



链滴

[学习分享] 基于 python 实现的汉诺塔递归实现与分析 (内容偏水)

作者: [yaoqiwood](#)

原文链接: <https://ld246.com/article/1568363452471>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

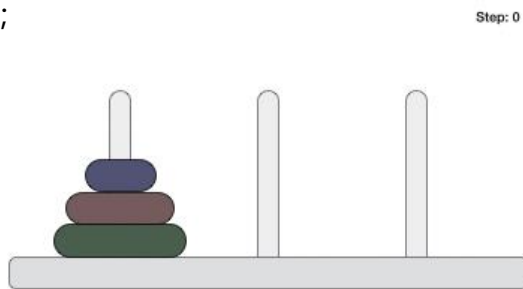
规则

以下代码基于Python3 实现

这段时间工作较之过去会少些，也就多了点时间研究和学习python，教程是跟着廖雪峰大佬的，之前学得还是蛮顺利的，直到遇到了递归那一章的练习，内容是关于汉诺塔移动过程，要求给出个变量 n,a,b,c

突然忘记介绍,首先，汉诺塔英文名又叫 tower of hanoi

玩法如下：



规则是，以上图为例，分别设定三个柱子为A,C,B。游戏的目的是将左边柱子上的1, 2, 3 碟子，由小至大按原样移动到柱子C上，每次只能一次移动一个碟子，且大碟子不可以堆放到比它本小的碟子上。

首先贴上代码：

```
def move(x, y):  
    print('%c ----> %c' % (x, y))  
  
def plat_move(n, a, b, c):  
    if n == 1:  
        move(a, c)  
    else:  
        plat_move(n - 1, a, c, b)  
        plat_move(1, a, b, c)  
        plat_move(n - 1, b, a, c)
```

然后我们写个调用执行一下，第一个n的参数是代表有多少个碟子，第二三个参数则是分别表每个柱子的名称。先假设有3个大小不同且从小到大自上而下排列在A柱子，又有B,C两个柱子。

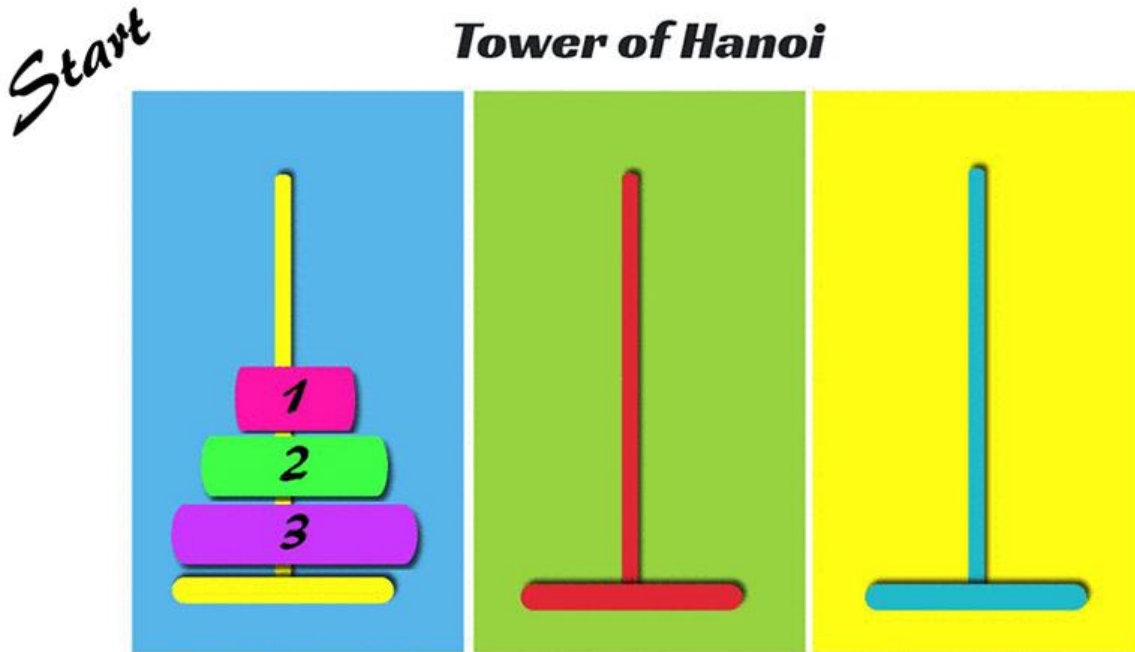
```
plat_move(3, 'A', 'B', 'C')
```

之后查看控制台结果会发现

```
A ----> C  
A ----> B  
C ----> B  
A ----> C  
B ----> A  
B ----> C
```

A ----> C

 找来gif图与结果对比:



 可以看到步骤是一致的。那么这个递归的实现就这样实现了，但是很多人大概看到这里会很以理解，为什么通过是调换调用的两次参数的位置就可以实现整个汉诺塔的移动过程（且还符合规则，起初这里我也卡了两天，三个还能尝试分解出整个过程，四个就完全就没有办法，大概查看了许多乎上的答案后，这才有了基本的理解。

分析

 整体实际上从过程分析，我们可以先对碟子的个数进行3，2，1分别拆解：

首先我们可以将碟子的个数设定为n
n=3 的状况

A ----> C
A ----> B
C ----> B
A ----> C
B ----> A
B ----> C
A ----> C

n=2 的状况

A ——> B
A ——> C
B ——> C

n=1 的状况
A ——> C

 可以发现当 $n=1$ 时，直接将盘子移动到了目标柱子上，那么也就有了递归中的那一步：

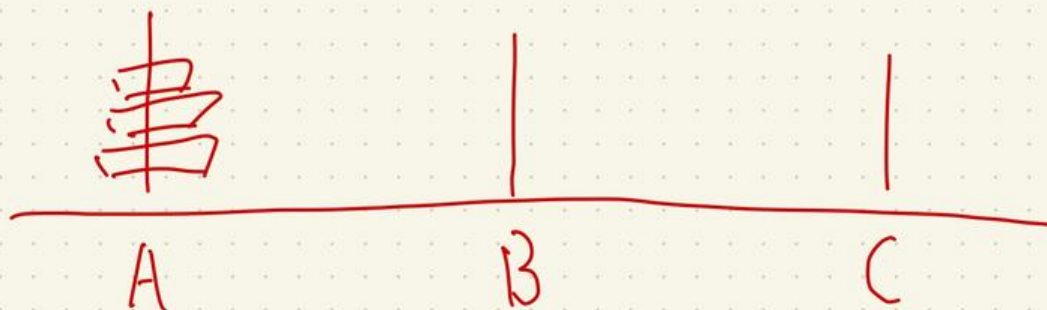
```
if n==1:  
    move(A,C)
```

 有了这个，我们就可以实现将最大的底部碟子移动到C（目标柱子）的必要动作，汉诺塔本其实就是将除了最大底盘（ n 本身）移动到C（目标柱子）的游戏，然后再将剩余的盘子全部移动到 n ，这个过程实际上就是移动 n 和移动 $n-1$ 的过程，只是 $n-1$ 后可以使用递归的方式重复移动，那么就可简单理解成， n 告诉 $n-1$ 你要开始移动盘子到B了，然后 $n-1$ 告诉 $n-1-1$ 你要开始移动开始移动盘子到B，直到 $n=1$ 时才正在开始移动盘子到B,但是对于每个 n 来说，移动盘子到那里了都是基于上个 $n+1$ 告自己的，所以会出现移动错位的情况，但最终都会移动到 b 上，当以上过程全部完成后， n 的最大值就是最底层的盘子看到所以其他的盘子都移动到了 b 时，便开始将自己移动到了C，此过程也就是函数开始调用后返回时 $A \rightarrow C$ 的情况，之后再调用 $B \rightarrow C$ 的函数即可，过程和前面一样)也就是代码所写的：

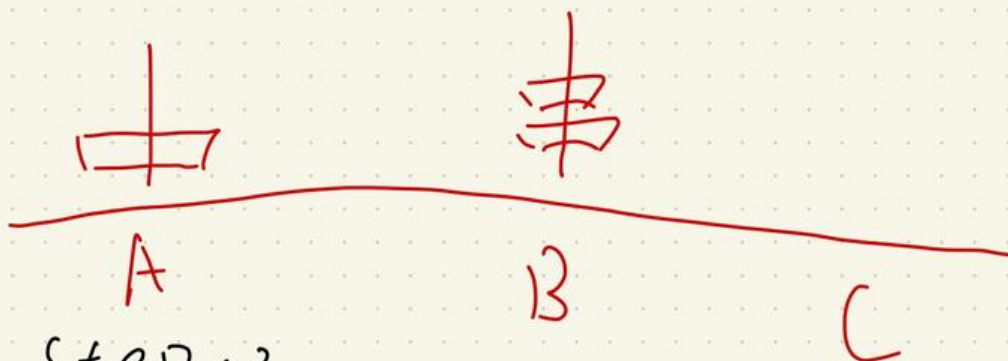
```
plat_move(n-1,a,c,b) # n-1进行A通过C移动至B 由于使用的是递归，可以理解成对整体进行移动至  
, 除n本身外  
plat_move(1,a,b,c) # 可以理解成打印整个移动过程  
plat_move(n-1,b,a,c) # 可以理解成将剩余的B由A移动至B  
# 为了方便理解 我在底下用画笔画了张图（临时画的）
```

Step: 1

$n=3$



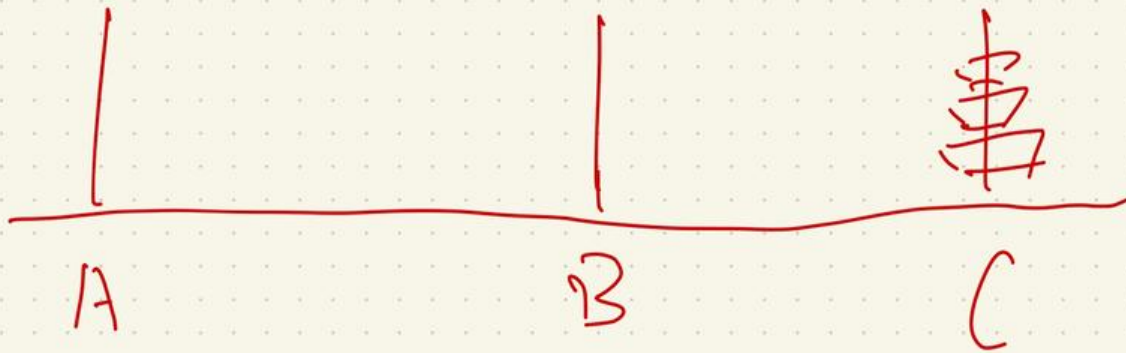
Step: 2



Step: 3



Step: 4



总结

 以后再写，对了，以上过程仅为个人参考各路大佬的答案后根据自己的理解写出来的，如果

误的话，欢迎留言告知讨论~