

2025 年 05 月软件设计师真题回忆版

综合知识

部分真题还原如下：

利用栈对算术表达式 $a * (b + c) - e$ 求值，设栈初始时空，则暂存操作数（或运算结果）的栈中最多需存放（ ）个操作数（或运算结果）。

- A. 3
- B. 1
- C. 2
- D. 4

答案：A

解析：执行过程：

1. 压入 a, b, c （栈中有 3 个操作数）；
2. 计算 $b + c$ 后，压入结果（栈剩 $a + \text{结果}$ ）；
3. 计算 $a * (\text{结果})$ 后，压入最终值；
4. 压入 e 并计算减法。

最大栈深度为 3。

在二维平面最近点对问题中，分治法的步骤不包括以下（ ）。

- A. 计算所有点对的欧氏距离
- B. 递归求解左右两半中点集的最近点对问题
- C. 按 x 坐标排序并将点集划分为左右两半
- D. 合并时仅需检查距离中线 δ 范围内的点

答案：A

解析：分治法的核心步骤为：排序并划分点集（C）、递归求解子问题（B）、合并时检查中线附近有限范围内的点（D）。而选项 A “计算所有点对的欧氏距离” 是暴力解法的步骤，不属于分治法

已知字符集为 $\{a, b, c, d, e, f, g, h\}$ ，若各字符的哈夫曼编码如下表所示，则对编码序列 010101100011001001001111 的译码结果为（ ）。

字符	a	b	c	d	e	f	g	h
哈夫曼编码	0100	10	0000	0101	001	011	11	0001

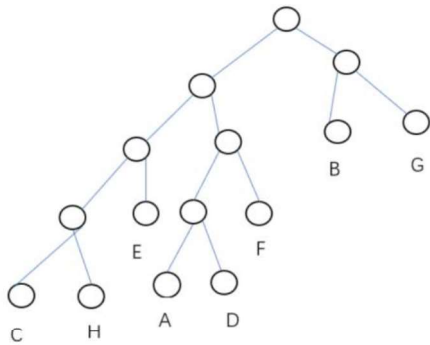
- A. daceabdg
- B. dabccabg

C. dfhbabfg

D. dfaecbfg

答案：C

解析：哈夫曼树如图所示：



在以阶段划分的编译器中，语义分析阶段的任务包括（）。

- ①识别记号 ②识别句子结构 ③检查程序中的语法错误
④填写符号表 ⑤生成中间代码

- A. ②③
B. ③④
C. ①②
D. ④⑤

答案：D

解析：语义分析阶段任务：语义分析主要负责检查源程序的语义正确性，包括类型检查、符号表管理等。

选项④：填写符号表是语义分析阶段的重要任务，用于记录程序中各种标识符的信息。

选项⑤：生成中间代码也是语义分析阶段的任务，将源程序转换为中间表示形式，便于后续的代码生成和优化。

选项①：识别记号是词法分析阶段的任务。

选项②：识别句子结构是语法分析阶段的任务。

选项③：检查程序中的语法错误是语法分析阶段的主要任务。

浮点数加减运算时，（）可能导致阶码上溢。

- A. 右规
B. 对阶
C. 尾数运算
D. 左规

答案：C

解析：选项 C：在浮点数加减运算中，尾数运算可能会产生进位，如果进位导致阶码超出了所能表示的最大范围，就会出现阶码上溢。

选项 A：右规是尾数右移，阶码加 1，一般不会导致阶码上溢。

选项 B：对阶是小阶向大阶看齐，阶码不会超出范围。

选项 D：左规是尾数左移，阶码减 1，不会导致阶码上溢。

() 用于在网络中向一组特定的设备发送数据包。

- A. 网络地址
- B. 广播地址
- C. 组播地址
- D. 单播地址

答案：C

解析：选项 C：组播地址用于在网络中向一组特定的设备发送数据包，只有加入了相应组播组的设备才能接收到组播数据包。

选项 A：网络地址是标识网络的地址，不是用于向特定设备组发送数据包的。

选项 B：广播地址用于向网络中的所有设备发送数据包，而不是特定的一组设备。

选项 D：单播地址用于在网络中向单个设备发送数据包。

某软件系统在测试过程中，() 最适合发现模块之间的接口问题。

- A. 单元测试
- B. 系统测试
- C. 验收测试
- D. 集成测试

答案：D

解析：选项 D：集成测试是将多个模块组合在一起进行测试，主要目的是发现模块之间的接口问题，如数据传递错误、模块间调用关系异常等。

选项 A：单元测试主要针对单个模块进行测试，侧重于模块内部的功能和逻辑。

选项 B：系统测试是对整个系统进行测试，验证系统是否满足需求规格说明书的要求，更关注系统的整体功能和性能。

选项 C：验收测试是由用户或客户进行的测试，用于确认系统是否满足业务需求。

用户 A 通过 SMTP/MIME 协议在邮件客户端（如 Outlook）中撰写邮件正文，并添加一个 Excel 附件（财务数据）发送给用户 B，该邮件采用的是 ()。

- A. 正文加密，附件明文传输
- B. 正文、附件均加密传输
- C. 正文明文，附件加密传输
- D. 正文、附件明文传输

答案：D

解析：本题答案选 D。SMTP（简单邮件传输协议）和 MIME（多用途互联网邮件扩展）协议本身并不对邮件正文和附件进行加密。当用户 A 通过这些协议在邮件客户端中撰写邮件正文并添加附件发送给用户 B 时，邮件正文和附件通常是以明文形式传输的，除非使用了额外的加密措施。

对某有序表进行折半查找时，参与比较的关键字顺序不可能是 ()。

- A. 43, 65, 99, 85, 78

B. 99, 85, 78, 65, 43

C. 43, 99, 85, 65, 78

D. 99, 85, 65, 78, 43

答案：D

解析：本题答案选 D。折半查找每次将查找区间缩小一半，参与比较的关键字应该是逐渐逼近目标关键字的。在有序表中进行折半查找，每次比较的关键字应该是当前查找区间的中间元素。如果有序表是升序排列，每次比较的关键字应该是先中间元素，然后根据比较结果缩小查找区间，再取新的中间元素比较。选项 D 中，65 之后是 78，再之后应该是比 65 大的数，不可能是 43，不符合折半查找的比较顺序。

用数据流图对某医院挂号系统如下需求进行建模：

- (1) 患者通过终端提交挂号请求；
- (2) 系统验证患者信息；
- (3) 系统生成挂号单，并存储到挂号记录。

以下（）不是数据流。

- A. 挂号请求
- B. 患者信息
- C. 挂号记录
- D. 挂号单

答案：C

解析：数据流：是指数据在系统中流动的路径，它代表了数据的传递和处理过程。

选项 A：挂号请求是患者通过终端提交给系统的数据，是数据流。

选项 B：患者信息在系统验证过程中流动，属于数据流。

选项 C：挂号记录是系统存储挂号单的地方，是数据存储，不是数据流。

选项 D：挂号单是系统生成并在系统内流动的数据，是数据流。

设链队列 Q 用含头结点的循环单链表表示，且仅设尾指针 rear，假设队列中有 n 个元素结点，则入队列和出队列运算的时间复杂度分别为（）。

- A. $O(1)$ 、 $O(1)$
- B. $O(1)$ 、 $O(n)$
- C. $O(n)$ 、 $O(1)$
- D. $O(n)$ 、 $O(n)$

答案：A

解析：入队列操作：因为链队列 Q 用含头结点的循环单链表表示且仅设尾指针 rear，入队列时只需在尾指针 rear 之后插入新结点，然后更新尾指针，操作时间复杂度为 $O(1)$ 。

出队列操作：可以通过尾指针 rear 找到头结点的下一个结点（即队头结点），删除该结点并更新头结点的指针，时间复杂度也为 $O(1)$ 。

编译器将频繁使用的临时变量（如循环计数器）优化到（）中，可以提升访问速度。

- A. 栈

- B. 堆
- C. 静态存储区
- D. 寄存器

答案：D

解析：选项 D：寄存器是 CPU 内部的高速存储单元，编译器将频繁使用的临时变量（如循环计数器）优化到寄存器中，CPU 可以直接快速访问这些变量，从而提升访问速度。

选项 A：栈主要用于存储函数调用的局部变量、参数等，访问速度相对寄存器较慢。

选项 B：堆用于动态分配内存，其访问速度也不如寄存器。

选项 C：静态存储区用于存储全局变量和静态变量，访问速度也没有寄存器快。

QoS 是网络的一种安全机制，确保重要业务量不会延迟或丢弃。通常情况下 QoS 被部署在（）上保证网络的高效运行。

- A. 网闸
- B. 防火墙
- C. 路由器
- D. IDS

答案：C

解析：选项 C：QoS（Quality of Service）即服务质量，路由器可以根据网络流量情况和业务优先级，对数据包进行分类、调度和队列管理等操作，从而保证重要业务量不会延迟或丢弃，所以通常将 QoS 部署在路由器上。

选项 A：网闸主要用于实现不同安全级别网络之间的安全隔离和数据交换，不主要用于 QoS 部署。

选项 B：防火墙主要用于控制网络访问，防止非法入侵，和 QoS 的主要功能不同。

选项 D：IDS（入侵检测系统）主要用于检测网络中的入侵行为，与 QoS 的作用不同。

下列各项中，具有法定时间性的知识产权是（）。

- ①商业秘密权 ②专利权 ③商标权 ④著作权

- A. 仅①③
- B. 仅②④
- C. ①②③④
- D. 仅②③④

答案：D

解析：专利权：专利权有明确的保护期限，如发明专利权的期限为二十年，实用新型专利权的期限为十年，外观设计专利权的期限为十五年，自申请日起计算。

商标权：商标权的有效期为十年，自核准注册之日起计算，期满可以续展。

著作权：著作权的保护期有一定的时间限制，如公民的作品，其发表权、使用权和获得报酬权的保护期为作者终生及其死亡后五十年。

商业秘密权：商业秘密权没有法定的时间限制，只要商业秘密符合秘密性、价值性和保密性等条件，其权利就受法律保护。

在一个气象监测系统中，天气变化会影响多种显示设备（如 PC 应用和手机应用）。这些设备需要及时更新。这一需求适合采用（ ）设计模式进行设计。

- A. 观察者（Observer）
- B. 备忘录（Memento）
- C. 策略（Strategy）
- D. 状态（State）

答案：A

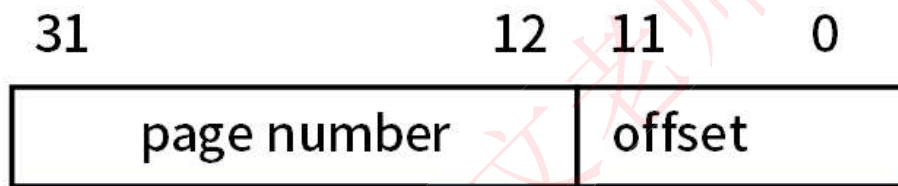
解析：选项 A：观察者模式定义了一种一对多的依赖关系，让多个观察者对象同时监听一个主题对象。在气象监测系统中，天气变化这个主题对象发生改变时，PC 应用和手机应用等观察者对象可以及时收到通知并更新显示，符合本题需求。

选项 B：备忘录模式主要用于保存一个对象的某个状态，以便在适当的时候恢复对象，与本题及时更新显示设备的需求无关。

选项 C：策略模式定义了一系列的算法，并将每个算法封装起来，使它们可以相互替换，和本题场景不相关。

选项 D：状态模式允许一个对象在其内部状态改变时改变它的行为，本题重点是主题对象通知观察者更新，并非对象状态改变导致行为改变。

在分页管理系统中，逻辑地址由页面编号和偏移值构成。如下所示的逻辑地址表示，（）。



- A. 页面大小为 2K，每个页面最大为 8M
- B. 页面大小为 4K，每个页面最大为 1M
- C. 页面大小为 8K，每个页面最大为 2M
- D. 页面大小为 1K，每个页面最大为 16M

答案：B

解析：1. 页内偏移量的位数决定页面大小：

偏移量占 0~11 位，共 12 位，对应的页面大小为 $2^{12}=4KB=4KB$

2. 页号的位数决定页面总数：

页号占 12~31 位，共 20 位，因此最大页数为 $2^{20}=1M$

关于开源软件的著作权，以下说法正确的是（）。

- A. 开源软件不涉及著作权问题
- B. 开源软件的著作权归使用者所有
- C. 开源软件的著作权归开发者所有，使用者不能进行任何修改
- D. 开源软件的著作权归开发者所有，但使用者可以自由使用、修改和分发

答案：D

解析：开源软件是指其源代码可以被公众使用、修改和分发的软件。虽然开源软件的著作权归开发者所有，但是遵循开源许可协议，使用者可以自由使用、修改和分发该软件。

A 选项：开源软件同样涉及著作权问题，只是其著作权管理遵循特定的开源许可协议。

B 选项：开源软件的著作权归开发者所有，而非使用者。

C 选项：使用者在遵循开源许可协议的前提下，可以对开源软件进行修改。所以本题选 D。

以下关于数据流图分层结构的叙述中，不正确的是（）。

A. 各层数据流图之间应该保持“平衡”关系

B. 顶层数据流图只包含一个处理框，表示待开发的系统

C. 数据流图的层次越多，细节程度越低

D. 分层的数据流图可以清晰的表达系统的层次结构，使得系统更易于理解

答案：C

解析：A 选项：各层数据流图之间应该保持“平衡”关系，即子图的输入输出数据流必须和父图中相应处理的输入输出数据流一致，该说法正确。

B 选项：顶层数据流图只包含一个处理框，表示待开发的系统，它抽象地描述了系统的功能，该说法正确。

C 选项：数据流图的层次越多，细节程度越高，而不是越低。随着分层细化，会逐步展现更多的处理过程和数据流动细节。所以 C 选项说法错误。

D 选项：分层的数据流图可以清晰的表达系统的层次结构，使得系统更易于理解，通过将复杂系统分解为多个层次的子图，便于不同人员对系统进行分析和设计，该说法正确。

（）攻击的技术实现路径主要是通过合法开发流程渗透，利用代码混淆、数字证书伪装绕过审查。

A. 零日漏洞

B. 撞库攻击

C. 供应链投毒

D. AI 赋能攻击

答案：C

解析：A 选项：零日漏洞是指被发现后立即被恶意利用的安全漏洞，主要是利用系统未修复的漏洞进行攻击，并非通过合法开发流程渗透等方式。

B 选项：撞库攻击是黑客通过收集互联网已泄露的用户和密码信息，生成对应的字典表，尝试批量登录其他网站，与题干中描述的技术实现路径不符。

C 选项：供应链投毒攻击的技术实现路径主要是通过合法开发流程渗透，利用代码混淆、数字证书伪装绕过审查，将恶意代码注入到软件供应链中。所以 C 选项正确。

D 选项：AI 赋能攻击是利用人工智能技术来增强攻击的效果和效率，和题干中的技术实现方式不同。

在关系数据库中，第三范式的主要目的是消除（ ）类型的问题。

A. 多值依赖

B. 传递依赖

C. 外键冗余

D. 主键冗余

答案：B

解析：在关系数据库中，第一范式（1NF）要求数据库表的每一列都是不可分割的基本数据项；第二范式（2NF）在满足 1NF 的基础上，消除了非主属性对主键的部分依赖；第三范式（3NF）是在满足 2NF 的基础上，消除非主属性对主键的传递依赖。多值依赖是第四范式（4NF）要处理的问题；外键冗余和主键冗余并非第三范式主要解决的问题。所以本题选 B。

某模块的各个部分，前一部分处理的输出是后一部分处理的输入，则该模块的内聚类型为（ ）。

A. 顺序内聚

B. 功能内聚

C. 通信内聚

D. 巧合内聚

答案：A

解析：顺序内聚：一个模块中各个处理元素和同一个功能密切相关，而且这些处理必须顺序执行，通常前一个处理元素的输出是后一个处理元素的输入。

功能内聚：模块内所有元素的各个组成部分全部都为完成同一个功能而存在，共同完成一个单一的功能，模块已不可再分。

通信内聚：指模块内所有处理元素都在同一个数据结构上操作或所有处理功能都通过公用数据而发生关联

巧合内聚：一个模块内的各处理元素之间没有任何联系，只是偶然地被凑到一起。

综上，本题答案选 A。

某电商平台“将订单处理、支付网关集成、物流跟踪等功能拆分为独立服务。每个服务通过 API 通信”，这是采用了（ ）。

A. 分层架构

B. 事件驱动架构

C. 面向对象架构

D. 微服务架构

答案：D

解析：微服务架构（D）：将系统拆分为独立部署的服务（订单处理、支付网关），通过 API 通信，符合高内聚低耦合原则。

对比其他架构：

分层架构（A）：按功能层划分（如展示层、业务层），不强调服务独立性。

面向对象（C）：以类和对象为核心，与分布式服务无关。

事件驱动（B）：通过事件队列通信，非本题描述的 API 直接调用场景。

下列事件中，会触发软件中断的是（ ）。

A. 电源故障

B. 按键输入

- C. 除零错误
- D. 定时器溢出

答案：C

解析：中断分类：

软件中断（C）：由程序错误（如除零、非法指令）触发，CPU 主动抛出异常。

硬件中断（B/D）：按键输入（B）、定时器溢出（D）属于外部设备触发。

电源故障（A）：系统级硬件故障，非程序逻辑引起。

某递归算法的时间复杂度计算公式为 $T(n) = 4T(n/2) + n \lg n$ ，其中 n 为问题规模，则该算法的时间复杂度是（ ）。

- A. $\Theta(n \lg n)$
- B. $\Theta(n^3)$
- C. $\Theta(n^2)$
- D. $\Theta(n^2 \lg n)$

答案：C

解析：递归式 $T(n) = 4T(n/2) + n \lg n$ 的时间复杂度可通过 主定理 分析：

1. 主定理形式：对形如 $T(n) = aT(n/b) + f(n)$ 的递归式，主定理通过比较 $f(n)$ 与 $n^{\log_b a}$ 的阶来确定时间复杂度。

2. 参数分析：

$a = 4, b = 2 \rightarrow \log_b a = \log_2 4 = 2$ ，因此 $n^{\log_b a} = n^2$ 。

$f(n) = n \lg n$ ，需与 n^2 比较阶。

3. 主定理情况判断：若 $f(n) = O(n^{\log_b a - \epsilon})$ （存在 $\epsilon > 0$ ），则 $T(n) = \Theta(n^{\log_b a})$ 。这里 $n \lg n$ 的阶低于 n^2 （因为 $\lg n$ 的增长速度远慢于 n ），且存在 $\epsilon = 0.5$ 使得 $n \lg n = O(n^{2 - 0.5}) = O(n^{1.5})$ 。因此满足 情况 1，时间复杂度为 $\Theta(n^2)$ 。

某银行开发核心交易系统，需严格遵循监管审计且需求变更极少，最适合的开发模型是（ ）。

- A. 敏捷开发（Scrum）
- B. 快速应用开发（RAD）
- C. 增量模型
- D. 瀑布模型

答案：D

解析：瀑布模型（D）：需求明确且变更少时，阶段性交付文档（如需求规格书、设计文档）便于监管审计，符合银行系统的高合规性要求。

敏捷模型（A）：适用于需求频繁变更的场景，与严格监管的文档固化需求矛盾。

RAD（B）与增量模型（C）：侧重快速迭代，难以满足核心系统的高稳定性要求。

在软件项目团队中，成员的技能水平和协作能力差异较大，导致项目进度受阻。为提高团队整体效率，项目经理应优先考虑（ ）。

- A. 重新分配团队成员任务

- B. 引入外部专家指导
- C. 增加项目奖励机制
- D. 开展内部技能分享与培训

答案：D

解析：技能差异的根本解决：成员技能差异直接影响代码质量和协作效率，内部培训（D）直接提升团队基线能力，长期效果最优。

其他选项对比：

任务重分配（A）：仅临时调整分工，未解决能力瓶颈。

外部专家（B）：短期指导难以转化为团队持续能力。

奖励机制（C）：未针对性提升技能，对复杂任务效果有限。

以下问题中，肯定可以用贪心算法求得最优解的是（ ）。

- A. 旅行商问题（TSP）
- B. 最大公共子序列（LCS）
- C. 部分（分数）背包问题
- D. 0-1 背包问题

答案：C

解析：贪心算法适用场景需满足贪心选择性质与最优子结构：

部分（分数）背包问题（C）：允许分割物品，贪心策略按单位价值排序选取，能保证全局最优。

旅行商问题（A）：贪心选择最短边可能陷入局部最优，无法保证全局最短路径。

0-1 背包问题（D）：不可分割物品，贪心策略可能错过更高总价值的组合。

最大公共子序列（B）：需动态规划递推求解，贪心无法处理重叠子问题

ISO/IEC 25010：2023 标准中，（ ）不属于可维护性的子特性。

- A. 可修改性（Modifiability）
- B. 可分析性（Analysability）
- C. 模块化（Modularity）
- D. 容错性（Fault Tolerance）

答案：D

解析：可维护性子特性包括可修改性、可分析性、模块化等，容错性属于可靠性范畴

采用面向对象设计方法开发电商平台，设计负责处理用户订单的类 OrderService，它不仅处理订单创建，还负责记录订单日志和发送确认邮件。OrderService 类违反了面向对象设计（ ）原则。

- A. 里氏替换
- B. 单一职责
- C. 接口隔离
- D. 开放-封闭

答案：B

解析：OrderService 同时负责订单创建、日志记录和邮件发送，承担了多个职责，违反单一职责原则

对 n 阶上三角矩阵 A 按行压缩存储，将元素 A_{ij} ($0 \leq i, j < n$ 且 $i \leq j$) 存储在 $B[k]$ ($k \geq 1$) 中，则 k 的值为 ()。

- A. $-i^2/2 + (n+1/2)i + j - n$
- B. $-i^2/2 + (n+1/2)i + j - 1$
- C. $-i^2/2 + (n-1/2)i + j$
- D. $-i^2/2 + (n-1/2)i + j + 1$

答案：D

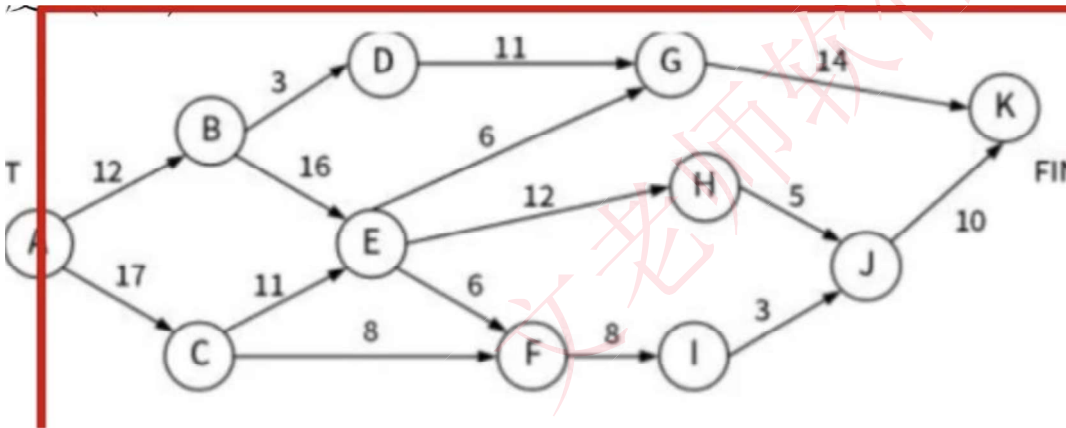
解析：代入法直接验证。

元素 $A[0][0]$, $A[0][1]$ 应是分别对应 $B[1]$, $B[2]$ 。

依次代入验证即可。

部分知识点回忆如下：

1. 给一个图，哪一点不在关键路径上，最迟完成时间
(题目有四条路径都是 28，求 JK 最迟完成时间)



(不是原图，找了一个类似的，有好几个关键路径)

2. 斐波那契数列时间复杂度 (题目给出递归公式)

fibonacci(n):

```
if n == 1:
```

```
    return 1
```

```
if n == 2:
```

```
    return 1
```

```
    return fibonacci(n-1) + fibonacci(n-2)
```

求它的时间复杂度 ()

A $O(n)$ B $O(n \log n)$ C $O(n^2)$ D $O(n^2 \log n)$

3. osi 数据库 JDBC 或 ODBC 模型实现功能，数据格式转换在哪一层

JDBC/ODBC 属于哪一层：

A 应用层 B 会话层 C 表示层 D (传输层)

客户端通过 JDBC/ODBC 发送 SQL 给数据库服务器，收到查询结果后转换成客户端可用的显示方

式,是属于哪一层:

A 会话层 B 表示层 C 应用层

4. linux 里的外设常被当做特殊的文件,一般存放在() 路径里。

A. /etc B. /var C /dev D/root

Linux 里的 sdc 文件是什么设备:

A IDE 磁盘 B 光盘 C SCSI 磁盘 D Floppy 盘

5. 循环链队列,只有尾指针和头结点,则入队列和出队列的时间复杂度分别是多少?

A. $O(1)$ 和 $O(n)$ B. $O(1)$ 和 $O(1)$ C. $O(n)$ 和 $O(n)$ D. $O(n)$ 和 $O(1)$

6. 在计算机网络中, A 类地址第一个字节地址范围是 ()

A. 1-127 B. 0-127 C. 1-126 D 0-126

7. 用 SMTP 和 MIME 协议传送 outlook 文档,他的正文,还带一个 Excel 附件(财务信息),如何传输更合适

A. 正文明文,附件加密传输, B 正文附件均加密 C. 正文加密,附件不加密 D. 附件、正文都明文

8. 在数据库中,以下那个不能成为关系模式的属性 ()

A. 简单属性, B. 组合属性 C. 多值属性 D, 派生属性

9. 在数据库中,第三范式可以消除什么 ()

A. 传递依赖, B. 多值依赖 C 主键冗余 D 外键冗余

10. 在 python 中, list={1, 2, 3}

new_list=list.append(4)

print(new_list)

A [1, 2, 3, [4]] B [1, 2, 3, 4] C[1, 2, 3] D

选择那个说法错误() -- 应是选择正确的

A, Python 字典的键和值可以为任意类型 B, 字符串可以切片索引不可以修改

C. 列表, 元组都是不可变的数据类型 D, 忘了 (D 集合的元素是无序且唯一, 可以像列表一样索引)

银行开发系统需求变更极少, 适合用() 模型

A. 敏捷开发 B. 瀑布模型 C. 增量模型

在一个项目开发中, 因为开发组内的成员能力参差不齐, 导致项目计划严重受阻, 假如你是项目经理, 你应该先去做什么

A. 调整开发组的成员结构、重新安排成员的任务和职责 B 在项目组开展培训和组内交流经验

C. 请外部专家协助 D 建立奖励机制

Jdbc/ ODBC 的应用属于 ISO/OSI 的参考模型中的哪一个

A, 应用层 B 传输层 C 会话层 D 表示层

在传输过程中为了客户端更好的展示，需要解析格式处理属于 ISO/OSI 的参考模型的哪一个

A, 应用层 B 传输层 C 会话层 D 表示层

某算法的递归的表达式， $T(n) = 4T(n/2) + n\log n$ ，则时间复杂度是多少（）

() 是技术让 程序计数器 PC 不能指向下一条指令

A. 高速缓存 cache B. 流水线 C. 内存管理 D. 虚拟内存

什么情况下会使浮点数运算发生上溢出（）

A. 对阶 B. 尾数相加 C. 左规 D 右规

下列那个说法是在传值调用中的应用（）

A. 将实参的值传给形参
B. 将形参的值传给实参
C. 实参地址传给形参
D. 形参地址传给实参

关于语义分析，下列说法正确的有

(1) 分析句子结构， (2) 分析语法的完整性 (3) 检查语法错误 (4) 存入符号表 (5) 生成中间文件

A 仅 34 B 仅 45 C D

时间戳对防止什么攻击最有用

A 重放攻击 B C D

以下哪种方式会引起软件中断？

A 键盘 B 定时器溢出 C 除 0 异常 D 电源故障

某个攻击有意设计流程渗透、伪装数字证书，是什么攻击？

A 零日漏洞攻击 B 供应链投毒 C AI 赋能攻击 D 撞库攻击

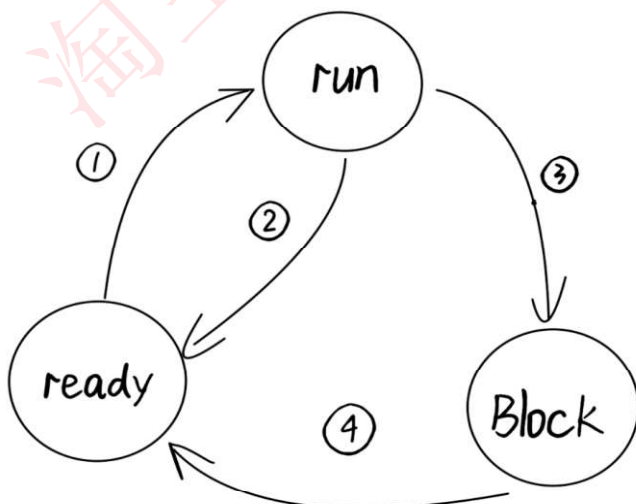
以下哪个不是二分查找可能的顺序（选项实在记不清了）

A 99 65 78 42

有 6 个进程，每个进程至少需要 3 个资源，求不发生死锁的最小资源数（）

A 13 B C D

某三态图如下 发生④的过程时说明（），（图大概就是长这样的）



A. 时间片到 B. 阻塞进程任务已经加载完成 C. 运行中的进程执行了 P 操作
实现关系运算 $a * (b - c) + d$ 栈里最多有多少个字符, 满足条件的最少位数

A 3 B C D

数据库分组排序后, 采用的筛选条件语句是 ()

A. Group by B. having C. Order by D. select

计算机中自动分配 IP 地址的协议是 ()

A. DHCP B. DNS C. ARP D. http

事物管理中互不干涉事务的特性是 ()

A. 隔离性 B. 一致性 C. 原子性 D. 持续性

单 CUP, 32 位什么频率是 133MHZ, 不考虑时钟频率的情况下, 每秒传输速率是多少 (), 假设时钟周期为 5, 则一秒钟的传输速率是多少 ()

A 133 B: 4256, C: 1064, D, 532

A: 851.2, B: 133 C: 532, D: 128

段页号: 32-12, 11-0, 问: 可以存多少页, 每页大小为多少

A: 4096 页, 每页大小 1M

B: 1024 页, 每页大小 4M,

C: 4096 页, 每页大小 16M

D: 2048 页, 每页大小 8M

题的意思如下, 描述不全

给出了一个无向图 (a, b, c, d, e, f), 且给出了无向图的边 (ab, cd, ef, de, 等), 问下列那种方位顺序不正确

A: abcdef B: cdefab, C: debcaf, D: acdefb

题的意思如下, 描述不全, 应是核心考试**这符号

Python a=[1, 2, 3, 4, 5]

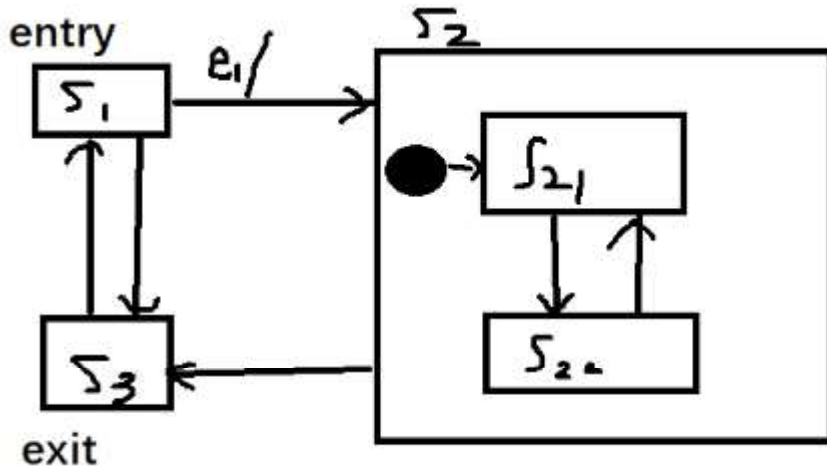
for(i**2 from if (i%2==0))

问, 输出多少

A 2, 4 B 4, 8, C 1, 2, 3, 4, 5 D: 1, 4, 9, 16, 25

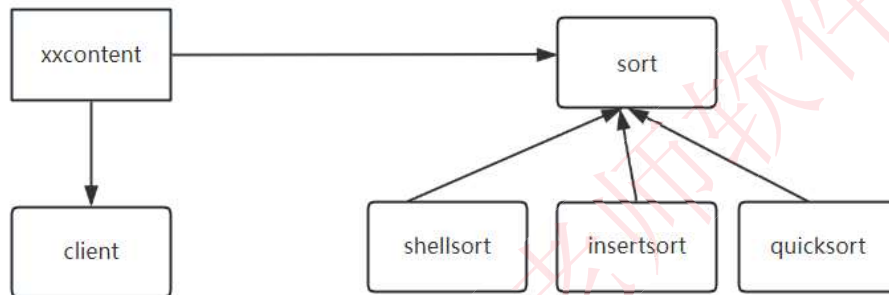
球面欧拉分治法(大概考的是这个, 选项都不记得了)

状态图, 若是 e1, e2, e3, 执行顺序 (), 若是 e2、e3、e4, 执行顺序 (), (大概是这样, 感觉考的是状态图的结束状态)



根据 ISO 标准，（）不是可维护性的子属性 A 可分析 B 可修改 C 模块化 D 容错性

sort 接口实现多个不同的排序类，这种模式是（）模式，是（）类型，应用场景是（），控制 sort 使用哪种排序的是（）



QOS 解决了网络延迟和阻塞的问题，部署在？

A, 路由器 B. IDS C. 防火墙 D. 网闸

周期为 133GHZ ， 32 位计算机，计算总线宽度（），如果考虑来回消耗时间（5 个时钟周期），速度是（）

折半查找不可能产生的数组的顺序（）

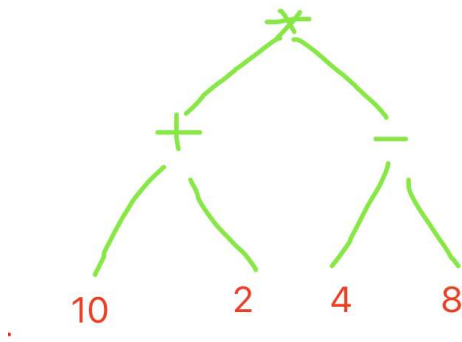
- A
- B
- C
- D

（）协议自动为主机分配地址

- A:ARP
- B:DHCP
- C:

D:

下图二叉树代表的式子



A: $(10+2) * (4-8)$

B:

C:

D:

分析案例使用了（）设计模式，选择这个设计模式是因为（）（给了一个案例让分析，答案是每个案例的定义，大概意思是这样）

下面关于 python 语言，说法正确的是（）

A: B: C: D:

案例分析

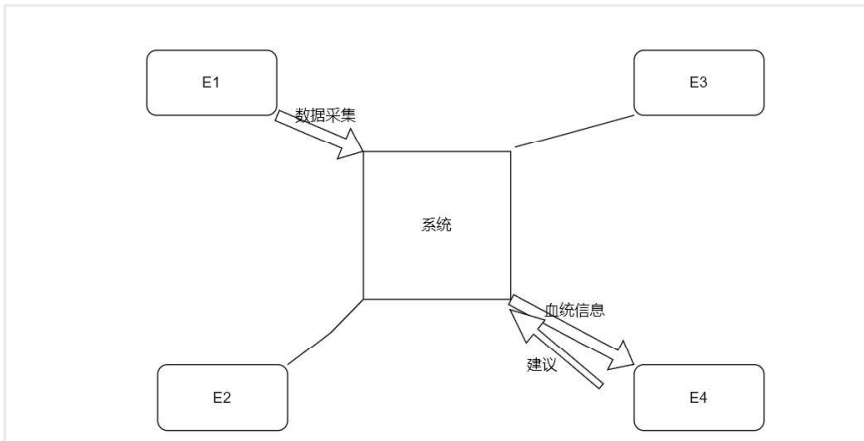
部分知识点回忆如下：

一、一个艺术品拍卖网站，有多个拍卖行（拍卖行 ID，费率），每个艺术品（艺术品 ID，艺术家）只有一个艺术家制作，只能在一个拍卖行进行拍卖（拍卖行 ID，(a)，(b)，成交金额，成交时间），拍卖要记录拍卖行和买家（买家 ID，名字，电话，声誉）之间的分成比例。

1、请画出实体之间的联系。补充 (a) (b) 的内容和主外键约束

2、如果一个艺术品有多个艺术家制作，一个艺术家也可以创作多个作品，请更新此图和关系模型

二、设计一款腕表，从患者获取血压数据，保存到历史血压数据，发给 AI 大模型，大模型判断血压趋势以及解析患者自然语言咨询，如果判断血压趋势危险，则给患者和医生发送提醒。医生可查看患者血压数据，并根据情况给出远程干预（对患者自然语音咨询的意见回复或远程干预）。患者可通过可视界面查看血压历史记录、医生意见、提醒、血压趋势、咨询回复。

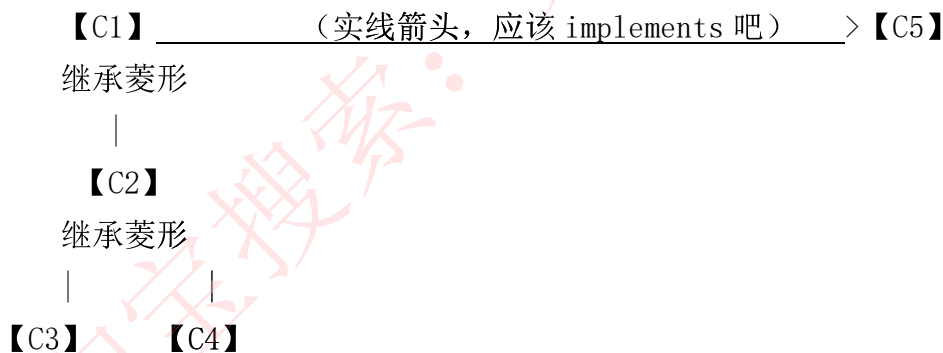


- 1、填写上述 E1-E4 的名称（我填：患者、腕表、AI 模型、医生）
- 2、填写 D1-D5 的保存数据库名称（血压历史记录表、提醒记录表、血压趋势表、建议表、）
- 3、补充缺失的数据流（4 分）

三、 开发一个农作物生长全生命周期监测系统，环境监测设备能从各种监测器采集环境数据，如土壤监测器采集土壤湿度，光照监测器采集光照强度、CO2 浓度等，发给大模型，也通过移动终端发给农业技术员。无人机采集数据给 AI 大模型进行训练，大模型判断是否需要灌溉，然后开启灌溉。支持农业技术员手工调整监测的阈值，超过阈值就发报警信息给农业技术员。大模型根据病虫害判断病虫害的严重程度，支持农业专家手工标注病虫害情况，以提升模型的精度。

这里有一个用例图

- 1、完善 U1-U4、A1-A4



- 1、填写 C1-C5 的名称
 - 3、策略有节能模式、高效模式等，用什么设计模式？为什么要用这种模式？
- 四、 求和数组。手工输入排好序的一串数字 A，和一个目标数字 M，用回溯法找出数组里的子数组，使得子数组各数字之和等于目标数字。为了优化，给出减枝条件：
- 1、已得到的子数组之和+剩余所有数之和 < M，则不可能有结果；
 - 2、已得到的子数组之和+当前值>M，则不可能有结果
- 1、要求填写代码里缺失的（1）-（5）
 - 2、如果输入数组为[1, 2, 3, 4, 5]，目标数字为 5，则输出是什么？
 - 3、最坏情况下该算法的时间复杂度是多少？

- 今年的算法比较简单：1、不满足条件返回。2、做一个条件对比。3、满足什么条件进行赋值。
4、对下一个循环变量赋值。5、初始化对象赋值

C 代码：

全局变量声明

```
int *A;    //源数组
int M;     //目标数字
int found;
int *path; //子数组
int path_len; //子数组长度
```

```
print_subset(int *path, int path_len){
    循环打印 path 数组里的每个数。
    found = 1;
    return;
}
```

```
backtrack(int current_sum, int index, int* remaining_sum, int path_len){
    if ( (3) ){ //找到了
        print_subset( )
    }
    if ( current_sum + remaining_sum[index] < M ){
        (4)
    }

    if ( current_sum + remaining_sum[index+1] > M ){ // 包括 index+1
    }

    if ( (5) ){ // 包括 index
    }
}
```

.....

}

```
main(){
    分配内存
```

手工输入一串排好序的数字，保存到 A 数组

手工输入目标数字 M

它给出注释说明计算 remaining_sum 数组的各个值方法

```
/** remaining_sum[n] = 0;  
remaining_sum[n-1] = A[n-1];  
remaining_sum[n-2] = A[n-1]+A[n-2];  
remaining_sum[n-3] = A[n-1]+A[n-2]+A[n-3];  
.....**/
```

然后好像这里有个循环计算 remaining_sum 数组的各元素值

```
for ( i =n; i>=0; i--){
```

(1)

```
}
```

(2)

释放分配的内存。

```
}
```

五、 用 Adapter 模式设计一个播放器，默认可以播放 mp3 格式。继承子类后可以播放 mp4 或 Lrc 格式。

【AdvancedMediaPlayer】

【MediaPlayer】

继承菱形

| |

【Mp4Player? 】 【LrcPlayer? 】

下面是 Java 代码。请补充缺失的(1)–(5)。

```
interface MediaPlayer {  
    public (1)  
}
```

案例题 Java 比较简单，定义一个接口名称，定义两个类变量
扩展一个音乐格式为支持多个音乐格式