序列自动机

定义

序列自动机是接受且仅接受一个字符串的子序列的自动机。

本文中用 s 代指这个字符串。

状态

若 s 包含 n 个字符，那么序列自动机包含 $n+1$ 个状态。

令 t 是 s 的一个子序列，那么 $\delta(start,t)$ 是 t 在 s 中第一次出现时末端的位置。

也就是说，一个状态 i 表示前缀 $s[1..i]$ 的子序列与前缀 $s[1..i-1]$ 的子序列的差集。

序列自动机上的所有状态都是接受状态。

转移

由状态定义可以得到 $\delta(u,c)=\min\{i|i>u,s[i]=c\}$ 也就是字符 c 下一次出现的位置。

为什么是“下一次”出现的位置呢？因为若 $i>j$ 后缀 $s[i..|s|]$ 的子序列是后缀 $s[j..|s|]$ 的子序列的子集，一定是选尽量靠前的最优。

构建

从后向前扫描，过程中维护每个字符最前的出现位置 $$ \begin{array}{ll} 1 & \textbf{Input. } \text{A string } S\\ 2 & \textbf{Output. } \text{The state transition of the sequence automaton of }S \\ 3 & \textbf{Method. } \\ 4 & \textbf{for }c\in\Sigma\\ 5 & \qquad next[c]\gets null\\ 6 & \textbf{for }i\gets|S|\textbf{ downto }1\\ 7 & \qquad next[S[i]]\gets i\\ 8 & \qquad \textbf{for }c\in\Sigma\\ 9 & \qquad\qquad \delta(i-1,c)\gets next[c]\\ 10 & \textbf{return }\delta \end{array} $$ 这样构建的复杂度是 $O(n|\sum|)$

例题

计蒜客 Subsequence

题意：给定一个字符串 S 输入 N 个字符串 T_i 判断 T_i 是否为 S 的子序列。

题解：序列自动机模板题，对文本串 S 求出 $nxt[]$ 数组，对每个 T_i 跑 $nxt[]$ 数组即可，若提前跑出去了则不是子序列。

代码

```
#include<bits/stdc++.h>
using namespace std;
const int N=1e5+5,INF=0x3f3f3f3f;
int nxt[N][26];
char s[N],t[N];
int main(){
    scanf("%s",s+1);
    int len=strlen(s+1);
    for(int i=0;i<26;i++) nxt[len][i]=nxt[len+1][i]=INF;
    for(int i=len;i>=1;i--){
        for(int j=0;j<26;j++){
            nxt[i-1][j]=nxt[i][j];
        }
        nxt[i-1][s[i]-'a']=i;
    }
    int m;
    scanf("%d",&m);
    while(m--){
        scanf("%s",t+1);
        int lent=strlen(t+1);
        int pos=0;
        bool ok=1;
        for(int i=1;i<=lent;i++){
            pos=nxt[pos][t[i]-'a'];
            if(pos==INF){
                ok=0;break;
            }
        }
        if(ok) puts("YES");
        else puts("NO");
    }
    return 0;
}
```

From:
https://wiki.cvbbacm.com/ - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:lgwza:%E5%BA%8F%E5%88%97%E8%87%AA%E5%8A%A8%E6%9C%BA&rev=1601965657

Last update: **2020/10/06 14:27**