2021年全国计算机统考408考研真题回忆版

数据结构选择题

1. 已知指针指向一个带头结点的非空单循环链表，结点结构 data | next |，其 中 next 是指向直接后继结点的指针，p 是尾指针，q 为临时指针。现要删除该链 表的第一个元素，正确的语句序列是（ ）

   A.h→next=h→next→next；q=h→next；free（q）；

   B.q=h→next；h→next=h→next→next；free（q）；

   C.q=h→next：h→next=q→next；if（p！=q）p=h；free（q）；

   D.q=h→next：h→next=q→next；if（p==q）p=h；free（q）；

   【解析】D。本题考查链表中结点的删除操作，若通过判断确定该单循环链表只有一个结点，则该结点既是链表的第一个元素，也是最后一个元素，则删除该元素后，链表变成带头结点的空链表，尾指针即是头指针。

2. 已知初始为空的队列 Q 的一端能进行入队操作又能进行出队操作，若 a 的入队 序列是 1，2，3，4，5，则不能得到的出队序列是（ ）

   A. 5，4，3，1，2

   B. 5，3，1，2，4

   C. 4，2，1，3，5

   D. 4，1，3，2，5

   【解析】D。题目应是一种输出受限的双端队列，因此根据选项C和D，若4是第一个出队的元素，那么至少1，2，3，4已全部入队，则队列中2应该与1相邻，出队时顺序也应该相邻，所以选项D错误。

3. 已知二维数组 A 按行优先方法存储，每个元素占用 1 个存储单元，若元素 A[3] [3]的存储地址是 220，则元素 A[5] [5]的存储地址是（ ）

   A.295　B.300　C.301　D.306

【解析】B。按照行优先存储，计算A[3] [3]到A[5] [5]之间的元素个数，再加上A[3] [3]的存储地址即可。

4. 某森林 F 对应的二叉树为 T，若 T 的先序遍历序列是 a，b，d，c，e，g，f， 中序遍历序列是 b，d，a，e，g，c，f，则 F 中树的棵数是（ ）

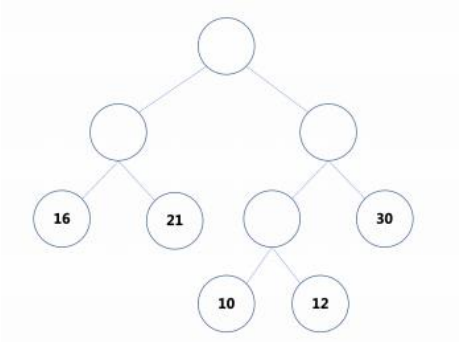
A.1　　B.2　　C. 3　　D.4

【解析】C。由于二叉树的前序遍历和中序遍历序列能确定唯一的一棵二叉树，因此本题可以确定二叉树T的结构，逐层断掉T中最右侧的右子树，即得到三棵树，因此森林F中树的棵树是3。

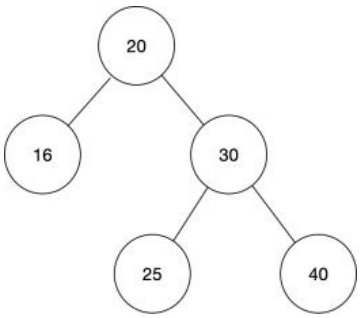
5. 若某二叉树有 5 个叶子结点，其权值分别为 10，12，16，21，30.则其最小的带权路径长度（WPL）是（ ）

A.89　　B.200　　C.208　　D.289

【解析】B。要求最小的WPL，即考查哈夫曼树的构造，构造结果如下图所示，因此WPL=2 x (16+21+30)+3 x (10+12)=200。

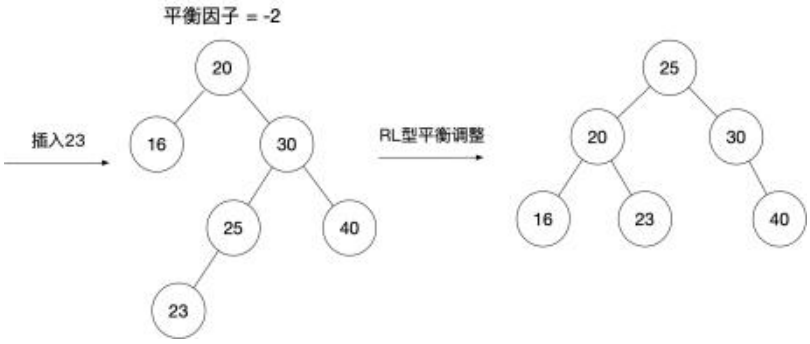


6. 给定平衡二叉树如下图所示，插入关键字 23 后根节点中的关键字是（ ）

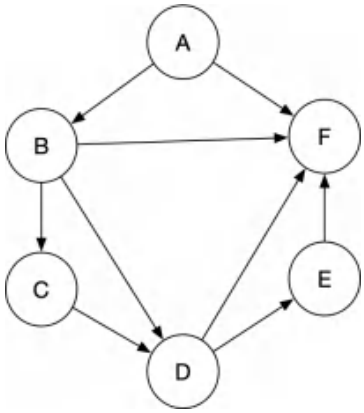


A.16　　B.20　　C.23　　D.25

【解析】D。将关键字23插入作为关键字25所在结点的左孩子后，该二叉树失衡，经过平衡调整后，该平衡二叉树变为如下图所示结构，根节点中的关键字为25。



7. 给定如下有向图，给定如下有向图，该图的拓扑有序序列的个数是（ ）



A.1　　B.2　　C.3　　D4

【解析】A。该有向图的拓扑序列唯一，为ABCDEF。

8. 使用 Dijkstra 算法求下图中从顶点 1 到其余各顶点的最短路径，将当前找到 的从顶点 1 到顶点 2，3，4，5 的最短路径长度保存在数组 dist 中，求出第二条 最短路径后，dist 中的内容更新为（ ）

A. 26，3，14，6

B. 25，3，14，6

C. 21，3，14，6

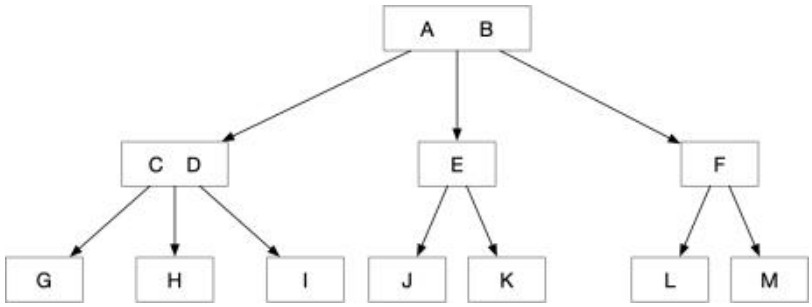
D. 15，3，14，6

【解析】C。题目信息不完全。

9. 在一棵高度为3、阶数为 3 的 B 树中，根为第一层，若第二层有 4 个关键字，则该树的结点个数最多是（ ）

A.11　　B.10　　C.9　　D.8

【解析】A。阶数为3的B树中，每个结点包含的关键字可以为1个或2个。本题中，要求第二层有4个关键字，则结点数最多的情况如下图所示，其中A、B、C、D... 表示关键字，最多可能有11个结点。



10. 设数组 S[ ]（93，946，372，9，146，151，301，485，236，372，43，892） 采用最低位优先（LSD）基数排序将 S 排列成升序序列，第 1 趟分配收集后元素 372 之前，之后紧邻的元素是（ ）

A.43，892

B.236，301

C.301，892

D.485，301

【解析】C。由于采用最低位优先，即第1趟按照个位从小到大排序，因此第一趟排序后的结果是(151，301，372，892，93，43，485，946，146，236)，选择C。

11. 将关键字 6，9，1，5，8，4，7 依次插入到初始为空的大根堆 H 中，得到的 H 是（ ）

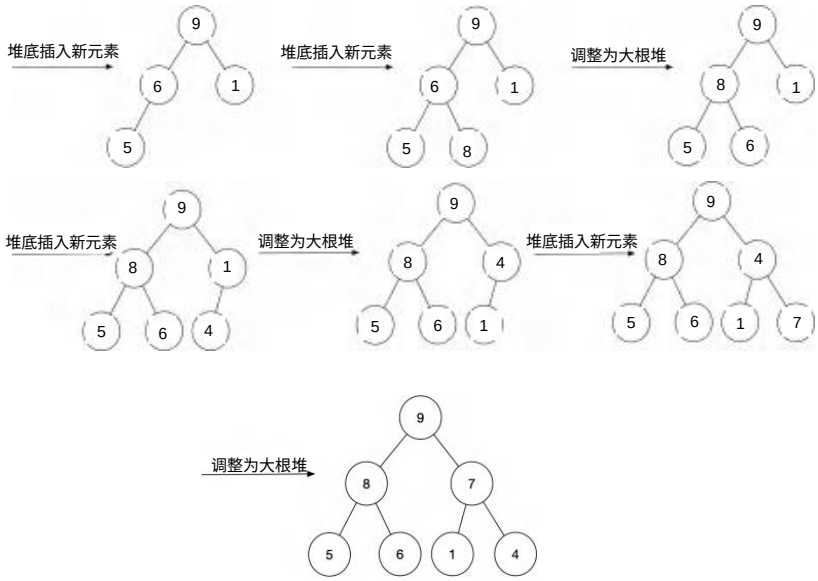
A. 9，8，7，6，5，4，1

B. 9，8，7，5，6，1，4

C. 9，8，7，5，6，4，1

D. 9，6，7，5，8，4，1

【解析】B。类似于408统考2018年11题，本题考查大根堆的构建及调整过程，大根堆中父结点的值大于或等于子结点的值，刚插入后从最后一个非叶子结点开始调整，调整后的结果为B。



计算机组成原理选择题

1. 计算器浮点运算速度为 93.0146PFLOPS，这说明该计算器每秒完成的浮点操作次数为（ ）

   A. $9.3x10^{13}$

   B. $9.3x10^{15}$

   C. 9.3 千万亿次

   D. 9.3 亿亿次

   > 【解析】D。K—M—G—T—P—E—Z，都是乘 10^3 的递增关系，P = 10^{15}。93.0146 PFLOPS = 每秒 $9.3×10^{16}$ 次浮点操作，即 9.3 亿亿次

2. 已知带符号整数用补码表示。变量 X，Y，Z 的机器数分别为 FFFDH，FFDFH， 7FFCH，下列结论中，正确的是（ ）

   A.若 X，Y，Z 为无符号整数，则 Z<X<Y

   B.若 X，Y，Z 为无符号整数，则 X<Y<Z

   C.若 X，Y，Z 为带符号整数，则 X<Y<Z

   D.若 X，Y，Z 为带符号整数，则 Y<X<Z

   > 【解析】D。若按无符号整数解读，显然 Z<Y<X。

若按带符号整数补码解读，则 Y<X<Z：

$X_{补}$ = FFFDH = 1111 1111 1111 1101B　　$X_{原}$ = 1000 0000 0000 0011B

$Y_{补}$ = FFDFH = 1111 1111 1101 1111B　　$Y_{原}$ = 1000 0000 0010 0001B

$Z_{补}$ = 7FFCH = 0111 1111 1111 1100B　　$Z_{原}$ = 0111 1111 1111 1100B

3. 下列数值中，不能用 IEEE754 浮点精确表示的（ ）

A.1.2　　B.1.25　　C.2.0　　D.2.5

【解析】A。IEEE754 浮点数，分为"数符、阶码、尾数"三个部分。

尾数可表示为 1.XXXXXXX 的形式，而阶码反映了尾数应该左移/右移多少位。因此，只要能用二进制定点数精确表示的真值，就一定可以转化为浮点数的表示形式。

1.25D = 1.01B　　2.0D = 10.0B　　2.5D = 10.1B

只有 1.2D 无法用浮点数精确表示

4. 某计算机的存储总线中有 24 位地址线和 32 位数据线，按字编制，字长为 32 位，若 000000H~3FFFFFH 为 RAM 区，则需要 512K*8 位的 RAM 芯片数为（ ）

A.8　　B.16　　C.32　　D.64

【解析】C。000000H~3FFFFFH，总共对应 2^{22} 个地址。由于系统采用“按字编制“，因此每个地址对应一个字。一个字 = 32bit = 4B，因此RAM区的大小为 2^{24} B。每块芯片的大小为 2^{19} B，共需要 2^{24}/ 2^{19}=2^5 =32 块芯片。

5. 若计算机主存地址为 32 位，按字节编址，cache 数据区大小为 32KB，主存块大小为 32B，采用直接映射方法和回写（Write Back）策略，则 cache 行的位 数至少是（ ）

A.275　　B.274　　C.258　　D.257

【解析】A。块大小 = 32B，因此Cache被分为 = 32KB/32B = 2^{10} 块，需要用 10 bit 表示Cache块号。主存地址为 32 位，因此主存大小为 2^{32}B，主存被分为 2^{32}B/32B = 2^{27} 块，因此需要用 27bit 表示主存块号。由于采用直接映射方式，因此 27bit 主存块号的低位 10bit 用于表示主存块在Cache 中的位置，高位17bit 作为 Tag

标记位。由于采用回写策略，因此需要添加 1bit 脏位，用于标记 Cache 块中的副本数据是否被修改过。另外，还需要 1bit 有效位，用于标记Cache块中的数据是否有效。

因此，一个Cache行由以下几个部分组成：1bit 有效位、1bit 脏位、17bit Tag位、32B 存放块数据。总共 275 bit。

6. 下列寄存器中，汇编语言程序员可见的是

①. 指令寄存器

②. 微指令寄存器

③. 基址寄存器

④. 标志状态寄存器

【解析】D ③④。

分类	寄存器	功能
用户可见	通用寄存器	存放操作数和地址信息；作为基址寄存器、变址寄存器等
	程序状态字寄存器（PSWR）	保留由逻辑运算指令在主存中的存放地址
	程序寄存器（PC）	指出下一条指令在主存中的存放地址
	累加寄存器(ACC)	暂时存放ALU运算的结果信息，用于实现加法运算
	指令寄存器（IR）	保存当前正在执行的那条指令
	暂存寄存器（DR）	暂存从主存读来的数据
用户不可见	存储器地址寄存器（MAR）	存放所要访问的主存单元的地址
	存储器数据寄存器（MDR）	存放向主存写入的信息或从主存中读出的信息

7. 下列关于数据通路的叙述中，错误的是

A. 数据通路包含 ALU 等组合逻辑（操作）元件

B. 数据通路包含计时器等时序逻辑（状态）元件

C. 数据通路不包含用于异常事件检测及响应的电路

D. 数据通路中的数据流动路径由控制信号进行控制

【解析】C。通常将指令执行过程中数据所经过的部件，包括路径上的部件称为数据通路。ALU、通用寄存器、状态寄存器、cache、MMU、浮点运算逻辑、异常和中断处理逻辑等都是指令执行过程中数据流经的部件，都属于数据通路的一部分。数据通路由控制部件进行控制。控制部件根据每条指令功能的不同生成对数据通路的控制信号，并正确控制指令执行流程。

8. 下列关于总线的叙述中，错误的是（ ）

A. 总线是在两个或多个部件之间进行数据交换的传输介质

B. 同步总线由时钟信号定时，时钟频率不一定等于工作频率

C. 异步总线由握手信号定时，一次握手过程完成一位数据交换

D. 突发（Barst）传送总线事务可以在总线上连续传送多个数据

【解析】C。A选项显然正确。B选项考察总线的性能指标，时钟频率，反映了每秒钟有几个时钟信号。工作频率指总线一秒内可以传送几次数据。D选项就是突发传送方式的基本概念。C选项，异步总线由握手信号定时，一次握手可以完成多位数据交换。

9. 下列选项中不属于 I/O 接口的是

A. 磁盘驱动器

B. 打印机适配器

C. 网络控制器

D. 可编程中断控制器

【解析】D。中断控制器集成在CPU内部，属于处理器微架构的一部分，比如 Intel 的 APIC，不属于I/O接口。I/O接口作为I/O设备和主机之间数据交换的“桥梁“，通常在CPU外部。本题B答案有些迷惑性，在某东某宝搜“打印机适配器”，通常指电源适配器。

操作系统选择题

1. 若系统中 n（n>=2）个进程，每个进程均需使用某类临界资源 2 个，则系统不会发生死锁所需的该类资源总至少是（ ）

A.2　　B.n　　C.n+1　　D.2n

【解析】C。若资源总数只有 n 个，则每个进程持有一个资源并等待另一个资源时，会发生死锁。资源总数有 n+1 个，则至少会有一个进程可以得到2个资源并顺利运行下去，不会发生死锁。

2. 通过系统调用完成的操作是

A.页面置换

B.进程调度

C.创建新进程

D.生成随机整数

【解析】C。“系统调用“由用户进程发起，请求操作系统服务。A选项，当内存中空闲页框不够时，操作系统会将某些页面置换出外存，这个过程由操作系统主动完成，并不是用户进程的系统调用直接引发的。B选项，用户进程只能通过系统调用申请阻塞、终

止自身，操作系统会在需要时主动进行进程调度，并不是由系统调用直接引发的。C选项，通过 Linux 中的 fork 系统调用，父进程可创建子进程。D选项，生成随机数只需要普通的函数调用，无需使用系统调用。

3. 实现时间片轮转算法必须有（ ）

   ①PCB ②中断机制 ③就绪队列 ④阻塞队列

   【解析】①②③。

   ①PCB是进程存在的唯一标志，任何一种进程调度算法必然需要使用PCB。②时间片轮转中，当时钟部件会定期发出时钟中断信号，CPU检测到该中断信号后检查当前运行的进程时间片是否已用完。显然，中断机制是必须要有的。③排队等待上处理机的进程PCB，都需要放在就绪队列中。④时间片轮转算法每次给就绪队列队头的进程分配时间片，和阻塞队列没什么关系。

4. 下列哪些状态会触发调度程序执行的是：

   ①中断处理结束 ②时间片用完 ③进程阻塞 ④进程执行结束

   【解析】①②③④。在时间片轮转算法中，当CPU检测到时钟中断信号后，开始中断处理，会检查当前进程的时间片是否已用完，若已用完，则中断处理结束后进行进程调度。因此，①②都可能触发调度程序。而③④都会导致当前正在运行的进程下处理机，显然也会触发进程调度。

5. 删除一个文件后，下列不会发生的是（ ）

   A. 快捷方式被删除

   B. 文件控制块被回收

   C. 磁盘空间被释放

   D. 删除目录项? （回忆载入失败 ┐(´∀`)┌）

   【解析】A。创建某文件的“快捷方式“，本质上是创建了另一个新文件，其中记录了原文件的存放路径。删除源文件并不会导致快捷方式被删除。

6. 使用SSTF磁盘调度算法，一个磁道访问序列，磁头刚开始在 184号磁道（待访问磁道序列信息不完整）

   A．41 大多数同学反馈正确答案选41，考察磁盘调度算法，常规题型。

7. 给了一个表格，使用clock算法，给出了一个虚拟地址，求映射的物理地址（）

   A.20　B.60　C.80　D.100

   【解析】B。考察页面置换算法，常规题型。

8. 二级页表中基址寄存器存放的是（）

   A. 一级页表物理地址

   B. 二级页表物理地址

   C. 一级页表虚拟地址

   D. 二级页表虚拟地址

   【解析】A。二级页表中基址寄存器存放的是一级页表的物理地址。根据一级页号、一级页表的起始物理地址可以找到一级页表中的表项

9. 不能在用户态下发生的是（）

   A. trap指令

   B. 系统调用

   C. I/O指令

   D. 库函数?　（回忆载入失败 ┐(´∀`) ┌）

   【解析】C。C选项 I/O指令 属于特权指令，不能在用户态下执行。

10. 创建进程需要做的是（）

    ①创建一个进程控制块　　②初始化一个进程控制块　　③创建就绪队列

    【解析】①②。显然，①②需要做的。而进程的就绪队列是操作系统启动的时候就创建好的。

11. 题目信息不全，考察多重中断（欢迎各位记忆小天才留言补充）

计算机网络选择题

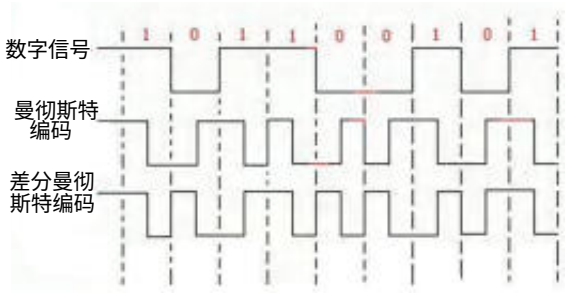
1. 在 TCP/IP 模型中，由传输层相邻的下一层实现的主要功能

   A.对话管理　　B.路由选择　　C.端到端报文段传输　　D.结点到结点流量控制

   【解析】B。在TCP/IP模型中，传输层相邻下一层为网际层，实现的主要功能包括有：1、处理来自传输层的分组发送请求，将分组装入IP数据报，填充报头，选择去往目的节点的路径，然后将数据报发送适当的端口；2、处理输入数据报，首先检查数据报的合法性，然后进行路由选择；3、处理ICMP报文，处理路由的选择，流量控制和拥塞控制。本题选择B。

2. 根据差分曼彻斯特编码的图形，选择对应的码串（ ）。

   【解析】差分曼彻斯特编码的特点在于， 在每个时钟周期的起始处：跳变则说明该比特是0，不跳变则说明该比特是1。



3. 子网划分，某网段划分为三个网段

   A. x.x.9.0/25　　B x.x.9.128/26

   C. x.x.9.192　　D.x.x.9.128/27

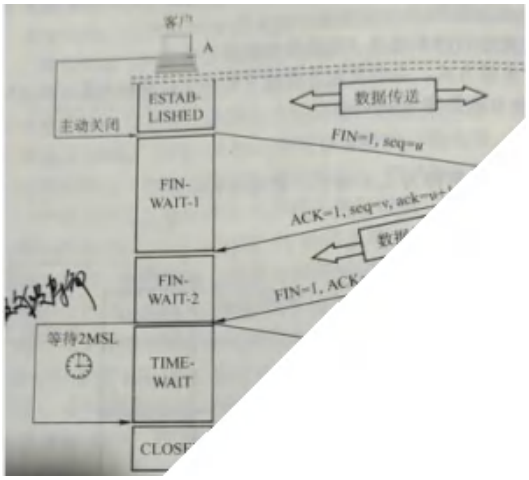
   【解析】题目不完全。

4. 链路层MTU=800B，已知一个IP数据报为1500B，求第二个分片的片偏移和MF为（ ）

   A. 796 1　　B.800 1　　C.796 0　　D.800 0

   【解析】题目回忆有问题。本题类似于2018年47题。链路层MTU指数据帧可封装数据的上限，IP分组头部长20B，最大IP分片分装数据的字节数为776，至少需要2个分片，第二个分片的偏移量为776/8=97，MF表示More Fragment，如果为1表示后面还有分片，否则表示最后一片或者没分片。

5. 题目不完全（呼叫各位记忆小天才）
6. 若客户首先向服务器发送 FIN 段请求断开 TCP 连接，则当客户收到的服务器发送的 FIN 段并向服务器发送 ACK 段后，TCP 状态转换为（　）



A.CLOSE_WAIT　B.TIME_WAIT　C.FIN_WAIT_1　D.FIN_WAIT_2

【解析】B。根据题干可知，TCP连接释放的四次挥手过程已经结束，客户的TCP状态切换为TIME-WAIT，表示客户再等到时间等待计时器设置的2MSL（最长报文段寿命）后，连接才算彻底关闭。

7. 若大小为 12B 的应用层数据通过 1 个 UDP 和 1 个 TCP，则 UDP 数据报和 TCP 段实现的有效载荷（应用层数据）最大传输效率为（　）

A.37.5% 16.7%　B.37.5% 37.5%　C.60.0% 16.7%
D.60.0% 37.5%

【解析】D。为达到最大传输效率，UDP首部长度8B，TCP首部长度20B，分别计算传输效率得12/(12+8) x 100%=60%，12/(12+20) x 100%=37.5%。

40.甲发送了seq=501，大小为200B的报文，乙发送了一个ack=501，接收窗口为500，甲在收到下一个ack之前还能发送的数据大小（　）

A.501~1000　B.501~700　C.701~1000　D.801~100

【解析】C。甲发完200B报文后，继续发送的报文段中序号字段seq=701，由于乙通告接收窗口为500，且甲未收到乙对于seq=501报文段的确认，则甲还能发送的报文段字节数为500-200=300B，因此甲在收到下一个ack之前还能发送的数据序号范围是701~1000。

综合应用题

1. （15）已知无向连通图G由顶点集V和边集E组成 |E|>0,当G中度为奇数的顶点个数为不大于2的偶数时，G存在包含所有边且长度为|E|的路径（称为EL路径），设图G采用邻接矩阵存储，类型定义如下：

```
typedef struct{                     //图的定义
    int numVertices, numEdges;      //图中实际的顶点数和边数
    char VerticesList [MAXV];       //顶点表，MAXV 为已定义常量
    int Edge [MAXV][MAXV];          //邻接矩阵
} MGraph;
```

请设计算法：`int IsExistEL (MGraph G)`，判断 G 否存在 EL 路径，若存在，则返回 1，否则，返回 0，要求：

（1）给出算法的基本设计思想

（2）根据设计思想，采用 C 或 C++描述算法，关键之处给出注释

（3）说明算法的时间复杂度和空间复杂度

解析：

(1) 从0号顶点出发，对图G进行广度优先遍历，并统计当前已遍历的顶点个数、度为奇数的顶点个数。遍历过程中，一旦度为奇数的顶点个数超过2个，立即返回0。一趟广度优先遍历结束后，若已遍历的顶点个数等于 numVertices，则说明图G连通，同时，若度为奇数的顶点个数为偶数，则说明存在EL路径，返回1。否则，返回0。

(2)算法描述如下：

```
int vCount=0;           //目前已经遍历的顶点个数
int oddCount=0;         //度为奇数的顶点个数
Queue Q;                //辅助队列，用于实现广度优先遍历
MGraph G;               //图G的数据结构，假设已经完成图的初始化
bool visited[G.numVertices];      //标记每个顶点是否被访问过，初始全为false

//访问顶点v
void visit (int v){
```

```
  visited[v]=TRUE;       //对v做已访问标记
  vCount++;              //已访问顶点数+1
}

//广度优先遍历算法
int BFS(Graph G,int v){ //从顶点v出发，广度优先遍历图G
  visit(v);              //访问顶点v
  Enqueue(Q,v);          //顶点v入队列Q
  while(!isEmpty(Q)){
    DeQueue(Q,v);          //队头元素出队，v指向出队元素
    degree=0;              //统计顶点v的度数
    for(int w=0; w<numVertices; w++){   //访问顶点v的所有邻接顶点
      if (G.Edge[v][w]>0) {            //存在顶点v到顶点w的边
          degree++;                    //顶点v的度数+1
           if(!visited[w]){            //w为v的尚未访问的邻接顶点
               visit(w);                 //访问顶点w
               EnQueue(Q,w);             //顶点w入队列
           }
       }
    }//for
    if(degree%2==1){      //顶点v的度数为奇数
      oddCount++;        //度数为奇数的顶点数+1
      if (oddCount>2)
         return 0;    //若度数为奇数的顶点数超过2个，一定不存在EL路径
    }
  }//while

  //无向图连通，且度数为奇数的顶点数为偶数个，必存在EL路径
  if(vCount==G.numVertices && oddCount%2==0){
    return 1;
  } else {
    return 0;
  }
```

```
}

int IsExistEL (MGraph G) {
  return BFS (G, 0);    //从0号顶点出发，进行广度优先遍历
}
```

(3) 记图中实际的顶点数numVertices = n。

时间复杂度：主要来自遍历整个邻接矩阵， $O(n^2)$

空间复杂度：主要来自辅助队列，最坏情况下所有顶点可能同时入队，因此空间复杂度为 O(n)

注：这个题目前还存在争议，不少同学反馈说题目已经明确告知图G是连通的，所以只需要简单地遍历邻接矩阵来统计每个顶点的度即可解决问题，似乎过于简单，由于这是408有史以来第一次考图的算法，因此也不排除这种可能性。即便出题老师没有默认图G是连通的，只要能用遍历邻接矩阵的方法统计结点的度并解决问题，扣分也不会太多。

42. 已知某排序算法如下

```
void cmpCountSort (int a[], int b[], int n) {
  int i,j,*count;
  count = (int *)malloc(sizeof(int)*n);   //C++语言: count=new int[n];
  for (int i=0; i<n; i++) count[i]=0;
  for (int i=0; i<n-1; i++){
    for (int j=i+1; j<n; j++) {
      if (a[i]<a[j])    count[j]++;
      else              count[i]++;
    }
  }
  for (int i=0; i<n; i++)   b[count[i]]=a[i];
  free(count);    //C++语言: delete count;
}
```

(1) 若有 int a[]={25, -10, 25, 10, 11, 19}，n=6，调用 comCountSort (a, b, n) 后数组b的内容是什么?

(2) 若a中有n个元素则算法执行过程中元素之间比较次数是多少?

(3) 该算法稳定吗？若稳定则阐述理由，否则，修改为稳定排序算法。

参考答案：

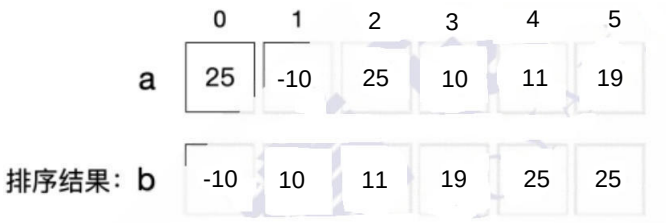
(1) 调用 comCountSort (a, b, n) 后数组b = {-10, 10, 11, 19, 25, 25}

(2) 比较次数 n*(n-1)/2

(3) 该排序算法不稳定，只需将 `if (a[i]<a[j])` 改为 `if (a[i]≤a[j])` 即可保证排序结果稳定。

解析：

（1）算法运行结果如下



其中，a为原始数组；b数组用于存放最终的排序结果；count[i]的大致含义是，在数组a中，不大于 a[i] 的元素有几个。

（2）从for循环的逻辑可知，第一趟排序，用 a[0] 与a[1~n-1] 比较；第二趟排序，用 a[1] 与a[2~n-1] 比较；第三趟排序，用 a[2] 与a[3~n-1] 比较..... 总共进行 n-1 趟，总的关键字比较次数为 (n-1)+(n-2)+(n-3)+.....+1 = n*(n-1)/2。题目没有让我们说明理由，因此可以不写分析过程。

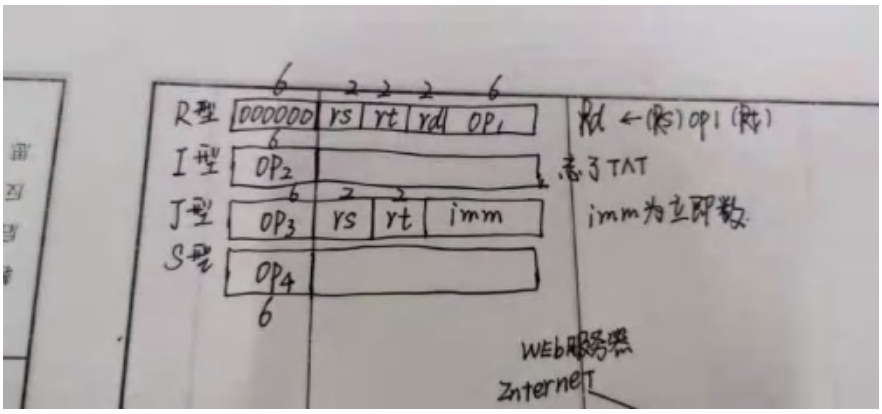
（3）如上图所示，用第一小问给的例子验证就能发现算法是不稳定的

43. 题目给出了四种指令格式，机器字长，指令字长，地址线位数，数据线位数

（1）确定 MAR、MDR、ALU、IR 的位数

（2）确定指令条数

(3) 求两个数相减，相乘的结果，问是否溢出。



解析：

（1）MAR位数和地址线位数相同、MDR位数和数据线位数相同、ALU位数与机器字长相同、IR位数与指令字长相同；

（2）

（3）减法未溢出，乘法溢出

注：本题信息不完整

44. M 的主存地址为 24 位，按字节编址，采用分页存储管理方式，虚拟地址为 30 位，页大小为 4KB，TLB 采用 2 路组相联方式（共8组）和 LRU 替换策略

（1）虚拟地址中有哪几位表示虚页号？哪几位表示页内地址？

（2）已知访问 TLB 时虚页号高位部分用作 TLB 标记，低位部分用做 TLB 组号，M 的虚拟地址中哪几位 TLB 标记？TLB 组号？

（3）初始 TLB 为空，访问虚页号为 10、12、16、7、26、4、12、20 哪一个号对应被替换，说明理由。若 M 中虚拟地址位数增加 32 位，TLB 表项位数增加几位？

解析：

（1）页面大小=2^{12} B，虚拟地址共30位，高18位表示虚页号、低12位表示页内地址

（2）TLB被分为8组，即2^3组，因此可用虚拟页号的低3位作为组号，高15位作为TLB标记

（3）先确定每个虚页号所属的分组：10%8=2，12%8=4，16%8=0，7%8=7，26%8=2, 4%8=4, 12%8=4, 20%8=4。显然，只有4号分组的TLB表项可能被装满。第一次访问12号页，将12号页表项调入TLB的4号分组；之后访问4号页，将4号页表项调入TLB的4号分组；再次访问12号页，TLB命中；最后访问20号页，此时TLB的4号分组已满，根据LRU替换策略，最近最久未访问的是4号页表项，因此将4号页表项替换出去，并换入20号页表项。虚拟地址增加到32位，则Tag标记位需要17bit，因此TLB表项位数增加2bit。

45. 给一些操作系统相关的工作步骤？排序?

    (1) ROM中的初始化程序；硬盘的引导程序；分区引导程序；操作系统初始化引导

    (2) 物理格式化；磁盘分区；逻辑格式化；装系统

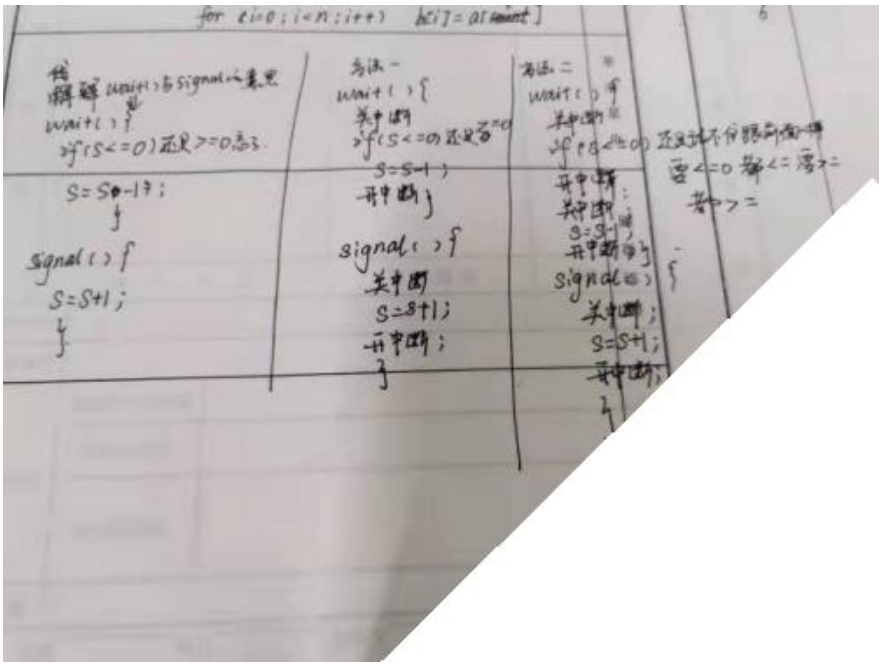
    (3) 对于(2)，分配扇区在哪一步：物理格式化；创建文件系统根目录在那一步：逻辑格式化

    解析：

    据反馈，这个题目给了一堆背景信息帮助大家理解。自己动手装过系统的同学做这个题会很占便宜，安装系统时，要先进BIOS选择启动盘（硬盘启动？U盘启动？光盘启动之类的），而BIOS是存储在ROM芯片里的。确定了启动盘之后，要确定启动分区（就是我们熟悉的C盘），然后就可以“开机”（操作系统初始化）。

    本题第一小问考察的较偏，预计普遍的得分率不高。而二三小问是常规的问题，在王道书和课程中都讲过

46. PV操作题，最左边是在解释什么是wait和signal，右边是给的两个试图保证wait、signal对S进行互斥操作的方法



（1）为什么要对S进行互斥访问，不互斥会怎么样

（2）题目给出的这两种方法哪一个能够实现对S的互斥访问，并解释原因

（3）利用开关中断是否可以实现互斥

解析：（这是某热心同学的手写回忆版，wait 里边应该是 `if (S>0)` 吧？）这个题就是考察 wait、siganl 原语的实现原理，原语要用开关中断来实现。

（1）S对应某种资源的数量，wait、signal 执行过程可能会被中断，导致各个进程对wait、signal对S的读写不互斥，从而导致错误。

（2）第一种方法更合理，第二种方法中，wait 对S值的判断没有“一气呵成”，也可能出错。比如两个进程都在wait 函数中判断 S>0，于是都会让 S--，导致S的值为负。

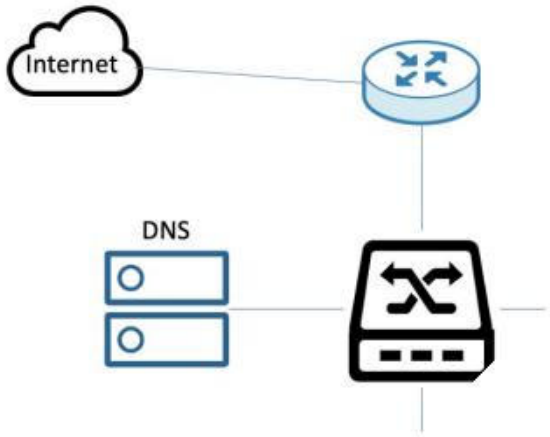
（3）可以，关中断之后，开中断之前，进程执行过程可以一气呵成

47. 某网络拓扑图如下所示，t0时刻，H1的ARP和交换机的转发表为空，t1时刻交换机收到H1的http报文，t0到t1期间没有其他信息出现在链路上。

    (1) 向Internet请求访问某网页时，除了 http 还有什么协议，以及这个申请访问的http帧叫什么？从传输层到链路层，dns封装的协议分别是哪些?

(2) t1时刻，交换机的转发表？转发表格式：{地址，端口}

(3) H2至少收到的与此相关的帧，目的地址是？



【解析】

（1）还需要TCP、DNS协议。申请访问的是HTTP请求报文。从传输层到链路层，dns封装的协议分别是传输层：tcp协议，网络层：ip协议、ARP协议、ICMP协议，链路层：PPP协议。

（2）（3）题目不完全。