

【软考达人】

软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



微信扫一扫，立马获取



6W+ 免费题库



免费备考资料

PC版题库: ruankaodaren.com

全国计算机技术与软件专业技术资格（水平）考试

2016 年下半年 软件设计师 上午试卷

（考试时间 9：00～11：30 共 150 分钟）

请按下述要求正确填写答题卡

1. 在答题卡的指定位置上正确写入你的姓名和准考证号，并用正规 2B 铅笔在你写入的准考证号下填涂准考证号。
2. 本试卷的试题中共有 75 个空格，需要全部解答，每个空格 1 分，满分 75 分。
3. 每个空格对应一个序号，有 A、B、C、D 四个选项，请选择一个最恰当的选项作为解答，在答题卡相应序号下填涂该选项。
4. 解答前务必阅读例题和答题卡上的例题填涂样式及填涂注意事项。解答时用正规 2B 铅笔正确填涂选项，如需修改，请用橡皮擦干净，否则会导致不能正确评分。

例题

● 2016 年下半年全国计算机技术与软件专业技术资格（水平）考试日期是 (88) 月 (89) 日。

- (88) A. 9 B. 10 C. 11 D. 12
(89) A. 4 B. 5 C. 6 D. 7

因为考试日期是“11 月 4 日”，故 (88) 选 C，(89) 选 A，应在答题卡序号 88 下对 C 填涂，在序号 89 下对 A 填涂（参看答题卡）。

●在程序运行过程中，CPU 需要将指令从内存中取出并加以分析和执行。CPU 依据 (1) 来区分在内存中以二进制编码形式存放的指令和数据。

- (1) A. 指令周期的不同阶段 B. 指令和数据的寻址方式
C. 指令操作码的译码结果 D. 指令和数据所在的存储单元

●计算机在一个指令周期的过程中，为从内存读取指令操作码，首先要将 (2) 的内容送到地址总线上。

- (2) A. 指令寄存器 (IR) B. 通用寄存器 (GR) C. 程序计数器 (PC) D. 状态寄存器 (PSW)

●设 16 位浮点数，其中阶符 1 位、阶码值 6 位、数符 1 位、尾数 8 位。若阶码用移码表示，尾数用补码表示，则该浮点数所能表示的数值范围是 (3)。

- (3) A. $-2^{64} \sim (1-2^{-8}) 2^{64}$
B. $-2^{63} \sim (1-2^{-8}) 2^{63}$
C. $-2^{64} \sim (1-2^{-(1-2^{-8})}) 2^{64} \sim (1-2^{-8}) 2^{64}$
D. $-(1-2^{-8}) 2^{63} \sim (1-2^{-8}) 2^{63}$

●已知数据信息为 16 位，最少应附加 (4) 位校验位，以实现海明码纠错。

- (4) A. 3 B. 4 C. 5 D. 6

●将一条指令的执行过程分解为取指、分析和执行三步，按照流水方式执行，若取指时间 $t_{\text{取指}}=4\Delta t$ 、分析时间 $t_{\text{分析}}=2\Delta t$ 、执行时间 $t_{\text{执行}}=3\Delta t$ ，则执行完 100 条指令，需要的时间为 (5) Δt 。

- (5) A. 200 B. 300 C. 400 D. 405

●以下关于 Cache 与主存间地址映射的叙述中，正确的是 (6)。

- (6) A. 操作系统负责管理 Cache 与主存之间的地址映射
B. 程序员需要通过编程来处理 Cache 与主存之间的地址映射
C. 应用软件对 Cache 与主存之间的地址映射进行调度
D. 由硬件自动完成 Cache 与主存之间的地址映射

●可用于数字签名的算法是(7)。

- (7)A.RSA B.IDEA C.RC4 D.MD5

●(8)不是数字签名的作用。

- (8)A.接收者可验证消息来源的真实性 B.发送者无法否认发送过该消息
C.接收者无法伪造或篡改消息 D.可验证接收者合法性

●在网络设计和实施过程中要采取多种安全措施，其中(9)是针对系统安全需求的措施。

- (9)A.设备防雷击 B.入侵检测 C.漏洞发现与补丁管理 D.流量控制

●(10)的保护期限是可以延长的。

- (10)A.专利权 B.商标权 C.著作权 D.商业秘密权

●甲公司软件设计师完成了一项涉及计算机程序的发明。之后，乙公司软件设计师也完成了与甲公司软件设计师相同的涉及计算机程序的发明。甲、乙公司于同一天向专利局申请发明专利。此情形下，(11)是专利权申请人。

- (11)A.甲公司 B.甲、乙两公司 C.乙公司 D.由甲、乙公司协商确定的公司

●甲、乙两厂生产的产品类似，且产品都使用“B”商标。两厂于同一天向商标局申请商标注册，且申请注册前两厂均未使用“B”商标。此情形下，(12)能核准注册。

- (12)A.甲厂 B.由甲、乙厂抽签确定的厂 C.乙厂 D.甲、乙两厂

●在FM方式的数字音乐合成器中，改变数字载波频率可以改变乐音的(13)，改变它的信号幅度可以改变乐音的(14)。

- (13)A.音调 B.音色 C.音高 D.音质

- (14)A.音调 B.音域 C.音高 D.带宽

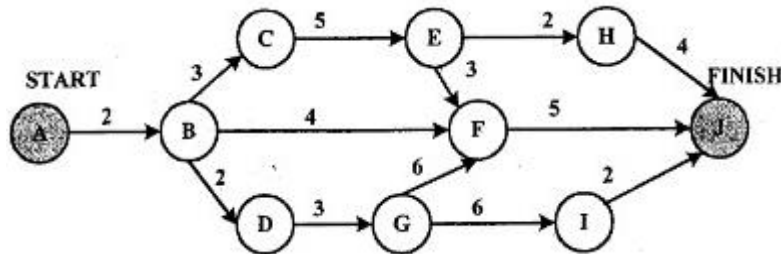
●结构化开发方法中，(15)主要包含对数据结构和算法的设计。

- (15) A. 体系结构设计 B. 数据设计 C. 接口设计 D. 过程设计

●在敏捷过程的开发方法中，(16)使用了迭代的方法，其中，把每段时间（30 天）一次的迭代称为一个“冲刺”，并按需求的优先级别来实现产品，多个自组织和自治的小组并行地递增实现产品。

- (16) A. 极限编程 XP B. 水晶法 C. 并列争球法 D. 自适应软件开发

●某软件项目的活动图如下图所示，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的数字表示相应活动的持续时间（天），则完成该项目的最少时间为(17)天。活动 BC 和 BF 最多可以晚开始(18)天而不会影响整个项目的进度。



- (17) A. 11 B. 15 C. 16 D. 18
 (18) A. 0 和 7 B. 0 和 11 C. 2 和 7 D. 2 和 11

●成本估算时，(19)方法以规模作为成本的主要因素，考虑多个成本驱动因子。该方法包括三个阶段模型，即应用组装模型、早期设计阶段模型和体系结构阶段模型。

- (19) A. 专家估算 B. Wolverton C. COCOMO D. COCOMO II

●逻辑表达式求值时常采用短路计算方式。“&&”、“||”、“!” 分别表示逻辑与、或、非运算，“&&”、“||” 为左结合，“!” 为右结合，优先级从高到低为 “!”、“&&”、“||”。对逻辑表达式 “x&& (y||!z)” 进行短路计算方式求值时，(20)。

- (20) A. x 为真，则整个表达式的值即为真，不需要计算 y 和 z 的值
 B. x 为假，则整个表达式的值即为假，不需要计算 y 和 z 的值
 C. x 为真，再根据 z 的值决定是否计算 y 的值
 D. x 为假，再根据 y 的值决定是否计算 z 的值

●常用的函数参数传递方式有传值与传引用两种。(21)。

(21)A.在传值方式下，形参与实参之间互相传值

B.在传值方式下，实参不能是变量

C.在传引用方式下，修改形参实质上改变了实参的值。

D.在传引用方式下，实参可以是任意的变量和表达式。

●二维数组 $a[1..N, 1..N]$ 可以按行存储或按列存储。对于数组元素 $a[i,j]$ ($1 \leq i,j \leq N$)，当(22)时，在按行和按列两种存储方式下，其偏移量相同。

(22)A. $i \neq j$

B. $i = j$

C. $i > j$

D. $i < j$

●实时操作系统主要用于有实时要求的过程控制等领域。实时系统对于来自外部的事件必须在(23)。

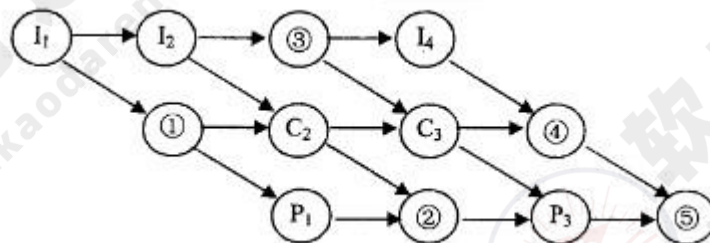
(23)A.一个时间片内进行处理

B.一个周转时间内进行处理

C.一个机器周期内进行处理

D.被控对象规定的时间内做出及时响应并对其进行处理

●假设某计算机系统中只有一个 CPU、一台输入设备和一台输出设备，若系统中有四个作业 T_1 、 T_2 、 T_3 和 T_4 ，系统采用优先级调度，且 T_1 的优先级 $> T_2$ 的优先级 $> T_3$ 的优先级 $> T_4$ 的优先级。每个作业 T_i 具有三个程序段：输入 I_i 、计算 C_i 和输出 P_i ($i=1, 2, 3, 4$)，其执行顺序为 $I_i \rightarrow C_i \rightarrow P_i$ 。这四个作业各程序段并发执行的前驱图如下所示。图中①、②分别为(24)，③、④、⑤分别为(25)。



(24)A. I_2 、 P_2

B. I_2 、 C_2

C. C_1 、 P_2

D. C_1 、 P_3

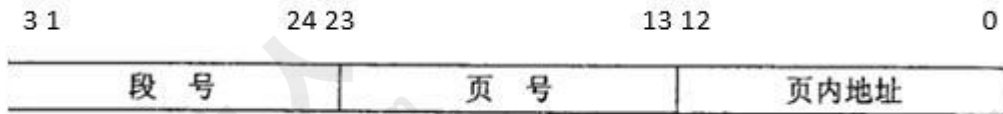
(25)A. C_2 、 C_4 、 P_4

B. I_2 、 I_3 、 C_4

C. I_3 、 P_3 、 P_4

D. I_3 、 C_4 、 P_4

●假设段页式存储管理系统中的地址结构如下图所示，则系统(26)。



- (26)A.最多可有 256 个段，每个段的大小均为 2048 个页，页的大小为 8K
 B.最多可有 256 个段，每个段最大允许有 2048 个页，页的大小为 8K
 C.最多可有 512 个段，每个段的大小均为 1024 个页，页的大小为 4K
 D.最多可有 512 个段，每个段最大允许有 1024 个页，页的大小为 4K

●假设系统中有 n 个进程共享 3 台扫描仪，并采用 PV 操作实现进程同步与互斥。若系统信号量 S 的当前值为 -1，进程 P_1 、 P_2 又分别执行了 1 次 $P(S)$ 操作，那么信号量 S 的值应为(27)。

- (27)A.3 B.-3 C.1 D.-1

●某字长为 32 位的计算机的文件管理系统采用位示图(bitmap)记录磁盘的使用情况。若磁盘的容量为 300GB，物理块的大小为 1MB，那么位示图的大小为(28)个字。

- (28)A.1200 B.3200 C.6400 D.9600

●某开发小组欲为一公司开发一个产品控制软件，监控产品的生产和销售过程，从购买各种材料开始，到产品的加工和销售进行全程跟踪。购买材料的流程、产品的加工过程以及销售过程可能会发生变化。该软件的开发最不宜采用(29)模型，主要是因为这种模型(30)。

- (29)A.瀑布 B.原型 C.增量 D.喷泉
 (30)A.不能解决风险 B.不能快速提交软件
 C.难以适应变化的需求 D.不能理解用户的需求

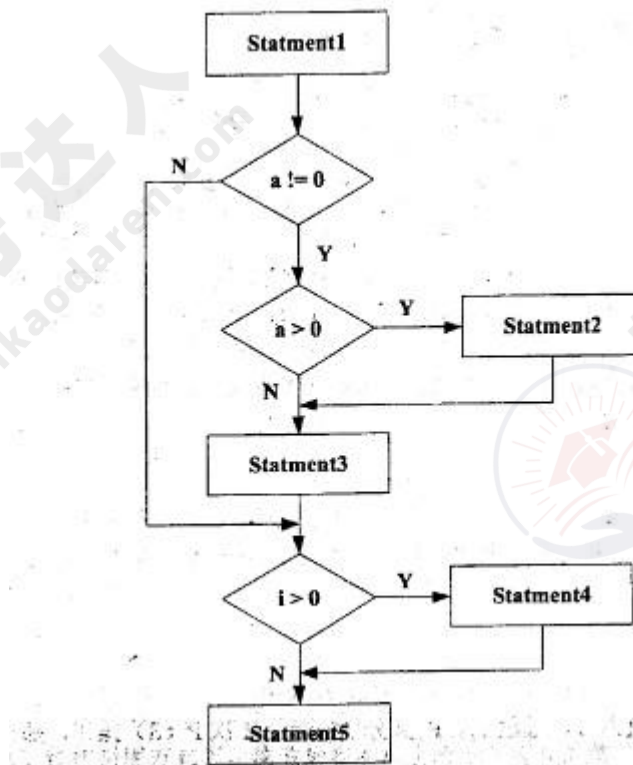
●(31)不属于软件质量特性中的可移植性。

- (31)A.适应性 B.易安装性 C.易替换性 D.易理解性

●(32)对下图所示流程图采用白盒测试方法进行测试，若要满足路径覆盖，则至少需要

2016 年下半年 软件设计师 上午试卷 第 6 页 (共 14 页)

(32) 个测试用例。采用 McCabe 度量法计算该程序的环路复杂性为 (33)。



(32) A.3 B.4 C.6 D.8

(33) A.1 B.2 C.3 D.4

● 计算机系统的 (34) 可以用 $MTBF / (1 + MTBF)$ 来度量，其中 MTBF 为平均失效间隔时间。

(34) A.可靠性 B.可用性 C.可维护性 D.健壮性

● 以下关于软件测试的叙述中，不正确的是 (35)。

(35) A.在设计测试用例时应考虑输入数据和预期输出结果

B.软件测试的目的是证明软件的正确性

C.在设计测试用例时，应该包括合理的输入条件

D.在设计测试用例时，应该包括不合理的输入条件

● 某模块中有两个处理 A 和 B，分别对数据结构 X 写数据和读数据，则该模块的内聚类型为 (36) 内聚。

(36) A.逻辑 B.过程 C.通信 D.内容

●在面向对象方法中，不同对象收到同一消息可以产生完全不同的结果，这一现象称为(37)。在使用时，用户可以发送一个通用的消息，而实现的细节则由接收对象自行决定。

- (37)A.接口 B.继承 C.覆盖 D.多态

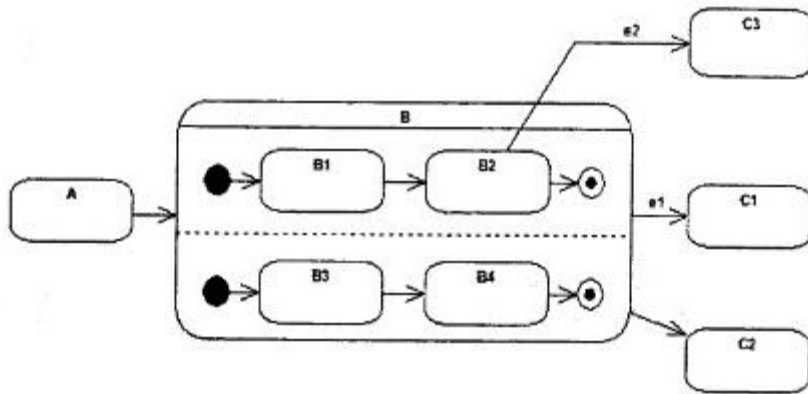
●在面向对象方法中，支持多态的是(38)。

- (38)A.静态分配 B.动态分配 C.静态类型 D.动态绑定

●面向对象分析的目的是为了获得对应用问题的理解，其主要活动不包括(39)。

- (39)A.认定并组织对象 B.描述对象间的相互作用
C.面向对象程序设计 D.确定基于对象的操作

●如下所示的 UML 状态图中，(40)时，不一定会离开状态 B。



- (40)A.状态 B 中的两个结束状态均达到 B.在当前状态为 B2 时，事件 e2 发生
C.事件 e2 发生 D.事件 e1 发生

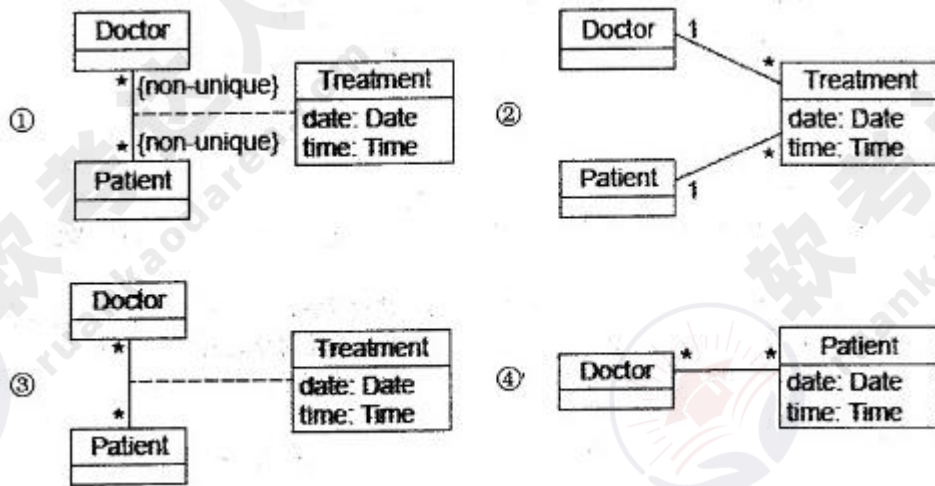
●以下关于 UML 状态图中转换（transition）的叙述中，不正确的是(41)。

- (41)A.活动可以在转换时执行也可以在状态内执行
B.监护条件只有在相应的事件发生时才进行检查
C.一个转换可以有事件触发器、监护条件和一个状态
D.事件触发转换

●下图①②③④所示是 UML(42)。现有场景：一名医生（Doctor）可以治疗多位病人（Patient），一位病人可以由多名医生治疗，一名医生可能多次治疗同一位病人。要记录哪

2016 年下半年 软件设计师 上午试卷 第 8 页 （共 14 页）

名医生治疗哪位病人时，需要存储治疗（Treatment）的日期和时间。以下①②③④图中（43）。
是描述此场景的模型。



- (42) A.用例图 B.对象图 C.类图 D.协作图
(43) A.① B.② C.③ D.④

●（44）模式定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换，使得算法可以独立于使用它们的客户而变化。以下（45）情况适合选用该模式。

- ①一个客户需要使用一组相关对象 ②一个对象的改变需要改变其它对象
③需要使用一个算法的不同变体 ④许多相关的类仅仅是行为有异

- (44) A.命令（Command） B.责任链（Chain of Responsibility）
C.观察者（Observer） D.策略（Strategy）

- (45) A.①② B.②③ C.③④ D.①④

●（46）模式将一个复杂对象的构建与其表示分离，使得同样的构建过程可以创建不同的表示。以下（47）情况适合选用该模式。

- ①抽象复杂对象的构建步骤
②基于构建过程的具体实现构建复杂对象的不同表示
③一个类仅有一个实例
④一个类的实例只能有几个不同状态组合中的一种

- (46) A.生成器（Builder） B.工厂方法（Factory Method）
C.原型（Prototype） D.单例（Singleton）

(47)A.①② B.②③ C.③④ D.①④

●由字符 a、b 构成的字符串中，若每个 a 后至少跟一个 b，则该字符串集合可用正规式表示为 (48)。

(48)A. $(b|ab)^*$ B. $(ab^*)^*$ C. $(a^*b^*)^*$ D. $(a|b)^*$

●乔姆斯基 (Chomsky) 将文法分为 4 种类型，程序设计语言的大多数语法现象可用其中的 (49) 描述。

(49)A. 上下文有关文法 B. 上下文无关文法 C. 正规文法 D. 短语结构文法

●运行下面的 C 程序代码段，会出现 (50) 错误。

```
int k=0;
for(;k<100;);
{k++;}
```

(50)A. 变量未定义 B. 静态语义 C. 语法 D. 动态语义

●在数据库系统中，一般由 DBA 使用 DBMS 提供的授权功能为不同用户授权，其主要目的是为了保证数据库的 (51)。

(51)A. 正确性 B. 安全性 C. 一致性 D. 完整性

●给定关系模式 $R(U, F)$ ，其中：U 为关系模式 R 中的属性集，F 是 U 上的一组函数依赖。假设 $U=\{A_1, A_2, A_3, A_4\}$ ， $F=\{A_1 \rightarrow A_2, A_1 A_2 \rightarrow A_3, A_1 \rightarrow A_4, A_2 \rightarrow A_4\}$ ，那么关系 R 的主键应为 (52)。函数依赖集 F 中的 (53) 是冗余的。

(52)A. A_1 B. $A_1 A_2$ C. $A_1 A_3$ D. $A_1 A_2 A_3$

(53)A. $A_1 \rightarrow A_2$ B. $A_1 A_2 \rightarrow A_3$ C. $A_1 \rightarrow A_4$ D. $A_2 \rightarrow A_4$

●给定关系 $R(A, B, C, D)$ 和关系 $S(A, C, E, F)$ ，对其进行自然连接运算 $R \bowtie S$ 后的属性列为 (54) 个；与 $\sigma_{R.B > S.E}(R \bowtie S)$ 等价的关系代数表达式为 (55)。

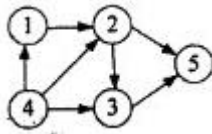
(54)A. 4 B. 5 C. 6 D. 8

- (55) A. $\sigma_{2>7}(R \times S)$ B. $\pi_{1,2,3,4,7,8}(\sigma_{1=5 \wedge 2>7 \wedge 3=6}(R \times S))$
 C. $\sigma_{2>'7'}(R \times S)$ D. $\pi_{1,2,3,4,7,8}(\sigma_{1=5 \wedge 2>'7' \wedge 3=6}(R \times S))$

●下列查询 B=“大数据”且 F=“开发平台”，结果集属性列为 A、B、C、F 的关系代数表达式中，查询效率最高的是 (56)。

- (56) A. $\pi_{1,2,3,8}(\sigma_{2='大数据' \wedge 1=5 \wedge 3=6 \wedge 8='开发平台'}(R \times S))$
 B. $\pi_{1,2,3,8}(\sigma_{1=5 \wedge 3=6 \wedge 8='开发平台'}(\sigma_{2='大数据'}(R) \times S))$
 C. $\pi_{1,2,3,8}(\sigma_{2='大数据' \wedge 1=5 \wedge 3=6}(R \times \sigma_{4='开发平台'}(S)))$
 D. $\pi_{1,2,3,8}(\sigma_{1=5 \wedge 3=6}(\sigma_{2='大数据'}(R) \times \sigma_{4='开发平台'}(S)))$

●拓扑序列是有向无环图中所有顶点的一个线性序列，若有向图中存在弧 $\langle v, w \rangle$ 或存在从顶点 v 到 w 的路径，则在该有向图的任一拓扑序列中， v 一定在 w 之前。下面有向图的拓扑序列是 (57)。



- (57) A. 41235 B. 43125 C. 42135 D. 41325

●设有一个包含 n 个元素的有序线性表。在等概率情况下删除其中的一个元素，若采用顺序存储结构，则平均需要移动 (58) 个元素；若采用单链表存储，则平均需要移动 (59) 个元素。

- (58) A. 1 B. $(n-1)/2$ C. $\log n$ D. n
 (59) A. 0 B. 1 C. $(n-1)/2$ D. $n/2$

●具有 3 个节点的二叉树有 (60) 种形态。

- (60) A. 2 B. 3 C. 5 D. 7

●以下关于二叉排序树（或二叉查找树、二叉搜索树）的叙述中，正确的是 (61)。

- (61) A. 对二叉排序树进行先序、中序和后序遍历，都得到结点关键字的有序序列

- B.含有 n 个结点的二叉排序树高度为 $(\log_2 n) + 1$
- C.从根到任意一个叶子结点的路径上，结点的关键字呈现有序排列的特点
- D.从左到右排列同层次的结点，其关键字呈现有序排列的特点

●下表为某文件中字符的出现频率，采用霍夫曼编码对下列字符编码，则字符序列“bee”的编码为(62)；编码“110001001101”的对应的字符序列为(63)。

字符	a	b	c	d	e	f
频率(%)	45	13	12	16	9	5

- (62) A.10111011101 B.10111001100 C.001100100 D.110011011
- (63) A.bad B.bee C.face D.bace

●两个矩阵 $A_{m \times n}$ 和 $B_{n \times p}$ 相乘，用基本的方法进行，则需要的乘法次数为 $m \times n \times p$ 。多个矩阵相乘满足结合律，不同的乘法顺序所需要的乘法次数不同。考虑采用动态规划方法确定 $M_i, M_{(i+1)}, \dots, M_j$ 多个矩阵连乘的最优顺序，即所需要的乘法次数最少。最少乘法次数用 $m[i, j]$ 表示，其递归式定义为：

$$m[i, j] = \begin{cases} 0 & i \geq j \\ \min_{i \leq k \leq j} \{m[i, k] + m[k+1, j] + p_{i-1}p_kp_j\} & i < j \end{cases}$$

其中 i, j 和 k 为矩阵下标，矩阵序列中 M_i 的维度为 $(p_{i-1}) \times p_i$ 采用自底向上的方法实现该算法来确定 n 个矩阵相乘的顺序，其时间复杂度为(64)。若四个矩阵 M_1, M_2, M_3, M_4 相乘的维度序列为 2、6、3、10、3，采用上述算法求解，则乘法次数为(65)。

- (64) A. $O(n^2)$ B. $O(n^2 \lg n)$ C. $O(n^3)$ D. $O(n^3 \lg n)$
- (65) A.156 B.144 C.180 D.360

●以下协议中属于应用层协议的是(66)，该协议的报文封装在(67)。

- (66) A.SNMP B.ARP C.ICMP D.X.25
- (67) A.TCP B.IP C.UDP D.ICMP

●某公司内部使用 `wb.xyz.com.cn` 作为访问某服务器的地址，其中 `wb` 是(68)。

- (68) A.主机名 B.协议名 C.目录名 D.文件名

●如果路由器收到了多个路由协议转发的关于某个目标的多条路由，那么决定采用哪条路由的策略是(69)。

- (69)A.选择与自己路由协议相同的
B.选择路由费用最小的
C.比较各个路由的管理距离
D.比较各个路由协议的版本

●与地址 220.112.179.92 匹配的路由表的表项是(70)。

- (70)A.220.112.145.32/22
B.220.112.145.64/22
C.220.112.147.64/22
D.220.112.177.64/22

● Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a (71), open or closed. In this respect software systems differ profoundly from computers, buildings, or automobiles, where repeated elements abound.

Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard. Software systems have orders of magnitude more (72) than computers do.

Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some (73) fashion, and the complexity of the whole increases much more than linearly.

The complexity of software is a(an) (74) property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence. Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties experimentally. This worked because the complexities (75) in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.

Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management

problems as well come from the complexity.

- | | | | |
|---------------|-------------|--------------|---------------|
| (71)A.task | B.job | C.subroutine | D.program |
| (72)A.states | B.parts | C.conditions | D.expressions |
| (73)A.linear | B.nonlinear | C.parallel | D.additive |
| (74)A.surface | B.outside | C.exterior | D.essential |
| (75)A.fixed | B.included | C.ignored | D.stabilized |