

全国计算机应用软件人员任职  
资格考试电视函授教学丛书

# 1987-1990年 软件人员 水平考试 试题分析

李 维 主编

北京航空航天大学出版社

# 1987—1990年软件人员水平 考试试题分析

李 维 主编

北京航空航天大学出版社

## 内 容 提 要

本书是“全国计算机应用软件人员水平考试”1987—1990年高级程序员级上、下午试题及程序员级下午部分试题的分析与解答。内容有：硬件知识、软件知识、计算机应用的综合基础知识（其中包括离散数学、事务处理、实时处理、计算机辅助系统）以及程序设计方面（包括 BASIC、FORTRAN、C、PASCAL、COBOL语言及程序流程图）的试题分析。由参加过水平考试辅导、学员反映好的教师编写。步骤清楚、可读性好。本书可作为电视教学、函授教学及全国各地举办的有关辅导班的教材和参考书。

### 1987—1990年软件人员水平考试试题分析

1987—1990 NIAN

RUANJIAN RENYUAN SHUIPING KAOSHI SHITI FENXI

李 维 主编

责任编辑 马晓虹

\*

北京航空航天大学出版社出版

新华书店总店科技发行所发行 各地新华书店经售

北京密云华都印刷厂印装

\*

787×1092 1/16 印张：15.75 字数：403 千字

1991年3月第一版 1991年3月第一次印刷 印数：15000册

ISBN 7-81012-230-4/TP·042

定价：7.00元

## 前 言

本书是“全国计算机应用软件人员水平考试电视教学辅导丛书”之一，适合于程序员和高级程序员复习、考试之用。亦可作为各类应用软件人员学习计算机软件技术基础的辅导材料。

全书根据《中国计算机应用软件人员水平考试的考试大纲》规定的内容，对1987—1990年高级程序员级上、下午试题及程序员级下午部分试题进行分析说明和作出解答。全书共分四章：第一章为软件知识的试题分析和解答，其中包括数据结构、程序语言、语言处理程序、文件系统、操作系统、数据库系统、网络软件、软件工程；第二章为硬件知识的试题分析和解答，其中包括计算机组成、存储器系统、外围设备、系统性能评价，体系结构；第三章为计算机应用的综合基础知识的试题分析与解答，其中包括数学、数值计算、事务处理、实时处理、计算机辅助系统；第四章为程序设计的试题分析和解答，其中包括程序设计基础、FORTRAN程序设计、C程序设计、PASCAL程序设计、COBOL程序设计、CASL程序设计。

鉴于上述试题未将考试大纲中的内容全部覆盖到，我们根据近几年国内外有关的文献资料，选编了一部分试题，使本书内容更为充实、全面。这部分试题在本书中均以“\*”标记。

我们希望本书的出版能有助于读者提高分析问题和解决问题的能力，对于复习、考试有所帮助。

本书第一、二、四章由李维编写。第三章及第四章4.3节试题15、16，4.4节试题1~4，4.7节试题1由许耀昌组织编写。许耀昌还审阅了部分书稿。

由于时间限制，加上作者水平有限，书中一定存在不少缺点，希望读者提出宝贵意见，以便使本书在今后修订、再版时能及时改正。

作 者

1990年10月

# 目 录

## 第一章 软件知识

|                 |        |
|-----------------|--------|
| 1.1 数据结构.....   | ( 1 )  |
| 1.2 程序语言.....   | ( 8 )  |
| 1.3 语言处理程序..... | ( 9 )  |
| 1.4 文件系统.....   | ( 11 ) |
| 1.5 操作系统.....   | ( 12 ) |
| 1.6 数据库系统.....  | ( 19 ) |
| 1.7 网络软件.....   | ( 23 ) |
| 1.8 软件工程.....   | ( 27 ) |

## 第二章 硬件知识

|                |        |
|----------------|--------|
| 2.1 计算机组成..... | ( 36 ) |
| 2.2 存储器系统..... | ( 41 ) |
| 2.3 外围设备.....  | ( 46 ) |
| 2.4 系统性能.....  | ( 47 ) |
| 2.5 体系结构.....  | ( 53 ) |

## 第三章 计算机应用的综合基础知识

|                  |        |
|------------------|--------|
| 3.1 数学.....      | ( 55 ) |
| 3.2 数值计算.....    | ( 66 ) |
| 3.3 实时处理.....    | ( 68 ) |
| 3.4 事务处理.....    | ( 70 ) |
| 3.5 计算机辅助系统..... | ( 75 ) |

## 第四章 程序设计

|                     |         |
|---------------------|---------|
| 4.1 事务处理流程图.....    | ( 80 )  |
| 4.2 算法流程图.....      | ( 104 ) |
| 4.3 FORTRAN程序.....  | ( 134 ) |
| 4.4 COBOL程序.....    | ( 171 ) |
| 4.5 C程序.....        | ( 199 ) |
| 4.6 PASCAL程序.....   | ( 210 ) |
| 4.7 CASL汇编语言程序..... | ( 238 ) |
| 附录 CASL汇编语言文本.....  | ( 245 ) |

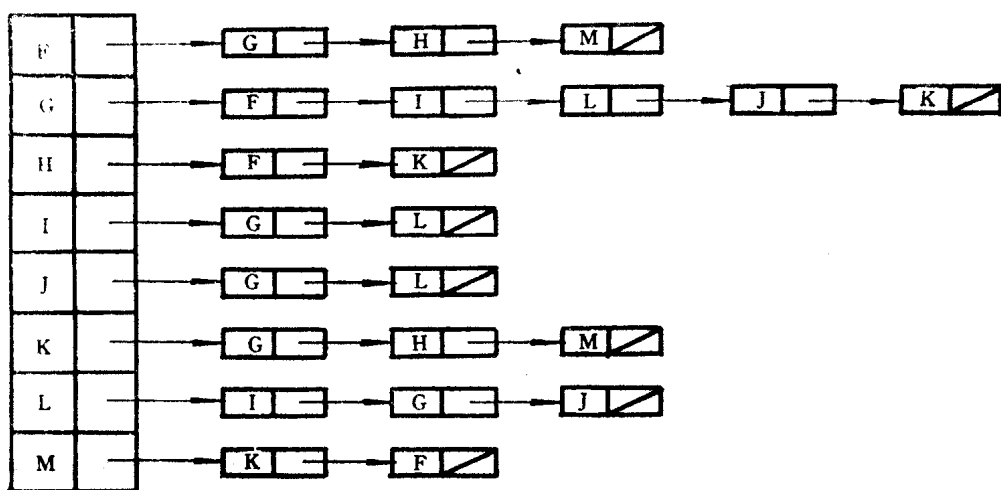
# 第一章 软件知识

## 1.1 数据结构

### 试题1

从供选择的答案中选出应填入下列叙述中的  内的正确答案，把编号写在答卷的对应栏内。

已知某图的邻接表为

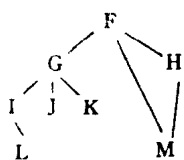


- ① 此邻接表所对应的无向图为  A  。
- ② 此图的由 F 开始的深度优先遍历为  B  。
- ③ 此图的由 F 开始的深度优先遍历的支撑树 (spanning tree) 为  C  。
- ④ 此图的由 F 开始的广度优先遍历为  D  。
- ⑤ 此图的由 F 开始的广度优先遍历的支撑树为  E  。

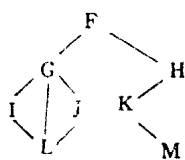
[供选择的答案]

A:

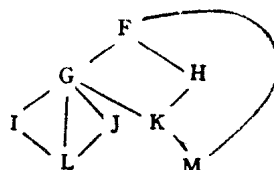
①



②



③

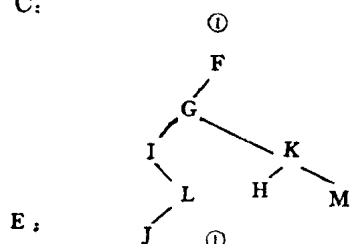


B, D:

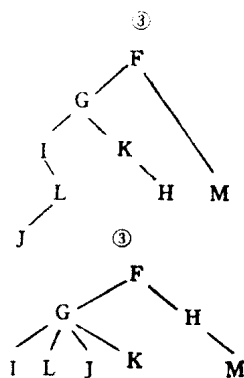
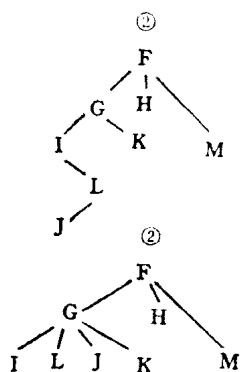
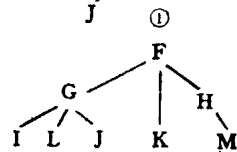
- |                   |                   |
|-------------------|-------------------|
| ① F G I L J M K H | ② F G I L J K H M |
| ③ F G I L J K M H | ④ F G H M I L J K |

⑤ F G H I L J K M      ⑥ F G H M K I L J

C:



E:



[试题分析]

A:

顶点i的邻接表是与顶点i邻接的所有顶点组成的表。

从试题中的邻接表看出:

顶点F与G, H, M相链。

顶点G与F, I, L, J, K相链。

⋮

⋮

依此类推。

因此, 只有③满足此邻接表。

B, C, D, E:

深度优先 (depth-first) 搜索法的遍历过程是: 假设从图  $G = (V, E)$  中某一顶点  $V_0$  出发, 在访问了任意一个和  $V_0$  邻接的顶点  $W_1$  之后, 再从  $W_1$  出发, 访问和  $W_1$  邻接且未被访问过的任意顶点  $W_2$ 。然后, 从  $W_2$  出发进行如上的访问。重复这种访问, 直到一个所有邻接点都被访问过的顶点为止。接着退回到尚有邻接点未被访问过的顶点, 再从该顶点出发, 重复上述搜索过程, 直到所有的被访问过的顶点的邻接点已被访问为止。

设  $E$  为图  $G$  中所有边的集合, 从图中任一顶点出发遍历图时, 必定将  $E$  分成两个集合  $T$  和  $B$ , 其中  $T$  是遍历图时所走边的集合,  $B$  则是剩余边的集合。这时,  $T$  和图  $G$  中所有顶点一起构成图  $G$  的极小连通子图。它也是一棵支撑树 (也称为生成树)。并且由深度优先搜索得到的是深度优先生成树; 由广度优先搜索得到的是广度优先生成树。

广度优先 (breadth-first) 搜索法的过程为: 从某顶点  $V_0$  出发, 在访问了  $V_0$  之后, 依次访问  $V_0$  的各个未曾访问过的邻接点, 然后分别从这些邻接点出发广度优先搜索遍历图, 直至图中尚有顶点未被访问, 则另选图中一个未曾访问的顶点作起点, 重复上述过程, 直到图中所有顶点都被访问过为止。

[答案]

A: ③      B: ②      C: ①      D: ④      E: ②

## 试题2

从供选择的答案中选出应填入下列叙述中的| |内的正确答案,把编号写在答卷的对应栏内。

二叉树是空,或由一个根结点和左、右两棵子树组成,其中每棵子树是二叉树。二叉树有多种形式:| A |是查找二叉树;| B |是平衡二叉树;二叉树中的所有结点个数为 $2^k - 1$  ( $k$ 是树的层数)时,是丰满二叉树。图1~图5是一些二叉树,其中图| C |是查找二叉树、图| D |是平衡二叉树而不是丰满二叉树、图| E |是丰满二叉树的实例。

### [供选择的答案]

A, B: ① 二叉树中每个结点的两棵子树的高度差不大于1。

② 二叉树中每个结点的两棵子树的高度差等于1。

③ 二叉树中每个结点的两棵子树是有序的。

④ 二叉树中每个结点有两棵非空子树,或有两棵空子树。

⑤ 二叉树中每个结点的关键字值大于其左非空子树(如果存在的话)所有结点的关键字值,且小于其右非空子树(如果存在的话)所有结点的关键字值。

⑥ 二叉树中的所有结点个数是 $2^{k-1} - 1$ ,其中 $k$ 是树的层数。

⑦ 二叉树中的所有结点,如果不存在非空左子树,则不存在非空右子树。

C, D, E: ① 图1

② 图2

③ 图3

④ 图4

⑤ 图5

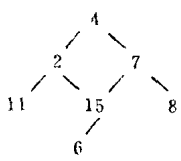


图 1

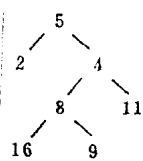


图 2

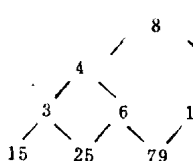


图 3

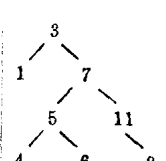


图 4

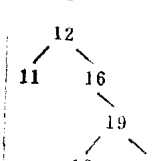


图 5

### [试题分析]

查找二叉树也称为排序二叉树 (binary sort tree), 它或者是一棵空树, 或者是具有下列性质的二叉树: (1) 若它的左子树不空, 则左子树上所有结点的值均小于它的根结点的值; (2) 若它的右子树不空, 则右子树上所有结点的值均大于等于它的根结点的值; (3) 它的左、右子树也分别为查找二叉树。通常, 查找二叉树是依次输入一个数据元素的序列构造而成的。具体构造过程是: 每读入一个元素, 建立一个新的结点, 按下列原则插入到二叉查找树中去: 若查找二叉树是空, 则新结点为查找二叉树的根结点; 若查找二叉树非空, 则将新结点的值和根结点的值比较, 若小于, 则插入到左子树中去; 否则插入到右子树中去。显然这个构造过程是递归的。

平衡二叉树 (balanced binary tree) 又称AVL树, 它或者是一棵空树, 或者是具有下列性质的二叉树: 它的左子树和右子树都是平衡二叉树, 且左子树和右子树的深度之差的绝对值不超过1。

### [答案]

A: ⑤    B: ①    C: ⑤    D: ①    E: ③

### 试题3

图1所示是个边带权的有向图G。请从供选择的答案中选出应该填入下列| |中的关于有向图G的正确答案，并将编号写在答卷的对应栏内。

有向图G的相邻矩阵表示为| A |，邻接表法表示如| B |所示。图2所示是图G的一棵| C |。| D |是有向图G的结点的一个拓扑序列。| E |是有向图G中从结点 $V_1$ 到结点 $V_6$ 的最短路径。

[供选择的答案]

A:

$$\textcircled{1} \quad A = \begin{bmatrix} 0 & 3 & 0 & 2 & 9 & 0 \\ 0 & 0 & 8 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 2 & 9 & 0 \\ 3 & 0 & 8 & 0 & 2 & 0 \\ 0 & 8 & 0 & 0 & 2 & 2 \\ 2 & 0 & 0 & 0 & 1 & 0 \\ 9 & 2 & 2 & 1 & 0 & 5 \\ 0 & 0 & 2 & 0 & 5 & 0 \end{bmatrix}$$

$$\textcircled{2} \quad A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

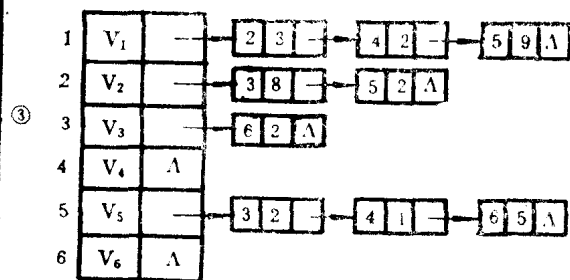
$$\textcircled{3} \quad A = \begin{bmatrix} 0 & 3 & 0 & 2 & 9 & 0 \\ 3 & 0 & 8 & 0 & 2 & 0 \\ 0 & 8 & 0 & 0 & 2 & 2 \\ 2 & 0 & 0 & 0 & 1 & 0 \\ 9 & 2 & 2 & 1 & 0 & 5 \\ 0 & 0 & 2 & 0 & 5 & 0 \end{bmatrix}$$

$$\textcircled{4} \quad A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

B:

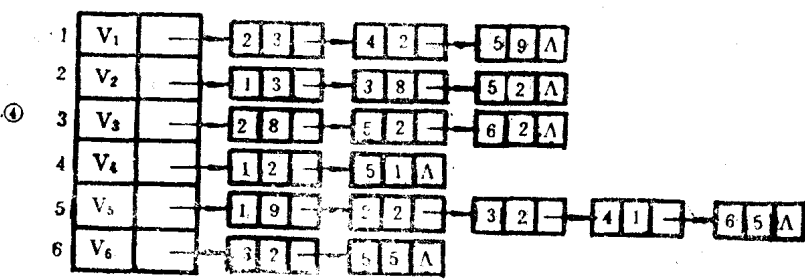
①

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 1 | 4 | 2 |
| 1 | 5 | 9 |
| 2 | 3 | 8 |
| 2 | 5 | 2 |
| 3 | 6 | 2 |
| 5 | 3 | 2 |
| 5 | 4 | 1 |
| 5 | 6 | 5 |



②

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 3 | 8 |
| 5 | 3 | 2 |
| 1 | 4 | 2 |
| 5 | 4 | 1 |
| 1 | 5 | 9 |
| 2 | 5 | 2 |
| 3 | 6 | 2 |
| 5 | 6 | 5 |



C: ① 从 $V_1$ 出发的广度优先生成树;

② 从 $V_1$ 出发的深度优先生成树;

③ 从 $V_2$ 出发的广度优先生成树;

④ 从 $V_2$ 出发的深度优先生成树;

- D, E: ①  $(V_1, V_2, V_3, V_4, V_5, V_6)$   
 ②  $(V_1, V_4, V_5, V_6, V_3, V_2)$   
 ③  $(V_1, V_2, V_6, V_3, V_5, V_4)$   
 ④  $(V_1, V_4, V_3, V_5, V_2, V_6)$

- ⑤  $(V_1, V_2, V_5, V_3, V_6)$   
 ⑥  $(V_1, V_2, V_3, V_6)$   
 ⑦  $(V_1, V_4, V_5, V_6)$   
 ⑧  $(V_1, V_5, V_6)$

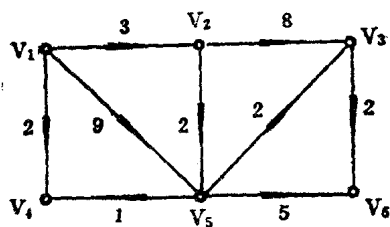


图1 有向图G

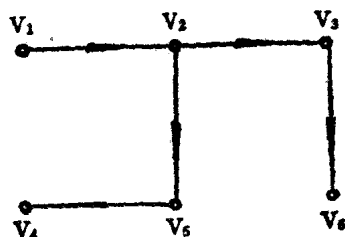


图2 生成树

### [试题分析]

对于有向图  $G = (V, R)$ , 其邻接矩阵  $A$  的定义为:

$$A[i, j] = \begin{cases} W_{ij} & \text{如果 } \langle V_i, V_j \rangle \in VR \\ 0 & \text{否则} \end{cases}$$

其中  $W_{ij}$  是弧  $\langle V_i, V_j \rangle$  上所带的权。

邻接表是图的一种链式存贮结构。在邻接表中, 对图中每个顶点建立一个单链表, 第  $i$  个单链表中的结点表示以顶点  $V_i$  为尾的弧。每个结点由三个域组成: 邻接点域(adjvex)指示与顶点  $V_i$  邻接的点在图中的位置, 链域(nextarc)指示下一条弧的结点, 数据域(info)存贮和弧相关的权值等。每个链表上附设一个表头结点, 在表头结点中, 除了设有链域(firstarc)指向链表中第一个结点之外, 还没有存储顶点  $V_i$  的名或其它有关信息的数据域(vexdata)。这些表头结点通常以顺序结构的形式存储, 以便随机访问任一顶点的链表。

生成树的概念见试题2的分析。

拓扑排序是有向图的一种重要运算。对于有向图  $G = (V, E)$ ,  $V$  中结点的线性序列  $(V_{i1}, V_{i2}, \dots, V_{in})$  称为一个拓扑序列, 若此结点序列满足如下条件: 若有向图  $G$  中从结点  $V_i$  到结点  $V_j$  有一条路径, 则在序列中结点  $V_i$  必在结点  $V_j$  之前。其具体实现算法是:

- (1) 在有向图中选一个没有前驱的顶点(即该顶点的入度为0), 且输出之。
- (2) 从图中删除该顶点和所有以它为尾的弧。

重复上述两步, 直至全部顶点均已输出, 或者图中没有无前驱的顶点为止。后一种情况则说明有向图中存在环。例如, 对于图6所示的有向图, 其拓扑排序为  $V_1, V_2, V_5, V_3, V_6, V_4$ 。

求有向图  $G$  中一对顶点  $\langle V_i, V_j \rangle$  之间的最短路径, 可采用Floyd提出的算法, 该算法以有向图的带权邻接矩阵  $A$  为基础, 其基本思想是: 如果从  $V_i$  到  $V_j$  有弧, 则从  $V_i$  到  $V_j$  存在一条长度为  $A[i, j]$  的路径, 该路径不一定是最短路径, 尚需进行  $n$  次试探。首先考虑路径  $(V_i, V_1, V_j)$  是否存在。如果存在, 则比较  $\langle V_i, V_j \rangle$  和  $\langle V_i, V_1, V_j \rangle$  的路径长度, 取长度较短者为从  $V_i$  到  $V_j$  的中间顶点的序号不大于1的最短路径。假如在路径上再增加一个顶点  $V_2$ , 也就是说, 如果  $\langle V_i, \dots, V_2 \rangle$  和  $\langle V_2, \dots, V_j \rangle$  分别是当前找到的中间顶点的序

号不大于1的最短路径,那么 $(V_1, \dots, V_2, \dots, V_j)$ 就有可能是从 $V_1$ 到 $V_j$ 的中间顶点的序号不大于2的最短路径。将它和已经得到的从 $V_1$ 到 $V_j$ 中间顶点序号不大于1的最短路径相比较,从中选出中间顶点序号不大于2的最短路径之后,再增加一个顶点 $V_3$ ,继续进行试探。依次类推。在一般情况下,若 $(V_1, \dots, V_k)$ 和 $(V_k, \dots, V_j)$ 分别是从小 $V_1$ 到 $V_k$ 和从 $V_k$ 到 $V_j$ 的中间顶点序号不大于 $k-1$ 的最短路径,则将 $(V_1, \dots, V_k, \dots, V_j)$ 和已经得到的从 $V_1$ 到 $V_j$ 且中间顶点序号不大于 $k-1$ 的最短路径相比较,其长度较短者便是从 $V_1$ 到 $V_j$ 的中间顶点序号不大于 $k$ 的最短路径。这样,在经过 $n$ 次比较后,最后求得的必是从 $V_1$ 到 $V_j$ 的最短路径。例如,对于图6所示的有向图,从 $V_1$ 到 $V_6$ 的最短路径是 $(V_1, V_2, V_5, V_3, V_6)$ 。

[答案]

A: ①    B: ③    C: ②    D: ③    E: ⑤

#### 试题4

从供选择的答案中选出应填入下列叙述中的|     |内的正确答案,把编号写在答卷的对应栏内。

在查找算法中,可用平均查找长度(记为ASL)来衡量一个查找算法的优劣,其定义为:

$$ASL = \sum_{i=1}^n P_i C_i$$

此处 $P_i$ 为表中第 $i$ 个记录被查找的概率, $C_i$ 为查到第 $i$ 个记录时已进行的和关键字比较的次数, $n$ 为表中原有记录数。

以下叙述中均假定每一个记录被查找的概率相等,即 $P_i = 1/n$  ( $i = 1, 2, \dots, n$ )。

当表中的记录连续存放在一个一维数组中时,可采用顺序查找与折半查找方法(折半查找要求表是按关键字有序的)。顺序查找时的ASL为|    A    |,折半查找时的ASL为|    B    |。记录的关键字有序时,用二叉排序树查找记录,在最坏的情况下,ASL为|    C    |。当二叉排序树是一棵平衡树时,ASL为|    D    |。在平衡树上删除一个结点后,可以通过旋转使其平衡,最坏的情形下需|    E    |次旋转。

[供选择的答案]

A ~ E:    ①  $O(1)$     ②  $O(\log_2 n)$     ③  $O((\log_2 n)^2)$     ④  $O(n \log_2 n)$     ⑤  $O(n)$   
              ⑥  $O(n^2)$

[试题分析]

在顺序查找时, $C_i$ 取决于所查记录在表中的位置。如查找记录1时,仅需比较1次;而查找记录 $n$ 时,则需比较 $n$ 次。一般地, $C_i$ 等于 $i$ 。

因此,假设每一个记录被查找的概率相等,即 $P_i = \frac{1}{n}$ ,  $i = 1, \dots, n$ ,则顺序查找的平均查找长度为:

$$\begin{aligned} ASL &= \sum_{i=1}^n P_i C_i \\ &= \frac{1}{n} \sum_{i=1}^n i \end{aligned}$$

$$= \frac{1}{n} (1 + 2 + \dots + n - 1 + n)$$

$$= \frac{n+1}{2}$$

顺序查找法与其它查找法相比，其缺点是ASL较大，特别是当n较大时，查找效率很低。

在折半查找中，找到表中任一元素的过程就是走了一条描述折半查找的二叉树中的从根结点到与该元素相应的结点的路径，所进行的和关键字比较的次数为该结点在二叉树中的层数。为了讨论方便，假定元素个数 $n = 2^h - 1$ ，即 $h = \log_2(n+1)$ ，则描述折半查找的二叉树的深度为h的满二叉树。树中层次为1的结点有1个，层次为2的结点有2个，……，层次为h的结点有 $2^{h-1}$ 个。原题假设表中每个元素的查找概率相等( $P_i = 1/n$ )，因此折半查找的平均查找长度为：

$$\begin{aligned} \text{ASL} &= \sum_{i=1}^n P_i C_i = \frac{1}{h} \sum_{j=1}^h j \cdot 2^{j-1} \\ &= \frac{1}{h} \left( \sum_{k=0}^{h-1} 2^k + 2 \sum_{k=0}^{h-2} 2^k + \dots + 2^{h-1} \sum_{k=0}^0 2^k \right) \\ &= \frac{1}{h} [h \cdot 2^h - (2^0 + 2^1 + \dots + 2^{h-1})] \\ &= \frac{1}{h} [(h-1)2^h + 1] \\ &= \frac{1}{n} [(n+1)(\log_2(n+1) - 1) + 1] \\ &= \frac{n+1}{n} \log_2(n+1) - 1 \end{aligned}$$

当n较大时，近似为 $\log_2(n+1) - 1$ 。可见折半查找的效率较高，但只能适用于有序表。

二叉排序树的查找过程类似于折半查找。但含有n个结点的二叉排序树的平均查找长度和树的形态有关。当表中的元素按关键字有序时，构成的二叉排序树蜕化成单支树，树的深度为n，其平均查找长度同顺序查找时一样，为 $(n+1)/2$ 。这是二叉排序树查找的最坏情况。

在平衡二叉排序树上进行查找的过程和排序树上的相同，在查找过程中和关键字进行比较的次数不超过树的深度。那末，含有n个关键字的平衡二叉树的最大深度是多少呢？我们首先分析深度为h的平衡二叉树所具有的最少结点数。假设用 $N_h$ 表示深度为h的平衡树中含有的最少结点数，显然有 $N_1 = 1$ ， $N_2 = 2$ ，……， $N_h = N_{h-1} + N_{h-2} + 1$ 。这个关系与Fibonacci数极相似。用归纳法可以证明： $N_h = F_{h+2} - 1$ ，而 $F_h$ 约等于 $\phi^h / \sqrt{5}$ ，其中 $\phi = (1 + \sqrt{5})/2$ 。则 $N_h$ 约等于 $(\phi^{h+2} / \sqrt{5} - 1)$ 。因此，含有n个结点的平衡二叉树的最大深度为

$$\begin{aligned} &\log_2(\sqrt{5}(n+1)) - 2 \\ &= \log_2(\sqrt{5}(n+1)) / \log_2 \phi - 2 \end{aligned}$$

$$= -\frac{2}{3} \log_2(\sqrt{n+1}) - 2$$

所以,在平衡树上进行查找的ASL为 $O(\log_2 n)$ 。

在平衡树上删除一个结点后可以通过旋转使其平衡,最坏情况下需作 $O(n^2)$ 次旋转。

【答案】

A: ③ B: ② C: ⑤ D: ② E: ⑥

## 1.2 程序语言

### 试题1

从供选择的答案中选出同下列程序设计语言关系最密切的字句。

(A) 50年代末以欧洲和美国的学者和研究人员为中心开发的程序设计语言。其特点之一是用Backus Naur Form (BNF表示法)描述语法 (Syntax)。

(B) 50年代末由美国国防部发起组织的CODASYL委员会开发的事务处理用的程序设计语言。

(C) 50年代后半期,以美国IBM公司的Backus为中心,在IBM公司的科学计算用的计算机上实现 (implement) 的程序设计语言。

(D) IBM公司/哈佛大学的Iverson设计的程序设计语言。适用于非定型的会话处理。

(E) 60年代初期,达特摩斯大学的Kemeny等设计的面向分时系统的程序设计语言。最近已经成为个人计算机上主要的程序设计语言。

(F) 60年代中期,IBM公司与其用户团体共同开发的程序设计语言。由于应用广泛 (可用于事务处理、科学计算、……),语言规模变得很大。

(G) 60年代末期,由苏黎士大学的Wirth设计的语言。它继承了(A)的流派,广泛用于教学。

(H) 70年代中期,美国国防部制定的面向嵌入式系统 (Embedded System) 的程序设计语言。它继承了(G)的流派,但规模比(G)大得多。

【供选择的答案】

① Ada      ② ALGOL      ③ APL      ④ BASIC  
⑤ COBOL    ⑥ FORTRAN    ⑦ PASCAL    ⑧ PL/1

【试题分析】

此题是有关程序设计语言的常识,无需分析解释。

【答案】

A: ②    B: ⑤    C: ⑥    D: ③  
E: ④    F: ⑧    G: ⑦    H: ①

### 试题2

从供选择的答案中选出应填入下面关于高级语言叙述的[ ]内的正确答案,把编

号写在答卷的对应栏内。

一种扩充型的第四代语言是| A |，面向对象型的语言是| B |，便于进行模块程序设计和系统程序设计的语言是| C |，目前在人工智能研究方面应用最多的语言是| D |，日本第五代计算机开发计划中用作核心语言的逻辑型程序设计语言是| E |。

[供选择的答案]

- ① FORTH      ② SNOBOL      ③ FORTRAN      ④ BLISS      ⑤ PROLOG  
⑥ ALGOL 68      ⑦ MODULA-2      ⑧ LISP      ⑨ SMALLTALK 80  
⑩ BASIC

[试题分析]

此题是关于高级程序设计语言的一般知识，考生应了解常用高级语言的特点和用途。

[答案]

A: ①      B: ⑨      C: ⑦      D: ⑧      E: ⑤

### 1.3 语言处理程序

试题1

从供选择的答案中选出适当字句填入下列关于系统变换工具的叙述中的| |内。

| A |逐条分析其它机种程序中的指令，变换成本机的指令然后再执行。由于是由软件逐条分析和变换指令，其缺点是执行速度太慢。

为了克服这一缺点，采用| B |实现分析和变换指令的例程（叫做| C |），以便提高执行速度。

| C |解释地执行其它机种的程序。

| D |把其它机种的程序变换成本机种的程序，然后同通常的计算机程序一样执行。

[供选择的答案]

A, C, D:

- ① 追踪程序      ② 编译程序      ③ 模拟程序      ④ 翻译程序  
⑤ 监视程序      ⑥ 解释程序      ⑦ 生成程序      ⑧ 仿真程序

B: ① 微程序      ② 宏指令      ③ 接口

[试题分析]

模拟程序是用软件的方法分析其它机种的程序然后再执行，因此速度较慢。

仿真程序是用微程序的方法，速度较快。

翻译程序是将其它机器上的程序翻成本机器上功能对等的程序。

[答案]

A: ③      B: ①      C: ⑧      D: ④

试题2

从供选择的答案中选出应填入下列叙述中的| |内的正确答案，把编号写在答卷的对应栏内。

1. 对于允许过程递归调用的语言，在它的目标程序的运行环境中至少应有| A |。

2. 在编译程序中安排优化的目的是为了得到| B |的目标代码。

3. 在编译程序中进行语法检查的目的是为了 | C |。
4. 在编译程序中安排生成中间代码的目的是 | D |。
5. 甲机上的某编译程序在乙机上能直接使用的必要条件是 | E |。

**[供选择的答案]**

- A: ① 静态存储分配    ② 栈式存储分配    ③ 堆式存储分配  
      ④ 栈式和堆式存储分配
- B: ① 结构清晰    ② 较短    ③ 高效率    ④ 使用的存储空间最小
- C: ① 发现程序所用语言的语法错误    ② 便于优化  
      ③ 发现程序中的所有错误    ④ 发现程序中的语法错误
- D: ① 便于进行优化    ② 便于进行寄存器分配  
      ③ 为了产生正确的目标代码    ④ 便于进行存储空间的组织
- E: ① 甲机的指令系统包含了乙机的指令系统  
      ② 甲机和乙机的操作系统功能完全相同  
      ③ 乙机的容量大于甲机的容量  
      ④ 乙机的指令系统包含甲机的指令系统

**[分析与答案]**

1. 对于允许过程递归调用的语言，在它的目标程序的运行环境中至少应有栈式存储分配。
2. 在编译程序中安排优化的目的是为了得到高效率的目标代码。
3. 在编译程序中进行语法检查的目的是为了发现程序中的语法错误。
4. 在编译程序中安排生成中间代码的目的是便于进行优化。
5. 甲机上的某编译程序在乙机上能直接使用的必要条件是乙机的指令系统包含甲机的指令系统。

**试题3**

从供选择的答案中选出应填入下列叙述中的 |     | 内的正确答案，把编号写在答卷的对应栏内。

高级语言编译程序常用的语法分析方法中，递归子程序法属于 | A | 分析方法，算符优先法属于 | B | 分析方法。

编译程序中代码优化的目的在于使目标程序的执行速度较快。采用的优化方法中，合并已知量和删除多余运算等是在 | C | 范围内进行；代码外提、删除归纳变量和强度削弱等是在 | D | 范围内进行；全局优化中的复写传播等是在 | E | 范围内进行。

**[供选择的答案]**

- A, B: ① 自左至右    ② 自右至左    ③ 混合方式    ④ 自顶向下    ⑤ 自底向上
- C, D, E: ① 基本块    ② 循环语句    ③ 赋值语句    ④ 条件语句    ⑤ 整个程序

**[试题分析]**

语法分析是编译过程的核心部份，其任务是按照文法从源程序符号串中识别出各类语法成份，同时进行语法检查，为语义分析和代码生成作准备。按语法树的建立方法，语法分析方法分为自顶向下和自底向上两大类。前者有递归子程序和LL分析法。后者有算符

优先分析法和LR分析方法。

为了使目标程序执行速度快，常采用一些代码优化技术，例如对基本块可采用合并已知变量和删除多余运算的优化技术；对循环语句可采用代码外提，删除归纳变量和强度削弱等技术；对整个程序可采用复写传播优化技术。

〔答案〕

A: ④ B: ⑤ C: ① D: ② E: ⑤

## 1.4 文件系统

### 试题1

阅读下列关于分区文件的叙述，从供选择的答案中选出适当的答案填入| |内。

分区文件同时具备顺序文件和索引顺序文件的特点。分区文件划分成若干个称为成员的子文件，各成员都是| |。分区文件有一个用于存放标识各成员的| B |和| |的| D |。

从| D |中可以知道各成员存放在什么地方，以及一个文件中能包含多少个成员。必要时，可追加和删除成员。

这种文件通常作为| E |使用，很少用作| F |。

〔供选择的答案〕

A, E, F:

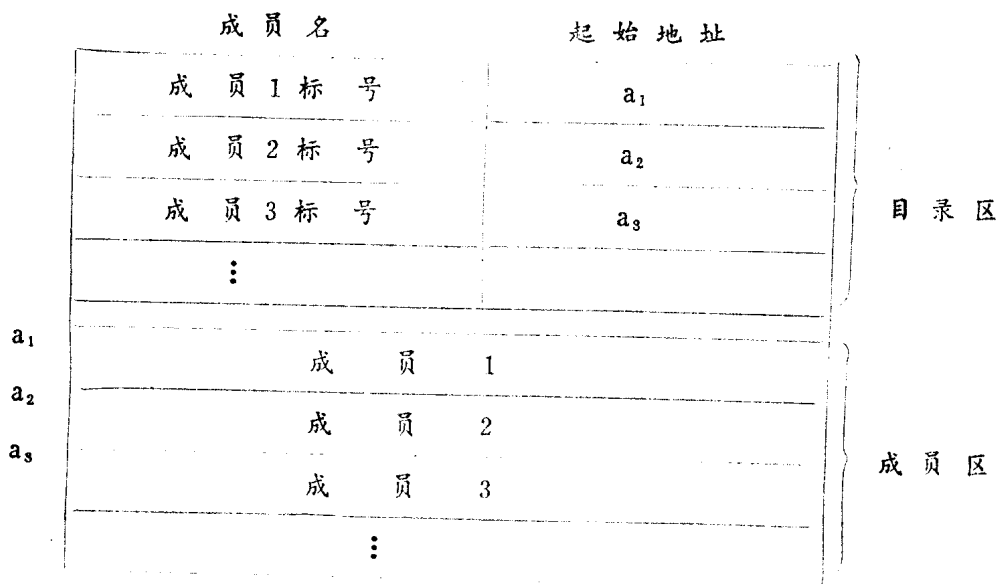
- ① 数据文件      ② 程序文件      ③ 顺序文件  
④ 直接文件      ⑤ 索引顺序文件      ⑥ 数据库

B, C, D:

- ① 记录长度      ② 名称      ③ 对照表      ④ 目录  
⑤ 起始地址      ⑥ 最终地址      ⑦ 索引      ⑧ 口令

〔试题分析〕

分区文件是文件的一种组织方式，其结构如下：



分区文件的特点是可以从文件的任意起始位置开始顺序存取成员。一般用于存储程序文件，很少用于存储数据文件。

〔答案〕

A: ③ B: ② C: ⑤ D: ④ E: ② F: ①

## 1.5 操作系统

### 试题1

从供选择的答案中选出应填入下列叙述中的  内的正确答案，把编号写在答卷的对应栏内。

A  是一种只能进行P操作和V操作的特殊变量。 A  可以用来实现异步并行进程间的  B  和  C 。 B  是指排它地访问共享数据， C  则是指进程间在逻辑上的相互制约关系。

D  是可以用来实现异步并行进程的  B  和  C  的特殊程序结构。 D  中的  E  用于实现进程间的  C 。

〔供选择的答案〕

A, B, C, D, E:

- |       |        |        |      |
|-------|--------|--------|------|
| ① 调度  | ② 类程   | ③ 进程   | ④ 互斥 |
| ⑤ 信号量 | ⑥ 控制变量 | ⑦ 条件变量 | ⑧ 管理 |
| ⑨ 同步  | ⑩ 共享变量 | ⑪ 规程   | ⑫ 分配 |

〔试题分析〕

操作系统中信号量是表示资源的物理实体，是一个与队列有关的特殊变量，其值仅能由P、V操作改变。操作系统利用它的状态对进程和资源进行管理，实现异步并发进程间的互斥与同步。利用信号量能方便地解决临界区问题。具体实现方法可参阅操作系统教材中的相关章节。信号量还能用于实现异步环境下的进程同步。所谓异步环境是指互相合作的一组并行进程，其中每个进程都能以各自独立的、不可预知的速度向前推进。所谓进程同步是指互相合作的进程之间须要交换一定信息。在某个进程未获得其合作进程所发来的信息之前，该进程等待，直到该信息到来之时才被唤醒继续执行，从而保证了诸进程的协调运行。因此，进程间的同步是指进程在逻辑上的相互制约关系。

随着操作系统的规模日趋庞大，结构越来越复杂，人们提出了结构化设计方法，采用了层次管理结构技术。在这种方法中，系统模块分成三大类：管程、类程和进程。管程定义了一个共享数据结构和各种进程在该数据结构上所能执行的全部操作。类程定义了一个专用的数据结构及在该数据结构上所有可能的操作。进程间的同步与互斥是通过管程实现的。管程中的条件变量用于实现进程间的同步。

〔答案〕

A: ⑤ B: ④ C: ⑨ D: ⑧ E: ⑦

### 试题2

从供选择的答案中选出应填入下列叙述中的  内的正确答案，把编号写在答卷的对应栏内。