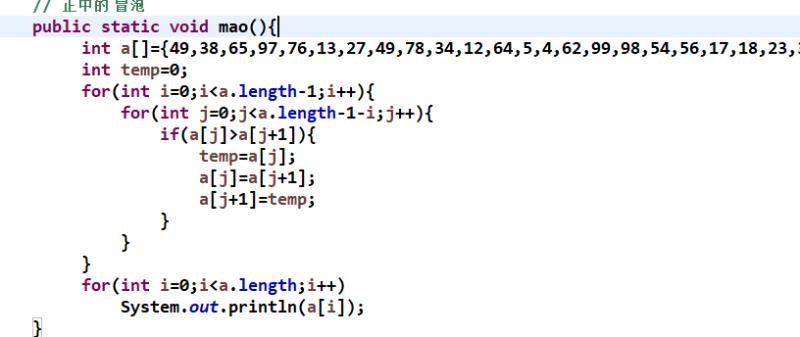


# 1.冒泡排序



# 二叉树遍历

先序遍历输出  
if left!=null treenode(left.data)  
if right!=null treenode(right.data)

]

# 大数据二叉树遍历

一棵树变多棵树，分布式遍历每棵树。

# 数据库表设计

ID，姓名，工资，上级ID，所在层次

# 5.SQL输出目录树

select \* from table order by 层次

# 6.SQL递归

以表Deparment为例

表结构为：

Id 部门内码，

DeptCode 部门编码，

ParentDeptId 上级部门内码

**使用T-SQL：**

with Dep as

(

select Id,DeptCode,DeptName from Department where Id=1

union all

select d.Id,d.DeptCode,d.DeptName from Dep

inner join Department d on dep.Id = d.ParentDeptId

)

select \* from Dep

简单解释一下：with as属于sql server新特性CTE（Common Table Expression）的关键字，用来储存临时结果集。常用于代替子查询。本例中可以理解为，找出Id=1的记录后，存放在临时表Dept中，然后临时表和Department进行内连接，找出它的子记录。子记录和第一条记录union后作为Dept新的结果集继续进行内连接，找出新的子记录。

**使用PL/SQL：**

select Id,DeptCode,DeptName from Department start with Id = 1 connect by prior Id = ParentDeptId;

start with 表示从哪一行记录开始递归查询，即根节点 connect by 表示进行递归，后面跟递归条件 prior 表示前一条记录，表示上一条记录的Id = 下一条记录的ParentDeptId 比如上面的SQL语句就可以理解为，以Id 为1的记录为根节点，递归查询下一条记录的ParentDeptId = 前一条记录的Id