

在java中实现递归算法的方法 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/284/2021_2022__E5_9C_A8ja_va_E4_B8_AD_c104_284357.htm 学过程序设计的朋友都知道，存在自调用的算法称作递归算法。递归往往能给我们带来非常简洁非常直观的代码形势，从而使我们的编码大大简化，然而递归的思维确实很我们的常规思维相逆的，我们通常都是从上而下的思维问题，而递归趋势从下往上的进行思维，正由于此，很多人对于递归有着深深的恐惧，我曾经也是如此，如今为把我的经验通过几个经典的例子与初学者共享，故作此文，希望能对需要者有所助益，如若如此，便是幸甚。。。二、递归算法设计的基本思想是：对于一个复杂的问题，把问题分解为若干个相对简单前类同的子问题，继续下去直到子问题简单到能够直接求解，也就是说到了递推的出口，这样原问题就有递推得解。关键要抓住的是：（1）递归出口（2）地推逐步向出口逼近三、具体实例1。汉诺塔这是递归的超经典的例子，几乎每本程序设计书上谈到递归都会介绍。具体情景不再赘述。以我上述的方法观之：（1）递归的出口在于disk数为一的时候（2）向出口逼近：如果不是一，是n，则我们先挪动上面n-1块disk，等上面挪完，即递归返回的时候，我们挪动最底下的disk。仅仅如此，一个貌似十分复杂的问题就解决了，因为挪动那n-1块disk的时候，会继续向上减少，直到disk的数量为一为止。下面给出 java 程序编码（已测试过，运行正常）：

```
import
javax.swing.JOptionPane. public class Hanoi { private static final
String DISK_B = "diskB". private static final String DISK_C =
```

```

"diskC". private static final String DISK_A = "diskA". static String
from=DISK_A. static String to=DISK_C. static String mid=DISK_B.
public static void main(String[] args) { String
input=JOptionPane.showInputDialog("please input the number of
the disks you want me move."). int num=Integer.parseInt(input).
move(num,from,mid,to). } private static void move(int num, String
from2, String mid2, String to2) { if(num==1){
System.out.println("move disk 1 from " from2 " to " to2). } else {
move(num-1,from2,to2,mid2). System.out.println("move disk "
num " from " from2 " to " to2). move(num-1,mid2,from2,to2). } } }

```

2。这是一个排列的例子，它所做的工作是将输入的一个字符串中的所有元素进行排序并输出，例如：你给出的参数

是"abc" 则程序会输出： abc acb bac bca cab cba (1) 算法的出口在于： l o w = h i g h 也就是现在给出的排列元素只有一个时。

(2) 算法的逼近过程：先确定排列的第一位元素，也就是循环中 i 所代表的元素，然后 l o w + 1 开始减少排列元素，如此下去，直到 l o w = h i g h

```

public static
void permute(String str) { char[] strArray = str.toCharArray().
permute(strArray, 0, strArray.length - 1). } public static void
permute(char[] list, int low, int high) { int i. if (low == high) { String
cout = "". for (i = 0. i cout = list[i]. System.out.println(cout). } else {
for (i = low. i char temp = list[low]. list[low] = list[i]. list[i] = temp.
permute(list, low 1, high). temp = list[low]. list[low] = list[i]. list[i] =
temp. } } }

```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com