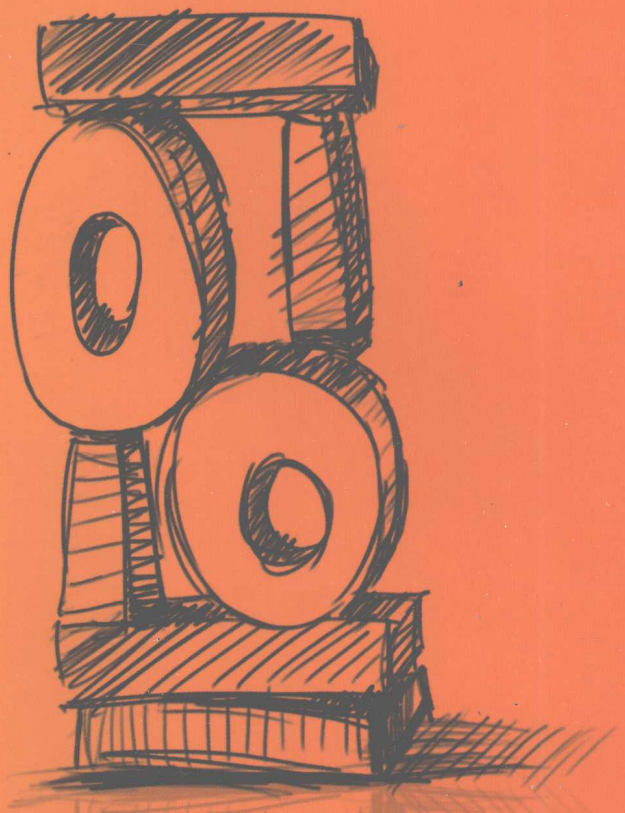
嵌入式系统软件设计基础丛书



嵌入式系统软件设计中的数据结构

陆　玲　周航慈　编著

北京航空航天大学出版社

嵌入式系统软件设计基础丛书

嵌入式系统软件设计中的数据结构

陆　玲　周航慈　编著

北京航空航天大学出版社

内 容 简 介

根据嵌入式系统软件设计需要的“数据结构”知识编写而成。书中基本内容有：常用线性数据结构在嵌入式系统中的实现和相关算法；树和图在嵌入式系统中的实现和相关算法；排序和查找算法等。

本书从嵌入式系统的实际硬件环境出发，用通俗易懂的语言代替枯燥难懂的理论解释，结合嵌入式系统的应用实例，使读者在比较轻松的条件下将“数据结构”的基本知识学到手。

本书可作为从事嵌入式系统软件设计的电子技术人员自学“数据结构”的教材，也可供高等院校电子技术类专业本科生、研究生作为教学参考书。

图书在版编目(CIP)数据

嵌入式系统软件设计中的数据结构/陆玲，周航慈编著. —北京：北京航空航天大学出版社，2008.8

ISBN 978-7-81124-356-7

Ⅰ.嵌… Ⅱ.①陆…②周… Ⅲ.微处理器—系统开发—数据结构 Ⅳ.TP332 TP311.12

中国版本图书馆 CIP 数据核字(2008)第 096720 号

嵌入式系统软件设计中的数据结构

陆 玲 周航慈 编著

责任编辑 李宗华 李开先

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话：010-82317024 传真：010-82328026

http://www.buaapress.com.cn E-mail:bhpress@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本：787 mm×960 mm 1/16 印张：11.5 字数：258 千字

2008 年 8 月第 1 版 2008 年 8 月第 1 次印刷 印数：5 000 册

ISBN 978-7-81124-356-7 定价：22.00 元

前 言

嵌入式系统在各行各业的应用越来越广，我国从事嵌入式系统开发的人员也越来越多，从我国主要的几种电子杂志上可以看出，有关嵌入式系统应用的文章也越来越多。

在开发一种嵌入式系统产品时，主要工作是做两方面的设计：硬件设计和软件设计。在硬件设计方面，各个半导体公司竞相推出各种高性能、低功耗、低成本的 CPU 和外围芯片，这使我们在进行硬件设计时可以很快地得到最先进的芯片。在这种情况下，硬件设计的外部条件越来越好，集成度越来越高，在实现相同功能的情况下线路越来越简化。在软件设计方面，虽然开发工具和程序设计语言也在不断提高，但技术人员本身的软件素质无疑起决定作用。因此，软件设计水平在嵌入式系统产品开发的过程中占有重要的地位，直接影响到产品的功能和竞争能力。我国目前绝大多数从事嵌入式系统开发的技术人员基本上不是计算机专业毕业的，有的可能没有上过大学，他们未接受过系统的软件基础理论教育，软件设计水平仍不太高。在软件开发过程中，他们只是不自觉地采用了一些规律性的设计方法，或者模仿别人的程序设计方法，而有更多成熟的基本方法没有被掌握，开发出来的软件水平不高，致使产品的功能和可靠性受到一定的制约。

软件设计是一门科学，有其自身的规律，也有很多成熟的理论和算法。要学习就要选教材，而目前能选到的软件方面的教材都是专为计算机专业编写的教材。这些教材起点较高，偏重理论叙述，未考虑嵌入式系统的硬件特点，对于广大嵌入式系统开发人员来说不是十分适合，学起来会感到比较抽象和吃力。

出于提高我国广大嵌入式系统开发人员软件素质的愿望，我们决定编写一本适合自学软件理论基础“数据结构”的书。该书起点要求不高，只要掌握 C 语言并已从事了一段时间嵌入式系统开发工作的人员就可以看懂。学完本书后，对软件设计的主要基础理论“数据结构”就能初步掌握，在进行软件设计时，可以减少很多盲目性，并为更系统、更深入地学习其他计算机软件设计理论打下良好基础。

本书第 1 章介绍数据结构的基本概念；第 2 章介绍线性表的知识和应用；第 3 章介绍队列的知识和应用；第 4 章介绍堆栈的知识和应用；第 5 章介绍串的知识和应用；第 6 章介绍数组的知识和应用；第 7 章介绍树的知识和应用；第 8 章介绍图的知识和应用；第 9 章介绍排序的知识和应用；第 10 章介绍查找的知识和应用。在编写过程中，我们的原则是：尽量结合嵌入式

系统的应用实例，采用通俗易懂的叙述方式，介绍最基本的核心内容。

陆玲编写了本书第1～8章的内容，周航慈编写第9章、第10章和前8章的应用实例。周航慈负责全书的策划、内容安排、文稿修改和审定。

在本书的编写过程中，得到北京航空航天大学出版社的大力支持，何立民教授给予了无私的帮助，在此表示衷心感谢！周立功先生在本书的策划过程中起了很大促进作用，在此也表示衷心感谢！

由于水平有限，书中错误及不足之处敬请广大读者予以指正，不胜感谢！

作　者

于东华理工大学

2008年3月

目 录

第1章 概 述

数据结构是一门研究“非数值计算”的程序设计的学科，它主要研究计算机操作对象和它们之间的关系以及操作方法等问题。它讨论的是那些不能用通常的数学分析来解决的、而且也无法用数学公式或方程来描述的问题。例如：图书馆的资料自动检索问题、学生各门课程成绩的排序问题、人机下棋游戏设计问题、两个城市之间多条交通道路选择问题等，这些无法用数学公式解决的实际问题都可使用“数据结构”知识来解决。

1.1 数据结构的基本概念

本节对数据结构所涉及的一些概念和术语进行简单的介绍。

1.1.1 数据和信息

数据是指能被输入到计算机中并被计算机处理的数值、字符等符号的总称，它是计算机程序加工的“原料”。例如用计算机求解一个线性方程组 $AX=B$ 时，必须输入矩阵 A 的值和向量 B 的值，这里的 A 与 B 都是输入数据，这些数据都能被输入到计算机中并被计算机所处理。对于我们常使用的文本编辑程序，例如 Windows 系统中的记事本，用户可从键盘中输入字母、数字、特殊符号等字符串，并且可以保存这些字符串，这些字符串也是数据。对于图像和声音等都可以将其编码转换成计算机可操作的数据。

信息指的是数据所表示的含义。同一数据在不同的程序中可表示不同的信息，例如：数字数据 85，在学生成绩管理程序中可表示学生的分数；在公交管理程序中可表示路程；在图像处理程序中可表示灰度值。而不同形式的数据可以传达同样的信息，例如：英文字符“A”、中文字符“优”、数字 5(5 分制)都表示学生的成绩为优秀。所以数据是信息的某种特定的符号表示形式，数据所能描述的信息是非常丰富的。

1.1.2 数据元素

数据元素是数据的基本单位。既然是单位，就可大可小，所以数据元素可以是不可分割的“原子”。例如一个整数“8”或一个字符“A”；也可以由若干个数据项组成，例如在学籍管理程

序中一个学生的基本情况也是一个数据元素，它包括学号、姓名、性别和出生日期等数据项。当然学号、姓名、性别、出生日期也可以分别是数据元素，因此是否是数据元素取决于是否在程序设计中作为一个整体进行处理。

1.1.3 数据对象

数据对象是具有相同性质的数据元素的集合，是数据的一个子集。在高级程序设计语言中，同一类型的数据就是同一种数据对象。因为计算机不可能同时处理一切类型的数据，总是对特定的问题处理一种或几种对象。例如用计算机求素数问题只涉及“整数”这种数据对象。

1.1.4 数据结构

数据结构是彼此具有一定关系的数据元素的集合。事实上，这些关系反映了客观世界事物之间的联系。由于客观事物存在着各种不同的联系形式，所以反映在数据关系上也就各不相同。简单地说，数据结构中的“数据”是指特性相同的数据元素的集合，“结构”就是指的数据元素之间存在的一种或多种特定关系。一般来说，数据结构包括以下 3 个方面的内容：

① 数据元素之间的逻辑关系，也称为数据的逻辑结构。逻辑结构是对数据元素之间的逻辑关系描述，它可以用一个数据元素的集合和定义在此集合上的若干关系来表示。例如：在一个家庭中，每个家庭成员（可用名字表示）就是一个数据元素，这些数据元素的集合就是一个家庭，父亲、母亲、儿子、丈夫、妻子等就是家庭中的若干个“逻辑”关系。

② 数据元素及其关系在计算机内存中的表示，也称为数据的物理结构或数据的存储结构。它包括数据元素的表示和数据元素之间关系的表示。同样以家庭为例，每个家庭都住着一套房间，每个成员（数据元素）都有自己的床位或房间，这就相当于数据元素“存储在计算机中”的物理位置。当然，每个成员的床位或房间都有独特的“物质”标志，可以反映成员在家庭中的关系。

③ 数据的运算及实现，即对数据元素可以施加的操作以及这些操作在相应的存储结构上的实现方法。还是以家庭为例，每个家庭在不同的时间家庭成员会发生变化，如出生与死亡会增加成员与减少成员，还会更换家庭成员的床位或房间。

1.2 逻辑结构

按照数据元素之间存在的逻辑关系的不同数学特性，通常有下列 4 类基本逻辑结构。

1.2.1 线性结构

在这种结构中数据元素之间存在着“一对一”的线性关系，也即数据元素存在着依次排列的先后次序关系，且只有一个起始数据元素和一个终止数据元素，如图 1-1(a)所示。

(a) 线性结构

17路：汽车站 中山路 迎宾广场 紫金路 服装厂 中心商场 滨江大道 南开发区 火车站

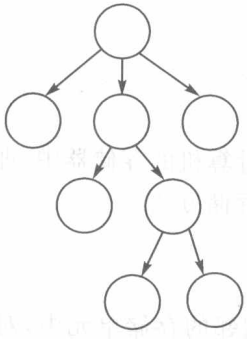
(b) 某公交线路

图 1-1 线性结构示意图

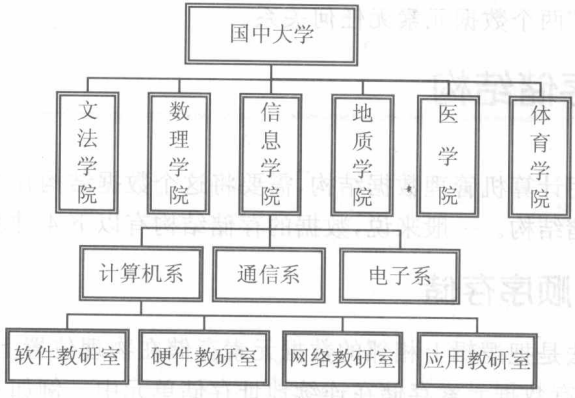
类似这种结构的如城市某路公交线路，数据元素为站名，每一路公交线路都有一个起点和终点，中间各站都按先后次序排列，如图 1-1(b)所示。

1.2.2 树形结构

在这种结构中，数据元素之间存在着“一对多”的树形关系，也即数据元素之间存在着层次关系或分支关系。这种结构只有一个起始数据元素(称为树根)，其他数据元素称为树叶，如图 1-2(a)所示。例如大学学校行政组织机构可用树形结构表示，树根的数据元素为学校，学校的下一层(第 1 层)对应各学院，每个学院下一层(第 2 层)对应各系，每个系下一层(第 3 层)对应各教研室等。所以可将大学行政机构抽象成为树形结构来处理，如图 1-2(b)所示。



(a) 树形结构

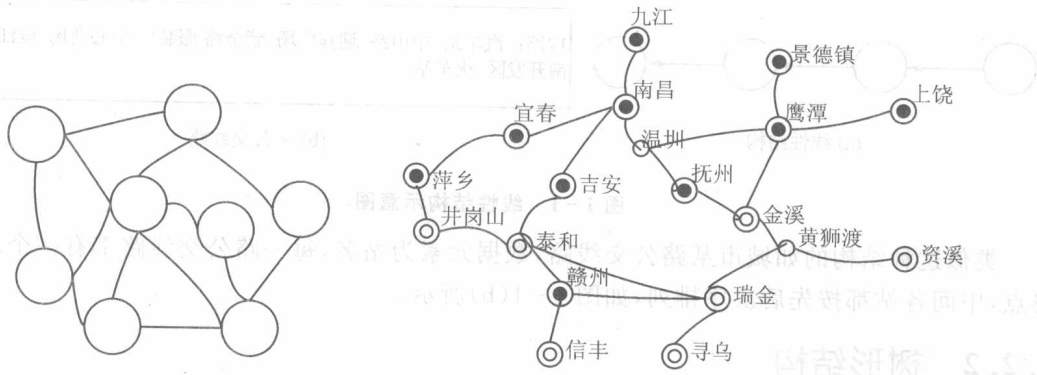


(b) 某大学部分行政机构示意图

图 1-2 树形结构示意图

1.2.3 图状或网状结构

在这种结构中，数据元素之间存在着“多对多”的网络关系，也即数据元素之间相互连接成网状。如图 1-3(a)所示。例如公路交通图可采用网状结构表示：每个地点为数据元素；公路是地点之间的关系；一个地点与多个其他地点之间有公路；通往一个地点的公路有多条；地点之间相互通过公路连接成网状。如图 1-3(b)所示。



(a) 网状结构　　(b) 江西省部分国道示意图

图 1-3 网状结构示意图

1.2.4 纯集合结构

这个结构中数据元素之间除了“同属一个集合”之外，别无其他关系，例如性别集合中的“男”与“女”两个数据元素无任何关系。

1.3 存储结构

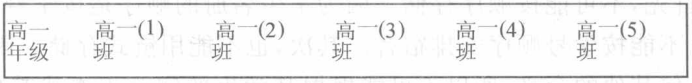
为了用计算机管理数据结构，需要将这个数据结构存入到计算机的存储器中，即需要考虑数据的存储结构。一般来说，数据的存储结构有以下 4 种基本存储方法。

1.3.1 顺序存储

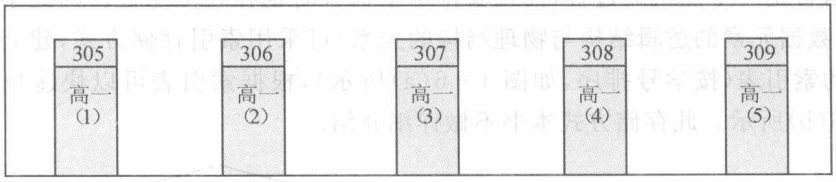
该方法是把逻辑上相邻的数据元素存储在物理位置上也相邻的存储单元中，对于线性结构来说，所有数据元素存储在连续地址存储单元中。例如对某中学的高一年级而言，共有 5 个班，每个班是一个数据元素，其逻辑顺序是高一(1)班、高一(2)班、高一(3)班、高一(4)班、高一(5)班，如图 1-4(a)所示。如果是顺序存储，这些班的教室(物理地址)也是按班的顺序在同一层排列，且班与班之间是隔壁房间，如图 1-4(b)所示。在计算机高级语言中通常用数组来表示顺序存储。

1.3.2 链状存储

该方法不要求逻辑上相邻的结点在物理位置上也相邻，也就是说数据元素可以存储在任意位置，但是需要用一个“指针”来联系不同存储位置而逻辑关系相邻的数据元素。例如：某旅游景区的所有景点(数据元素)的最佳参观线路是逻辑顺序，如图 1-5(a)所示。但所有景点



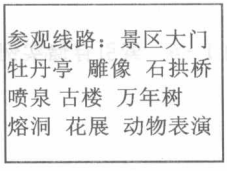
(a) 高一年级5个班逻辑结构



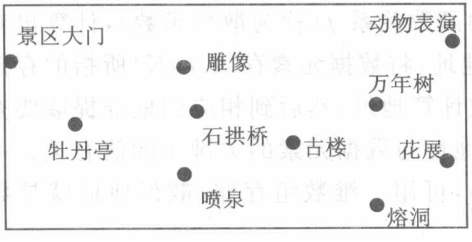
(b) 高一年级5个班物理结构

图 1-4 顺序存储实例图

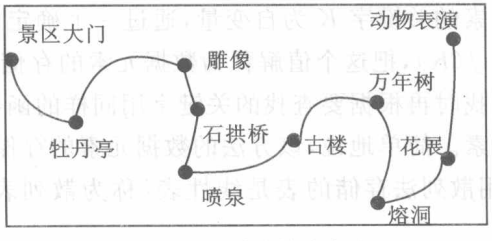
的地理位置并不是按顺序排列，而是随机分布在景区的不同位置上，如图 1-5(b)所示。游客从第一个景点开始参观，进入下一个景点时，都有路标指示牌指示下一个景点的方向与位置，所以游客通过当前景点都可以快速找到下一个景点，直到最后一个景点参观完毕，如图 1-5(c)所示，所有景点通过路标指示牌链成一个线路。路标指示牌相当于“指针”，用它来联系不同位置而逻辑关系(最佳路线)相邻的景点。在计算机程序设计语言中通常用指针类型来描述“指针”。



(a) 参观线路逻辑结构



(b) 景点分布图



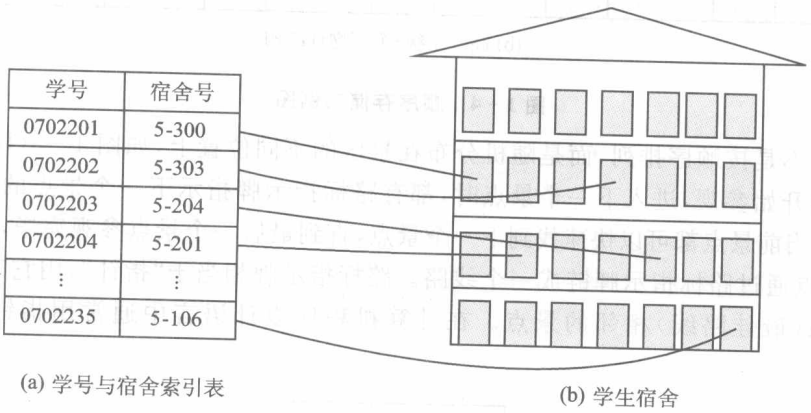
(c) 参观线路链式表示

图 1-5 链式存储实例图

1.3.3 索引存储

该方法通常是在存储数据元素信息的同时，建立附加的索引表，索引表中一般包括关键字和地址等信息，其中关键字是指可唯一标识一个数据元素的数据项。例如大学里学生名册与

学生宿舍的关系。首先，不可能按顺序存储。因为学生名册的顺序是按学号顺序排列，且不考虑男生与女生，所以不能按学号顺序安排宿舍。其次，也不能用链式存储。因为学号只是不同学生的标志，没有任何其他的含义，所以不可能根据某学生的宿舍来查找其他学生的宿舍，一般查找学生宿舍是一个随机的过程，而且安排学生宿舍时也存在随意性。因此学生与宿舍的关系（相当于数据元素的逻辑结构与物理结构的关系）可采用索引存储方式，建立一个学号与宿舍号关系的索引表（按学号排序，如图 1-6(a)所示），根据索引表可以快速找到学生的宿舍，如图 1-6(b)所示。此存储方式本书不做详细介绍。



(a) 学号与宿舍索引表　　(b) 学生宿舍

图 1-6　索引存储实例图

1.3.4　散列存储

该方法是依据数据元素的关键字直接计算出该数据元素的存储位置。其基本思想是：以数据元素的关键字 K 为自变量，通过一个确定的函数关系 f（称为散列函数），计算出对应的函数值 $f(K)$，把这个值解释为数据元素的存储地址，将数据元素存入 $f(K)$所指的存储位置上。查找时再根据要查找的关键字用同样的函数计算地址，然后到相应的地址提取要查找的数据元素。简单地说，该方法的数据元素的存储地址与数据元素的关键字的值有关。一般情况下，用散列法存储的表是线性表（称为散列表），可用一维数组存储，散列地址就是数组的下标。

例如：已知 60 个数据元素的线性表，其关键字 K 都是小于 100 的正整数，则可定义散列函数为 $f(K)=K$。散列表的定义为 datatype $H[100]$，$H[i]$存放关键字为 i 的数据元素。此存储方式本书不做详细介绍。

一般来说，以上 4 种基本的存储方法，既可单独使用，也可以组合起来对数据结构进行存储。同一种逻辑结构采用不同的存储方法，可得到不同的存储结构。选择何种存储结构来存储表示相应的逻辑结构，要视具体问题的要求而定，也可依据运算是否方便和算法的时间效率

和空间要求而定。

1.4 算 法

用计算机完成解题任务，除了要选取适当的数据结构外，还需要制定出解决问题的方法和步骤，这就是所谓的计算机算法。算法是对问题求解过程的一种描述，是为解决一个或一类问题给出的一个确定的、有限长的操作序列。

1.4.1 算法的描述

算法可以用自然语言、数学语言或约定的符号语言来描述，也可用计算机高级程序语言描述。例如对于“从 3 个整数中选出最大的一个整数并输出”的问题，可采用以下 3 种方法：

1. 自然语言描述算法

第①步：. 输入 3 个整数 x，y，z；

第②步：先从 2 个数 x、y 中选出最大的一个，设为 m；

第③步：再从 m 与 z 选出最大的一个，设为 n；

第④步：输出 n。

用自然语言描述算法比较容易看懂，对读者要求不高。但算法过程的流向不清晰，特别是算法中含有多处转向时，容易使人弄不明白。

2. 符号语言描述算法

符号语言有许多类型，这里仅介绍用程序流程图描述算法，如图 1-7 所示。

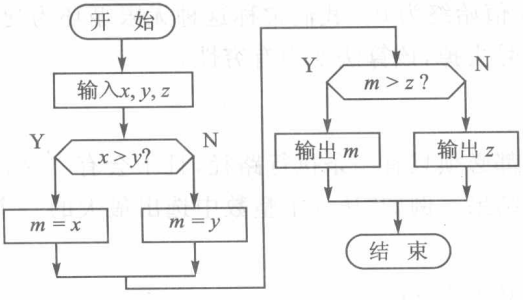


图 1-7 算法流程图

程序流程图描述的算法比较清晰，而且容易转换成计算机语言。

3. 高级语言描述算法

目前有多种程序设计语言，本文仅介绍用 C 语言的描述。

```
scanf("%d%d%d",&x,&y,&z);
if (x > y)  m = x;
else   m = y;
if (m > z)   printf("%d",m);
else  printf("%d",z);
```

用高级语言描述的算法对读者要求较高，必须具备高级语言的知识。但这种算法直接可输入到计算机中进行调试，容易验证其正确性。本书中的数据结构算法大部分采用C语言描述。

1.4.2 算法的特征

一个正确的算法必须满足以下5个重要特征：

1. 有穷性

对于任意一组合法的输入值，在执行有穷步骤之后一定能结束，即算法中的操作步骤为有限个，且每个步骤都能在有限时间内完成。例如如下算法：

```
    scanf("%d",&n);
    s = 0;i = 0;
    while(i <= n);
    {s = s + i; i ++ ;}
printf("%d",s);
```

上述算法应该是求和：1+2+3+…+n。但算法中有一个错误，在while语句后面加了一个分号，使得while(i<=n)语句成为一个独立语句。当输入的n大于0时，while(i<=n)语句无穷地循环执行，因为i值始终为0。我们常称这种无限循环为死循环，程序不能在有限的时间内完成。将多余的分号去掉，该算法变为有穷性。

2. 确定性

对于每种情况下算法都必须只有一条执行路径，且不会有二义性。对相同的输入在任何时候执行都应得出相同的结果。例如“从3个整数中选出最大的一个整数或最小的一个整数并输出”的算法如下：

第①步：输入3个整数x，y，z；

第②步：先从两个数x、y中选出最大的或最小的一个，设为m；

第③步：再从m与z中选出最大的或最小一个，设为n；

第④步：输出n。

上述算法的第2步与第3步是不确定的，对于相同的输入会有不同的结果。因此算法应改为：