] 阅读理解

### 字典树写法

[参考1](#)

[参考2](#)

[bitset用法](#)

[854ms/45.96MB](#)

```

1  #include<bits/stdc++.h>
2  #define FOR(i,a,b) for(int i=(a);i<=(b);++i)
3  using namespace std;
4
5  int nex[300007][26],n,cnt=0;
6  bitset<1001> b[500007]; //bool b[500007][1001];
7
8  void insert(char* s,int rol){
9      int len=strlen(s+1);
10     int now=0;
11     FOR(i,1,len){
12         int p=s[i]-'a';
13         if(!nex[now][p]) //如果$Trie$树中没有这个单词的前缀就进行编号
14             nex[now][p]=++cnt; //上文中说到的编号
15         now=nex[now][p]; //接着深入一层，更新现在的位置
16     }
17     b[now][rol]=1; //这个单词在第x行出现了
18 }
19
20 void check(char* s){
21     int len=strlen(s+1);
22     int now=0,flag=1;
23     FOR(i,1,len){
24         int p=s[i]-'a';
25         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
26             flag=0;
27             break;
28         }

```

```

29     now=nex[now][p]; //否则就更新位置
30 }
31 if(flag){
32     FOR(i,1,n) //题面上说按字典序输出
33         if(b[now][i]) cout<<i<<" "; //输出在哪些句子中出现过
34 }
35 putchar('\n');
36 }
37
38 int main(){
39     cin>>n;
40     char s[25];
41     FOR(i,1,n){
42         int l;cin>>l;
43         FOR(j,1,l){ //一个单词一个单词的插入Trie树里
44             cin>>(s+l);
45             insert(s,i);
46         }
47     }
48     int m;cin>>m;
49     FOR(i,1,m){
50         cin>>(s+l);
51         check(s);
52     }
53     return 0;
54 }

```

## STL: MAP + VECTOR 写法

### [参考](#)

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](#)

```

1  #include<bits/stdc++.h>
2  #define FOR(i,a,b) for(int i=(a);i<=(b);++i)
3  #define mem(a) memset(a,0,sizeof(a))
4  using namespace std;
5
6  const int maxn = 100001;
7  int n,m,num,cnt[maxn];
8  string s;
9  map<string,vector<int>> a;
10
11 int main(){
12     cin>>n;
13     FOR(i,1,n){
14         cin>>num;
15         FOR(j,1,num){

```

```

16         cin>>s;
17         a[s].push_back(i); //每一个单词就是一个vector,存句子的编号
18     }
19 }
20 cin>>m;
21 FOR(i,1,m){
22     cin>>s;
23     mem(cnt); //cnt就是去重的桶。每用一次输出不同的查询要清零。
24     int len=a[s].size();
25     FOR(j,0,len-1)
26         if(cnt[a[s][j]] == 0){
27             cout<<a[s][j]<<" ";
28             cnt[a[s][j]]++; //用桶去重
29         }
30     cout<<endl;
31 }
32 return 0;
33 }

```

## STL: MAP + SET 写法

### 参考

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](#)

```

1  #include<bits/stdc++.h>
2  #define FOR(i,a,b) for(int i=(a);i<=(b);++i)
3  using namespace std;
4
5  map<string,set<int> > m;
6
7  int main(){
8      int n,p,l;
9      string s;
10     cin>>n;
11     FOR(i,1,n){
12         cin>>l; //单词个数
13         FOR(j,0,l-1){
14             cin>>s; //单词
15             m[s].insert(i);
16         }
17     }
18     cin>>p;
19     while(p--){
20         cin>>s;
21         if(m.count(s)){ //如果m中存在元素s
22             for(auto iter=m[s].begin();iter!=m[s].end();++iter)
23                 cout<<*iter<<" ";

```

```
24     }
25     cout<<endl;
26 }
27 return 0;
28 }
```

# 字符串哈希

[P3370 【模板】字符串哈希](#)

[参考](#)

```
1  #include<bits/stdc++.h>
2  #define ll long long
3  #define FOR(i,a,b) for(int i=(a);i<=(b);++i)
4  using namespace std;
5
6  const int p=131;//13331
7  const int N=1e7+7;
8  int n,m;
9  ll a[N];
10 char str[N];
11
12 ll get_hash(char s[]){
13     ll res=0;
14     int len=strlen(s);
15     FOR(i,0,len-1)
16         res=res*p+(ll)s[i];
17     return res;
18 }
19
20 int main(){
21     cin>>n;
22     FOR(i,1,n){
23         cin>>str;
24         a[i]=get_hash(str);
25     }
26     int ans=1;
27     sort(a+1,a+1+n);
28     FOR(i,1,n-1)
29         if(a[i]!=a[i+1]) ans++;
30     cout<<ans;
31     return 0;
32 }
```