

全国计算机技术与软件专业
技术资格（水平）考试用书

软件设计师考试 习题集

希赛教育软考学院 桂阳 陈勇军 主编

7年的精华试题汇总·
10余位专家经验集结· 10万考友在线交流·



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

全国计算机技术与软件专业
技术资格（水平）考试用书

软件设计师考试

习题集

希赛教育软考学院 桂阳 陈勇军 主编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书由希赛教育软考学院组织编写，作为全国计算机技术与软件专业技术资格（水平）考试中的软件设计师级别考试的辅导与培训教材。本书根据最新的软件设计师考试大纲，对2005~2011年考试真题进行了归类总结，并给出了解答，使考生更有针对性地进行复习和应考，实践性较强。

考生可通过做本书的习题，掌握考试大纲规定的知识点、考试重点和难点，熟悉考试方法、试题形式、试题的深度和广度、考试内容的分布，以及解答问题的方法和技巧。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

软件设计师考试习题集 / 桂阳，陈勇军主编. —北京：电子工业出版社，2011.9
全国计算机技术与软件专业技术资格（水平）考试用书
ISBN 978-7-121-14401-1

I. ①软… II. ①桂… ②陈… III. ①软件设计—工程技术人员—资格考试—习题集 IV. ①TP311.5-44

中国版本图书馆 CIP 数据核字（2011）第 168436 号

策划编辑：孙学瑛

责任编辑：葛娜

印刷：北京东光印刷厂

装订：三河市皇庄路通装订厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开本：787×1092 1/16 印张：19.5 字数：486 千字

印次：2011 年 9 月第 1 次印刷

印数：4000 册 定价：39.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前 言

全国计算机技术与软件专业技术资格（水平）考试（以下简称“软考”）是一个难度很大的考试，平均通过率很低。主要原因是考试范围十分广泛，涉及计算机专业的每门课程，还有数学、外语、系统工程、信息化和知识产权等知识领域，且注重考查新技术和新方法的应用。考试不但注重广度，而且还有一定深度。

《软件设计师考试习题集》是为软考中的软件设计师级别而编写的考试用书，全书对2005~2011年的软件设计师考试的所有考题进行了归类，并给出了解答，使考生更有针对性地进行复习和应考，实践性较强。考生可通过做本书的习题，掌握考试大纲规定的知识点、考试重点和难点，熟悉考试方法、试题形式、试题的深度和广度、考试内容的分布，以及解答问题的方法和技巧。

作者权威，阵容强大

希赛教育（www.educity.cn）专业从事人才培养、教育产品开发、教育图书出版，在职业教育方面具有极高的权威性。特别是在在线教育方面，稳居国内首位，希赛教育的远程教育模式得到了国家教育部门的认可和推广。

希赛教育软考学院（www.csairk.com）是全国计算机技术与软件专业技术资格（水平）考试的顶级培训机构，拥有近20名资深软考辅导专家，负责高级资格的考试大纲制定工作，以及软考辅导教材的编写工作，共组织编写和出版了60多本软考教材，内容涵盖了初级、中级和高级的各个专业，包括教程系列、辅导系列、考点分析系列、冲刺系列、串讲系列、试题精解系列、疑难解答系列、全程指导系列、案例分析系列、指定参考用书系列、一本通11个系列的书籍。希赛教育软考学院的专家录制了软考培训视频教程、串讲视频教程、试题讲解视频教程、专题讲解视频教程4个系列的软考视频，希赛教育软考学院的软考教材、软考视频、软考辅导为考生助考、提高通过率做出了不可磨灭的贡献，在软考领域有口皆碑。特别是在高级资格领域，无论是考试教材，还是在线辅导和面授，希赛教育软考学院都独占鳌头。

本书由希赛教育软考学院的桂阳和陈勇军主编，参加编写的人员有张友生、谢顺、施游、李雄、何玉云、胡钊源、王勇、胡光超、刘洋波和左水林。

在线测试，心中有数

上学吧（www.shangxueba.com）在线测试平台为考生准备了在线测试，其中有数十套全真模拟试题和考前密卷，考生可选择任何一套进行测试。测试完毕，系统自动判卷，立即给出分数。

对于考生做错的地方，系统会自动记忆，待考生第二次参加测试时，可选择“试题复

习”。这样，系统就会自动把考生原来做错的试题显示出来，供考生重新测试，以加强记忆。

如此，读者可利用上学吧在线测试平台的在线测试系统检查自己的实际水平，加强考前训练，做到心中有数，考试不慌。

诸多帮助，诚挚致谢

在本书出版之际，要特别感谢全国软考办的命题专家们，编者在本书中引用了部分考试原题，使本书能够尽量方便读者阅读。在本书的编写过程中，参考了许多相关的文献和书籍，编者在此对这些参考文献的作者表示感谢。

感谢电子工业出版社的孙学瑛老师，她在本书的策划、选题的申报、写作大纲的确定，以及编辑、出版等方面，付出了辛勤的劳动和智慧，给予了我们很多的支持和帮助。

感谢参加希赛教育软考学院辅导和培训的学员，正是他们的想法汇成了本书的源动力，他们的意见使本书更加贴近读者。

由于编者水平有限，且本书涉及的内容很广，书中难免存在错漏和不妥之处，编者诚恳地期望各位专家和读者不吝指正和帮助，对此，我们将十分感激。

互动讨论，专家答疑

希赛教育软考学院（www.csairk.com）是中国最大的软考在线教育网站，该网站论坛是国内人气最旺的软考社区，在这里，读者可以和数十万考生进行在线交流，讨论有关学习和考试的问题。希赛教育软考学院拥有强大的师资队伍，为读者提供全程的答疑服务，在线回答读者的提问。

有关本书的意见反馈和咨询，读者可在希赛教育软考学院论坛“软考教材”版块中的“希赛教育软考学院”栏目上与作者进行交流。

希赛教育软考学院
2011年6月

目 录

第 1 章 数据结构与算法 1	第 9 章 安全性、可靠性 及系统性能评价 93
1.1 习题..... 1	9.1 习题.....93
1.2 习题参考答案..... 16	9.2 习题参考答案.....97
第 2 章 程序语言 17	第 10 章 数据通信与计算机网络 98
2.1 习题..... 17	10.1 习题.....98
2.2 习题参考答案..... 26	10.2 习题参考答案..... 105
第 3 章 操作系统 27	第 11 章 知识产权 106
3.1 习题..... 27	11.1 习题..... 106
3.2 习题参考答案..... 37	11.2 习题参考答案..... 108
第 4 章 软件工程 39	第 12 章 标准化知识 109
4.1 习题..... 39	12.1 习题..... 109
4.2 习题参考答案..... 55	12.2 习题参考答案..... 109
第 5 章 面向对象方法学 56	第 13 章 信息化与信息系统 110
5.1 习题..... 56	13.1 习题..... 110
5.2 习题参考答案..... 68	13.2 习题参考答案..... 110
第 6 章 数据库系统 70	第 14 章 专业英语 111
6.1 习题..... 70	14.1 习题..... 111
6.2 习题参考答案..... 81	14.2 习题参考答案..... 121
第 7 章 多媒体技术及其应用 82	第 15 章 数据流图的设计 122
7.1 习题..... 82	15.1 习题..... 122
7.2 习题参考答案..... 84	15.2 习题参考答案..... 142
第 8 章 计算机组成与体系结构 85	第 16 章 程序流程图 148
8.1 习题..... 85	16.1 习题..... 148
8.2 习题参考答案..... 91	16.2 习题参考答案..... 163

第 17 章 测试用例设计	166	第 20 章 数据库设计	243
17.1 习题	166	20.1 习题	243
17.2 习题参考答案	168	20.2 习题参考答案	263
第 18 章 面向对象程序设计	169	第 21 章 常用算法设计	272
18.1 习题	169	21.1 习题	272
18.2 习题参考答案	213	21.2 习题参考答案	298
第 19 章 UML 分析与设计	219	第 22 章 其他设计	302
19.1 习题	219	22.1 习题	302
19.2 习题参考答案	238	22.2 习题参考答案	304

第1章 数据结构与算法

1.1 习题

● 循环链表的主要优点是___(1)___。

- (1) A. 不再需要头指针了
 B. 已知某个结点的位置后,能很容易找到它的直接前驱结点
 C. 在进行删除操作后,能保证链表不断开
 D. 从表中任一结点出发都能遍历整个链表

● 表达式 $a*(b+c)-d$ 的后缀表达式为___(2)___。

- (2) A. $abcd*+-$ B. $abc+*d-$ C. $abc*+d-$ D. $-+*abcd$

● 若二叉树的先序遍历序列为 ABDECF, 中序遍历序列为 DBE AFC, 则其后序遍历序列为___(3)___。

- (3) A. DEBAFC B. DEFBCA C. DEBCFA D. DEBFCA

● 无向图中一个顶点的度是指图中___(4)___。

- (4) A. 通过该顶点的简单路径数 B. 通过该顶点的回路数
 C. 与该顶点相邻的顶点数 D. 与该顶点连通的顶点数

● 利用逐点插入法建立序列 (50,72,43,85,75,20,35,45,65,30) 对应的二叉排序树以后, 查找元素 30 要进行___(5)___次元素间的比较。

- (5) A. 4 B. 5 C. 6 D. 7

● 在常用的描述二叉排序树的存储结构中, 关键字值最大的结点___(6)___。

- (6) A. 左指针一定为空 B. 右指针一定为空
 C. 左、右指针均为空 D. 左、右指针均不为空

● 一个具有 n ($n>0$) 个顶点的连通无向图至少有___(7)___条边。

- (7) A. $n+1$ B. n C. $n/2$ D. $n-1$

● 由权值为 9,2,5,7 的 4 个叶子结点构造一棵哈夫曼树, 该树的带权路径长度为___(8)___。

- (8) A. 23 B. 37 C. 44 D. 46

● 在最好和最坏情况下的时间复杂度均为 $O(n\log_2 n)$ 且稳定的排序方法是___(9)___。

- (9) A. 基数排序 B. 快速排序 C. 堆排序 D. 归并排序

● 已知一个线性表 (38,25,74,63,52,48), 假定采用散列函数 $h(\text{key}) = \text{key} \% 7$ 计算散列地址, 并散列存储在散列表 $A[0, \dots, 6]$ 中, 若采用线性探测方法解决冲突, 则在该散列表上进行等概率成功查找的平均查找长度为___(10)___。

- (10) A. 1.5 B. 1.7 C. 2.0 D. 2.3

● 为了在状态空间树中___(11)___, 可以利用 LC-检索 (Least Cost Search) 快速找到一个答案结点。在进行 LC-检索时, 为避免算法过分偏向于纵深检查, 应该___(12)___。

- (11) A. 找出任一个答案结点 B. 找出所有的答案结点

- C. 找出最优的答案结点
D. 进行遍历
- (12) A. 使用精确的成本函数 $c(\cdot)$ 来做 LC-检索
B. 使用广度优先检索
C. 使用深度优先检索
D. 在成本估计函数 $\hat{e}(\cdot)$ 中考虑根结点到当前结点的成本 (距离)

● 以比较为基础的排序算法在最坏情况下的计算时间下界为 (13)。

- (13) A. $O(n)$ B. $O(n^2)$ C. $O(\log_2 n)$ D. $O(n \log_2 n)$

● 利用动态规划方法求解每对结点之间的最短路径问题 (all pairs shortest path problem) 时, 设有向图 $G = \langle V, E \rangle$ 共有 n 个结点, 结点编号 $1 \sim n$, 设 C 是 G 的成本邻接矩阵, $D_k(i, j)$ 即为图 G 中结点 i 到 j 并且不经过编号比 k 还大的结点的最短路径长度 ($D_n(i, j)$ 即为图 G 中结点 i 到 j 的最短路径长度), 则求解该问题的递推关系式为 (14)。

- (14) A. $D_k(i, j) = D_{k-1}(i, j) + C(i, j)$
B. $D_k(i, j) = \min\{D_{k-1}(i, j), D_{k-1}(i, j) + C(i, j)\}$
C. $D_k(i, j) = D_{k-1}(i, k) + D_{k-1}(k, j)$
D. $D_k(i, j) = \min\{D_{k-1}(i, j), D_{k-1}(i, k) + D_{k-1}(k, j)\}$

● 在活动图中, 结点表示项目中各个工作阶段的里程碑, 连接各个结点的边表示活动, 边上的数字表示活动持续的时间。在下面的活动图 1-1 中, 从 A 到 J 的关键路径是 (15), 关键路径长度是 (16), 从 E 开始的活动的启动的最早时间是 (17)。

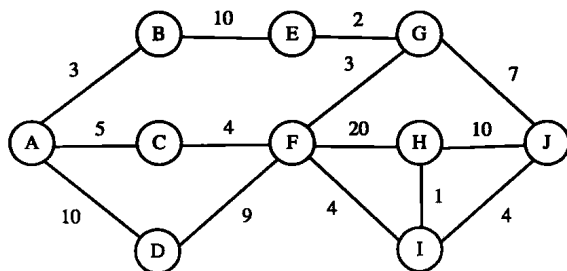


图 1-1 活动图

- (15) A. ABEGJ B. ADFHJ C. ACFGJ D. ADFIJ
(16) A. 22 B. 49 C. 19 D. 35
(17) A. 10 B. 12 C. 13 D. 15

● 已知某二叉树的中序、层序序列分别为 DBAFCE、FDEBCA, 则该二叉树的后序序列为 (18)。

- (18) A. BCDEAF B. ABDCEF C. DBACEF D. DABECF

● 在二叉树的顺序存储中, 每个结点的存储位置与其父结点、左右子树结点的位置都存在一个简单的映射关系, 因此可与三叉链表对应。若某二叉树共有 n 个结点, 采用三叉链表存储时, 每个结点的数据域需要 d 个字节, 每个指针域占用 4 个字节, 若采用顺序存储, 则最后一个结点下标为 k (起始下标为 1), 那么 (19) 时采用顺序存储更节省空间。

- (19) A. $d < \frac{12n}{k-n}$ B. $d > \frac{12n}{k-n}$ C. $d < \frac{12n}{k+n}$ D. $d < \frac{12n}{k+n}$

● 简单无向图的邻接矩阵是对称的, 可以对其进行压缩存储。若无向图 G 有 n 个结点, 其邻接矩阵为 $A[1..n, 1..n]$, 且压缩存储在 $B[1..k]$ 中, 则 k 的值至少为 (20)。若按行压缩存储对称矩阵的上三角元素, 则当 n 等于 10 时, 边 (V_6, V_3) 的信息存储在 $B[$ (21) $]$ 中。

- (20) A. $\frac{n(n+1)}{2}$ B. $\frac{n^2}{2}$ C. $\frac{(n-1)(n+1)}{2}$ D. $\frac{n(n-1)}{2}$

- (21) A. 18 B. 19 C. 20 D. 21

● 在 11 个元素的有序表 $A[1..11]$ 中进行折半查找 ($\lfloor (low + high) / 2 \rfloor$), 查找元素 $A[11]$ 时, 被比较的元素的下标依次是 (22)。

- (22) A. 6,8,10,11 B. 6,9,10,11 C. 6,7,9,11 D. 6,8,9,11

● 由元素序列 $(27, 16, 75, 38, 51)$ 构造平衡二叉树, 则首次出现的最小不平衡子树的根 (即离插入结点最近且平衡因子的绝对值为 2 的结点) 为 (23)。

- (23) A. 27 B. 38 C. 51 D. 75

● 若排序前后关键字相同的两个元素相对位置不变, 则称该排序方法是稳定的。(24) 排序是稳定的。

- (24) A. 归并 B. 快速 C. 希尔 D. 堆

● 设求解某问题的递归算法如下:

```
F(int n){
    if (n==1){
        Move(1);
    }else{
        F(n-1);
        Move(n);
        F(n-1);
    }
}
```

求解该算法的计算时间时, 仅考虑算法 $Move$ 所做的计算为主要计算, 且 $Move$ 为常数级算法。则算法 F 的计算时间 $T(n)$ 的递推关系式为 (25); 设算法 $Move$ 的计算时间为 k , 当 $n=4$ 时, 算法 F 的计算时间为 (26)。

- (25) A. $T(n)=T(n-1)+1$ B. $T(n)=2T(n-1)$
 C. $T(n)=2T(n-1)+1$ D. $T(n)=2T(n+1)+1$

- (26) A. $14k$ B. $15k$ C. $16k$ D. $17k$

● 利用贪心法求解 0-1 背包问题时, (27) 能够确保获得最优解。用动态规划方法求解 0-1 背包问题时, 将“用前 i 个物品来装容量是 X 的背包”的 0-1 背包问题记为 $KNAP(1, i, X)$, 设 $f_i(X)$ 是 $KNAP(1, i, X)$ 最优解的效益值, 第 j 个物品的重量和放入背包后取得效益值分别为 W_j 和 p_j ($j=1 \sim n$)。则依次求解 $f_0(X), f_1(X), \dots, f_n(X)$ 的过程中使用的递推关系式为 (28)。

- (27) A. 优先选取重量最小的物品 B. 优先选取效益最大的物品
 C. 优先选取单位重量效益最大的物品 D. 没有任何准则

- (28) A. $f_i(X)=\min\{f_{i-1}(X), f_{i-1}(X)+p_i\}$
 B. $f_i(X)=\max\{f_{i-1}(X), f_{i-1}(X-W_i)+p_i\}$
 C. $f_i(X)=\min\{f_{i-1}(X-W_i), f_{i-1}(X-W_i)+p_i\}$

D. $f_i(X) = \max\{f_{i-1}(X - W_i), f_{i-1}(X) + p_i\}$

● 与逆波兰式 $ab + c * d$ 对应的中缀表达式是 (29)。

- (29) A. $a - b - c * d$ B. $-(a + b) * c - d$
 C. $-a + b * c - d$ D. $(a + b) * (-c - d)$

● 拓扑序列是无环有向图中所有顶点的一个线性序列，图中任意路径中的各个顶点在该图的拓扑序列中保持先后关系，(30) 为图 1-2 所示有向图的一个拓扑序列。

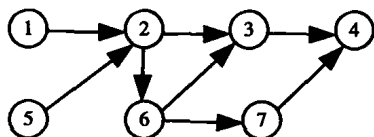


图 1-2 有向图拓扑序列

- (30) A. 1 2 3 4 5 6 7 B. 1 5 2 6 3 7 4
 C. 5 1 2 6 3 4 7 D. 5 1 2 3 7 6 4

● 为了便于存储和处理一般树结构形式的信息，常采用孩子-兄弟表示法将其转换成二叉树（左子关系表示父子，右子关系表示兄弟），与图 1-3 所示的树对应的二叉树是 (31)。

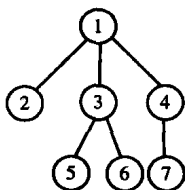
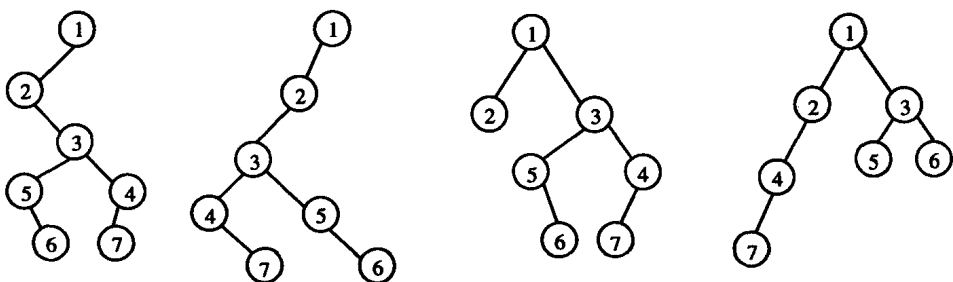


图 1-3 树

- (31) A. B. C. D.



● 给定一个有 n 个元素的有序线性表。若采用顺序存储结构，则在等概率前提下，删除其中的一个元素平均需要移动 (32) 个元素。

- (32) A. $(n+1)/2$ B. $n/2$ C. $(n-1)/2$ D. 1

● 在平衡二叉树中，(33)。

- (33) A. 任意结点的左、右子树结点数目相同
 B. 任意结点的左、右子树高度相同
 C. 任意结点的左、右子树高度之差的绝对值不大于 1
 D. 不存在度为 1 的结点

● 在 (34) 存储结构中, 数据结构中元素的存储地址与其关键字之间存在某种映射关系。

- (34) A. 顺序 (Sequence) B. 链表 (Link)
 C. 索引 (Index) D. 散列 (Hash)

● 对于求取两个长度为 n 的字符串的最长公共子序列 (LCS) 问题, 利用 (35) 策略可以有效地避免子串最长公共子序列的重复计算, 得到时间复杂度为 $O(n^2)$ 的正确算法。串 $\langle 1,0,0,1,0,1,0,1 \rangle$ 和 $\langle 0,1,0,1,1,0,1,1 \rangle$ 的最长公共子序列的长度为 (36)。

- (35) A. 分治 B. 贪心 C. 动态规划 D. 分支限界
 (36) A. 3 B. 4 C. 5 D. 6

● 设某算法的计算时间可用递推关系式 $T(n)=2T(n/2)+n$ 表示, 则该算法的时间复杂度为 (37)。

- (37) A. $O(\lg n)$ B. $O(n \lg n)$ C. $O(n)$ D. $O(n^2)$

● (38) 在其最好情况下的算法时间复杂度为 $O(n)$ 。

- (38) A. 插入排序 B. 归并排序 C. 快速排序 D. 堆排序

● 表达式 “ $X=(A+B) \times (C - D/E)$ ” 的后缀表示为 (39)。

- (39) A. $XAB+CDE/-\times=$ B. $XAB-C-DE/\times=$
 C. $XAB+CDE/-\times=$ D. $XAB-CD-E/\times=$

● 结点数目为 n 的二叉查找树 (二叉排序树) 的最小高度为 (40), 最大高度为 (41)。

- (40) A. n B. $n/2$ C. $\lceil \log_2 n \rceil$ D. $\lceil \log_2(n+1) \rceil$
 (41) A. n B. $n/2$ C. $\lceil \log_2 n \rceil$ D. $\lceil \log_2(n+1) \rceil$

● 某双向链表中的结点如图 1-4 所示, 删除 t 所指结点的操作为 (42)。

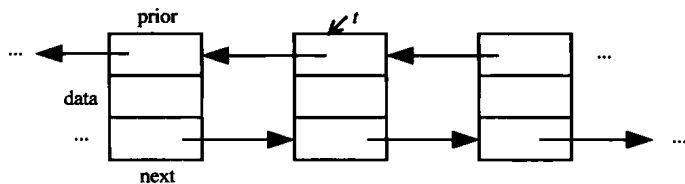


图 1-4 双向链表

- (42) A. $t \rightarrow \text{prior} \rightarrow \text{next} = t \rightarrow \text{next}; t \rightarrow \text{next} \rightarrow \text{prior} = t \rightarrow \text{prior};$
 B. $t \rightarrow \text{prior} \rightarrow \text{prior} = t \rightarrow \text{prior}; t \rightarrow \text{next} \rightarrow \text{next} = t \rightarrow \text{next};$
 C. $t \rightarrow \text{prior} \rightarrow \text{next} = t \rightarrow \text{prior}; t \rightarrow \text{next} \rightarrow \text{prior} = t \rightarrow \text{next};$
 D. $t \rightarrow \text{prior} \rightarrow \text{prior} = t \rightarrow \text{next}; t \rightarrow \text{next} \rightarrow \text{prior} = t \rightarrow \text{prior};$

● 对于二维数组 $a[0..4,1..5]$, 设每个元素占 1 个存储单元, 且以列为主序存储, 则元素 $a[2,2]$ 相对于数组空间起始地址的偏移量是 (43)。

- (43) A. 5 B. 7 C. 10 D. 15

● 对于 n 个元素的关键字序列 $\{k_1, k_2, \dots, k_n\}$, 当且仅当满足关系 $k_i \leq k_{2i}$ 且 $k_i \leq k_{2i+1}$ ($2i < n, 2i+1 \leq n$) 时称其为小根堆, 反之则称为大根堆。以下序列中, (44) 不符合堆的定义。

- (44) A. (4,10,15,72,39,23,18) B. (58,27,36,12,8,23,9)

C. (4,10,18,72,39,23,15)

D. (58,36,27,12,8,23,9)

● 求单源点最短路径的迪杰斯特拉 (Dijkstra) 算法是按 (45) 的顺序求源点到各顶点的最短路径的。

(45) A. 路径长度递减

B. 路径长度递增

C. 顶点编号递减

D. 顶点编号递增

● (46) 算法策略与递归技术的联系最弱。

(46) A. 动态规划

B. 贪心

C. 回溯

D. 分治

● 对于具有 n 个元素的一个数据序列, 若只需得到其中第 k 个元素之前的部分排序, 最好采用 (47), 使用分治 (Divide and Conquer) 策略的是 (48) 算法。

(47) A. 希尔排序

B. 直接插入排序

C. 快速排序

D. 堆排序

(48) A. 冒泡排序

B. 插入排序

C. 快速排序

D. 堆排序

● 表达式 $(a+b)*(c-d)$ 的后缀表示为 (49)。

(49) A. $ab+cd*$

B. $abcd+*$

C. $ab+*cd-$

D. $abcd*+-$

● 输入受限的双端队列是指元素只能从队列的一端输入, 但可以从队列的两端输出, 如图 1-5 所示。若有 8,1,4,2 依次进入输入受限的双端队列, 则得不到输出序列 (50)。



图 1-5 双端队列

(50) A. 2,8,1,4

B. 1,4,8,2

C. 4,2,1,8

D. 2,1,4,8

● 已知某二叉树的中序序列为 CBDAEFI, 先序序列为 ABCDEFI, 则该二叉树的高度为 (51)。

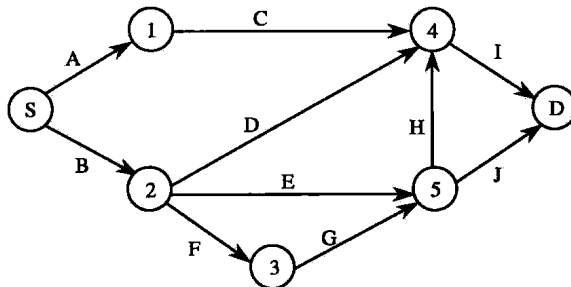
(51) A. 2

B. 3

C. 4

D. 5

● 某工程计划如图 1-6 所示, 各个作业所需的天数如下表所示, 设该工程从第 0 天开工, 则该工程的最短工期是 (52) 天, 作业 J 最迟应在第 (53) 天开工。



作业	A	B	C	D	E	F	G	H	I	J
所需天数	7	6	8	10	7	3	2	4	3	7

图 1-6 工程计划图

(52) A. 17

B. 18

C. 19

D. 20

(53) A. 11

B. 13

C. 14

D. 16

● 在如图 1-7 所示的平衡二叉树（树中任一结点的左右子树高度之差不超过 1）中，结点 A 的右子树 AR 高度为 h ，结点 B 的左子树 BL 高度为 h ，结点 C 的左子树 CL、右子树 CR 高度都为 $h-1$ 。若在 CR 中插入一个结点并使得 CR 的高度增加 1，则该二叉树 (54)。

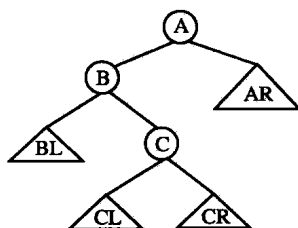


图 1-7 平衡二叉树

- (54) A. 以 B 为根的子二叉树变为不平衡
 B. 以 C 为根的子二叉树变为不平衡
 C. 以 A 为根的子二叉树变为不平衡
 D. 仍然是平衡二叉树

● 设商店有 10 元、5 元、2 元和 1 元的零币，每种零币的数量充足。售货员给顾客找零钱时，零币的数量越少越好。例如给顾客找零 29 元：先选 2 张 10 元币，然后选择 1 张 5 元币，再选择两张 2 元币。以上的找零钱方法采用了 (55) 策略。

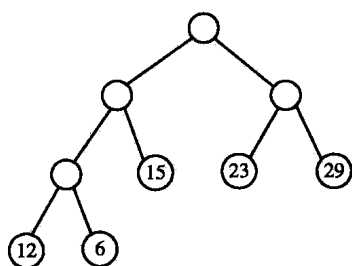
- (55) A. 分治 B. 贪心 C. 动态规划 D. 回溯

● 对 n 个元素的数组进行 (56)，其平均时间复杂度和最坏情况下的时间复杂度都是 $O(n \log n)$ 。

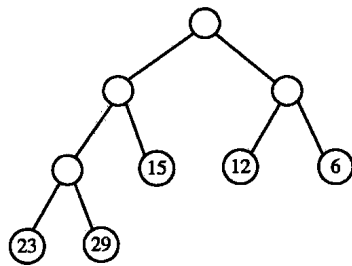
- (56) A. 希尔排序 B. 快速排序 C. 堆排序 D. 选择排序

● 由权值为 29,12,15,6,23 的 5 个叶子结点构造的哈夫曼树为 (57)，其带权路径长度为 (58)。

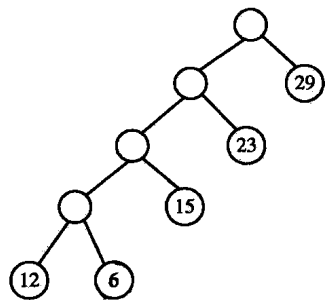
(57) A.



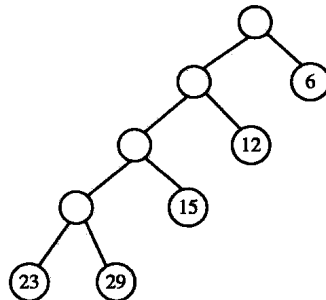
B.



C.



D.



- (58) A. 85 B. 188

- C. 192 D. 222

- (69) A. 8 B. 12 C. 33 D. 48

● 已知一个线性表 (16,25,35,43,51,62,87,93)，采用散列函数 $H(\text{Key})=\text{Key} \bmod 7$ 将元素散列到表长为 9 的散列表中。若采用线性探测的开放定址法解决冲突（顺序地探查可用存储单元），则构造的哈希表为 (70)，在该散列表上进行等概率成功查找的平均查找长度为 (71)（为确定记录在查找表中的位置，需和给定关键字值进行比较的次数期望值称为查找算法在查找成功时的平均查找长度）。

(70) A.

0	1	2	3	4	5	6	7	8
35	43	16	51	25		62	87	93

B.

0	1	2	3	4	5	6	7	8
35	43	16	93	25	51	62	87	

C.

0	1	2	3	4	5	6	7	8
35	43	16	51	25	87	62	93	

D.

0	1	2	3	4	5	6	7	8
35	43	16	51	25	87	62		93

- (71) A. $(5*1+2+3+6)/8$ B. $(5*1+2+3+6)/9$
 C. $(8*1)/8$ D. $(8*1)/9$

● 若将某有序树 T 转换为二叉树 $T1$ ，则 T 中结点的后（根）序序列就是 $T1$ 中结点的 (72) 遍历序列。例如，如图 1-8 (a) 所示的有序树转化为二叉树后如图 1-8 (b) 所示。

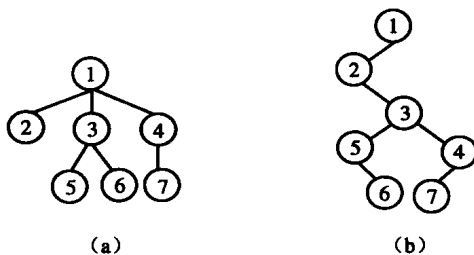


图 1-8 有序树转化为二叉树

- (72) A. 先序 B. 中序 C. 后序 D. 层序

● 设一个包含 N 个顶点、 E 条边的简单有向图采用邻接矩阵存储结构(矩阵元素 $A[i][j]$ 等于 1/0 分别表示顶点 i 与顶点 j 之间有/无弧)，则该矩阵的元素数目为 (73)，其中非零元素数目为 (74)。

- (73) A. E^2 B. N^2 C. $N^2 - E^2$ D. $N^2 + E^2$
 (74) A. N B. $N+E$ C. E D. $N - E$

● 一个算法是对某类给定问题求解过程的精确描述，算法中描述的操作都可以通过将已经实现的基本操作执行有限次来实现。这句话说明算法具有 (75) 特性。

(75) A. 有穷性 B. 可行性 C. 确定性 D. 健壮性

● 斐波那契 (Fibonacci) 数列可以递归地定义为:

$$F(n) = \begin{cases} 1, & n=0 \\ 1, & n=1 \\ F(n-1) + F(n-2), & n>1 \end{cases}$$

用递归算法求解 $F(5)$ 时需要执行 (76) 次 “+” 运算, 该方法采用的算法策略是 (77)。

(76) A. 5 B. 6 C. 7 D. 8

(77) A. 动态规划 B. 分治 C. 回溯 D. 分支限界

● 若总是以待排序列的第一个元素作为基准元素进行快速排序, 那么在最好情况下的时间复杂度为 (78)。

(78) A. $O(\log_2 n)$ B. $O(n)$ C. $O(n \log_2 n)$ D. $O(n^2)$

● 表达式 $(a-b)*(c+5)$ 的后缀表示是 (79)。

(79) A. $abc5+*-$ B. $ab-c+5*$

C. $abc-*5+$ D. $ab-c5+*$

● 一个具有 m 个结点的二叉树, 其二叉链表结点 (左、右孩子指针分别用 $left$ 和 $right$ 表示) 中的空指针总数必定为 (80) 个。为形成中序 (先序、后序) 线索二叉树, 现对该二叉链表所有结点进行如下操作: 若结点 p 的左孩子指针为空, 则将该左指针改为指向 p 在中序 (先序、后序) 遍历序列的前驱结点; 若 p 的右孩子指针为空, 则将该右指针改为指向 p 在中序 (先序、后序) 遍历序列的后继结点。假设指针 s 指向中序 (先序、后序) 线索二叉树中的某结点, 则 (81)。

(80) A. $m+2$ B. $m+1$ C. m D. $m-1$

(81) A. $s \rightarrow right$ 指向的结点一定是 s 所指结点的直接后继结点

B. $s \rightarrow left$ 指向的结点一定是 s 所指结点的直接前驱结点

C. 从 s 所指结点出发的 $right$ 链可能构成环

D. s 所指结点的 $left$ 和 $right$ 指针一定指向不同的结点

● (82) 的邻接矩阵是一个对称矩阵。

(82) A. 无向图 B. AOV 网 C. AOE 网 D. 有向图

● 将一个无序序列中的元素依次插入到一棵 (83), 并进行中序遍历, 可得到一个有序序列。

(83) A. 完全二叉树 B. 最小生成树 C. 二叉排序树 D. 最优二叉树

● 广义表中的元素可以是原子, 也可以是表, 因此广义表的适用存储结构是

(84)。

(84) A. 链表 B. 静态数组 C. 动态数组 D. 散列表

● 某一维数组中依次存放了数据元素 12, 23, 30, 38, 41, 52, 54, 76, 85, 在用折半 (二分) 查找方法 (向上取整) 查找元素 54 时, 所经历 “比较” 运算的数据元素依次为 (85)。

(85) A. 41, 52, 54 B. 41, 76, 54 C. 41, 76, 52, 54 D. 41, 30, 76, 54

● 具有 n 个顶点、 e 条边的图采用邻接表存储结构, 进行深度优先遍历和广度优先遍历运算的时间复杂度均为 (86)。

(86) A. $O(n^2)$ B. $O(e^2)$ C. $O(n*e)$ D. $O(n+e)$