



NE4C育才

NORTHEAST YUCUI SCHOOL

东北育才学校优才教育精品校本课程系列教材

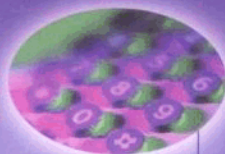
SUANFAYISHU

数学思维训练

数学思维训练

算法艺术

主编/高琛 副主编/刘子军



本册主编/邱桂香



辽宁教育出版社



NEYC育才
NORTHEAST YOUTH SCHOOL

东北育才学校优才教育精品校本课程系列教材

算法艺术

主编 / 高琛 副主编 / 刘子军

本册主编 / 邱桂香



辽宁教育出版社

图书在版编目 (CIP) 数据

算法艺术/邱桂香主编. —沈阳: 辽宁教育出版社, 2009.3
(东北育才学校优才教育精品校本课程系列教材/高琛主编)
ISBN 978-7-5382-8201-6

I. 算… II. 邱… III. 电子计算机—算法理论—中学—教学参考资料
IV. G634.673

中国版本图书馆 CIP 数据核字 (2008) 第 208624 号

辽宁教育出版社出版、发行
(沈阳市和平区十一纬路 25 号 邮政编码 110003)
沈阳航空发动机研究所印刷厂印刷

开本: 787 毫米×1092 毫米 1/16 字数: 245 千字 印张: 13 1/4 插页: 1
印数: 1—2000 册

2009 年 3 月第 1 版

2009 年 3 月第 1 次印刷

责任编辑: 崔 崇 王 莹 关 鑫

责任校对: 王 静

封面设计: 熊 飞

版式设计: 熊 飞

ISBN 978-7-5382-8201-6

定 价: 27.00 元

继《中国古代文化常识》《影视制作及多媒体技术初步》《团体心理训练》《陶林漫步》和《有机化学》之后，东北育才学校又集中出版了《成功心理训练》《天文观测》《高中生职业生涯规划》《茶艺》《中学生单片机技术》《算法艺术》《咨询员培训》《稚拙的世界》《动态数理探究课程》《中国古代小说简史》《初中综合数学》《新语文》等一套共 12 本精品校本课程系列教材，可喜可贺。这是东北育才学校校本课程开发和建设的又一丰硕成果。它伴随着育才文化建设的时代脚步，承载着育才文化的基因面世，从一个侧面表明育才的改革在深化，育才的事业在发展。

比较而言，这一套校本教材在内容上，更加关注受教育者的成功立志，关注意志品格的锤炼，关注创造精神和实践能力的形成，关注思维品质和学习品质的培养，关注素养、人格和终身发展，既体现了东北育才学校“培养各方面领军人物领袖资质”的培养目标，也反映了该校认真落实“建设人力资源强国”的基本国策，使学生成为强国兴邦之栋梁的高度自觉。把打基础“全面、有个性”的做法，又提高到了一个新的层次。这启示我们，贯彻校本课程开发与建设的基础性原则，既要注意从本校实际出发，又要紧跟时代的脚步。

这套教材更加注重多样性和补充性，从实施操作角度，大大增加了学生自主选择的机会，拓展了他们涉猎更多课外领域知识和参与互动对话的空间，这在“校本课程主要服务于学生获取综合性知识和直接经验，主要满足于学生兴趣需

求”的宗旨定位上是一大进步。这启示我们，校本课程的独到之处与独特作用的呈现，有赖于“校本课程开发与建设必须坚持以学生发展为本，追求校本课程最大限度贴近学生个性发展需要”方针的认真贯彻。开发是为了应用，应用也有助于开发。这套教材中的许多内容就是源自学生社团活动的诉求和活动的感悟与经验的升华。

依托包括学生社团在内的各种组织，特别是场所设施并作为资源加以充分的开发与利用，是东北育才学校校本课程开发与建设的一个显著特点，也是该校不断开创校本课程开发与建设新局面并使之上层的重要原因。东北育才学校伴随新课程改革和校园文化建设，有计划地建设了一大批能满足学生个性发展需要，培养探索、实践和动手能力的诸如机器人技术实验室、软件科学研究实验室、辽宁资源展馆、生命科学展馆、生态科普教育基地、天文观测站、校史馆、陶艺室、茶艺室、服装及设计教室、建筑及设计教室、汽车模拟驾驶室等高水平的实验基地和实习场所，催生了类似茶艺、天文观测等一些新的校本课程，同时又为这些校本课程的增效，发挥了不可替代的支撑作用。例如，在堪称国内一流的茶艺室讲茶道，通过文雅凝重的文化氛围，弘扬中华传统美德，帮助学生修身养性，提高人文素养；在本校的天文台上进行“天文观测”，感受浩瀚星空的浩瀚和神秘，自有亲切别样情趣；在辽宁资源展馆流连，感受家乡的史之悠久、地之广博、物之丰饶；在校史馆中徜徉，回望那一段如歌的岁月，品味育才踟躅前行的艰难与沧桑，更幸福于作为一名育才人的自豪……所有这些启示我们，注重资源开发是校本课程开发与建设的重要条件，也是取得育人实效的有利保障，特别是校内资源的充分开发与利用，不可或缺，它具有独到性，有时是无法替代的。

以上各点表明，坚持优才教育的办学理念和追求卓越的行为方式的东北育才学校校本课程的开发与建设本身也在前进和发展。本套校本教材的出版是育才文化在闪光！其品位价值的客观存在，对我市中小学的借鉴意义不可低估。我们确信，在东北育才学校绚丽的校本教材的引领下，我市也会迎来校本课程开发的万紫千红。希望东北育才学校开发出更好更多的校本课程系列丛书，祝愿全市中小学的校本课程开发与建设的局面越来越喜人！

苏文捷

编者寄语

信息学奥赛的内容以程序设计为主，而程序设计的核心是算法设计。

算法是艺术，是思维的艺术，是科学的艺术，是创造的艺术。只有深入其中、勇于探索、追求完美的人，才能感受到算法艺术的无穷魅力。

经历了十余年的信息学奥赛辅导，本人接触过众多的信息学奥赛精英，他们的灵感、他们的算法、他们的代码，常常令人叹为观止。于是，本人时时有一种冲动，要将算法之美呈现给更多的人，要让更多的人在学的过程中，感受到算法艺术之美。今年年初，本人终于有了一个实现夙愿的机会。

《算法艺术》是在整理多年的信息学奥赛辅导讲义基础上形成的，在行文上有三个特点：

第一，按照基本算法、排序算法、搜索算法、图的算法和动态规划的顺序，由浅入深地对信息学奥赛中经常出现的算法进行了研究和解析，不仅降低了学习难度，更顺应了学习进程，便于读者掌握多种算法。

第二，针对每一种算法，从算法思想、典型例题、相关习题三个角度进行描述，有利于读者逐步加深对算法的理解，进而应用算法解题。

第三，对每一道例题的解析，并没有单纯给出代码，而是先进行算法设计，然后撰写程序代码。这种结构设计，目的在于引导读者经历思维的过程，使其形成良好的思维习惯，进而培养并提高其解题能力。

本书适合具备程序设计语言基础的读者在深入研究算法时使用，也适合信息学奥赛教练作为参考资料。需要说明的是，本书解题中提供的算法仅为抛砖引玉。深入思考、实践探究是学好算法的最佳方法，更是实现和提升本书价值的必由之路。

目 录

序言

编者寄语

第一章 算法概述	1
1.1 算法的概念	1
1.2 算法的描述	1
1.3 算法的分析	3
第二章 基本算法	7
2.1 穷举法	7
2.2 贪心法	13
2.3 分治法	21
2.4 递推法	29
2.5 递归法	39
2.6 回溯法	46
第三章 排序算法	54
3.1 排序算法	54
3.2 排序算法的比较和选择	64
第四章 搜索算法	77
4.1 深度优先搜索	77
4.2 广度优先搜索	89
第五章 图的算法	110
5.1 最短路径	110
5.2 最小生成树	128
5.3 拓扑排序	144

第六章 动态规划	161
6.1 线性转移	162
6.2 阶段性转移	170
6.3 树型转移	181
6.4 多维空间转移	192



第一章 算法概述

1.1 算法的概念

图灵奖获得者 D.E.Knuth 在他的巨著《计算机程序设计技巧》中说：“算法的现代意义十分类似于处方、过程、方法、技术、规程、程序，只不过是算法一词的含义有时稍微有一点特殊。一个算法，就是一个有穷规则的集合，其中之规则规定了一个解决某一特定类型的问题的运算序列。”

算法是在有限步骤内求解某一问题所使用的一组定义明确的规则。通俗点说，就是计算机解题的过程。

算法是程序设计的精髓，程序设计的实质就是构造解决问题的算法，将其解释为计算机语言。

一个算法应该具有如下特征：

1. 有穷性：一个算法必须保证它的执行步骤是有限的。
2. 确切性：算法中的每一个步骤必须有确切的定义。
3. 能行性：算法中的每一个步骤都要足够简单，能够精确地运行，并且能在有限的时间内完成。
4. 输入：一个算法有 0 个或多个输入，以刻画运算对象的初始情况。
5. 输出：一个算法有一个或多个输出，以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

1.2 算法的描述

一个算法可以用多种不同的方法来描述，例如自然语言法、流程图法、计算机语言法等。同一算法，无论用何种方法描述，其功用是一样的，只是在算法的描述和实现方法上有所不同。

流程图是一种比较直观易用的、用图形来描述算法的方法。结构化程序设计中三种程序结构的流程图（N—S 图）如下：

1. 顺序结构

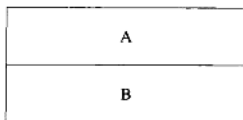


图 1-2-1 顺序结构 N-S 图

2. 选择结构

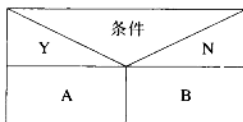


图 1-2-2 选择结构 N-S 图

3. 循环结构

(1) 当型循环

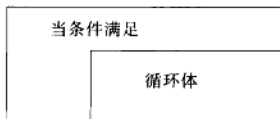


图 1-2-3 当型结构 N-S 图

(2) 直到型循环

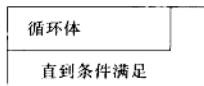


图 1-2-4 直到型结构 N-S 图

【例题】画出下述问题的流程图。

打印任意 10 个数中的正数。



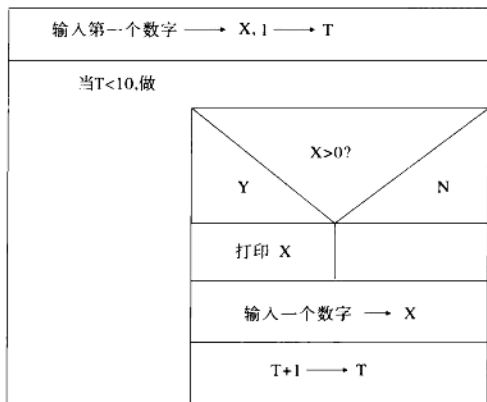


图 1—2—5 例题的流程图

1.3 算法的分析

同一问题可以使用不同的算法解决，而一个算法的质量优劣将影响到算法乃至程序的效率。算法分析的目的就在于选择合适算法和改进算法，算法分析的内容是分析算法的复杂度。

一个算法的复杂度高低体现在运行该算法所需要的计算机资源的多少。所需的资源越多，该算法的复杂性越高；反之，则该算法的复杂性越低。计算机的资源，最重要的是时间和空间（即存储器）资源。因而，算法的复杂度有时间复杂度和空间复杂度之分。

一、时间复杂度

（一）时间频度

执行一个算法所耗费的时间，从理论上讲是必须上机运行测试才能知道。但我们不可能也没有必要对每个算法都上机测试，只需知道算法花费的时间多少就可以了，并且一个算法花费的时间与算法中语句的执行次数成正比例。一个算法中的语句执行次数称为语句频度或时间频度，记为 $T(n)$ 。其中 n 称为问题的规模，当 n 不断变化时，时间频度 $T(n)$ 也会不断变化。

(二) 时间复杂度

若有某个辅助函数 $f(n)$, 使得当 n 趋近于无穷大时, $T(n)/f(n)$ 的极限值为不等于零的常数, 则称 $f(n)$ 是 $T(n)$ 的同数量级函数。记作 $T(n)=O(f(n))$, 称 $O(f(n))$ 为算法的渐近时间复杂度, 简称时间复杂度。

在各种不同的算法中, 若算法中语句执行次数为一个常数, 则时间复杂度为 $O(1)$ 。另外, 在时间频度不相同, 时间复杂度有可能相同, 如 $T(n)=n^2+3n+4$ 与 $T(n)=4n^2+2n+1$, 它们的频度不同, 但时间复杂度相同, 都为 $O(n^2)$ 。

按数量级递增排列, 常见的时间复杂度有: 常数阶 $O(1)$ 、对数阶 $O(\log_2 n)$ 、线性阶 $O(n)$ 、线性对数阶 $O(n \log_2 n)$ 、平方阶 $O(n^2)$ 、立方阶 $O(n^3)$ 、 k 次方阶 $O(n^k)$ 、指数阶 $O(2^n)$ ……随着问题规模 n 的不断增大, 上述时间复杂度也不断增大, 算法的执行效率不断降低。这些算法时间复杂度由小到大依次为:

$$O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < \dots < O(2^n) < O(n!)$$

(三) 求解算法的时间复杂度

根据算法, 求解其时间复杂度的具体步骤是:

1. 找出算法中的基本语句

算法中执行次数最多的那条语句就是基本语句, 通常是最内层循环的循环体。

2. 计算基本语句的执行次数的数量级

只需计算基本语句执行次数的数量级, 这就意味着只要保证基本语句执行次数的函数中的最高次幂正确即可, 可以忽略所有低次幂和最高次幂的系数。这样能够简化算法分析, 并且使注意力集中在最重要的增长率上。

3. 用大 O 记号表示算法的时间性能

将基本语句执行次数的数量级放入大 O 记号中。

如果算法中包含嵌套的循环, 则基本语句通常是最内层的循环体; 如果算法中包含并列的循环, 则将并列循环的时间复杂度相加。

例如, 在以下三个程序段中:

```
(1) x:=x+1
(2) for i:=1 to n do
    x:=x+1
(3) for i:=1 to n do
    for j:=1 to n do
        x:=x+1
```

基本操作 “ $x:=x+1$ ” 出现的次数分别为 1、 n 和 n^2 , 则这三个程序段的时间复杂度分别为 $O(1)$, $O(n)$, $O(n^2)$ 。



二、空间复杂度

与时间复杂度类似，空间复杂度是指算法在计算机内执行时所需存储空间的度量。记作：

$$S(n)=O(f(n))$$

我们一般所讨论的是除正常占用内存开销外的辅助存储单元规模。

由于当今计算机硬件技术发展很快，程序所能支配的自由空间一般比较充足，所以空间复杂度就不如时间复杂度那么重要了。在研究算法效率时，人们主要是研究其算法的时间复杂度，而很少讨论它的空间耗费。

时间复杂度和空间复杂度在一定条件下是可以相互转化的。在中学生信息学奥赛中，对程序的运行时间作出了严格的限制，如果运行时间超出了限定就会判错。因此在设计算法时首先要考虑的是时间因素，必要时可以牺牲空间来换取时间。

三、算法的分析

【例题】下面是两个“求 $n!$ 所产生的数末尾有多少个连续的 0（中间的 0 不计）”的算法，比较其时间复杂度。

算法一的 pascal 程序如下：

```
var i,t,n,sum:longint;
begin
  t:=0;
  sum:=1;
  readln(n);
  for i:=1 to n do
  begin
    sum:=sum*i;
    while (sum mod 10=0) do
    begin
      sum:=sum div 10;
      inc(t);
    end;
    sum:=sum mod 1000;
  end;
```

```

        writeln(t:6);
    end.
    算法二的 pascal 程序如下:
    var t,n:integer;
    begin
        readln(n);
        t:=0;
        repeat
            n:=n div 5 ;
            inc(t,n);
        until n<5;
        writeln(t:6);
    end.

```

分析：从程序描述可以看出，算法一的思路是：从 1 乘到 n ，每乘一个数判断一次，若后面有 0 则去掉后面的 0，并记下 0 的个数。为了不超出数的表示范围，去掉与生成 0 无关的数，只保留有效位数，当乘完 n 次后就得到 0 的个数。算法二的思路是：因为此题中生成 0 的个数只与含 5 的个数有关， $n!$ 的分解数中含 5 的个数就等于末尾 0 的个数，因此问题转化为直接求 $n!$ 的分解数中含 5 的个数。

对比两种算法就不难看出，它们的时间复杂度分别为 $O(n)$ 、 $O(\log_2 n)$ ，算法二的执行时间远远小于算法一的执行时间。



第二章 基本算法

2.1 穷 举 法

一、算法思想

所谓穷举法，也称枚举法，指的是从可能解的集合中一一枚举各个元素，用题目给定的检验条件判定哪些是能使条件成立的元素，这些元素就是问题的解。

二、典型例题

(一) 构造三角形

1. 问题描述

将 A 、 B 、 C 、 D 、 E 、 F 这六个变量排成如图 2—1—1 所示的三角形，这六个变量分别取 $[1, 6]$ 中的整数，且均不相同。求使三角形三条边上的变量之和相等的全部解。

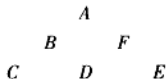


图 2—1—1 三角形

如图 2—1—2 就是一个解。

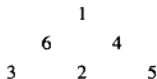


图 2—1—2 三角形的一个解



2. 算法设计

用变量 a 、 b 、 c 、 d 、 e 、 f 分别顺序取 1 至 6 的整数，在它们互不相同的条件下，判断由它们排成的如图 2—1—1 所示的三角形，三条边上的变量之和是否相等。如果相等就是一种满足要求的排列，把它们输出。当这些变量取尽所有的组合后，程序就可得到全部可能的解。

3. 程序代码

```
var a,b,c,d,e,f:1..6;
begin
    for a:=1 to 6 do
        for b:=1 to 6 do
            if a<>b then
                for c:=1 to 6 do
                    if (a<>c)and(b<>c)then
                        for d:=1 to 6 do
                            if (a<>d)and(b<>d)and(c<>d)then
                                for e:=1 to 6 do
                                    if (a<>e)and(b<>e)and(c<>e)and(d<>e)then
                                        begin
                                            f:=(1+2+3+4+5+6)-(a+b+c+d+e);
                                            if (a+b+c=c+d+e)and(a+b+c=a+f+e)then
                                                writeln(a,b,c,d,e,f)
                                            end;
                                        end;
                                    end;
                                end;
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
```

按穷举法编写的程序通常不能适应变化的情况。如问题改成有 9 个变量排成三角形，每条边有 4 个变量的情况，程序的循环重数就要相应改变。

(二) 背包问题

1. 问题描述

有 n ($n < 10$) 件不同价值、不同重量的物品，现在要从这 n 件物品中选取一部分物品，使选中物品的总重量不超过指定的限制重量，且选中物品的价值之和最大。请找出选择方案。

2. 算法设计

n 件物品的选择会有 2^n 种情况，直接使用穷举法，会有多重循环。那么，我们换一个思路来想问题：既然每一种物品只有选或不选两种情况，我们不妨对每一件物品用 0 表示不选，用 1 表示选取，那么每一种背包选取方案就是 0 和 1 组成的一个 n 元组，而这个 n 元组也可以理解成一个 n 位的二进制数，我

们也不难得出这个 n 位的二进制数的取值范围为 $0 \sim 2^n - 1$ 。那么，把 $0 \sim 2^n - 1$ 分别转化为相应的二进制数，一一枚举，就可以得到我们所需要的 2^n 个 n 元组，穷举之后，就可以找到价值最大的一组。

3. 程序代码

```
var n,m,i,j,temp: integer;
    maxv,maxw,temv,temw:real;
    v,w:array[1..10] of real;
    a,b:array[1..10] of 0..1;
begin
    assign(input,'bag.in');
    reset(input);
    assign(output,'bag.out');
    rewrite(output);
    readln(maxw);
    readln(n);
    m:=1;
    for i:=1 to n do
        begin
            readln(v[i],w[i]);
            m:=m*2;
        end;
    maxv:=0;
    for i:=0 to m-1 do
        begin
            temp:=i;
            temw:=0;
            temv:=0;
            for j:=1 to n do
                begin
                    if temp mod 2=0 then
                        b[j]:=0
                    else
                        begin
                            b[j]:=1;
                            temw:=temw+w[j];

```