

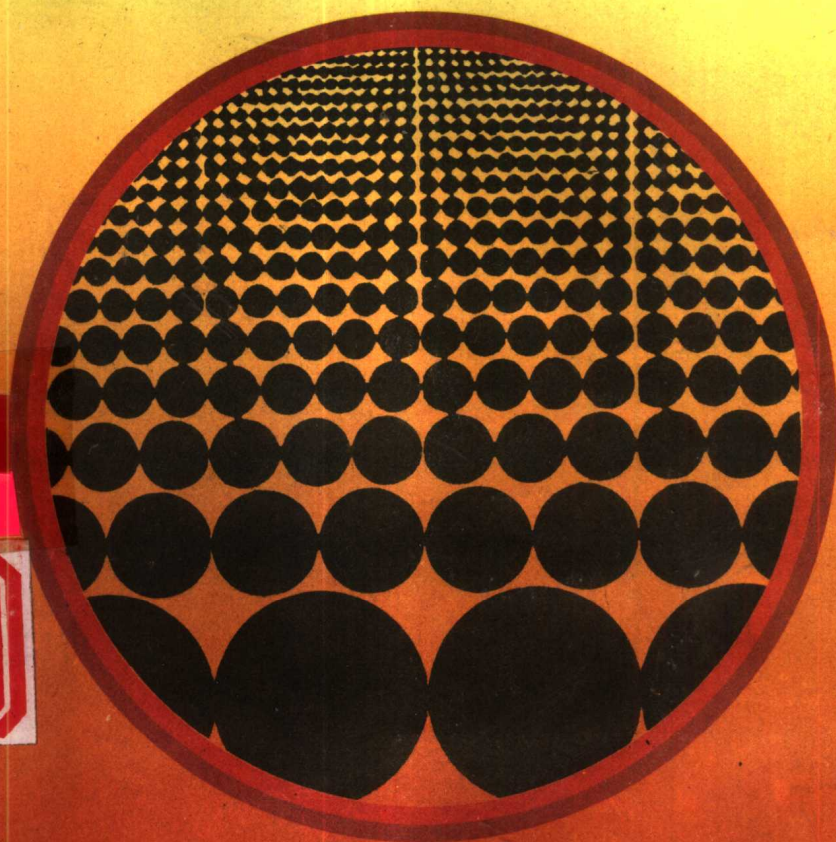
软件资格水平考试

热身训练

(三) 程序编制

吕志孔 编著

山东科学技术出版社



# 软件资格水平考试热身训练

(三)

程 序 编 制

吕志孔 编著

山东科学技术出版社

(鲁)新登字 05 号

软件资格水平考试热身训练

(三)

程序编制

吕志孔 编著

\*

山东科学技术出版社出版  
(济南市玉函路 邮政编码 250002)

山东省新华书店发行

山东新华印刷厂潍坊厂印刷

华光 IV 型计算机——激光汉字编辑排版系统排版

\*

787×1092 毫米 16 开本 12.75 印张 280 千字  
1992 年 5 月第 1 版 1992 年 5 月第 1 次印刷  
印数: 1—7,600

ISBN 7-5331-1019-6/G·151

定价: 6.80 元

# 前 言

计算机软件专业技术资格和水平考试已在全国统一举行,为配合这一考试,帮助广大软件人员做好应试准备和提高实际应用水平,我们编写了这套热身训练。

本套热身训练以程序员级和高级程序员级应试人员为读者对象,对我国自1987年以来的5年试题进行了归纳整理,列举复习要点,提出应试对策。这套书以强化训练为宗旨,以大量的题解和习题覆盖考试大纲,熔讲解、自测、练习于一炉。

本套热身训练共分4册。(一)基础知识;(二)计算机英语;(三)程序编制;(四)1989~1991年度试题及解答汇编。

(一) 基础知识:本册内容针对考试大纲中的软件知识、软件设计能力、硬件知识与计算机应用的综合基础知识部分。书中分3篇:硬件基础知识,软件基础知识,综合基础知识。每篇的第一章为引言,概述本篇内容,其他各章为专题讲解,均分为知识要点、试题分布、试题倾向与应试对策、题例与分析、练习等5节,其中以题例与分析占的比重较大。

(二) 计算机英语:本册内容针对考试中上午试题的外语部分。书中总结了历年英语试题的特点,列举大量例题和习题,对考试所涉及的语法、专业术语、阅读理解进行训练。

(三) 程序编制:本册内容针对考试中的下午试题部分。共分6章:流程图、FORTRAN程序、COBOL程序、PASCAL程序、C语言程序、CASL汇编语言程序。每一章都列出历年试题在本章内容上的分布情况,总结命题倾向和应试对策,然后列举大量例题进行分析讲解,并给出一定量的习题。

(四) 1989~1991年度试题及解答汇编:本册汇集了我国自1989至1991三年的程序员和高级程序员试题(包括1991年的初级程序员试题)及答案,可使读者对近年来的考试内容和考试特点获得一个总体认识。

本套热身训练适合于应试人员复习使用,也可供计算机专业师生参考。根据计算机的发展和考试的变化情况,这套书计划每隔两三年修订一次,欢迎广大读者多提宝贵意见,以便再版时修正。

《软件资格水平考试热身训练》编写委员会

1991年12月

# 目 录

<b>第一章</b>	<b>流程图</b> .....	(1)
1.1	试题分布 .....	(1)
1.2	倾向与对策 .....	(2)
1.3	题例与分析 .....	(3)
1.4	练习.....	(21)
<b>第二章</b>	<b>FORTRAN 程序</b> .....	(30)
2.1	试题分布.....	(30)
2.2	倾向与对策.....	(31)
2.3	题例与分析.....	(32)
2.4	练习.....	(52)
<b>第三章</b>	<b>COBOL 程序</b> .....	(64)
3.1	试题分布.....	(64)
3.2	倾向与对策.....	(65)
3.3	题例与分析.....	(66)
3.4	练习.....	(88)
<b>第四章</b>	<b>PASCAL 程序</b> .....	(96)
4.1	试题分布.....	(96)
4.2	倾向与对策.....	(97)
4.3	题例与分析.....	(97)
4.4	练习 .....	(113)
<b>第五章</b>	<b>C 语言程序</b> .....	(121)
5.1	试题分布 .....	(121)
5.2	倾向与对策 .....	(121)
5.3	题例与分析 .....	(122)
5.4	练习 .....	(140)
<b>第六章</b>	<b>CASL 汇编语言程序</b> .....	(148)
6.1	试题分布 .....	(148)
6.2	倾向与对策 .....	(148)
6.3	题例与分析 .....	(149)
6.4	练习 .....	(175)
附录 I	CASL 汇编语言文本 .....	(185)
附录 II	练习答案.....	(190)
	主要参考书目.....	(197)

# 第一章 流程图

## 1.1 试题分布

我国与日本历年流程图试题的分布如表 1.1、表 1.2、表 1.3 所示。

表 1.1 我国历年程序员级流程图试题分布

题号 内容	年度	1987	1988	1989	1990	1991
求整数					1	1
数列					2	
线性表		1				
排序				1		
密文解密			1			

表 1.2 我国历年高级程序员级流程图试题分布

题号 内容	年度	1988	1989	1990	1991
二分法求根		5			
递归函数					1
矩阵				4	2
求子集合				5	
字符串变换			2		
元素换位			4		
排序		3			
查找		4			
遍历二叉树			3		
击柱游戏				1	

表 1.3

日本历年流程图试题分布

题号 内容	1985	1986 春	1986 秋	1987 春	1987 秋	1988 春	1988 秋	1989 春
求素数	1							
求极值					1			
初等计算			2,3	1		3	1	3
方程求解		2						
直线回归							2	
频度统计					2			
数据检查	3			3				
字符串处理		3	1					1
图表制作	2			2		1		
判断					3			2
排序						2		
合并							3	
文件更新		1						

## 1.2 倾向与对策

由试题分布可以看出,流程图的命题内容多集中于一些常用算法与操作,尤其是初等计算。读者应从培养阅读框图能力、熟悉常用算法与操作等方面进行热身训练。

1. 审好“说明”, 弄清题意。当“说明”较长时, 应仔细分析, 并标出关键点, 分清已知条件及求解条件。

2. 细读框图, 确定算法。明确各变量的含义及初值, 变量除输入变量、中间变量、结果变量外, 还有循环控制变量, 与数据结构有关的数组变量、下标变量、记录变量、链表、树与文件等。另外, 常量的含义也要明白。

3. 赋好初值, 仔细推敲。初值 0 与 1 大相径庭, 切勿粗心。

4. 循环结构, 认真对待。要弄清预置部分、循环体、修正部分与检查部分; 多重循环还要分清内循环与外循环。

5. 先易后难, 注意检查。一般要填的内容往往是: 赋初值, 条件判断(如出错判断、循环结束判断、题目要求的判断等), 循环变量增值、减值, 算法或公式的操作(链表的

摘链、挂链、链指针的移动, 栈的进栈、出栈, 队列的入队、出队、插队等), 打印输出, 文件操作等。依次先填容易的, 并把握住极值、端点与特殊值, 最后统查一遍。

6. 熟悉算法, 掌握操作。流程图设计涉及许多算法、操作、像科学计算中的初等计算(平均值、标准差、质因数分解、最大公约数、误差计算、相关系数等), 方程求解(二分法、牛顿法、消去法等), 数值积分(梯形公式、辛普生公式等), 矩阵运算(矩阵的和、差、积及转置等)以及表查找(顺序查找、二分查找、区间查找等), 排序等操作。应尽量熟悉、掌握, 应试时就能事半功倍。

### 1.3 题例与分析

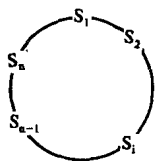
#### 题例 1.1(中 88 程下午题一)\*

阅读下列说明和流程图, 把应该填入其中①~⑤处的字句, 写在答卷的对应栏内(每处只填一个“a→b”型的字句), 使之成为完整的流程图。

#### 【说明】

本流程图是对某种简单密码文(密文)解密。密文由字符序列组成, 解密后产生的字符序列称为原文。解密算法如下:

把密文  $S_1S_2\cdots S_n$  按顺时针方向看成一个环, 如下所示。



解密时按读入的整数值  $KEY (KEY > 1)$ , 从  $S_1$  起顺时针计数, 当计数到第  $KEY$  个字符时, 取出该字符作为原文的第一个字符, 并把它从环中删去。接着从下一个字符起继续计数, 取出第  $KEY$  个字符作为原文的第二个字符, 并从环中删去。依次类推, 直至  $N$  个字符全部取完。由上述算法依次取出的字符序列即为原文。

例如, 当  $KEY=3$  时, 密文 NUI TP 的原文为 INPUT。

开始解密时, 密文存放在字符数组  $S$  中, 长度为  $N (N > 1)$ , 所得到的原文也存放在数组  $S$  中。为了从  $S(1)$  起依次存放原文字符, 在必要时部分未解密的字符需作适当的移动。

\* 括号内为题目的出处, 即中国 1988 年度程序员级下午试题一。其余括号类似。



**【流程图】**

**【分析】**

流程图中用变量  $M$  指向下一个待从环中取出字符的下标。其初值为 0，然后循环执行  $M+KEY \rightarrow M$ ，决定下一个待取字符的下标。若  $M$  的值不大于  $N$  且不等于 1，则将  $S(M)$  暂存于变量  $T$  中。由于原题要求所得到的原文仍存放在  $S$  中，并以  $S(I)$  开始存放，故有必要将部分未解密的字符按顺时针方向移动，以腾出地方存放暂存于  $T$  的字符。程序中使用变量  $K$  组成循环移动  $S$  中的字符， $K$  的初值应为  $M$ 。若  $K > I$ ，则移动  $S$  中的字符，即  $S(K-1) \rightarrow S(K)$ ，若  $K \leq I$ ，则循环结束，将  $T$  中的字符送入腾出的位置。不难看出，在流程图的前部，当  $M > N$  时，应对  $M$  作如下修正： $M - N + I - 1 \rightarrow M$ 。

**【答案】**

- ①  $0 \rightarrow M$
- ②  $M - N + I - 1 \rightarrow M$
- ③  $M \rightarrow K$
- ④  $S(K-1) \rightarrow S(K)$
- ⑤  $T \rightarrow S(I)$  或  $T \rightarrow S(K)$

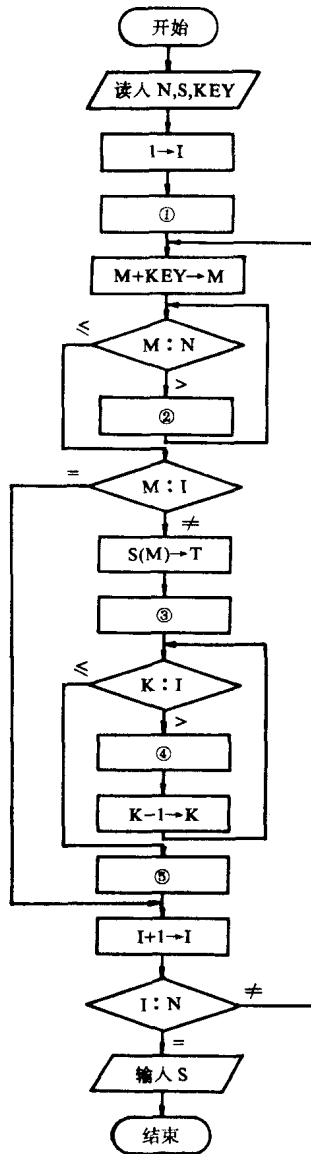
**题例 1.2** (中 88 高程下午题四)

阅读下列说明及流程图 A 和 B，回答问题 1 至问题 3，把解答填入答卷的对应栏内。

**【说明】**

流程图 A 和流程图 B 分别给出两种查找  $P$  是否是  $S$  的子序列的方法。对于两个长度分别为  $m$  和  $n$  ( $n \geq m > 0$ ) 的字符序列

$$P = p_0 p_1 \cdots p_{m-1}$$



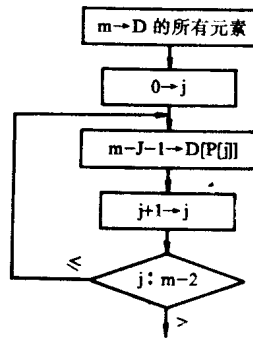
$S = s_0s_1 \dots s_{n-1}$

当 P 是 S 的子序列时, 输出 P 在 S 中首次出现的起始位置, 否则输出“失败”信息。

流程图 A 的查找方法是将 P 与 S 左端对齐, 自左至右逐个比较 P 与 S 的对应字符。如不相同, P 就向右移动一个字符位置。

流程图 B 的查找方法是将 P 与 S 左端对齐, 自右至左逐个比较 P 与 S 的对应字符。如不相同, P 就向右移动  $D[s_k]$  个字符位置。其中,  $s_k$  是发现不相同, 与字符  $P_{m-1}$  对应的 S 中的那个字符。

数组  $D[\text{字符}]$  的值用以下流程图所示的算法确定:



例如: 当 P 为 book 时,  $m = 4$ , D 的值如下表所示:

ch	D[ch]
b	3
o	1
k	4
其他字符	4

若 S 为 ipokebookw, 则查找过程如下:

位置	0	1	2	3	4	5	6	7	8	9
S	i	p	o	k	e	b	o	o	k	w
P	b	o	o	k						

失败,  $D[k] = 4$ , 右移 4 位

				b	o	o	k			
--	--	--	--	---	---	---	---	--	--	--

失败,  $D[o] = 1$ , 右移 1 位

					b	o	o	k		
--	--	--	--	--	---	---	---	---	--	--

成功, P 在 S 中的起始位置为 5

### 【问题 1】

填充流程图 A 和流程图 B 中的①~⑥, 使之成为完整的流程图(每处只填一个“a→b”或“a : b”型的字句)。

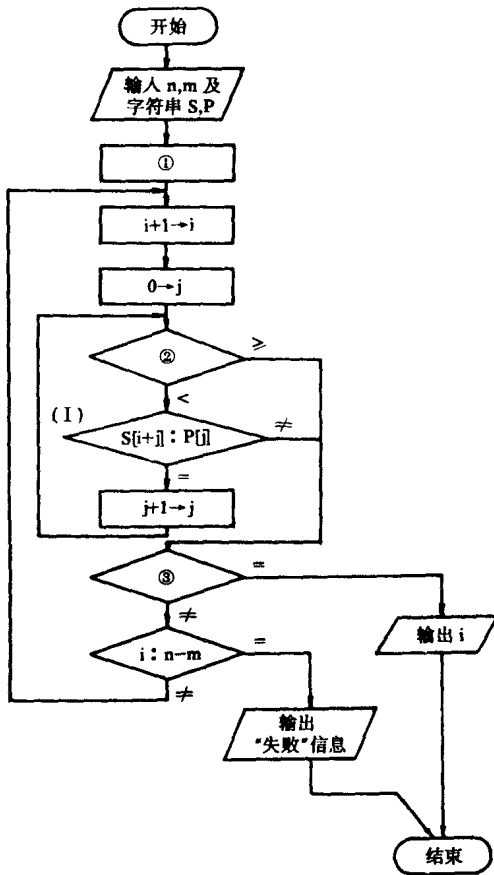
**【问题 2】**

在查找失败时,流程图 A 中(I)处的字符比较次数最少是多少?最多是多少?(用  $m$  和  $n$  表示)

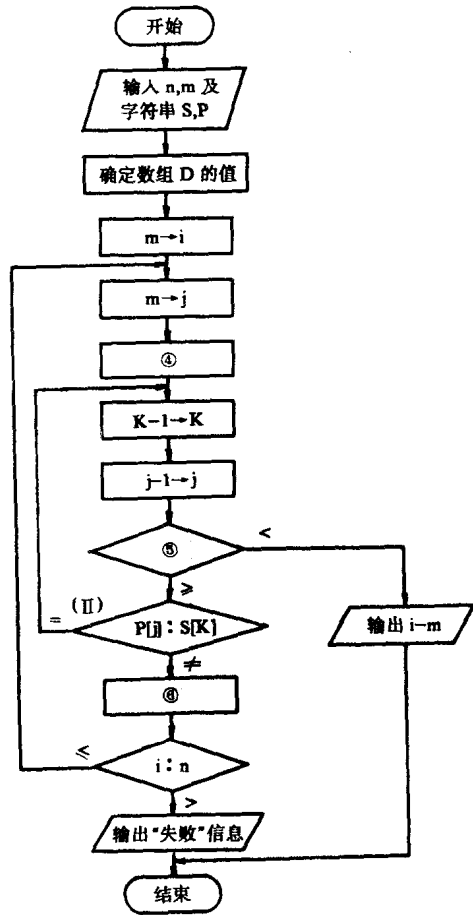
**【问题 3】**

在查找失败时, $S$  和  $P$  必须满足什么条件,才能使流程图 B 中(I)处字符比较的次数为  $\lfloor n/m \rfloor$ ? (其中  $\lfloor \cdot \rfloor$  为取整符号)

**【流程图 A】**



**【流程图 B】**



**【分析】**

在流程图 A 中,变量  $i$  指出移动的位置。由于在循环前要做  $i+1 \rightarrow i$ ,故  $i$  的初值应为  $-1$ 。变量  $j$  用作遍历字符串  $p$  的下标,其初值为  $0$ ,当  $j < m$  时,比较  $S[i+j]$  与  $P[j]$ 。若相等,则继续比较下一对字符;否则跳出内层循环。由于判断  $i:m$  与  $S[i+j]:P[j]$  共用一个出口,故在③处要判断控制来自哪一个判断;是全部比较成功还是有不同的字符。

当字符串  $P$  中的首字符  $P_0$  不出现在字符串  $S$  的  $S_0, S_1, S_2, \dots, S_{n-m}$  中时,流程图 A 中  $S[i+j]$  与  $P[j]$  的比较次数最少,共  $n-m+1$  次。当字符串  $P_0, P_1, \dots, P_{m-1}$  具有如下特点:

$$P[j]=S[i+j], P[m-1] \neq S[i+m-1]$$

其中,  $j=0, 1, 2, \dots, m-2, i=0, 1, 2, \dots, n-m$ 。此时, 流程图 A 中  $S[i+j]$  与  $P[j]$  的比较次数最多, 共  $m \times (n-m+1)$  次。

流程图 B 中, 变量  $i$  指出当前字符串  $P$  与  $S$  的位置关系, 当比较失败后,  $P$  向后移到  $i$  所指示的位置。变量  $j, k$  是比较  $P[j]$  与  $S[k]$  时所用的循环变量。 $j$  的初值应为  $m, k$  的初值应为  $i$ 。若比较  $P[j]$  与  $S[k]$  每次都成功, 出现循环时应有  $j < 0$  或  $k < i-m$ 。

当每次与  $P_{m-1}$  相比较的  $S$  中的字符都不出现在  $P_0, P_1, \dots, P_{m-2}$  中时, 流程图 B 中  $P[j]:S[k]$  比较的次数为  $\lfloor n/m \rfloor$ 。

### 【答案】

问题 1:

①  $-1 \rightarrow i$

②  $j: m$

③  $j: m$

④  $i \rightarrow k$

⑤  $j: 0$

⑥  $i + D[S[i-1]] \rightarrow i$

问题 2:

比较次数最少为  $n-m+1$ ; 最多为  $m \times (n-m+1)$ 。

问题 3:

每次与  $P_{m-1}$  相比较的  $S$  中的字符都不出现在  $P_0 P_1 P_2 \dots P_{m-2}$  中时, 流程图 B 中的比较次数为  $\lfloor n/m \rfloor$ 。

### 题例 1.3 (中 88 高程下午题五)

阅读下列说明和流程图, 回答问题 1 至问题 3, 把解答填入答卷的对应栏内。

### 【说明】

本流程图是一种奇特的用二分法求实系数多项式  $P(x)$  在区间  $[a, b]$  上的实根的流程图。精度为  $\text{eps} (\text{eps} > 0)$ 。设

$$P(x) = a_1 x^n + a_2 x^{n-1} + \dots + a_n x + a_{n+1} (n > 0)$$

### 【问题 1】

假定  $P(x)$  在  $[a, b]$  上有实根, 且  $P(a) * P(b) < 0$ , 填充流程图中的 ①~⑦, 使之成为完整的流程图 (每处只填一个“ $a \rightarrow b$ ”或“ $a: b$ ”型的字句)。

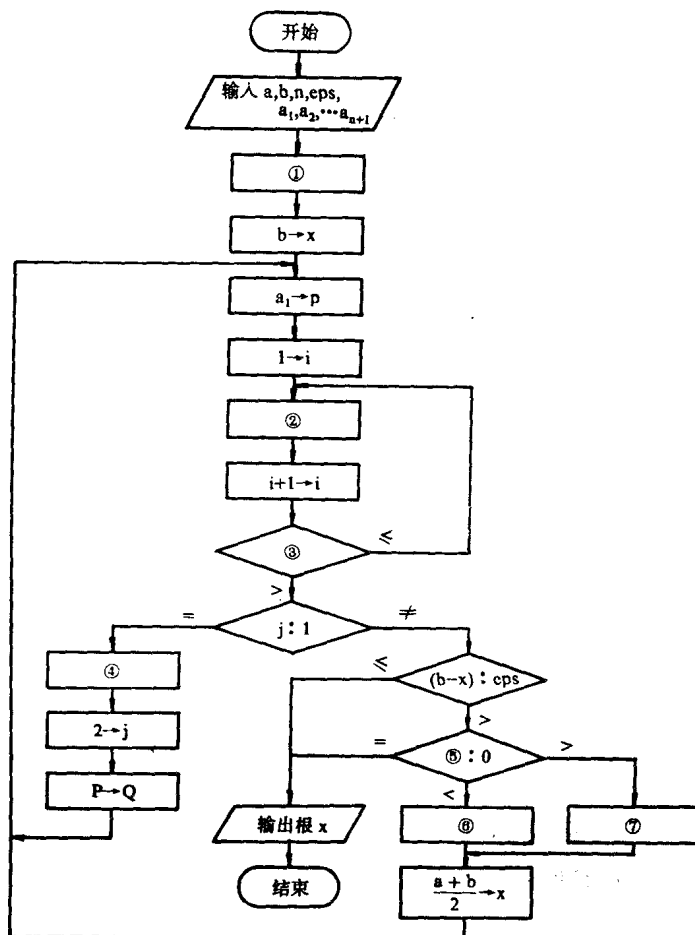
### 【问题 2】

假定  $P(x)$  在  $[a, b]$  上没有实根, 由流程图输出的  $x$  的值是什么?

### 【问题 3】

假定  $P(x)$  在  $[a, b]$  上有实根, 在什么情况下流程图输出的根是不正确的?

## 【流程图】



## 【分析】

设  $P(x)$  在  $[a, b]$  上有实根, 且  $P(a) * P(b) < 0$ 。用奇特二分法求  $P(x)$  在区间  $[a, b]$  上的实根。首先计算  $P(a)$  与  $P(b)$ , 然后在区间  $[a, b]$  上求多项式的根。若  $b-x < \text{eps}$ , 即  $x$  接近  $b$ , 则将  $x$  当作  $P(x)$  的根输出, 程序结束。若  $p(x) * p(b) < 0$  ( $x$  的初值为  $a$ ), 则缩小求解区间, 置  $a$  为  $x$ 。然后取  $x = \frac{a+b}{2}$ , 即  $x$  为新区间的中间点, 重复求解过程; 若  $P(x) * P(b) > 0$ , 则缩小求解区间, 置  $b$  为  $x$ 。然后取  $x$  为新区间的中间点, 重复求解过程; 若  $P(x) * P(b) = 0$ , 则表示  $x$  是多项式的根, 输出之, 程序结束。

不难看出, 程序一开始是计算  $P(x)$  的值且存入  $P$  中, 即  $p * x + a_{i+1} \rightarrow P$ 。下标  $i$  要与  $n$  比较。  $j$  的初值应为 1, 若  $j=1$  则  $a \rightarrow x, 2 \rightarrow j, P \rightarrow Q$ , 此时,  $Q=P(b)$ 。下一次循环再求  $P(a)$  的值。

对于上述的奇特二分法, 当  $P(x)$  没有  $[a, b]$  内的实根时, 程序将出现死循环,  $x$  的值永远是  $a$ 。即使  $P(x)$  在  $[a, b]$  上有实根, 但当  $P(b)=0$  且  $P(a) \neq 0$  时, 或  $P(a) * P(b) > 0$  时, 流程图输出的根都是不正确的。

**【答案】**

问题 1:

- ①  $1 \rightarrow j$
- ②  $p * x + a_{i+1} \rightarrow P$
- ③  $i : n$
- ④  $a \rightarrow x$
- ⑤  $P * Q$
- ⑥  $x \rightarrow a$
- ⑦  $x \rightarrow b$

问题 2:

x 的值为 a

问题 3:

$P(b) = 0$  且  $P(a) \neq 0$  或  $P(a) * P(b) > 0$

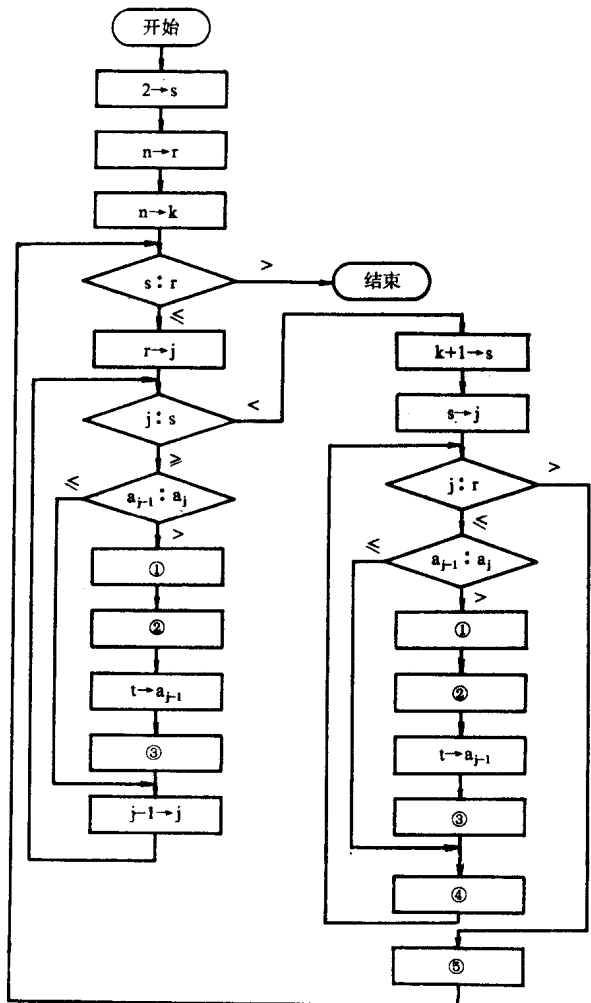
**题例 1.4** (中 89 程下午题一)

阅读下列说明和流程图, 把应填入其中①~⑤处的字句, 写在答卷的对应栏内(每处只填一个“a→b”型的句子), 使之成为完整的流程图。

**【说明】**

流程图实现了一个将一组无序数  $a_1, a_2, \dots, a_n$  排成递增序列的算法。该算法在  $s-1$  到  $r$  (初值为 1 到  $n$ ) 的区间内, 先冒泡后下沉进行排序, 直到该区间为空。图中变量  $k$  用来指出一次冒泡或下沉后  $a_1 \sim a_{k-1}$  或  $a_k \sim a_n$  已排序。

**【流程图】**



### 【分析】

流程图实现的算法,在  $s-1$  到  $r$  (初值为 1 和  $n$ ) 的区间  $a_{s-1}, a_{s-2}, \dots, a_{r-1}, a_r$  内,先冒泡后下沉进行排序,直至该区间为空。冒泡时,  $j$  为循环变量,当  $a_{j-1} > a_j$  时,应交换位置,流程中使用了临时变量  $t$  交换  $a_{j-1}, a_j$ 。变量  $k$  用来指出一次冒泡后  $a_1, a_2, \dots, a_{k-1}$  已排序。下沉时,工作过程同冒泡时类似,所不同的是循环变量  $j$  的变化规律。 $j$  的初值为  $k+1$ , 终值为  $r$ 。每次下沉后,  $j$  应向后移动一个位置,故应有  $j+1 \rightarrow j$ 。当一次下沉完毕后,只要还有未排好序的元素,就应继续排序,即开始下一次冒泡,冒泡区间为  $(S, K-1)$ 。

### 【答案】

- ①  $a_j \rightarrow t$
- ②  $a_{j-1} \rightarrow a_j$
- ③  $j \rightarrow k$
- ④  $j+1 \rightarrow j$
- ⑤  $k-1 \rightarrow r$

### 题例 1.5 (中 89 高程下午题三)

阅读下列说明和流程图,回答问题 1 和问题 2,把解答填入答卷的对应栏内。

### 【说明】

流程图用来实现中序遍历二叉树的算法,二叉树存放在数组  $tree$  中,每个数组元素存放树中的一个结点,每个结点的形式如下所示:

值	左 指 针	右 指 针
---	-------	-------

分别用  $tree[i].v, tree[i].l, tree[i].r$  来表示第  $i$  个结点的“值”、“左指针”、“右指针”。其中左、右指针的值为所指结点在数组中的下标,若指针的值为零,表示它指向空树。图中指针  $root$  用以指向二叉树的根结点。

### 【问题 1】

填充流程图中的①~③,使其按中序遍历二叉树(每处只填一个“ $a \rightarrow b$ ”或“ $a : b$ ”型的字句)。

### 【问题 2】

将流程图中的(A)框移至哪个位置(用图中所示的 I ~ K 回答),使该流程图的算法从中序遍历变成前序遍历。

**【流程图】**

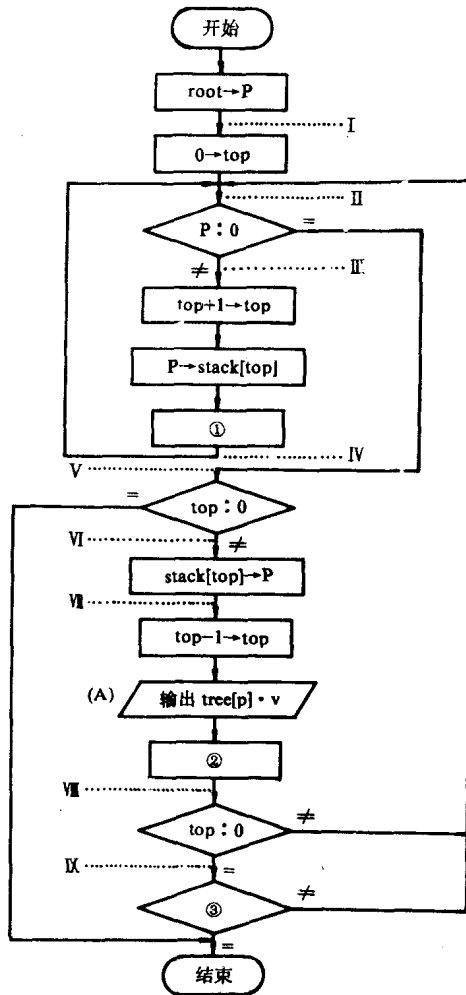
**【分析】**

二叉树的遍历次序是递归定义的。中序遍历二叉树,首先向左下降进入根的左子树,在左子树中继续下降,直至遇到叶结点,访问该结点,然后向上访问其父结点,访毕,再访右子树,此时遍历了最左边的一个子树,之后上访根结点、右子树。所以,要设计一个实现二叉树遍历的非递归算法,在向左下降的过程中保存结点的地址,以便当周游完一个子树后能顺利转移到右子树继续遍历下去。通常是用栈来完成此项任务的。在流程图中用到了栈 stack,其指针为 top,由中序遍历的上述分析可知,①处应填  $tree[p] \cdot l \rightarrow P$ ,表示首先遍历左子树,然后算法开始退栈,并访问根结点。此时应开始遍历右子树,故②处应填  $tree[p] \cdot r \rightarrow P$ 。算法结束条件应是当前子树为空,即③处应填  $P : 0$ 。

前序遍历二叉树的次序是:先访问根,再向左下降进入根的左子树,在其中又先访根,再向左下降进入根的左子树,如此做下去,直至遇到叶结点,也即遍历完了某一根的左子树,然后再向上,去遍历此根的右子树。具体实现时,只需将流程图中的 A 框移至 II 处,即先访根结点,便使得原中序遍历变为前序遍历。

**【答案】**

问题 1:





①  $tree[p] \cdot l \rightarrow p$

②  $tree[p] \cdot r \rightarrow p$

③  $P : O$

问题 2: 将 A 框移至 II 处

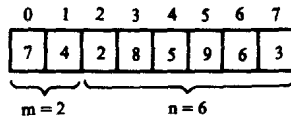
题例 1.6(中 89 高程下午题四)

阅读下列说明和流程图, 回答问题 1 至问题 3, 把解答填入答卷的对应栏内。

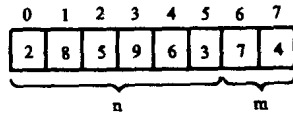
**【说明】**

在一维数组 A 中存放着  $m+n$  个整数( $m, n \geq 1$ )。流程图给出了一种算法, 使 A 中前  $m$  个元素( $A[0] \sim A[m-1]$ )与后  $n$  个元素( $A[m] \sim A[m+n-1]$ )互换位置, 并保持其各自原有的内部顺序。

例如, 若 A 为



则互换后 A 的内容变成



**【问题 1】**

若  $m=3, n=4$ , 数组 A 中各元素的值依次为 5, 7, 4, 8, 1, 2, 3, 则执行该流程图后, 变量 t 和 x 的值是什么?

**【问题 2】**

若  $m=6, n=4$ , 数组 A 中各元素的值依次为 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 则执行该流程图后, 将输出什么(每次输出的结果写成一行)?

**【问题 3】**

$m, n$  满足什么条件时, 该流程图中虚线部分的 k 循环只执行一遍。