

数据结构作业第四章——串

庄震丰 22920182204393

Nov. 19th, 2019

4-12

题目要求：编写一个实现串的置换操作 Replace(&S,T,V) 的算法。

算法分析：由于与 4-17 一样，这里采用普通算法求解子串位置，即一个个位置相比。

空间复杂度 $O(N+M)=O(N)$ ，时间复杂度 $O(NM)$ 。

4-12

```

1 //常规方法替换子串
2 #include<bits/stdc++.h>
3 using namespace std;
4 #define maxn 1000
5 string S;
6 void Replace(string &S,string T, string V)
7 {
8     int flag[maxn]={0};
9     int p=0,d=V.length()-T.length();
10    for (int i=0;i<S.length();i++)
11    {
12        if (S.substr(i,T.length())==T)
13            flag[++p]=i+1+d*(p-1);
14    }
15    for (int i=1;i<=p;i++)
16        S.replace(flag[i]-1,T.length(),V);
17 }
18 int main()
19 {
20     string t,v;
21     int operatenum;
22     cout<<"Please input the operaternumber:"<<endl;
23     cin>>operatenum;
24     getchar();
25     getline(cin,S);
26     for (int i=1;i<=operatenum;i++)
27     {
28         getline(cin,t);
29         getline(cin,v);
30         Replace(S,t,v);
31     }
32     cout<<S;
33     return 0;
34 }
```

4-17

题目要求：编写算法，实现串的置换操作 Replace(&S,T,V)。

算法分析：采用多次匹配的 KMP 算法，求解所有匹配点，当且仅当当前模式串在源串中没有重叠，若有重叠只考虑

第一个。

先求 nextval 数组，求解完后，利用 nextval 数组进行匹配，匹配失败的时候直接把当前点 j 跳到模式串中的 nextval[j]，去匹配，减少无效匹配次数。

空间复杂度 $O(N)$ ，匹配时间复杂度 $O(N+M)$ ，初始化为 $O(M)$ ，总体实现是线性时间。

4-17.cpp

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define maxn 1000
4  string S;
5  int n;
6  void getNext(string b, int *nextval)//求数组next
7  {
8      int len = b.length();
9      int j = 0 , k = -1 ;
10     nextval[0] = -1 ;
11     while(j < len)//查找多次且可重叠时不能减一，因为该单词的末尾加一的也需要被下一次查询用到。lennext
12     {
13         if(k == -1 || b[k] == b[j])
14         {
15             k++;
16             j++;
17             // 下面数组的优化next
18             if(b[k] != b[j])
19                 nextval[j] = k ;
20             else
21                 nextval[j] = nextval[k];
22         }
23         else
24         {
25             k = nextval[k];
26         }
27     }
28 }
29 void Replace(string &S,string b,string a)
30 {//子串替换
31     int flag[maxn]={0};
32
33     getline(cin,S);
34     int nextval[10009];
35     getline(cin,b);
36     getline(cin,a);
37     getNext(b, nextval);
38
39     int lena = a.length() , lenb = b.length();
40
41     int i = 0 , j = 0;
42     int ans = 0 ,d=lena-lenb;
43     while(i < lena)
44     {
45         if(j == -1 || S[i] == b[j])

```

```
46         {
47             i++ ;
48             j++ ;
49         }
50         else
51         {
52             j = nextval[j];
53         }
54         if(j == lenb)
55         {
56             flag[++ans]=i+1+d*(ans-1);
57             j = nextval[j] ;
58         }
59     }
60     printf("%d\n" , ans);
61     for (int i=1;i<=ans;i++)
62         S.replace(flag[i]-1,lenb,a);
63 }
64 int main()
65 {
66     string A,B;
67     cout<<"please input the S string:"<<endl;
68     getline(cin,S); //输入源串
69     cout<<"please input the operate number:"<<endl;
70     cin>>n; //输入操作总数
71     getchar();
72     for (int i=1;i<=n;i++)
73     {
74         getline(cin,A);
75         getline(cin,B);
76         Replace(S,A,B);
77     }
78     return 0 ;
79 }
```