例子六

剖析递归行为和递归行为时间复杂度的估算

一个递归行为的例子

master公式的使用

T(N) = a*T(N/b) + O(N^d)

1) log(b,a) > d -> 复杂度为O(N^log(b,a))
2) log(b,a) = d -> 复杂度为O(N^d * logN)
3) log(b,a) < d -> 复杂度为O(N^d)

补充阅读：www.gocalf.com/blog/algorithm-complexity-and-master-theorem.html

题目三

堆排序的细节和复杂度分析

时间复杂度O(N*logN)，额外空间复杂度O(1)

堆结构非常重要

1，堆结构的heapInsert与heapify
2，堆结构的增大和减少
3，如果只是建立堆的过程，时间复杂度为O(N)
4，优先级队列结构，就是堆结构

用数组表示堆
左孩子：2*i+1
右孩子：2*i+2
父亲：（i-1）/2

插入，删除
O（logN）

题目九

介绍一下工程中的综合排序算法

若你需要排序的是基本数据类型，则选择快速排序。
若你需要排序的是引用数据类型，则选择归并排序。
(基于稳定性考虑)
你需要排序的样本数量小于60，直接选择插入排序。

虽然插入排序的时间复杂度为$O(N^2)$，我们是忽略常数项得出来的$O(N^2)$，但在数量在60以内，插入排序的时间复杂度为$O(N^2)$的劣势体现不出来，反而插入排序常数项很低，导致在小样本情况下，插入排序极快。 如果一开始数组容量很大，但可以分治处理，分治后如果数组容量$(L>R - 60)$小于60，可以直接选择插排。

题目一

实现二叉树的先序、中序、后序遍历，包括递归方式和非递归方式

非递归

先序：打印，先右后左

后序：入栈，先左后右
打印

中序：栈节点不空，
节点不空，入栈，去左孩子
空：出栈，打印，去右孩子

题目三

在二叉树中找到一个节点的后继节点

【题目】 现在有一种新的二叉树节点类型如下：

```
public class Node { public int value; public Node left;
public Node right; public Node parent;
public Node(int data) { this.value = data; }
}
```

有右子树：为右子树最左节点

无右子树：最近父节点（他是这父节点的左孩子）

题目三

认识布隆过滤器

$$M(bit) = \frac{-n\ (样本量) * \ln(p(容错率))}{(\ln 2)^2}$$

$$K = \ln 2 * m/n$$

$$P真 = (1-e^{(-n*k/m)})^k$$

认识一致性哈希