

新编

奥林匹克基础知识及素质教育丛书

计算机(下)

主编：吕 品

展开思想的翅膀
活跃在奥林匹克广场上
为了明天的成功
哪怕今天摸爬滚打
让我们手挽手
深挖智慧的力量
奋斗——前进——
这里是练兵的战场



科学技术文献出版社

◇ 新编奥林匹克基础知识及素质教育丛书

计 算 机

(下 册)

主 编 吕 品
编 著 滕秋莉 庄燕文
黎 巛

科学技术文献出版社

Scientific and Technical Documents Publishing House

北 京

图书在版编目(CIP)数据

计算机/吕品主编.-北京:科学技术文献出版社,1999.3
(新编奥林匹克基础知识及素质教育丛书)

ISBN 7-5023-3231-6

I. 计… II. 吕… III. 计算机课-中学-教学参考资料 IV. G633.673

中国版本图书馆 CIP 数据核字(98)第 38983 号

出 版 者:科学技术文献出版社

图 书 发 行 部:北京市复兴路 15 号(公主坟)中国科学技术信息研究所大
楼 B 段/100038

图 书 编 务 部:北京市西苑南一院 8 号楼(颐和园西苑公汽站)/100091

邮 购 部 电 话:(010)68515544-2953

图书编务部电话:(010)62878310,(010)62877791,(010)62877789

图书发行部电话:(010)68515544-2945,(010)68514035,(010)68514009

门 市 部 电 话:(010)68515544-2172

图书发行部传真:(010)68514035

图书编务部传真:(010)62878317

E-mail:stdph@istic.ac.cn

策 划 编 辑:王亚琪 王 琦

责 任 编 辑:王亚琪

责 任 校 对:赵文珍

责 任 出 版:周永京

封 面 设 计:宋雪梅

发 行 者:新华书店北京发行所

印 刷 者:北京国马印刷厂

版 (印) 次:1999 年 3 月第 1 版 1999 年 3 月第 1 次印刷

开 本:850×1168 32 开

字 数:188 千

印 张:7

印 数:1—10000 册

定 价:24.00 元(上册 14.00 元 下册 10.00 元)

© 版权所有 违法必究

购买本社图书,凡字迹不清、缺页、倒页、脱页者,本社发行部负责调换。

(京)新登字 130 号

内 容 简 介

本书是为广大青少年计算机爱好者和信息学奥林匹克学校的学生了解信息学奥林匹克和深入学习计算机知识而编写。其主要内容为:PASCAL 语言的基础知识;基本数据结构(如栈、队列、链表、数组、树、图等);常用算法;人工智能的一些初步知识(如深度优先搜索、广度优先搜索、启发式搜索、分枝定界、动态规划和博弈等)。

全书有三个特点:一是以算法为中心,语言为工具,结合算法和程序的设计学习 PASCAL 语言的有关知识。二是理论与实际密切结合,从实例出发引出理论,再用理论指导实践,解决实际问题。内容有一定深度但不脱离学生实际。三是便于自学。书中有较多的实例,有详尽的分析,并配有相当数量的习题来加以巩固。

科学技术文献出版社 向广大读者致意

科学技术文献出版社成立于 1973 年,国家科学技术部主管,主要出版科技政策、科技管理、信息科学、农业、医学、电子技术、实用技术、培训教材、教辅读物等图书。

我们的所有努力,都是为了使您增长知识和才干。

前 言

近些年来,世界范围内的学科奥林匹克竞赛方兴未艾。我国自参赛以来,不断取得优异成绩。1997年,我国参加在阿根廷布宜诺斯艾利斯举办的第37届世界数学奥林匹克竞赛,6名选手均获金牌,并取得了团体第一名的好成绩。学生参加各学科的奥林匹克竞赛活动,不但为国家争得了荣誉,也已成为他们丰富学习的内容、增长知识、提高各门功课学习成绩的重要方式之一。

为了帮助广大中小学生完整、准确、全面地掌握各门功课的学习内容,在日常的学习和参加奥林匹克竞赛活动中取得好的成绩,同时为了配合目前中小学素质教育,我们邀请了京内外著名奥校具有多年教学与辅导经验的权威老师,编写了这套《新编奥林匹克基础知识及素质教育丛书》。

参加本丛书编写工作的老师,全部来自于教学第一线,具有扎实的基础理论功底和丰富的教学实践经验。他们结合自己多年教学、科研和奥校辅导的经验,在总结各类奥林匹克竞赛教学讲义、习题解答及辅导材料的基础上,博采众家之长,形成了本丛书独具特色的风格和特点:

(1)学科门类齐全。全套丛书共18分册,涵盖数学、物理、化学、生物、计算机5个学科,跨越小学、初中、高中三个阶段,是目前此类图书中覆盖学科最广、教学内容最全、实用性最强的奥林匹克竞赛系列丛书之一。

新编奥林匹克基础知识及素质教育丛书

主要作者简介

- 吴文虎** 中国计算机学会普及委员会主任
国际信息学奥林匹克中国队总教练
清华大学计算机系教授
- 吕 品** 全国计算机教材审查委员会委员
北京信息学奥林匹克学校副校长
中学高级教师
- 刘 尧** 北京教育学院化学教研室主任、教授
- 陆 禾** 北京 14 中化学特级教师
北京市有突出贡献的专家
- 黄儒兰** 北京教育局化学教研室主任
中学特级教师
- 冯士腾** 北京宣武区教育学会秘书长
中学特级教师
- 李方烈** 北京宣武区中学数学教研室主任
中学特级教师
- 赵欣如** 北京师范大学生物系教授
中国生物奥林匹克竞赛委员会委员
- 曹保义** 北京师范大学二附中副校长
生物教研组组长
中学高级教师

- 高建军** 湖南长沙一中生物教研组组长
中学高级教师
- 石长地** 首都师范大学研究生处教师
数学奥林匹克专业研究生毕业
教育学硕士
- 贺贤孝** 辽宁师范大学数学系教授
辽宁数学教育学会副会长
- 杨 骞** 辽宁师范大学数学系副教授
大连市奥林匹克学校校长
- 由 峻** 北京市宣武区中学教研室主任
- 秦家达** 北京市 66 中物理教研组组长
中学高级教师
- 高玉镶** 北京师范大学附中物理高级教师
- 马凌风** 北京市 15 中物理教研组组长
中学高级教师
- 王健子** 北京市 15 中物理高级教师

目 录

☞	第七章 深度优先搜索	(299)
☞	第八章 广度优先搜索	(341)
☞	第九章 启发式搜索	(378)
☞	第十章 分枝定界	(403)
☞	第十一章 动态规划	(435)
☞	第十二章 博弈	(459)

第七章 深度优先搜索

§ 7.1 深度优先搜索

我们在对一些问题进行求解时,会发现有些问题很难找到规律,或者根本无规律可寻。对于这样的问题,可以利用计算机运算速度快的特点,先搜索查找所有可能出现的情况,再根据题目条件从所有可能的情况中,删除那些不符合条件的解。

【例 7.1】 有 A、B、C、D、E 5 本书,要分给张、王、刘、赵、钱 5 位同学,每人只能选 1 本。每个人都将自己喜爱的书填写在下表中。请你设计一个程序,打印出让每个人都满意的所有分书方案。

	A	B	C	D	E
张			✓	✓	
王	✓	✓			✓
刘		✓	✓		
赵				✓	
钱		✓			✓

☞ 问题分析:

题目中每人喜爱哪本书是随意的,无规律可循,所以用穷举方法解较为合适。按穷举法的一般算法,可以暂不考虑一些条件,先求出满足部分条件的解,即可行解。然后,再加上尚未考虑的条件,从可行解中删除不符合这些条件的解,留下的就是问题的解。具体到本题中,我们可以先不考虑“让每人都满意”这个条件,这样,就只剩“每人选一本且只能选一本”这一个条件了。在这个条件下,可行解是5本书的所有全排列,一共有 $5! = 120$ 种情况。从这120种可行解中删去不符合“每人都满意”这一条件的解,剩下的就是本题的解。

为编程方便,我们用1、2、3、4、5分别表示这5本书。这5个数字的一种全排列就是5本书的一种分法。例如54321就表示第五本书(即E)分给张,第四本书(即D)分给王……,第一本书(即A)分给钱。

每个人“喜爱书表”,在程序中我们用二维数组Like来表示,1表示喜爱,0表示不喜爱。排列的产生可以用穷举法,也可以用专门算法。

☞ 算法设计:

第一步:产生5个数字的一个全排列;

第二步:检查所产生的全排列是否符合“喜爱书表”,如果符合就输出;

第三步:检查是否所有排列都产生了,如果没有产生完,则返回第一步;

第四步:结束。

根据题目给出的条件,还可以对上面算法进行一些改进。例如产生一个全排列12345时,第一个数1表示将第一本书给小张。但从表中可以看出,这是不可能的,因为小张只喜欢第三、第四本书。也就是说, $1 \times \times \times \times$ 这一类分法是不符合条件的。由此使我们想到,如果选定第一本书后,就立即检查一下是否符合条件,当发现第一个数的选择不符合条件时,就不必再产生后面的4个数了,这样做

可以减少很多的运算量。换句话说,第一个数只在 3 和 4 中选择,这样就可以减少 $3/5$ 的运算量。同理,在选定了第一个数后,其他 4 个数字的选择也可以用类似的方法处理,即选择第二个数后,立即检查是否符合条件。例如,第一个数选 3,第二个数选 4 后,立即进行检查,发现不符合条件,就另选第二个数。这样就又把 $34 \times \times \times$ 一类的分法删去了,从而又减少了一部分运算量。

综上所述,改进后本题算法应该是:在产生各种排列时,每增加一个数字,就检查一下该数的加入是否符合条件,如不符合,就立刻换一个;若符合条件,则再产生下一个数。因为从第 i 本书到第 $i+1$ 本书的寻找过程是相同的,所以可以用递归方法编程。

算法框图:

PROCEDURE TRY(i);

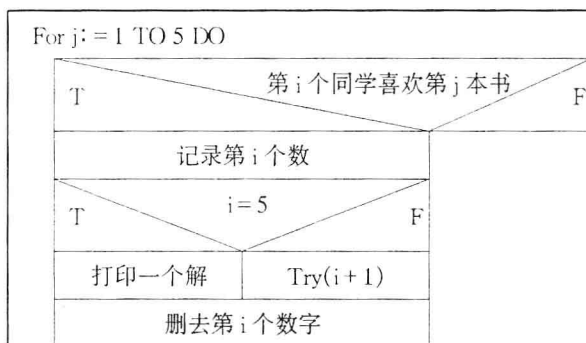


图 7-1

我们用二维数组 like 存放“喜爱书表”,用集合 flag 存放已分出书的编号,数组 book 存储各人所分得书的编号,如 $\text{book}[1] = 3$,则表示第一个同学(小张)分得编号为 3 的书。

递归程序如下:

```
Program allot_book(output);
```

```
type five = 1..5;
```

```

const like:array[five,five] of 0..1 =
    ((0,0,1,1,0),(1,1,0,0,1),(0,1,1,0,0),(0,0,0,1,0),
    (0,1,0,0,1));
                                {个人对各种书的喜爱情况}
    name:array[five] of string[5]
= ('zhang','wang','liu','zhao','qian');
                                {数组 name 存放学生姓名}
    var book:array[1..5] of 0..5;
                                {存放各人分配到的书的编号}

    flag:set of five;
    c:integer;

procedure print;                {打印分配方案}
var i:integer;
begin
    inc(c);                      {计数,统计得到分配方案数}
    writeln('answer',c,':');
    for i:=1 to 5 do
        writeln(name[i]:10,':',chr(64+book[i]));
    end;

procedure try(i:integer); {判断第 I 个学生,分得书的编号}
var j:integer;
begin
    for j:=1 to 5 do
        if not(j in flag) and (like[i,j]>0) then
            begin
                {第 j 本书未选过,且第 I 个学生喜爱第 j 本书}
                flag:=flag+[j]; {修改已选书编号集合,加入第 j 本书}
                book[i]:=j;      {记录第 I 个学生分得的书号}
            end
        if i=5 then print      {I=5,5 个学生都分到自己喜爱的书}

```

```

        else try(i + 1);
            I < 5, 继续搜索下一个学生可能分到书的情况
flag := flag - [j];    {后退一步, 以便查找下一种分配方案}
book[i] := 0;
end
end;
{ ===== main prg ===== }
begin
    flag := [ ];
    c := 0;
    try(1);
    readln
end.

```

运行结果为:

```

zhang:C
wang:A
liu:B
Zhao:D
qian:E

```

另外, 此题也可以用非递归的算法解。非递归算法的基本思想是用栈来存放被选中书的编号。设 dep 表示搜索深度, r 为待选书号, p 为搜索成功标志。算法表示如下:

```
procedure dfs_unrecursion;
```

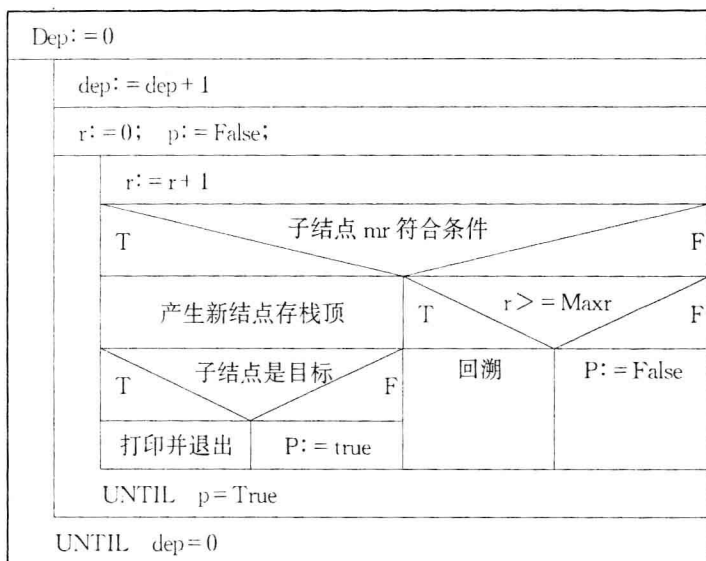


图 7-2

其中回溯的过程设计如下:

```
procedure 回溯;
```

```
    dep := dep - 1;
```

```
    if dep = 0 then p := true
```

```
        else 取回栈顶元素(出栈);
```

使用非递归算法,完整程序如下:

```
Program allot_book2(output);
```

```
type five = 1..5;
```

```
const like:array[five,five] of 0..1 =
```

```
    ((0,0,1,1,0),(1,1,0,0,1),(0,1,1,0,0),
     (0,0,0,1,0),(0,1,0,0,1));
```

```

name:array[five] of string[5] =
    ('zhang','wang','liu','zhao','qian');
var book:array[1..5] of 0..5;
    flag:set of five;
    c,dep,r:integer;
    p:boolean;

```

```

procedure print;
    var i:integer;
    begin
        inc(c);
        writeln('answer',c,':');
        for i:=1 to 5 do
            writeln(name[i]:10,':',chr(64+book[i]));
        end;

```

```

procedure backtracking;           {回溯过程}
    begin
        dep:=dep-1;               {退回上一步}
        if dep=0 then p:=true
            else begin             {栈顶元素出栈}
                r:=book[dep];
                flag:=flag-[r]
            end
        end;
    {===== main =====}
    begin
        flag:=[]; c:=0;dep:=0;

```

```

repeat
  dep := dep + 1;
  r := 0;
  p := false;
  repeat
    r := r + 1;
    if not(r in flag) and (like[dep,r] > 0) then
      begin
        {新产生的子结点满足条件}
        flag := flag + [r];    {新结点入栈}
        book[dep] := r;
        if dep = 5 then print   {搜索成功}
          else p := true
        end
      else if r >= 5 then backtracking
        else p := false
      until p = true;
    until dep = 0;
  readln
end.

```

上述程序运行产生结点的过程如下图 7-3 所示：

这是一棵搜索树。结点旁的编号为该子结点产生的顺序。从各子结点产生的顺序可以看出,这种搜索方法是,深度越大的结点越优先进行扩展,即先产生它的子结点。例如,产生结点(3)后,并不是继续产生与其深度相同(深度为 1)的结点(4),而是先产生其深度为 2 的子结点(3,1)。同样产生了结点(3,1)后,接着不是产生同深度的结点(3,2)或上一深度的结点,而是产生深度为 3 的子结点(3,1,2)。这是因为(3,2)的深度是 2,而结点(3,1,2)的深度是 3,(3,1,2)的深

度大,所以先扩展。

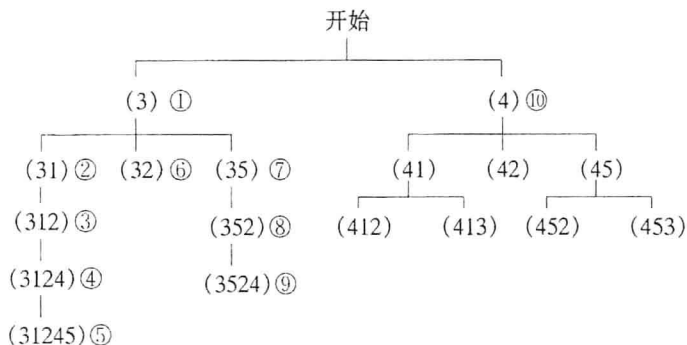


图 7-3

这种在搜索过程中,深度大的结点先进行扩展的算法,我们就称它为深度优先搜索法。英语称之为 Depth-First-Search,简称为 DFS 法。

上述算法实际是一个产生式系统。综合数据库就是一维数组 BOOK;产生规则是在 1~5 中选择与前面已选定的数不相同的数;控制策略就是深度大的结点先扩展,深度达到 5 即是目标结点。

§ 7.2 深度优先搜索基本算法(一)

从例 7.1 我们可以看出,深度优先搜索法有两个显著特点:

(1) 对已产生的结点按深度排序存储,深度大的先得到扩展,即先产生它的子结点;

(2) 深度大的结点是后产生的,但先得到扩展,即“后产生先扩展”。因此该算法应该用堆栈作为的主要数据结构存储产生的结点。