

知识精炼（二）



主讲人：邓哲也



BZOJ 1030 文本生成器

给定 n 个小串，构造一个长度为 m 的大串。

一个串合法当且仅当包含至少一个小串。

问合法串的数目。

$N \leq 60$ ，所有文本长度都不会超过100，只含有大写字母。

样例输入：

2 2

A

B

样例输出：

100

BZOJ 1030 文本生成器

把所有的小串插入 AC 自动机。

对于 AC 自动机的节点，计算出从根到每个节点是否会经过至少一个小串，然后在上面 dp 即可。

$f[v][n][x]$:

$v=1$ 当前经过至少一个小串， $v=0$ 表示没有

n 表示当前构造的串长

x 表示现在在 AC 自动机的 x 节点上

BZOJ 1030 文本生成器

首先计算出 $\text{cnt}[x]$ 表示 AC 自动机上从根走到 x 的路径上是否会经过至少一个小串。

这个可以在 BFS 计算 fail 指针的时候计算。

BZOJ 1030 文本生成器

插入 Trie 树:

```
for (int i = 1; i <= n; i++) {  
    scanf( "%s" , s );  
    int x = 0;  
    for (int j = 0; s[j]; j++) {  
        if (!ch[x][s[j] - 'A' ]) ch[x][s[j] - 'A' ] =  
tot ++;  
        x = ch[x][s[j] - 'A' ];  
    }  
    cnt[x] = 1;  
}
```

BZOJ 1030 文本生成器

BFS 遍历求 next 指针:

```
int s = 0, e = 1;
q[0] = next[0] = 0;
while(s < e) {
    int x = q[s ++];
    cnt[x] |= cnt[next[x]];
    for (int j = 0; j < 26; j ++)
        if (ch[x][j]) {
            q[e ++] = ch[x][j];
            next[ch[x][j]] = (!x) ? 0 : ch[next[x]][j];
        } else ch[x][j] = ch[next[x]][j];
}
```

BZOJ 1030 文本生成器

看 $f[v][n][x]$ 可以转移到哪些节点：

对于 26 条出边： $x \rightarrow \text{next}[x][k] \quad (0 \leq k \leq 25)$

记 $y = \text{next}[x][k]$

$f[v \mid \text{cnt}[y]][n + 1][y] += f[v][n][x]$

BZOJ 1030 文本生成器

DP 统计:

```
f[0][0][0]=1;
```

```
for(int i=0;i<m;i++)
```

```
    for(int o=0;o<=1;o++)
```

```
        for(int j=0;j<tot;j++)
```

```
            if(f[o][i][j])
```

```
                for(int k=0;k<26;k++) {
```

```
                    int x=ch[j][k];
```

```
                    f[o|cnt[x]][i+1][x]=(f[o|cnt[x]][i+1][x]+f[o][i][j])
```

```
%10007;
```

```
}
```


BZOJ 1030 文本生成器

统计答案：

```
int ans = 0;
for (int i = 0; i < tot; i ++)
    ans = (ans + f[1][m][i]) % 10007;
```

时间复杂度 $O(26mL)$ ， L 为所有字符串的长度之和。

BZOJ 1030 文本生成器加强版

可以注意到状态的转移可以写成矩阵乘法的形式。

用矩阵乘法快速幂来加速 dp 的转移。

时间复杂度 $O(26L^3 \log m)$ ， L 为所有字符串的长度之和。

下节课再见