

青少年信息学奥林匹克

初级竞赛辅导 (第二版)

方文祺 主编 黄福铭 主审



南开大学出版社

青少年信息学奥林匹克
初级竞赛辅导
(第二版)

方文祺 主编

黄福铭 主审

南开大学出版社

天津

图书在版编目(CIP)数据

青少年信息学奥林匹克初级竞赛辅导 / 方文祺主编. —2
版. —天津:南开大学出版社, 2007. 5

ISBN 978-7-310-02702-6

I. 青... II. 方... III. 计算机课—中小学—教学参考
资料 IV. G634. 673

中国版本图书馆 CIP 数据核字(2007)第 053817 号

版权所有 侵权必究

南开大学出版社出版发行

出版人: 肖占鹏

地址: 天津市南开区卫津路 94 号 邮政编码: 300071

营销部电话: (022)23508339 23500755

营销部传真: (022)23508542 邮购部电话: (022)23502200

*

天津泰宇印务有限公司印刷

全国各地新华书店经销

*

2007 年 5 月第 2 版 2007 年 5 月第 2 次印刷

787×1092 毫米 16 开本 16.25 印张 408 千字

定价: 23.00 元

如遇图书印装质量问题, 请与本社营销部联系调换, 电话: (022)23507125

内 容 简 介

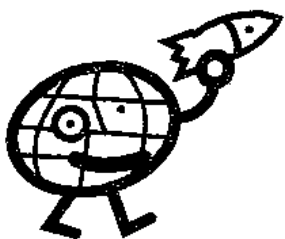
由特级教师方文祺老师主编的《青少年信息学奥林匹克》丛书，经过多年的使用，在吸收了老师们近几年来在竞赛辅导经验的基础上，进行了全面修订。

丛书修订后仍然分《青少年信息学奥林匹克——Quick BASIC 编程基础》与《青少年信息学奥林匹克——初级竞赛辅导》两册出版。

《青少年信息学奥林匹克——Quick BASIC 编程基础》修订版删除了有关计算机基础知识部分的内容，增加了有关 Quick BASIC 集成环境操作以及数据文件的内容。

《青少年信息学奥林匹克——初级竞赛辅导》修订版则更加突出了算法思想，增加了常用算法、程序优化以及程序调试与测试等内容，其中算法部分涵盖了中学数学课程中算法与程序设计课程的相关内容。

本丛书可以作为小学高年级学生以及初中学生参加青少年信息学奥林匹克竞赛活动（普及组）的培训与辅导用书，同时也可以作为中学生学习数学与信息技术课的参考用书。



再版前言

近年来，随着新课程改革的全面展开，信息技术成为中学必修课程之一，特别是基础课程数学已经把算法列为必修内容，算法思想将贯穿高中数学课程的相关部分。算法是计算科学的重要基础。实践证明，以算法与编程为核心的青少年信息学奥林匹克竞赛有利于培养中小学生的逻辑思维能力和信息能力，有利于促进学生智力的全面发展。

为了适应新时期我国青少年信息学奥林匹克竞赛活动的要求，提供更加适宜的竞赛辅导参考用书，编者吸收了近年来广大读者特别是一线辅导老师们提出的宝贵意见，适当提高难度和增加题量，对《青少年信息学奥林匹克——Quick BASIC 编程基础》与《青少年信息学奥林匹克——初级竞赛辅导》进行了全面修订。

《青少年信息学奥林匹克——Quick BASIC 编程基础》修订版去掉了关于计算机基础知识的部分内容，增加了 Quick BASIC 集成环境操作以及数据文件的相关内容。

《青少年信息学奥林匹克——初级竞赛辅导》修订版突出了算法思想，增加了常用算法、程序优化、程序调试与测试等内容，其中算法部分涵盖了中学数学课程中关于算法与程序设计方面的内容。

带有“*”的章节内容可作为选学内容。

本书除了作为小学高年级学生以及初中学生参加青少年信息学奥林匹克竞赛活动（普及组）的培训与辅导用书之外，也可以作为中学生数学“算法思想”或信息技术“算法与程序设计”等课程的参考用书。

本书由天津市青少年科技中心黄福铭老师主审。参与本书部分章节编写与修订的还有吕从中、何春英、崔玉忠等老师。读者在使用本书的过程中若发现错误之处，请发送电子邮件至 fangwq@tjty.net。本书再版过程中得到了南开大学出版社肖占鹏社长、张蓓副社长的全力支持，尹建国老师和李正明老师为本书付出了辛勤的劳动，这里一并致以真挚的感谢。

方文祺

2007.2



前言

伴随着计算机的普及,青少年信息学奥林匹克竞赛已经蓬勃开展了二十多年。青少年信息学奥林匹克竞赛活动是推动信息技术教育发展和深入的重要手段,他为中小學生提供了充分展示其个性、特长与能力的机会。实践证明,青少年信息学奥林匹克竞赛有利于培养中小学生的逻辑思维能力与信息处理能力,有利于促进学生智力的全面发展。

江泽民主席在致第 15 届国际信息学奥林匹克竞赛的贺信中指出:“在人类即将进入新世纪之际,以信息科技和生命科技为核心的科技进步与创新,正在深刻地改变着人类的生产方式和生活方式,推动着世界文明的发展。青年是人类的未来,也是世界科技发展的未来。国际信息学奥林匹克竞赛活动对年轻一代了解和掌握现代科学技术,养成创新精神,具有重要作用。”

国际信息学奥林匹克中国队总教练、清华大学博士生导师吴文虎教授在为笔者编著的《青少年程序设计的数学基础》(北京大学出版社,1997)一书所写的序言中指出:“从求解历年信息学奥林匹克国内外试题的情况看,都是一个将具体问题抽象为数学模型的思维过程。有了数学模型才有用计算机来求解的基础。这种将具体问题抽象为数学模型的建模过程,是一种创造性的思维过程,除了可以训练想象力、判断力、洞察力和逻辑思维能力之外,对于激发创造意识与欲望,培养创造能力也是大有裨益的”。

BASIC 语言是青少年学习编程的启蒙语言,伴随着计算机技术的飞速发展,从 DOS 环境到 Windows 环境,从基本 BASIC 到可视化编程环境 Visual BASIC, BASIC 语言是编程语言中发展最快的一种高级语言。Quick BASIC 就是一个 DOS 环境下的 BASIC 集成环境,它所支持的 Quick BASIC 语言既具有传统 BASIC 语言简单易学的特点,又具有现代结构化程序设计语言的特点。青少年在使用 Quick BASIC 语言编程的基础上,很容易掌握其他高级语言。由于 Quick BASIC 语言一直是我国青少年信息学奥林匹克竞赛规定采用的结构化的编程语言之一,所以,本书通过 Quick BASIC 语言来介绍程序设计的基本知识、基本思想和基本方法。

为了推动我国青少年信息学奥林匹克竞赛活动的进一步开展,为积极参与这项活动的学生提供更加适宜的竞赛辅导参考用书,编者在《少年信息学(计算机)奥林匹克》(南开大学出版社,1996)一书的基础上,

吸收了近几年来辅导学生参加全国青少年信息学奥林匹克分区联赛的经验，重新编写了这套竞赛辅导参考书。

本套书分两册出版，《青少年信息学奥林匹克——Quick BASIC 编程基础》一书，从电脑的基础知识讲起，由浅入深地介绍了 Quick BASIC 集成环境的基本操作和 Quick BASIC 语言的基本内容，并通过对大量精选例题的剖析，介绍结构化 BASIC 语言的编程方法；《青少年信息学奥林匹克——初级竞赛辅导》一书则围绕竞赛大纲，对大量精选的各种类型试题和例题进行分析，在加强基础训练的基础上，帮助学生掌握常用算法，提高运用计算机解决问题的能力，为更高层次的学习打下基础。

本书可以作为小学高年级学生以及初中学生参加青少年信息学奥林匹克竞赛活动的培训与辅导用书。

本书由方文祺老师主编、天津市青少年科技中心黄福铭老师主审。参与本书部分章节编写的还有吕从中、何春英、崔玉忠等老师。本书在出版过程中得到南开大学出版社张蓓老师、尹建国老师的全力帮助，责任编辑老师付出了辛勤的劳动，这里一并致以真挚的感谢。

方文祺

2003.2

目 录

再版前言

前言

1 第一章 基本算法举例

- 1 § 1.1 程序设计基础
- 9 § 1.2 解析法
- 13 § 1.3 穷举法
- 19 § 1.4 构造法
- 28 § 1.5 模拟与随机模拟
- 35 § 1.6 递推与迭代
- 43 § 1.7 递归
- 51 § 1.8 排序与查找

59 第二章 基础知识类型题例解

- 59 § 2.1 计算机基本知识
- 60 § 2.2 计算机中数的表示
- 62 § 2.3 汉字与编码基础知识
- 63 § 2.4 计算机软件知识
- 64 § 2.5 操作系统基本知识
- 65 * § 2.6 DOS 基本操作
- 67 § 2.7 网络基础知识
- 68 § 2.8 信息安全基础知识
- 69 § 2.9 程序设计基础知识
- 72 习 题

76	第三章 分析程序类型题例解
76	§ 3.1 循环与分支结构程序的分析
81	§ 3.2 使用数组程序的分析
84	§ 3.3 调用过程程序的分析
89	§ 3.4 完善程序类型题的分析
100	习 题
105	第四章 编程初级类型题例解
105	§ 4.1 有趣的自然数
110	§ 4.2 字符图形
114	§ 4.3 字符串处理
122	§ 4.4 数制及其转换
130	§ 4.5 高精度计算
133	§ 4.6 组合数学题例
139	§ 4.7 其他
145	习 题
149	*第五章 搜索算法初步
149	§ 5.1 回溯
154	§ 5.2 程序的优化
160	习 题
162	*第六章 程序的调试与测试
162	§ 6.1 程序的静态调试
165	§ 6.2 程序的动态调试
169	§ 6.3 程序动态调试实例
178	§ 6.4 程序测试
184	§ 6.5 生成可执行文件

186	附录一 练习题与习题参考答案
226	附录二 全国青少年信息学奥林匹克联赛 天津市（小学组）试题选
249	参考书目



第一章 基本算法举例

§1.1 程序设计基础

(一) 程序设计步骤

1. 分析问题

程序设计，首先应分析问题的要求，先解决需要“做什么”的问题。对要解决的问题，要明确题目的要求，列出所有已知量，找出题目的求解范围、解的精度等。

2. 建立数学模型

对实际问题进行分析之后，找出内在规律，将复杂问题简化、抽象，建立数学模型。只有建立了模型的问题，才有可能利用计算机来解决。

3. 设计算法

设计算法就是描述“如何做”，如何一步一步地解决问题。对程序设计而言，算法是对解题过程的精确而完整的描述，它可以是数学公式，也可以是规则。一个算法可以采用多种方式描述，并需要逐步求精，不断地把这些算法具体化。

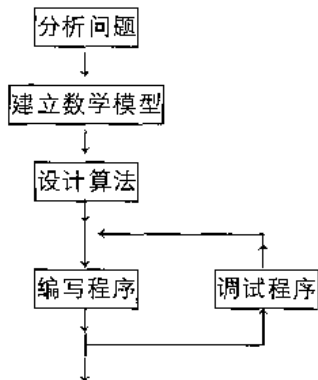
4. 编写程序

算法研究问题解决的方法，我们感兴趣的还是用计算机语言来描述算法，直到最终能把全部算法表示成可以由计算机实现的操作命令，即程序。

5. 调试程序

编写结束的程序必须能在计算机上执行，并能正确无误，这就需要经多次调试、修改，直到满意为止。

程序设计流程图如下：



（二）结构化设计方法

结构化程序设计方法是公认的面向过程编程应遵循的基本方法和原则。

1. 结构化程序设计的特征

① 以顺序、选择和循环三种基本结构的组合来描述程序。每个结构只有一个入口、一个出口，不允许从一个结构跳到另一个结构，各单元之间接口简单、逻辑清晰。

② 采用模块化结构。把一个较大的程序划分为若干子程序，每一个子程序成为一个模块，每一个模块又可继续划分为更小的子模块。

③ 利用自顶向下的方法设计算法。把整个程序看作一个整体，先全局后局部，自顶向下，一层一层分解处理，逐步求精。直到整个问题可用程序设计语言明确地描述出来为止。

④ 采用结构化程序设计语言书写程序。只使用三种基本结构，语句块采用缩进格式，可以适当使用注解，使程序结构清晰，易于阅读。

2. 结构化程序设计的步骤

第一步：首先对问题进行仔细分析，确定其输入、输出数据，写出程序运行的主要过程和任务。

第二步：把一个复杂的大问题分解为若干相对独立的小问题。如果小问题仍较复杂，则可以把这些小问题又继续分解成若干子问题，这样不断地分解，使得小问题或子问题简单到能够直接用程序的三种基本结构表达为止。

第三步：对应每一个小问题或子问题编写出一个功能上相对独立的程序模块来。模块是数据说明和可执行语句的集合，它有良好的接口(有名字，输入，输出)。一个模块只有一种用途或者只完成一个任务。

第四步：把这些模块集中起来组成一个整体，程序中的模块相互协作完成程序的功能，完成对问题的求解。这种用模块组装起来的程序被称为模块化结构程序。

可见，编制程序与建大楼一样，首先要考虑大楼的整体结构而忽略一些细节问题，待把整体框架搭起来后，再逐步解决每个房间的细节问题。使用自顶向下、逐步细化的设计方法符合人们解决复杂问题的一般规律：先全局后局部，先整体后细节，先抽象后具体、逐步求精。

在考虑如何解决一个问题时，往往不可能一下子就触及问题的细节，也无法立即提出解决问题的详细步骤。因此在详细地分析了该问题的条件和要求之后，首先考虑解决问题的策略，从而设计出一个用自然语言描述的算法，这个算法中的各个步骤是不具体的，数据也不作具体的表达。然后在下一步，再考虑这些抽象的数据和操作如何实现，即逐步求精。如此逐步进行设计，直到所得到的算法是用具体的赋值语句、条件语句、循环语句和输出语句表示为止。

【例 1】成绩统计。某小组 8 位同学的 6 门课的成绩如下：

101	85,69,67,78,86,89
102	90,98,95,89,96,90
103	72,60,75,84,94,89
104	81,78,90,80,76,78
105	69,68,91,70,66,70

106	92,80,84,79,90,56
107	99,92,65,75,61,74
108	75,79,81,90,85,55

试编程求个人总分，并按总分排序。



分析：这是一个比较典型的数据处理问题。

第一步：确定程序的输入数据和主要任务。

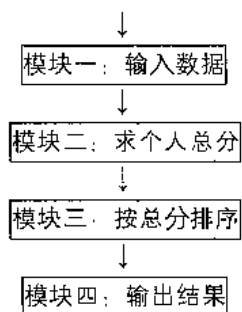
输入数据：n 个同学的学号和 k 门课的成绩；

主要任务：① 计算个人总分

② 按总分排序

第二步：自上而下，分解功能模块，建立任务框架。

我们可以把问题按要求分解成如下几个功能模块：



第三步：分而治之，分别根据各个模块需要，设计算法。

模块一：输入数据

```

SUB in
  FOR i=1 TO n
    读入每一个同学的数据
  NEXT i
END SUB

```

模块二：输入数据

```

SUB in
  FOR i=1 TO n
    计算每一个同学的总分
  NEXT i
END SUB

```

模块三：排序

```

SUB sort
  按个人总分排序
END SUB

```

模块四：输出结果

```

SUB out
  FOR i=1 TO n

```

输出每一个同学的数据

```
NEXT i
END SUB
```

第四步：逐步求精，编写独立功能模块。

说明变量：用变量 n 存放人数，用变量 k 存放科目数。

用数组 $\text{name}\$(n)$ 存放姓名，用数组 $x(n,k)$ 存放成绩。

用数组 $s(n)$ 存放个人总分，用数组 $mc()$ 存放名次。

为了简化过程参数，可以将上述变量定义为全局变量。

模块一：输入数据

```
SUB in
FOR i=1 TO n
    REM 读入每一个同学的数据
    READ name$(i)           '读入学号
    FOR j=1 TO k
        READ x(i,j)         '读入成绩
    NEXT j
NEXT i
END SUB
```

模块二：输入数据

```
SUB in
FOR i=1 TO n
    REM 计算每一个同学的总分
    FOR j=1 TO k
        s(i) = s(i) + x(i,j) '求总分（二维数组按行求和）
    NEXT j
NEXT i
END SUB
```

模块三：排序

```
SUB sort
REM 按个人总分排序
FOR i=1 TO n-1
    FOR j=i TO n
        REM 交换两个同学的成绩
    NEXT j
NEXT i
END SUB
```

模块四：输出结果

```
SUB out
FOR i=1 TO n
    REM 输出每一个同学的数据
```

```

FOR i = 1 TO n
    PRINT name$ (i);           '输出姓名
    FOR j = 1 TO k
        PRINT x (i,j);        '输出个人各科成绩
    NEXT j
    PRINT s (i);               '输出个人总分
    PRINT mc (i)               '输出名次
NEXT i
NEXT i
END SUB

```

第五步：模块进一步求精。

模块三：排序

```

SUB sort
    REM 按个人总分排序
    FOR i = 1 TO n - 1
        FOR j = i TO n
            REM 交换两个同学的成绩
            IF s (i) < s (j) THEN
                SWAP name$ (i), name$ (j) '交换学号
                SWAP s (i), s (j)         '交换总分
                FOR kk = 1 TO k
                    SWAP x (i, kk), x (j, kk) '交换成绩
                NEXT kk
            END IF
        NEXT j
    NEXT i
END SUB

```

模块四：考虑控制表格式输出；

考虑输出同名次问题。

```

SUB out
    FOR i=1 TO n
        REM 输出每一个同学的数据
        FOR i = 1 TO n
            PRINT name$ (i);           '输出姓名
            FOR j = 1 TO k
                REM 控制表格式输出个人各科成绩
                PRINT TAB (4 + 5 * j); x (i,j);
            NEXT j
            PRINT TAB (6 + 5 * j); s (i); '输出个人总分
            REM 考虑同名次输出

```

```

        IF s(i) = s(i-1) THEN kk = kk + 1: mc(i) = mc(i) - 1
        PRINT TAB(2 + 6 * (j + 2)); mc(i)
    NEXT i
NEXT i
END SUB

```

第六步，调用模块，组装程序。

主程序如下：

```

REM 为了简化过程参数起见，将有关变量定义为全局变量。
DIM SHARED n, k
n = 8: k = 6
DIM SHARED name$(n), s(n), mc(n)
DIM SHARED x(n, k)
DATA 101, 85, 69, 67, 78, 86, 89
DATA 102, 90, 98, 95, 89, 96, 90
DATA 103, 72, 60, 75, 84, 94, 89
DATA 104, 81, 78, 90, 80, 76, 78
DATA 105, 69, 68, 91, 70, 66, 70
DATA 106, 92, 80, 84, 79, 90, 56
DATA 107, 99, 92, 65, 75, 61, 74
DATA 108, 75, 79, 81, 90, 85, 55
CALL in '调用输入成绩过程
CALL sum '调用求个人总分过程
CALL sort '调用按总分排序过程
CALL pri '调用按名次输出过程
END

```

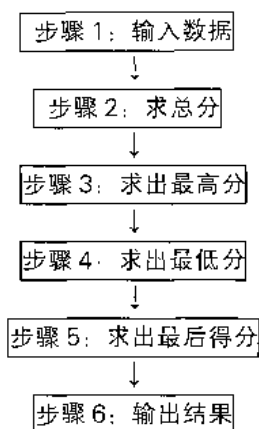
运行结果如下：

102	90	98	95	89	96	90	558	1
104	81	78	90	80	76	78	483	2
106	92	80	84	79	90	56	481	3
101	85	69	67	78	86	89	474	4
103	72	60	75	84	94	89	474	4
107	99	92	65	75	61	74	466	6
108	75	79	81	90	85	55	465	7
105	69	68	91	70	66	70	434	8

【例 2】为了丰富校园文化艺术生活，在一年一度举行的校园歌手大奖赛上，有 10 名裁判为每位选手的表演打分，分值在 1~100 分之间。评分的规则是：去掉一个最高分和一个最低分，选手最后得分是其余 8 个得分的平均值。请编写程序实现为每位选手打分。



分析：先粗略地划分解决问题的步骤，然后再对抽象出来的每一个步骤进一步设计算法，即逐步求精。流程图如下：



第一步，划分模块。

模块 1：输入数据。

```
REM 输入评委人数 n
FOR i=0 TO n
    REM 输入各评委的打分结果,保存到数组 a 中
NEXT i
```

模块 2：累加求总分。

用变量 s 存放总分。利用循环累加求出总分。

```
初始化  $s \leftarrow 0$ 
FOR i=1 TO n
     $s \leftarrow s + a(i)$ 
NEXT i
```

模块 3：求最高分。

从一个数组 a 中找出数组元素的最大值，可以利用循环通过比较来实现。依次把数组 a 中的数据与 mx 比较。每次比较后，就把较大数放入 mx 中。最后 mx 中的值，就是数组元素中的最大值。

```
 $mx \leftarrow a(1)$ 
FOR i=2 TO n
    IF  $mx < a(i)$  THEN  $mx \leftarrow a(i)$ 
NEXT i
 $mx$  为最大值
```

模块 4：求最低分。

用变量 mn 存放最低分。利用循环比较找出数组 $a(i)$ 的最小值。依次把数组 a 中的数据与 mn 比较。每次比较后，就把较小数放入 mn 中。最后 mn 中的值，就是数组元素中的最小值。