

原创

KMP & AC自动机

2019-09-21 15:51:36

Y-Y_-

阅读数 25

更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/weixin_44410512/article/details/101109082

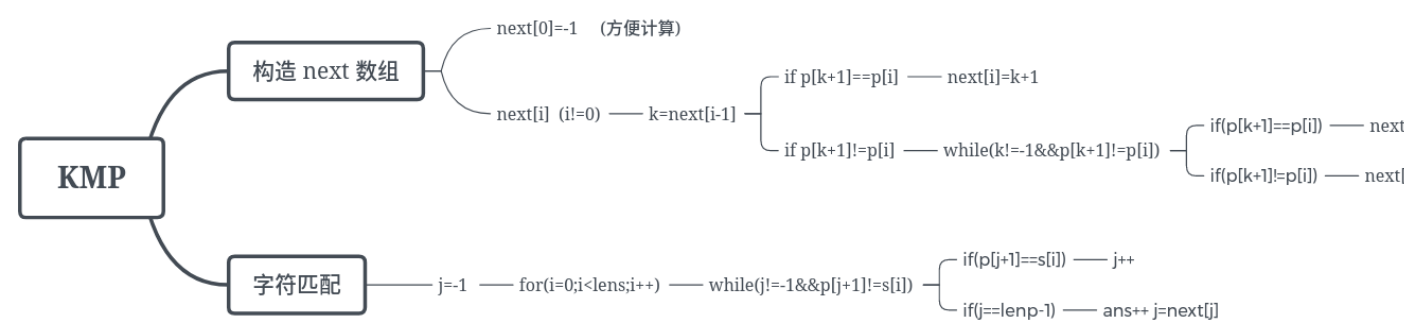
KMP 和 AC自动机 都可以解决字符串匹配问题
KMP是一对一匹配
AC自动机是多对一匹配

KMP

KMP核心思想是 利用字符串的前缀与后缀相同，失配时跳到等于后缀的前缀，可以不必从头开始。
这张图列举了字符"ABCDABD"所有的前缀和后缀 红色的表示前缀和后缀相等

模式串的各个子串	前缀	后缀
A	空	空
AB	A	B
ABC	A,AB	C,BC
ABCD	A,AB,ABC	D,CD,BCD
ABCD A	A,AB,ABC,ABCD	A,DA,CDA,BCDA
ABCD AB	A,A B,ABC,ABCD,ABCD A	B,A B,DAB,CDAB,BCDAB
ABCD ABD	A,AB,ABC,ABCD,ABCD A ABCD AB	D,BD,ABD,DABD,CDABD BCDAB D

例如：
用字符 P= "ABCDABCDE" 去匹配 字符 S = "ABCDABCDABCDABCD"
当匹配到 ABCDABCDABCDABCD时
发现' A ' 与' E ' 不匹配
那指针就可以不必从头开始，可以跳到ABCDABCDE 从A开始匹配
原理很简单，就是因为后缀等于前缀，那么本来后缀能匹配的上的前缀依然能匹配上
现在开始构造后缀等于前缀的指针数组 (next)
这里用到了DP的思想
当计算第i个字符的next指针
去找第i-1个字符的next指针所指向的字符
要是这个字符的下一个字符等于第i个字符 那么第i个字符的next指针指向这个字符的下标
否则继续去寻找这个字符反而next指针所指向的字符(开始循环)
原理还是很简单
因为要找第i个字符next指针，第i-1个字符已经找到到它的next
所以就在第i-1个字符所构成的后缀 与之相等的前缀的后面一个去找第i个字符



<https://blog.csdn.net/>

code

```

1  /*
2  https://www.luogu.org/problem/P3375
3  P3375
4  【模板】KMP 字符串匹配
5  */
6  #include<iostream>
7  #include<cstdio>
8  #include<queue>
9  #include<stack>
10 #include<cstring>
11 using namespace std;
12 #define maxn 1000005
13 char s[maxn];
14 char p[maxn];
15 int ans[maxn];
16 int next_[maxn];
17 int cnt;
18 int lens, lenp;
19 void build(){
20     next_[0]=-1;
21     int k=-1;
22     for(int i=1;i<lenp;i++){
23         while(k!=-1&&p[k+1]!=p[i]) k=next_[k];
24         if(p[k+1]==p[i]) k++;
25         next_[i]=k;
26     }
27 }
28 void kmp(){
29     cnt=0;
30     lenp=strlen(p);
31     lens=strlen(s);
32     build();
33     int j=-1;
34     for(int i=0;i<lens;i++){
35         while(j!=-1&&p[j+1]!=s[i]) j=next_[j];
36         if(p[j+1]==s[i]) j++;
37         if(j==lenp-1){
38             ans[cnt++]=i-lenp+2;
39             j=next_[j];
40         }
41     }
42 }
43 int main(){
44     scanf("%s%s", s, p);
45     kmp();
46     for(int i=0;i<cnt;i++){
47         printf("%d\n", ans[i]);
48     }
49     for(int i=0;i<lenp;i++){
50         printf("%d ", next_[i]+1);
51     }
52     return 0;
53 }
54

```

AC自动机

AC自动机 核心思想是 在由所有字符串p所构成的字典(前缀树)树上跑KMP

要跑KMP 就要构造next数组, 在这里叫它失配指针fail

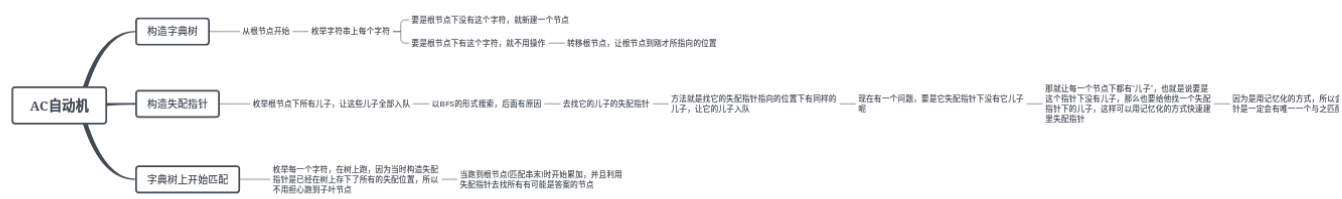
同样构造fail指针要用到dp思想

其实和KMP找next指针一样

依旧还是去找它的父亲节点的失配指针的字符的儿子有无这个字符

要是没有就是它的失配指针

要是没有就继续找它的失配指针 就和KMP一样



<https://blog.csdn.net/>

code

```
1  /*
2  https://www.luogu.org/problem/P3796
3  P3796
4  【模板】AC自动机（加强版）
5  */
6  #include <queue>
7  #include <cstdlib>
8  #include <cmath>
9  #include <cstdio>
10 #include <string>
11 #include <cstring>
12 #include <iostream>
13 #include <algorithm>
14 #include <map>
15 using namespace std;
16 char a[160][100];
17 const int maxn = 1*1e6+9;
18 int trie[maxn][26]; // 字典树
19 int fail[maxn]; // 失败时的回溯指针
20 int cnt = 0;
21 int ans[160];
22 int mp[maxn];
23 int anss;
24 void insertWords(char s[],int id){
25     int root = 0;
26     int len=strlen(s);
27     for(int i=0;i<len;i++){
28         int next = s[i] - 'a';
29         if(!trie[root][next])
30             trie[root][next] = ++cnt;
31         root = trie[root][next];
32     }
33     mp[root]=id;
34 }
35 void getFail(){
36     queue <int>q;
37     for(int i=0;i<26;i++){
38         if(trie[0][i]){
39             fail[trie[0][i]] = 0;
40             q.push(trie[0][i]);
41         }
42     }
43     while(!q.empty()){
44         int now = q.front();
45         q.pop();
46         for(int i=0;i<26;i++){
47             if(trie[now][i]){
48                 fail[trie[now][i]] = trie[fail[now]][i];
49                 q.push(trie[now][i]);
50             }
51             else{
52                 trie[now][i] = trie[fail[now]][i];
53             }
54         }
55     }
56 }
```

```
55     }
56 }
57 void query(char s[]){
58     int now = 0;
59     int len=strlen(s);
60     for(int i=0;i<len;i++){
61         now = trie[now][s[i]-'a'];
62         for(int j=now;j=j=fail[j]){
63             ans[mp[j]] ++;
64         }
65     }
66 }
67 char s[maxn];
68 int main() {
69     int n;
70     while(~scanf("%d", &n)&& n){
71         memset(ans,0,sizeof ans);
72         memset(fail,0,sizeof fail);
73         memset(trie,0,sizeof trie);
74         memset(mp,0,sizeof mp);
75         cnt=0;
76         anss=0;
77         for(int i=1;i<=n;i++){
78             scanf("%s", a[i]);
79             insertWords(a[i],i);
80         }
81         getFail();
82         scanf("%s", s);
83         query(s);
84         for(int i=1;i<=n;i++){
85             if(ans[i]>anss) anss=ans[i];
86         }
87         printf("%d\n", anss);
88         for(int i=1;i<=n;i++){
89             if(ans[i]==anss){
90                 printf("%s\n", a[i]);
91             }
92         }
93     }
94     return 0;
95 }
```

有 0 个人打赏

文章最后发布于: 201

©2019 CSDN 皮肤主题: 大白 设计师: CSDN官方博客