



本文内容是对递归算法的梳理和总结，本文内容包括：

递归(recursion)

概念

算法

示例

规则

1 递归(recursion)

1.1 概念

- 递归是一种自调用函数，也就是说函数内部调用函数本身，每次调用时传入不同的变量。
- 递归有助于编程者解决复杂的问题，同时让代码变得简洁。

1.2 算法

递归算法中包含：递推、回推、终止条件

- 方法内部调用方法自己的过程称为**递推**。
- 满足方法内部不在调用方法本身的条件称为递归的**终止条件**，
- 方法内部返回到上一次调用的过程称为**回推**。

1.3 示例

题目：求 $n!$

分析：

n 的阶乘可以等价于 $n * (n-1)!$

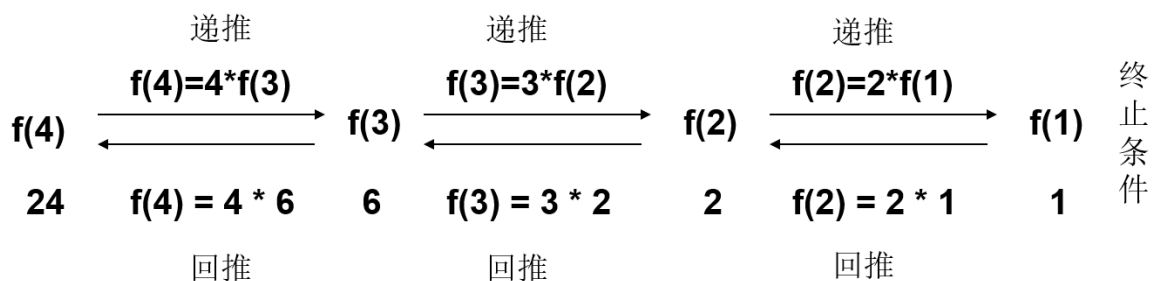
等价于 $n * (n-1) * (n-2)!$

等价于 $n * (n-1) * (n-2) * (n-3)!$

直到乘1为止

```
1 public static int f(int n){
2     if(n==1){ //终止条件
3         return 1; //回推
4     }
5     return n * f(n-1); //递推
6 }
7 public static void main(String[] args) {
8     int result = f(4);
9     System.out.println(result);
10 }
```

分析递归调用的过程



1.4 规则

1. 指向递归方法时，要创建一个新的受保护的独立空间（栈空间）
2. 方法的局部变量是独立的，不会相互影响，比如变量n
3. 如果方法中使用的是引用类型的变量（比如数组），就会共享该引用变量
4. **递归必须向终止条件逼近**，否则就是无限递归，就会出现 StackOverflowError（死龟）
5. 当递归方法遇到return，就开始回推，回推的过程中遵循**谁调用就将结果返回给谁**。