

算法讲解|如何直观理解递归算法

周行算协 周行算协 2020-10-18 10:07



什么是递归

递归是程序设计语言中的一种广泛应用的算法，能大大减少多次重复计算的代码量。

递归就是某个函数或者操作在运行中调用自己的现象，类似于一个连环套娃的过程。



////

///

//

递归的实现

当然，由于计算机的计算能力有限，并且最终需要通过深层的结果求得浅层的结果，我们不可能让递归程序无限进行下去。



禁止套娃

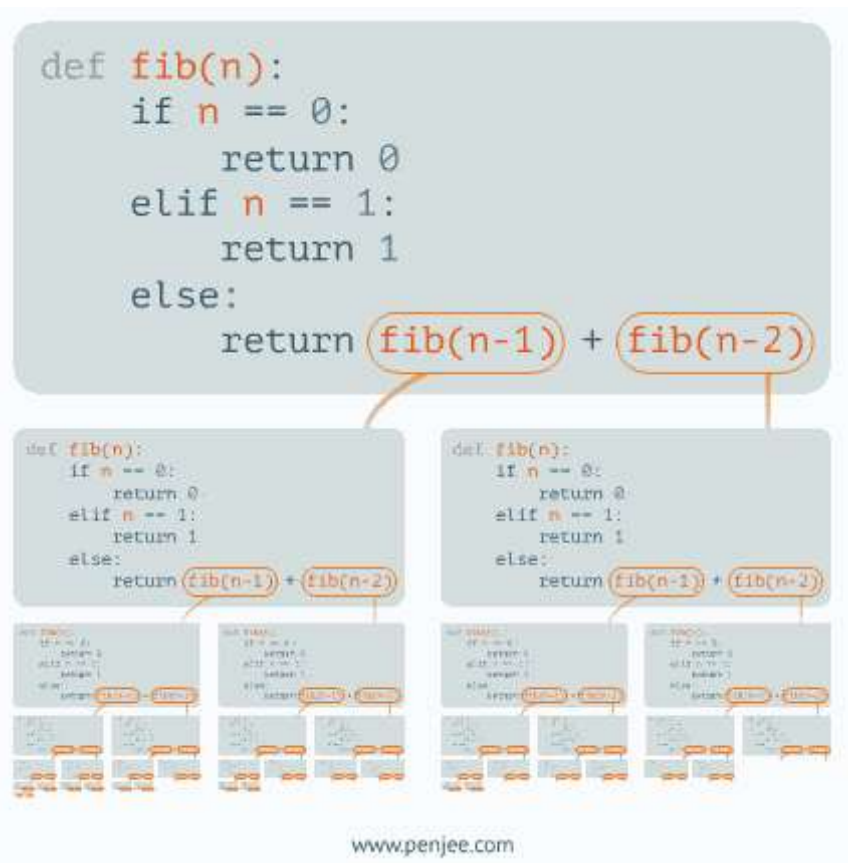
举个例子，我们熟悉的养兔子问题（斐波那契数列）中的项就可以用递归求解。
我们只需要知道

$$\begin{aligned} f(0) &= 0; \\ f(1) &= 1; \end{aligned}$$

$$f(i)=f(i-1)+f(i-2);(2\leq i\leq n)$$

就可以了。

程序会先运行到求 $f(n)$ 的环节，又因为此时 $f(n-1),f(n-2)$ 仍未求出，程序会递归到下一层求解，之后求 $f(n-1)$ 时又因为 $f(n-2),f(n-3)$ 未知，又会进入下一层递归...直到0和1时，不能继续往下递归，返回上层。用 $f(0)$ 和 $f(1)$ 求出 $f(2)$ 进而返回上层求出 $f(3)$ ，以此类推，直到求出 $f(n)$ 。



我们将上述求解过程分成许多小过程，每个小过程的小结果称为**状态**。

在递归过程中，我们求某一状态前需要之前的状态作为支撑来求解，该过程称为**状态转移**。

///

////

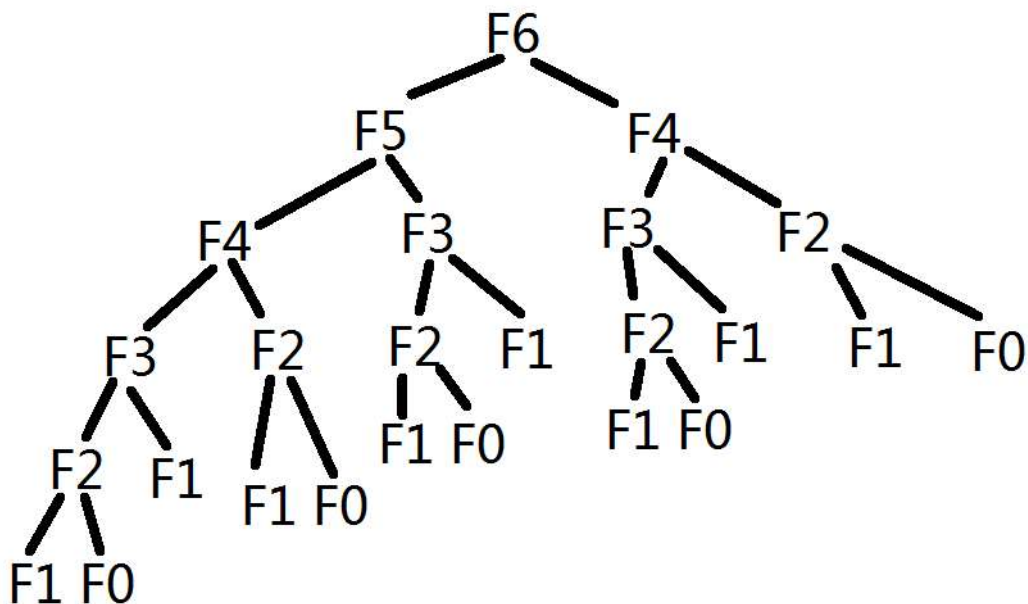
//

代码实现

我们来看看是如何用代码实现递归这一算法的吧，我们编写一个程序，输入小于等于30的非负整数*i*，输出*f(i)*，以下给出代码的C语言版本。

```
1  #include<stdio.h>
2  int Fibonacci(int n) {
3      if (n == 0 || n == 1) return n;
4      //f(0)==0, f(1)==1, 直接返回上层
5      return Fibonacci(n - 1) + Fibonacci(n - 2);
6      /*递归到下一层,
7      求出f(n-1), f(n-2)后再返回该层求f(n)*/
8  }
9  int main(void) {
10     int i, fi; //i为输入的数列的序数, fi为输出的f(i)
11     scanf("%d", &i); //输入
12     fi = Fibonacci(i); //调用递归函数进行求解
13     printf("%d\n", fi); //输出
14     return 0;
15 }
```

递归过程形成了一个*n*层的二叉树（每个节点最多有两个子节点），整棵树最多有2的*n*次方个节点，故该算法的时间复杂度为 $O(2^n)$





////

///

//

下面留下几道例题，感兴趣的小伙伴们可以挑战一下哦！

昆虫繁殖

问题描述

科学家在热带森林中发现了一种特殊的昆虫，这种昆虫的繁殖能力很强。每对成虫过 x 个月产 y 对卵，每对卵要过两个月长成成虫。假设每个成虫不死，第一个月只有一对成虫，且卵长成成虫后的第一个月不产卵（过 x 个月产卵），问过 z 个月以后，共有成虫多少对？

输入

包含三个整数，用空格隔开，分别表示 x 、 y 、 z 。

题目范围 $x \geq 1, y \geq 1, z \geq x$

输出

z 个月后成虫的对数

样例输入：

1 2 8

样例输出：

37

超级楼梯

问题描述

有一楼梯共 M 级，刚开始时你在第一级，若每次只能跨上一级或二级，要走上第 M 级，共有多少种走法？

...

输入

输入数据首先包含一个整数N，表示测试实例的个数，然后是N行数据，每行包含一个整数M ($1 \leq M \leq 40$)，表示楼梯的级数。

输出

对于每个测试实例，请输出不同走法的数量

样例输入

2

2

3

样例输出

1

2

解析我们会在下期推送放出，如果想获取更多资料欢迎加入周行算协招新群↓↓↓



end.

文字，代码，：李怡凡

配图：unsplash，penjee，网络

阅读 76



周行算协

分享

收藏

1

5