

实验三 传统软件工程的软件设计

姓名： 高树林

班级： 2020185

学号： 202018526

实验学时：2（必修）

一、目的与任务

目的：

- （1）通过 Internet 搜索与浏览，了解主流的软件工具和软件开发环境产品及发展和应用情况。
- （2）分析综合问题，利用软件总体设计技术，掌握软件结构设计的图形工具；
- （3）通过软件详细设计的过程描述，对目标系统进行过程设计，掌握详细设计工具。

任务：可采用不同的软件总体设计方法，完成对综合问题的软件设计过程，包括软件总体设计及软件详细设计，并给出系统的软件结构，程序流程图、盒图等

二、实验内容

（1）软件工具学习

通过网络搜索，体会软件工具的应用状况，重点了解下面几种工具的概况：

1) Microsoft Visio

2) Processon

3) Edrawmax

注：盒图（NS）绘制时，在流程图选项，搜索“纳西-施奈德曼”

4) PADFlowChart

（2）软件总体设计

任务：某培训中心要研制一个计算机管理系统，其业务如下：将学员发来的信件收集分类后，按几种不同的情况处理。

- a. 如果是报名的，将报名数据送给负责报名事务的职员，将查阅课程文件，检查该课程是否满额，在学生文件、课程文件上登记，开出报告单交财务部门，财务人员开出发票给学生。
- b. 如果注销原来已选修的课程，则由注销人员在课程文件、学生文件和账目文件上做相应修改，并给学生注销。
- c. 如果是付款的，则由财务人员在账目文件上登记，给学生收费收据。

请对上述描述画出其数据流程图，并画出该培训管理的软件结构图的主图

（3）软件详细设计

1. 求一个数组中的最大数，数组表示为 $A(n)$, $n=1, 2, \dots, n$ 的自然数。请画出该程序流程图；该算法 NS 图。用 PAD 图表示该算法。

2.1 冒泡法排序详细设计（二选一）

任务：论述冒泡法算法思想及步骤，绘制程序流程图，绘制 PAD 分析图

2.2 插入法排序详细设计（二选一）

任务：论述插入法算法思想及步骤，绘制程序流程图，绘制 PAD 分析图

三、实验结果：

(1) 软件工具分析

(2) 软件总体设计

(3) 软件详细设计

1. 数组最大数详细设计

1) 算法思想

通过逐个比较数组中的元素来找到最大数。算法从数组的第一个元素开始，将其作为当前的最大数（maxNumber）。然后，它遍历数组中的每个元素，将当前元素与 maxNumber 进行比较。如果当前元素大于 maxNumber，则更新 maxNumber 为当前元素的值，以确保 maxNumber 始终保存着数组中的最大数。如果当前元素小于或等于 maxNumber，则继续遍历下一个元素，保持 maxNumber 不变。通过这种方式，算法会逐个比较数组中的元素，最终找到数组中的最大数。这个算法的时间复杂度是 $O(n)$ ，其中 n 是数组的长度。

2) 算法步骤

1. 初始化一个变量 maxNumber，将其设为数组的第一个元素。
2. 遍历数组中的每个元素，从第二个元素开始。
3. 对于当前遍历到的元素，将其与 maxNumber 进行比较。如果当前元素大于 maxNumber，则将 maxNumber 更新为当前元素。
4. 如果当前元素小于或等于 maxNumber，则继续遍历下一个元素。
5. 当遍历完所有元素后，maxNumber 将保存数组中的最大数。
6. 返回 maxNumber。

3) 程序流程图绘制

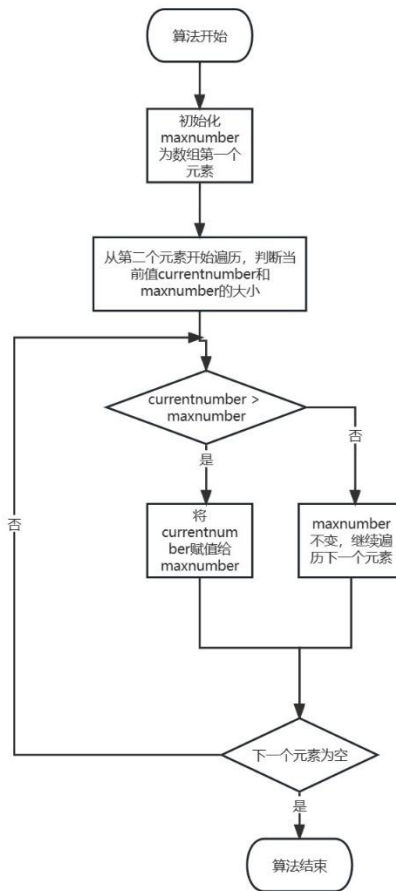


图 1 找出数组最大数的流程图

4) 盒图绘制

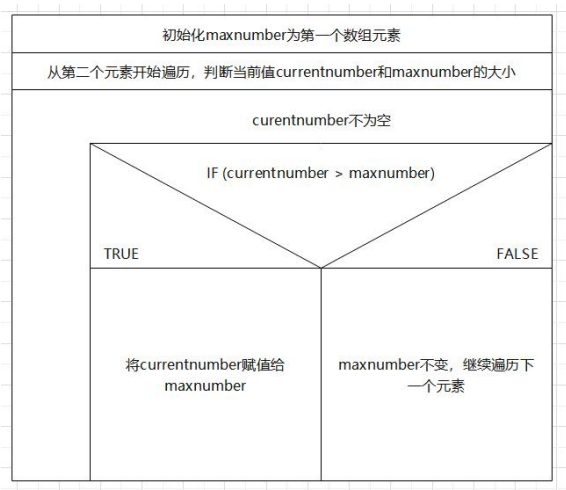


图 2 找出数组最大数的盒图

5) PAD 图绘制

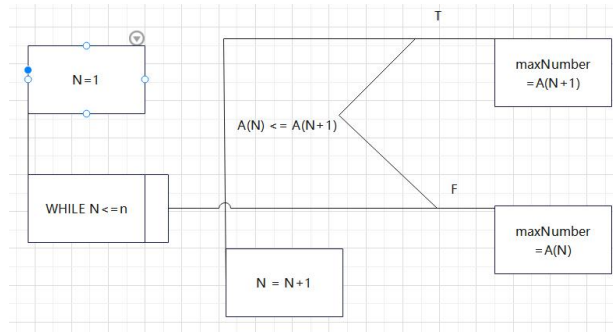


图 3 找出数组最大数的 PAD 图

2. 冒泡法排序详细设计（二选一）

1) 算法思想

通过相邻元素之间的比较和交换操作，将较大的元素逐步“冒泡”到数组的末尾。每一轮排序都会确定一个当前未排序部分的最大元素，所以需要重复执行多轮排序，直到整个数组排序完成。

2) 算法步骤

1. 从待排序的数组的第一个元素开始，假设数组长度为 n 。
2. 开始第一轮排序：
 - 2.1 依次比较相邻的两个元素，从第一个元素到倒数第二个元素。
 - 2.2 如果前一个元素大于后一个元素，则交换这两个元素的位置，将较大的元素往后移动。
 - 2.3 继续向后遍历，重复执行上述比较和交换操作，直到达到倒数第二个元素。
 - 2.4 第一轮排序结束后，最后一个元素将会是当前未排序部分的最大元素。
3. 开始第二轮排序：依次比较相邻的两个元素，从第一个元素到倒数第三个元素。重复执行上述比较和交换操作，直到达到倒数第三个元素。第二轮排序结束后，倒数第二个元素将会是当前未排序部分的第二大元素。
4. 重复执行步骤 3，每轮排序时减少比较的次数，直到执行完所有的轮数，也就是 $n-1$ 轮排序。
5. 当没有发生交换操作时，表示数组已经完全排序，排序过程结束。

3) 程序流程图绘制

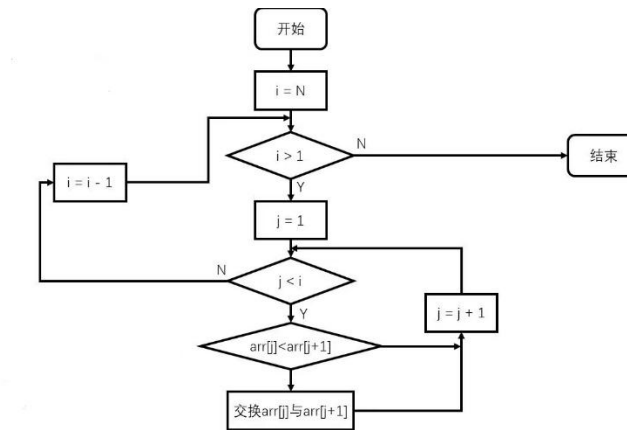


图 4 冒泡排序算法的流程图

4) PAD 图绘制

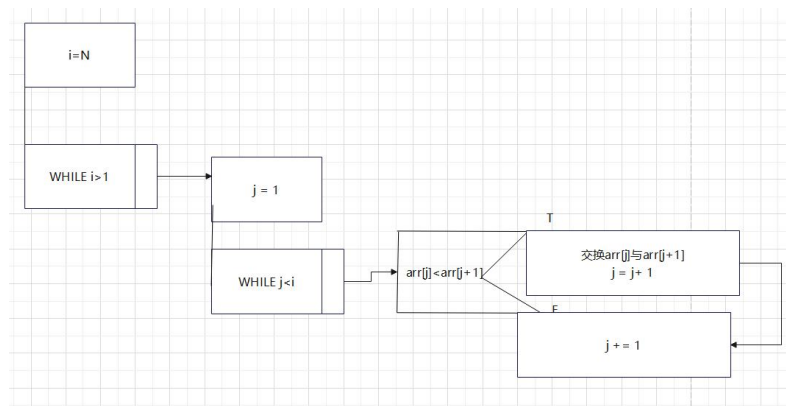


图 5 冒泡排序算法的 PAD 图

2. 插入法排序详细设计（二选一）

1) 算法思想

通过不断将未排序部分的元素插入到已排序部分的正确位置，逐步构建有序的结果。每次插入操作都会在已排序部分找到待插入元素的正确位置，并将其他元素向后移动，为待插入元素腾出位置。

具体思想为：将数组的第一个元素看作是已排序部分，剩余部分为未排序部分。从未排序部分取出第一个元素，将其插入到已排序部分的正确位置。比较已排序部分的最后一个元素与待插入元素的大小：如果已排序部分的最后一个元素大于待插入元素，则将该元素后移一位，为待插入元素腾出位置。继续向前比较，直到找到一个小于或等于待插入元素的元素，或者已经比较到已排序部分的第一个元素。将待插入元素插入到找到的位置。

重复步骤 2~4，直到未排序部分的元素全部插入到已排序部分。数组排序完成。

2) 算法步骤

具体的插入排序算法步骤如下：

1. 假设待排序的序列为 arr ，长度为 n 。
2. 从下标 1 开始遍历数组，直到下标 $n-1$ 。
3. 在每次遍历时，将当前元素 $arr[i]$ 插入到已排序序列中的正确位置。
4. 为了找到正确的插入位置，从当前元素往前比较，将比 $arr[i]$ 大的元素后移一位，直到找到小于等于 $arr[i]$ 的位置。
5. 将 $arr[i]$ 插入到找到的位置。
6. 重复步骤 2-5，直到遍历完所有元素。

3) 程序流程图绘制

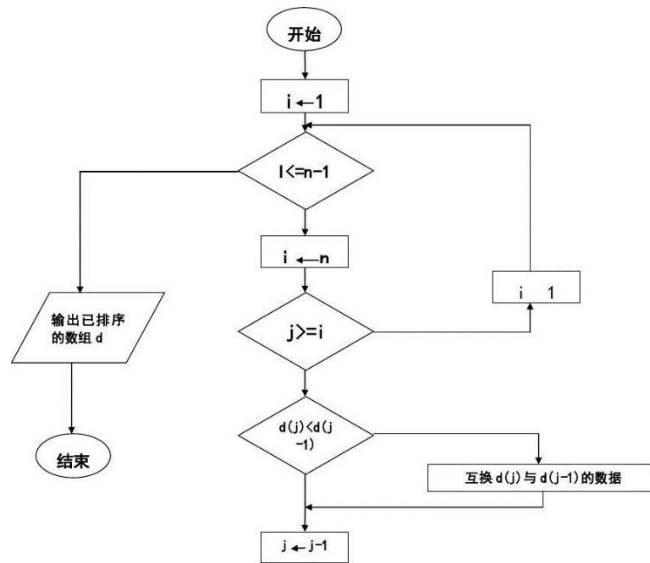


图 6 插入排序算法的流程图

4) PAD 图绘制

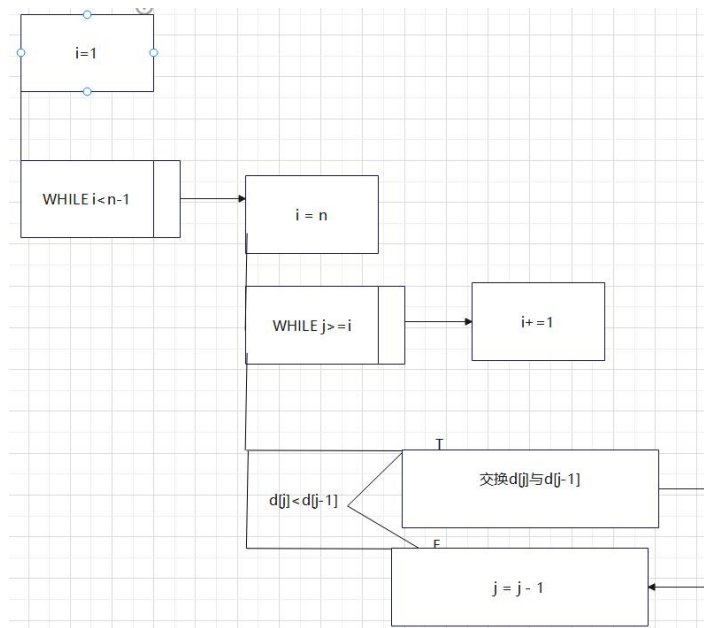


图 7 插入排序算法的 PAD 图