

# 知识精炼（一）



主讲人：邓哲也



# HDU 4628 Pieces

有一个长度不超过 16 的字符串。

每次你可以从中删除一个子序列，但是要求这个子序列是回文的。

问最少删除几次可以把这个字符串删光。

样例：

2

aa        (1次)

abb       (2次)

# HDU 4628 Pieces

每次可以选择一个子序列，而子序列是可以用二进制来表示的。

用一个  $n$  位的二进制数  $s$  来表示，如果第  $i$  位是 1，则表示第  $i$  个数在这个子序列中。

由此衍生出：用  $f[s]$  来表示把  $s$  这个子序列删完的最小步数。

答案自然就是  $f[(1 \ll n) - 1]$

## HDU 4628 Pieces

如果  $s$  对应的子序列是一个回文串，那么  $f[s] = 1$

其他的  $s$  应该怎么转移？

一个显然的想法，可以把  $s$  分成两个不相交的集合  $x$  和  $y$ ,

即  $x \cap y = \emptyset, x \cup y = s$

$f[s] = \min(f[s], f[s \setminus x] + 1)$  ( $x$  是回文子串)

直接枚举  $x$  和  $y$  的时间复杂度是  $O(2^n)$ ，总时间复杂度是  $O(4^n)$ ，无法承受。

# HDU 4628 Pieces

注意到  $x$  和  $y$  都是  $s$  的子集。

所有集合的子集个数之和的级别是  $O(3^n)$

(证明可以对每个集合考虑贡献, 二项式定理)

我们只要枚举  $s$  的子集  $x$ , 那么  $y$  自然就是  $s - x$

# HDU 4628 Pieces

枚举  $x$  的子集代码:

```
for(int i = x; i; i = (i - 1) & x) {  
    // i 就是 x 的子集  
}
```

首先  $x$  的子集一定不会比  $x$  大。

考虑二进制减法的意义：把最低位的 1 变成 0，在那一位后面全都补上 1。然后  $\& x$  则会快速跳到下一个子集。

# HDU 4628 Pieces

枚举  $x$  的子集代码:

```
for(int i = x; i; i = (i - 1) & x) {  
    // i 就是 x 的子集  
}
```

这样就可以不重不漏的枚举所有  $x$  的子集。

总的时间复杂度是  $O(3^n)$

# HDU 4628 Pieces

```
int judge(int s, int len){  
    char str[20];  
    int tot = 0;  
    for(int i = 0;i < len;i ++)  
        if (s >> i & 1) str[tot ++] = str[i];  
    for(int i = 0;i < len / 2;i ++)  
        if (str[i] != str[tot - i - 1])  
            return 0;  
    return 1;  
}
```



# HDU 4628 Pieces

```
scanf( "%s" , str);
int len = strlen(str);
for(int i = 1;i < (1 << len);i ++) flag[i] = check(i, len);
memset(f, 63, sizeof(f));
for(int i = 1;i < (1 << len);i ++)
    for(int s = i;s;s = (s - 1) & i)
        if(flag[s])
            f[i] = min(f[i], f[i^s] + 1);
printf( "%d\n" , f[(1 << len) - 1]);
```

下节课再见