

剑指Offer（七）：斐波那契数列

© 2017年11月24日 10:53:36 3 4,771 °C 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

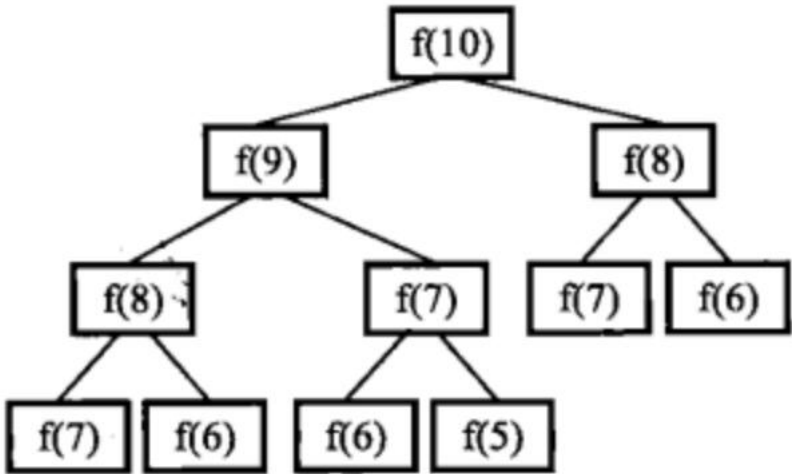
大家都知道斐波那契数列，现在要求输入一个整数n，请你输出斐波那契数列的第n项。（n<=39）

斐波那契数列公式为：

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

1、思路

这道题递归很好写，但是存在很严重的效率问题。我们以求解f(10)为例分析递归的求解过程。想求f(10)，需要先求得f(9)和f(8)。同样，想求得f(9)，需要先求的f(8)和f(7)....我们可以用树形结构来表示这种依赖关系，如下图所示：



我们不难发现在这棵树中有很多结点是重复的，而且重复的结点数会随着n的增加而急剧增加，这意味计算量会随着n的增加而急剧增大。事实上，递归方法计算的时间复杂度是以n的指数的方式递增的。

所以，使用简单的循环方法来实现。

2、代码

C++:


C++

```
1 class Solution {
2 public:
3     int Fibonacci(int n) {
4         if(n <= 0)
5             return 0;
6         if(n == 1)
7             return 1;
8         int first = 0, second = 1, third = 0;
9         for (int i = 2; i <= n; i++) {
10             third = first + second;
11             first = second;
12             second = third;
13         }
14         return third;
15     }
16 };
```

Python2.7:

Python

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def Fibonacci(self, n):
4         # write code here
5         if n <= 1:
6             return n
7         first, second, third = 0, 1, 0
8         for i in range(2, n+1):
9             third = first + second
10            first = second
11            second = third
12        return third
```



微信公众号

分享技术，乐享生活：微信公众号搜索「JackCui-AI」关注一个在互联网摸爬滚打的潜行者。

伟大的艺术品不必追随潮流，他本身就能引领潮流。--- 乔布斯