



# 递归

## 例2. Fibonacci数列求解.

$$f(n) = \begin{cases} 0, & \text{当 } n=0 \\ 1, & \text{当 } n=1 \\ f(n-1)+f(n-2), & \text{当 } n>1 \end{cases}$$

0, 1, 1, 2, 3, 5, 8, 13, 21, .....

请分别写出递归与非递归程序!

## 递归求解:

```
1. int f (int n)
2. {
3.     if (n<=1)
4.         return (n);
5.     else
6.         return (f(n-1)+f(n-2));
7. }
```

## 非递归求解:

```
1. int f (int n)
2. {      int pre, now, next, j;
3.      if (n<=1) return (n);
4.      else
5.      {      pre=0; now=1;
6.              for(j=2;j<=n;j++)
7.              {      next=pre+now;
8.                      pre=now;
9.                      now=next;
10.             }
11.             return(next);
12.     }
13.}
```

## 回文串

如果一个字符串从左向右读和从右向左读完全相同（不区分大小写），则这个字符串称为回文串（palindrome），例如“noon”、“madam”等都是回文串。

请分别写出递归与非递归程序！

可以用递归方法检测一个字符串S是否为回文串。

设  $S = "S_h S_{h+1} \dots S_{t-1} S_t"$  ,

$h$  是第一个字符的位置,  $t$  是最后一个字符的位置。

1) 若  $S$  是空串或长度为1, 则  $S$  为回文串, 否则:

2) 若  $S_h \neq S_t$ , 则  $S$  不是回文串, 否则, 递归地检测  $S$  的子串

$"S_{h+1} \dots S_{t-1}"$  .

## 回文串

```
1. bool palindrome(string &s,unsigned h,unsigned t)
2. {
3.     if (h>=t) return 1;
4.     if(tolower(s[h])==tolower(s[t]))
5.         return palindrome(s,h+1,t-1);
6.     else
7.         return 0;
8. }
```

非递归程序：设  $S = "S_h S_{h+1} \dots S_{t-1} S_t"$  ,

$h$  是第一个字符的位置,  $t$  是最后一个字符的位置。

1) 若  $S$  是空串或长度为1, 则  $S$  为回文串, 否则:

2) 若  $S_h \neq S_t$ , 则  $S$  不是回文串, 否则,  $h = h + 1, t = t - 1$ , 继续监测  $S$  串的 " $S_h \dots S_t$ ".



## 回文串

```
1. bool palindrome(string &s)
2. {
3.     int h=0,t=strlen(s)-1;
4.     while(h<=t)
5.     {
6.         if(s[h]!=s[t])return 0;
7.         h++;t--;
8.     }
9.     return 1;
10.}
```