3.6 栈的应用--栈与递归

递归是程序设计中的一个强有力的工具

- 很多数学函数是递归定义的;
- 有的数据结构，如二叉树、广义表等，由于结构本身固有的递归特性，则它们的操作可递归地描述;
- 有些问题，虽然问题本身没有明显的递归结构，但用递归求解比迭代求解更简单。如背包问题、Hanoi塔问题、八皇后问题

3.6 栈的应用--栈与递归

```
int factorial(int n)
{
        if(n == 1)
                return 1;
        else
                return(n*factorial(n-1));
};
```

- 函数自己调用自己称为递归调用
- 直接调用
- 间接调用

3.6 栈的应用--栈与递归

函数调用的过程

调用前：

（1）将所有的实参、返回地址传递给被调用函数保存

（2）为被调用函数的局部变量分配存储区

（3）将控制转移到被调用函数入口

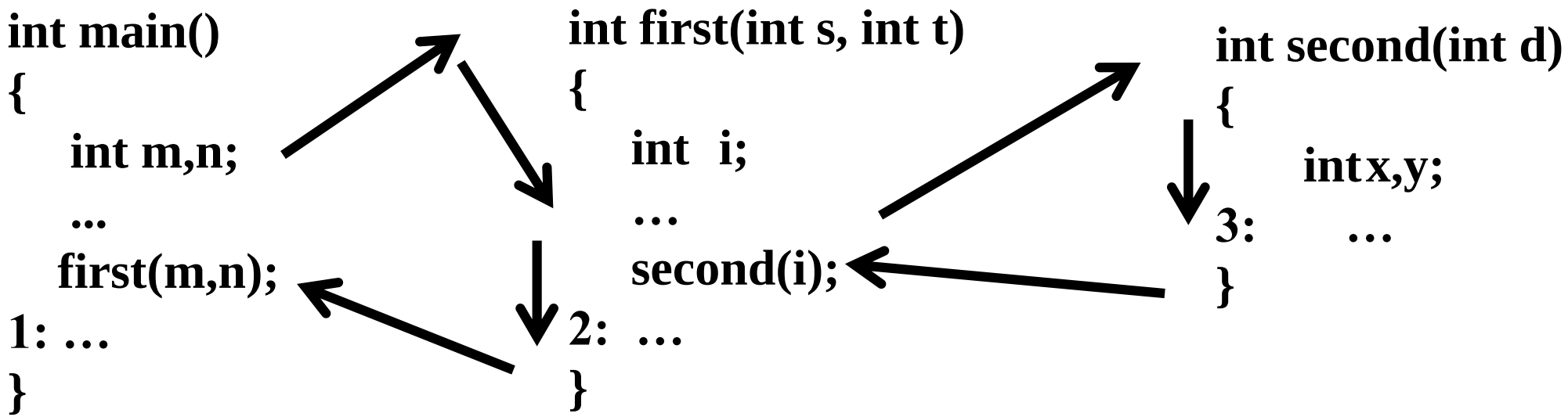
调用后：

（1）保存被调用函数的计算结果

（2）释放被调用函数的数据区

（3）依照被调用函数保存的返回地址将控制转移到调用函数

嵌套调用举例



- 多个函数嵌套调用时，按照“后调用先返回”的原则进行
- 内存管理实行“栈式管理”
- 运行栈：运行时动态分配的空间

算法3-15

```
1  int factorial_NR(int n)//阶乘函数
2  {
3          int res; LinkStack st;
4          st = SetNullStack_Link(n);
5          while(n>0){
6                  Push_link(st, n); n = n-1;
7          }
8          res = 1;
9          while (!IsNullStack_link(st)){
10                 res = res*Top_link(st);
11                 printf("当前栈顶元素是： %d\n", Top_link(st));
12                 Pop_link(st);
13         }
14         return(res);
15 }
```

思考： Koch 雪花

Koch 雪花是Helge VonKoch于1904 年提出来的一种模型,它将一根定长的线段side通过分割产生图案,由于每次分割的深度和Koch角的不同，每个图案都有自身很明显的不同特点。在每次的分割中，对side/N(N>0,N是整数)都三等分, 在中间的那一段上再凸起一个小正三角形, 这样一直下去, 所得图像的形状类似雪花。