

# 中国科学院大学

## 2016 年招收攻读硕士学位研究生入学统一考试试题

### 科目名称：计算机学科综合(专业)

一、单项选择题：第 1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项最符合试题要求。

1. 对于一个初始为空的栈  $s$  执行操作  $push(s,5)$ ,  $push(s,2)$ ,  $push(s,4)$ ,  $pop(s,x)$ ,  $getTop(s,x)$  后,  $x$  的值应是 ( )

- A. 5                      B. 2                      C. 4                      D. 0

2. 将一个  $n \times n$  的对称矩阵  $A$  的下三角部分按行存放在一个一维数组  $B$  中,  $A[0][0]$  存放在  $B[0]$  中, 那么第  $i$  行的对角元素  $A[i][i]$  在  $B$  中的存放位置是 ( )

- A.  $(i+3) \times i/2$       B.  $(i+1) \times i/2$       C.  $(2n-i+1) \times i/2$       D.  $(2n-i-1) \times i/2$

3. 设  $T$  是哈夫曼二叉树, 具有 5 个叶节点, 树  $T$  的高度最高可以是 ( )

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

4. 当采用邻接表方式存储带权连通图时, 求最小生成树的 Prim 算法的时间复杂度为 ( ) 设图有  $n$  个顶点和  $e$  条边。

- A.  $O(n)$                       B.  $O(e \log_2 e)$                       C.  $O(n^2)$                       D.  $O(n^3)$

5. 用快速排序对下列 4 个序列做升序排列, 各以序列第一个元素为轴点进行第一次划分, 则在本次划分过程中需要移动元素次数最多的序列是 ( )

- A. {10, 30, 50, 70, 90}                      B. {50, 70, 90, 10, 30}  
C. {50, 30, 10, 70, 90}                      D. {90, 70, 50, 30, 10}

6. 在内排序的过程中, 通常需要对待排序元素序列的排序码做多趟扫描。采用不同的排序方法将产生不同的排序中间结果, 设要将集合 {tang, deng, an, wan, shi, bai, fang, li} 中的排序码按升序排列, 则 ( ) 是二路归并排序一趟扫描的结果。

- A. wan, deng, tang, an, bai, fang, li, shi  
B. an, deng, bai, li, shi, tang, fang, wan  
C. deng, an, tang, shi, bai, fang, li, wan  
D. deng, tang, an, wan, bai, shi, fang, li

7. 已知一个线性序列 {38, 25, 74, 63, 52, 48}, 假定用散列函数  $Hash(key)=key \% 7$  计算散列地址, 散列存储在表  $A[10]$  中。如果采用线性探测法解决冲突, 且各元素的查找概率相等, 则在该散列表上查找不成功的平均查找长度为 ( )

- A. 2.60                      B. 3.14                      C. 3.71                      D. 4.33

8. 串的两种基本的存储方式是 ( )

- A. 顺序存储和链式存储                      B. 顺序存储和堆存储  
C. 堆存储和链式存储                      D. 堆存储和数组存储

9. 稀疏矩阵一般的压缩方法有 ( ) 两种。

- A. 二维数组和三维数组                      B. 三元组和散列  
C. 三元组和十字链表                      D. 散列和十字链表

10. 无向图  $G=(V,E)$ , 其中:  $V=\{a,b,c,d,e,f\}$ ,  $E=\{(a,e), (a,c), (b,e), (c,f), (f,d), (e,d)\}$ , 对该图进行深度优先遍历, 得到的顶点序列正确的是 ( )

- A. a, e, d, f, c, b                      B. a, b, e, c, d, f  
C. a, c, f, e, b, d                      D. a, e, b, c, f, d

11. 下列描述哪个论述是错误的：（ ）
- A. 并发性是指两个或多个事件在同一时刻发生
  - B. 并发性是指两个或多个事件在同一时间间隔内发生
  - C. 在单处理机系统中，程序只能并发地执行
  - D. 在多处理机系统中，程序智能并行地执行
12. 对属于同一个进程的多个线程，下列哪个对象是线程私有的？（ ）
- A. 页表
  - B. 栈
  - C. 代码段
  - D. 进程控制块
13. 多线程之间效率最高的通信方式是下列哪种方式？（ ）
- A. 共享变量
  - B. 消息传递
  - C. 管道
  - D. 共享文件
14. 以下哪个因素不会影响任务切换的实时性：（ ）
- A. 分时系统时间片大小
  - B. 当前进程是否可以抢占
  - C. 当前进程的优先级
  - D. 当前程序是否使用汇编优化
15. 当计算机的物理内存容量小于程序运行所需的存储空间时，（ ）
- A. 根本无法运行程序
  - B. 必须修改程序，使用文件等外部存储
  - C. 可以采用虚拟存储方式，提高物理内存使用率
  - D. 程序可以正常运行，无需系统支持
16. 某处理器具有 36 位虚拟地址空间，页面大小为 4KB。该处理器才用了多级页表，每级页表包含 12 位也好，那么该处理器具有：（ ）
- A. 1 级页表
  - B. 2 级页表
  - C. 3 级页表
  - D. 4 级页表
17. 关于分页和分段系统的对比，以下哪个描述是正确的？（ ）
- A. 页的大小固定且由系统决定，而分段的大小由程序决定，因此分页更容易产生内部碎片。
  - B. 分段可以方便多程序共享，分页无法实现多进程的共享。
  - C. 分页的地址空间是一维的，分段的地址空间是二维的，因此分页系统编程更方便。
  - D. 分页系统需要“页表”支持，分段系统需要“段表”支持，但通常段表项数比页表项数多。
18. 下列哪个对象不属于文件系统的一部分：（ ）
- A. 分区表
  - B. 目录
  - C. 超级块
  - D. 空闲块表
19. 下面是某 Unix 操作系统中对某个文件进行列示的内容：
- ```
$ls -l
-rw-r--r-- 1 root wheel 1152 Sep 9 01:07 /etc/hosts
```
- 说明该操作系统才用了（ ）方式保护文件访问的安全性：
- A. 强制访问控制（MAC）
  - B. 访问权限表（Capability）
  - C. 访问控制表（ACL）
  - D. 加密保护（Encryption）
20. 以下关于机群系统的描述哪个是错误的？（ ）
- A. 机群节点具有独立的内存地址空间
  - B. 每个机群节点具有独立的操作系统
  - C. 机群系统比多处理器系统具有更好的扩展性
  - D. 一个进程的多个线程可以分布在不同的节点上
21. 冯·诺依曼计算机工作方式的基本特点是（ ）
- A. 多指令流单数据流
  - B. 按地址访问并通常顺序执行指令
  - C. 堆栈操作
  - D. 机器以控制器为中心

22. 数字运算的操作次数最少的机器数表示形式是 ( )  
A. 原码      B. 反码      C. 补码      D. 三个都不是
23. 某字长为 8 位的计算机中, 已知变量  $x$ 、 $y$ 、 $z$  的类型为 `int` (用补码表示),  $[x]_{\text{补}}=11110100$ ,  $[y]_{\text{补}}=10110000$ 。若  $z=x/2+2y$ , 则  $z$  的机器数为 ( )  
A. 11000000      B. 00100100      C. 11011010      D. 溢出
24. 如果一个存储单元被访问, 那么与该单元临近的存储单元也可能被访问, 该特性称为 ( )  
A. 时间局部性      B. 空间局部性      C. 程序局部性      D. 数据局部性
25. 磁盘转速提高一倍, 则如下论述正确的是 ( )  
A. 平均寻道使劲缩小为原来的一半  
B. 其存取速度提高一倍  
C. 平均等待时间缩小为原来的一半  
D. 不影响磁盘的数据传输速率
26. 下列关于 RISC 的叙述中, 错误的是 ( )  
A. 丰富的指令格式和寻址方式  
B. 只有 `LOAD` 和 `STORE` 指令可以访问存储器  
C. 指令长度固定, 指令执行一般采用流水方式  
D. 控制器采用组合逻辑设计, 不用微程序控制
27. 相对于硬布线控制器, 微程序控制器的特点是 ( )  
A. 指令执行速度慢, 指令功能的修改和扩展容易  
B. 指令执行速度慢, 指令功能的修改和扩展难  
C. 指令执行速度快, 指令功能的修改和扩展容易  
D. 指令执行速度快, 指令功能的修改和扩展难
28. 某 CPU 采用 3 级指令流水线, 取指、分析、执行三个流水段的事件分别是  $2\text{ns}$ 、 $2\text{ns}$  和  $1\text{ns}$ 。假定 CPU 执行了 100 条指令, 在其执行过程中, 没有发现任何流水线阻塞, 则 100 条指令全部执行完毕需要 ( )  
A.  $500\text{ns}$       B.  $200\text{ns}$       C.  $204\text{ns}$       D.  $203\text{ns}$
29. 多总线结构优于单总线结构的原因是 ( )。  
A. 减少了寄存器数目      B. 提高总线的吞吐率  
C. 提高了互联能力      D. 减少内存需求
30. 关于 DMA 方式的论述, 正确的是 ( )  
A. DMA 方式利用软件实现数据传送  
B. DMA 方式请求的是总线使用权  
C. DMA 方式适用于数据传输率比较低的外部设备  
D. DMA 方式在传送过程中需要 CPU 的干预
31. 在 OSI 参考模型中, 负责提供可靠的端到端数据传输的是 ( ) 的功能。  
A. 数据链路层      B. 运输层      C. 网络层      D. 应用层
32. ( ) 代表以双绞线为传输媒介的快速以太网。  
A. 10BASE2      B. 10BASE5      C. 100BASE-F      D. 1000BASE-T
33. 以太网媒体访问控制技术 CSMA/CD 的机制是 ( )  
A. 循环使用带宽      B. 预约带宽      C. 争用带宽      D. 按优先级分配带宽
34. 令牌环网中某个站点能发送帧是因为 ( )  
A. 令牌到达      B. 优先级最高      C. 最先提出申请      D. 可随机发送
35. IP 数据包的最大长度为 ( )

- A. 1500                      B. 1520                      C. 65535                      D. 65526
36. 以太网中，在第 6 次碰撞之后，一个节点选择等待 8 时间槽的概率是（ ）  
A. 1/8                      B. 1/16                      C. 1/32                      D. 1/64
37. 当一台计算机从 FTP 服务器下载文件时，在该 FTP 服务器上对数据进行封装的五个转换步骤是（ ）  
A. 比特，数据帧，数据包，数据段，数据  
B. 数据，数据段，数据包，数据帧，比特  
C. 数据包，数据段，数据，比特，数据帧  
D. 数据段，数据包，数据帧，比特，数据
38. 如果主机所询问的本地域名服务器不知道被查询域名的 IP 地址，那么本地域名服务器就以 DNS 客户的身份向其他根域名服务器继续发出查询请求报文，这种查询方式称为（ ）  
A. 递归查询                      B. 迭代查询                      C. 迭代与递归查询相结合                      D. 高速缓存查询
39. 滑动窗口流控机制工作原理中，可直接传输并不需要确认的是（ ）  
A. 滑动窗口左边的数据                      B. 滑动窗口右边的数据  
C. 滑动窗口内的数据                      D. 滑动窗口收缩的数据
40. TCP 建立连接过程中，通过抓包软件 wireshark 捕获到一个数据包，其中 Seq: 0, Ack: 1, [SYN set 1, ACK set 1] (SYN, ACK 标志位均为 1) 则正常情况下，下一个数据包的 Seq, Ack 为（ ）  
A. Seq: 1, Ack: 1                      B. Seq: 1, Ack: 2  
C. Seq: 0, Ack: 2                      D. Seq: 0, Ack: 1

## 二、问答题（题号 41-47，共 70 分）

41. 设图书馆的读者记录包含下列数据项：图书证号，姓名，身份证号，性别，当前在借图书数量，借书记录和还书记录。其中，借书记录、换书记录均包含书号、书名、借出时间、归还日期。

1) 请设计一个合理的存储结构，以保存上述读者记录。要求给出该存储结构的数据类型定义，并给出存储结构示意图（3 分）

2) 设计一个尽可能高效的算法，用于为读者共借书（在借图书超过  $n$  时不能再借书）和换书功能。要求给出算法的设计思想。并采用 C 或 C++ 或 Java 语言描述算法（5 分）

3) 说明上述所设计算法的时间复杂度和空间复杂度（2 分）

42. 给定一棵用二叉链表表示的二叉树，给定其中的结点  $x$  和  $y$ ，编写算法，显示  $x$  和  $y$  结点的所有公共祖先结点的值。

1) 要求给出算法的设计思想（4 分）

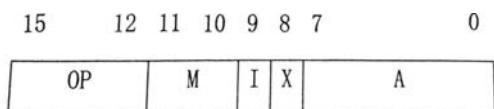
2) 采用 C 或 C++ 或 Java 语言描述算法（6 分）

43. 用单向链表实现一个 FIFO 队列的入队、出队函数，并用互斥锁(mutex)保证多线程下的数据一致性，给出程序代码（程序代码可以采用 C、C++ 或 Java 语言的任意一种实现，可不用考虑例外处理）。（7 分）

44. (1) 试从操作系统角度简单描述：进程在用户态执行过程中，从一个打开的文件中读取一个字节操作的全过程。（6 分）

(2) 这个过程中，数据最多进行了几次拷贝？（2 分）

45. 某计算机存储器容量为  $64K \times 16$  位，按字节编址。采用 16 位单一固定字长指令格式，指令各字段定义如下：



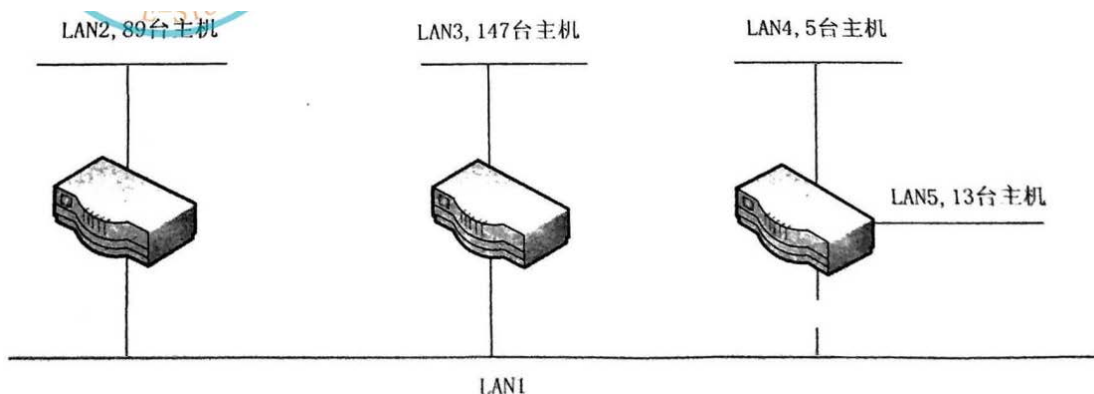
其中 M 为寻址模式，I 为间址特征，X 为变址特征。转移指令采用相对寻址方式，相对便宜用补码表示。转移执行时，转移目标地址为  $(PC)+2+A$ ；顺序执行时，下条指令地址为  $(PC)+2$ 。回答下列问题：（8 分，每问 2 分）

- 1) 该指令系统最多可有多少指令？
- 2) 存储地址寄存器（MAR）和存储数据寄存器（MDR）至少各需多少位？
- 3) 转移指令的目标地址范围是多少？
- 4) 某转移指令的地址为 2009H，要求成果转移以后的目标地址是 2000H，则该转移指令字段 A 的内容是多少？

46. 假设某计算机的 CPU 主频为 80MHz，每指令周期 CPI 为 4，并且平均每条指令访存 1.5 次，主存容量为 16MB。Cache 容量为 16KB，Cache 块大小为 32B，采用 4 路组相联映射。请回答以下问题：

- 1) 该计算机的 MIPS（百万指令/秒）是多少？（2 分）
- 2) 设 Cache 初始状态为空，CPU 以此从主存第 0、1、2、…、99 号单元读出 100 个字（主存一次读出一个字），每个字 32 位，并重复 10 次，则 Cache 的命中率是多少？（3 分）
- 3) 若 Cache 速度是主存速度的 6 倍，试问有 Cache 或无 Cache 相比，速度提高多少倍？（2 分）
- 4) 按照第二问算得的 Cache 命中率，在不考虑 DMA 传送的情况下，主存带宽至少达到多少才能满足 CPU 的方寸要求？（5 分）

47. 一个自治系统有 5 个局域网，如图所示 LAN2 至 LAN5 上的主机数分别为：89、147、5 和 13，该自治系统分配到 IP 地址块为 130.138.118.0/23，试给出每一个局域网的地址块（包括前缀）。（15 分）



2016 年参考答案

一、选择题

|   |                                                                                                                            |
|---|----------------------------------------------------------------------------------------------------------------------------|
| 1 | <p>解析：栈的储存顺序是先进后出，所以pop出栈操作将4出栈后，栈顶元素就是2。</p> <p>答案：B</p>                                                                  |
| 2 | <p>解析：第0行只有1个元素，第1行有2个元素.....第i-1行需要存储i个元素。第i行前的元素数为1+2+3...+i。第i行第i个元素，所以1+2+3+...+i+i。</p> <p>答案：A</p>                   |
| 3 | <p>解析：对于哈夫曼二叉树而言除了第一层和最后一层以外其余层只有一个叶子结点，最后一层有两个叶子结点。</p> <p>答案：C</p>                                                       |
| 4 | <p>解析：prim算法需要依次加入所有的顶点，所以时间复杂度为O(n)，对于每一个加入的顶点需要遍历所有的顶点的边的权值来判断是否确定加入，也需要O(n)。总时间复杂度O(n<sup>2</sup>)。</p> <p>答案：C</p>     |
| 5 | <p>解析：对于快速排序来说最好的情况就是每次都能选取需要排序序列的中间值作为比较子，最坏情况就是逆序排序或者顺序排序。逆序和顺序比较次数最多，但移动次数不是。选择中间值此时一趟排序的效果尽可能的好，移动次数最多</p> <p>答案：B</p> |
| 6 | <p>解析：二路归并排序第一步将n个元素分为n/2个子序列，第一趟二路归并排序是将相邻两个子序列合并成n/4个子序列。</p> <p>答案：D</p>                                                |
| 7 | <p>解析：</p> <p>查找不成功的平均查找次数：(6+5+4+3+2+1+1)/7</p> <p>答案：B</p>                                                               |
| 8 | <p>解析：计算机存储结构一般包括顺序存储，链式存储，索引存储，以及散列存储。串是一种节点元素仅由一个字符构成的线性表，存储时既可以使用顺序结构也可以使用链式结构。</p> <p>答案：A</p>                         |



|    |                                                                                                                                                          |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9  | <p>解析：稀疏矩阵压缩表示法常见的有三元组表示法、带辅助行向量的二元组表示法（也即行逻辑链表的顺序表），十字链表表示法等。</p> <p>答案：C</p>                                                                           |
| 10 | <p>解析：首先根据题目画出图，再进行深度优先遍历。深度优先遍历过程：首先访问出发点v，并将其标记为已访问过；然后依次从v出发搜索v的每个邻接点w。</p> <p>若w未曾访问过，则以w为新的出发点继续进行深度优先遍历，直至图中所有和源点v有路径相通的顶点均已被访问为止。</p> <p>答案：A</p> |
| 11 | <p>解析：并行是指两个或多个事件在同一时刻发生，并发是指两个或多个事件在同一时间间隔内发生。在单处理机系统中，程序只能并发执行，而在多处理机系统中，程序既能并行执行也能并发执行。</p> <p>答案：D</p>                                               |
| 12 | <p>解析：一个标准的线程由线程ID，当前指令指针(PC)，寄存器集合和堆栈组成。因此栈是线程私有的。</p> <p>答案：B</p>                                                                                      |
| 13 | <p>解析：线程之间效率最高的通信方式是管道。</p> <p>答案：C</p>                                                                                                                  |
| 14 | <p>解析：时间片、是否抢占和进程优先级影响任务切换的实时性。</p> <p>答案：D</p>                                                                                                          |
| 15 | <p>解析：采用虚拟存储方式，可以运行比物理内存容量大的程序。</p> <p>答案：C</p>                                                                                                          |
| 16 | <p>解析：页面大小4KB，页内偏移占低12位。虚页号占36-12=24位，每级页表包含12位页号，因此处理器具有2级页表。</p> <p>答案：B</p>                                                                           |
| 17 | <p>解析：段表项数通常比页表项数多，D正确。</p> <p>答案：D</p>                                                                                                                  |
| 18 | <p>解析：目录用于文件的逻辑组织，超级块包含文件系统的总体信息，空闲块表对磁盘的空闲块进行管理。它们都是文件系统的一部分。</p> <p>答案：A</p>                                                                           |
| 19 | <p>解析：由图知，不同的用户对文件的访问权限不同。例如只有用户自己能读写，而组成员和其他人员只能读，这是采用访问权限表的方式来保护文件访问的安全性。</p> <p>答案：B</p>                                                              |
| 20 | <p>解析：机群（cluster）系统是互相连接的多个独立计算机的集合，这些计算机可以是单机或多处理器系统（PC、工作站或SMP），每个结点都有自己的存储器、I/O设备和操作系统。机群系统比多处理器系统具有更好的扩展性。</p> <p>答案：D</p>                           |

| 21 | <p>答案：B</p> <p>解析：考察冯诺依曼计算机结果的特点。冯诺依曼机的特点为：</p> <p>(1) 计算机硬件系统由运算器、存储器、控制器、输入设备和输出设备五大部件组成。</p> <p>(2) 指令和数据以同等地位存放于存储器内。并可按地址访问。</p> <p>(3) 指令和数据均用二进制表示。</p> <p>(4) 指令由操作码和地址码组成。操作码表示操作的性质，地址码表示操作数在存储器中的位置。</p> <p>(5) 指令在存储器中按顺序存放。通常，指令是顺序执行的。在特殊情况下，可根据运算结果或指定的条件来改变运算顺序。</p> <p>(6) 机器以运算器为中心。运算器完成数据的算术运算和逻辑运算。输入输出设备和存储器之间的数据传送通过运算器和存储器完成。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |      |    |               |    |         |                    |         |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|------|----|---------------|----|---------|--------------------|---------|
| 22 | <p>答案：C</p> <p>解析：考察数据的表示和运算。在计算机中，通过补码表示机器数使得减法和加法都可以使用同一的加法器进行计算，使得数学运算的操作次数最少。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |    |      |    |               |    |         |                    |         |
| 23 | <p>答案：D</p> <p>解析：考察数据的运算。由于<math>z=x/2+2y</math>，所以首先需要计算<math>x/2</math>和<math>2y</math>，对补码表示的操作数进行乘以2或者除以2的操作实际上相当于对操作数进行左移或者右移一位的操作。</p> <p>不同机器数算术移位后的空位添补规则</p> <table border="1"> <thead> <tr> <th>数制</th><th>添补代码</th></tr> </thead> <tbody> <tr> <td>正数</td><td>原码、补码、反码<br/>0</td></tr> <tr> <td rowspan="3">负数</td><td>原码<br/>0</td></tr> <tr> <td>补码<br/>左移添0<br/>右移添1</td></tr> <tr> <td>反码<br/>1</td></tr> </tbody> </table> <p> <math>[X]_{补} = 11110100</math>    <math>[Y]_{补} = 10110000</math> </p> <p> <math>\Rightarrow [x/2]_{补} = 1111010</math>    <math>[2y]_{补} = 11100000</math> (溢出) </p> <p> <span style="margin-right: 100px;">右移一位</span> <span>左移一位</span> </p> <p>由于<math>2y</math>的结果溢出，而且<math>x/2</math>和<math>2y</math>均为负数，其相加结果<math>z &lt; 2y</math>，则<math>z</math>的机器数同样溢出。</p> | 数制 | 添补代码 | 正数 | 原码、补码、反码<br>0 | 负数 | 原码<br>0 | 补码<br>左移添0<br>右移添1 | 反码<br>1 |
| 数制 | 添补代码                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |    |      |    |               |    |         |                    |         |
| 正数 | 原码、补码、反码<br>0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |    |      |    |               |    |         |                    |         |
| 负数 | 原码<br>0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |      |    |               |    |         |                    |         |
|    | 补码<br>左移添0<br>右移添1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |    |      |    |               |    |         |                    |         |
|    | 反码<br>1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |      |    |               |    |         |                    |         |



|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 24 | <p><b>答案：B</b></p> <p><b>解析：</b>局部性原理是指CPU访问存储器时，无论是存取指令还是存取数据，所访问的存储单元都趋于聚集在一个较小的连续区域中。</p> <p><b>时间局部性：</b>如果一个存储单元正在被访问，那么在近期它很可能还会被再次访问。程序循环、堆栈等是产生时间局部性的原因。</p> <p><b>空间局部性：</b>如果一个存储单元被访问，那么该存储单元附近的区域在也可能被访问。</p>                                                                                                                                                                                                                      |
| 25 | <p><b>答案：C</b></p> <p><b>解析：</b>考察磁盘的性能指标。</p> <p>平均寻道时间指磁头移动到目的磁道的时间，与磁盘转速无关，排除A。</p> <p>存取速度=存取量/存取时间，而存取时间=平均寻道时间+旋转延迟时间+传输时间，磁盘转速提高一倍会使旋转延迟时间减半，而不能使存取速度提高一倍，排除B。</p> <p>平均等待时间是即旋转延迟时间，指当磁头已处于要访问的磁道时，等待所要访问的扇区旋转至磁头下方的时间。当磁盘转速提升一倍时，平均等待时间缩小一倍，C正确。</p> <p>数据传输率指磁盘存储器在单位时间内向主机传送数据的字节数。其计算公式为：数据传输率=转数(转/秒)*每条磁道的容量(字节)。当转速提升一倍时，数据传输率增加一倍，排除D。</p>                                                                                     |
| 26 | <p><b>答案：A</b></p> <p><b>解析：</b>考察RISC结构计算机。</p> <p><b>RISC：</b>精简指令系统计算机。特点是所有指令的格式都是一致的，所有指令的指令周期也是相同的，并且采用流水线技术。选取使用频率高的基础指令，指令条数少，复杂指令由简单指令组成。只有载入和存储指令可以访问内存，数据处理指令只对寄存器的内容进行操作。以硬布线控制为主。</p> <p><b>CISC：</b>复杂指令系统计算机。特点是指令系统复杂庞大，指令格式和寻址方式丰富，指令长度和格式不一致，可以访存的指令不受限制，大多数指令需要多个时钟周期才能完成。以微程序控制为主。</p>                                                                                                                                     |
| 27 | <p><b>答案：D</b></p> <p><b>解析：</b>考察硬布线控制器。常用的控制器设计方式有两种：硬布线控制器和微程序控制器。</p> <p>硬布线控制器的基本原理是根据指令的要求、当前的时序以及内部和外部的状态情况，按照时间的顺序发送一系列微操作控制信号。</p> <p>优点：控制器的速度取决于电路延迟，所以执行速度很快，常用于RISC CPU。</p> <p>缺点：由于将控制部件看做专门产生固定时序控制信号的逻辑电路，所以把用最少数件和取得最高速度作为设计目标，所以造成可拓展性极差，设计完成后，不可能通过修改增加新功能。</p> <p>微程序控制器采用存储逻辑实现，将微操作信号代码化，使每条机器指令换成为一段微程序并存入控制存储器（CM）中，执行指令时读出即可。</p> <p>缺点：由于微程序控制器采用了存储程序原理，所以每条指令的微程序从控存中读取，影响了速度。</p> <p>优点：具有规整性、灵活性、可拓展性、可维护性。</p> |

|    |                                                                                                                                                                                                                                                                                                |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28 | <p>答案：D</p> <p>解析：考察CPU指令流水线。</p> <p>时空图如下：</p> <p>总执行时间=2*2+100*2-1=203ns</p>                                                                                                                                                                                                                 |
| 29 | <p>答案：B</p> <p>解析：考察总线系统。</p> <p>单总线结构将CPU、主存、I/O设备（通过I/O接口）都挂在一组总线上，I/O设备与主存之间直接交换信息。</p> <p>优点：结构简单、成本低、拓展性强，缺点：带宽低、负载重，多个部件只能争用唯一的总线，且不支持并发。</p> <p>多总线结构采用多条总线，常见的双总线结构使用了主存总线用于CPU、主存、通道之间的数据传输；使用I/O总线用于多个外部设备与通道之间的数据传送。</p> <p>优点：将低速的I/O设备从单总线上分离出来，实现存储器总线和I/O总线分离；缺点：增加通道等硬件设备。</p> |
| 30 | <p>答案：B</p> <p>解析：考察DMA直接存储器访问。DMA直接存储器访问方式是一种完全由硬件进行成组信息传送的控制方式，直接在I/O设备与主存之间交换数据，交换过程有DMA控制器控制，无需CPU的干预，只需要在DMA传送开始前向CPU请求总线使用，结束后归还总线控制权。其目的是为了与磁盘等高速I/O设备进行成批交换数据。</p>                                                                                                                     |
| 31 | B                                                                                                                                                                                                                                                                                              |
| 32 | D                                                                                                                                                                                                                                                                                              |
| 33 | C                                                                                                                                                                                                                                                                                              |
| 34 | A                                                                                                                                                                                                                                                                                              |
| 35 | C                                                                                                                                                                                                                                                                                              |
| 36 | D                                                                                                                                                                                                                                                                                              |
| 37 | B                                                                                                                                                                                                                                                                                              |
| 38 | A                                                                                                                                                                                                                                                                                              |
| 39 | C                                                                                                                                                                                                                                                                                              |
| 40 | A                                                                                                                                                                                                                                                                                              |

## 二、问答题

41.

(1) struct Person

```
{
    int id;
```

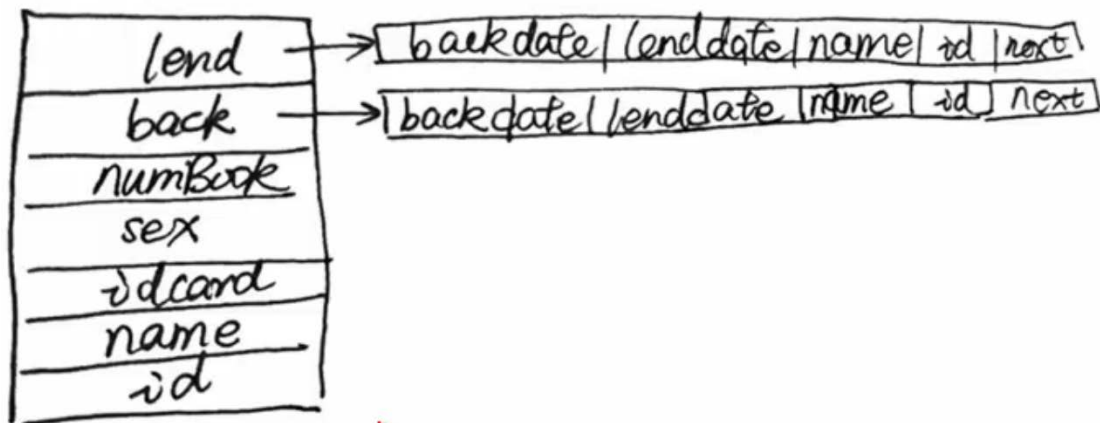
```

char name[];
int idCard[];
bool sex;
int numBook;
record *lend;
record *back;
};

struct record
{
    int id;
    char name[];
    tm lenddate;
    tm backdate;
    struct record* next;
};

```

存储结构示意图



(2)

```

void lendBook(int id, char name[], tm time1)
{
    if(numBook < n)
    {
        record *s = new record;
        s->id = id;
        s->name = name;
        s->lenddate = time1;
        s->next = lend->next;
        lend->next = s;
        numBook = numBook + 1;
    }
    else
    {
        cout << "You can't lend any book now." << endl;
    }
}

```



```

    }
}

void backBook(int id,char name[],tm time2)
{
    record *p=lend;
    while(p)
    {
        if(p->id==id)break;
        else p=p->next;
    }
    record *s=new record;
    s->id=id;
    s->name=name;
    s->lenddate=p->lenddate;
    p->backdate=time2;
    s->backdate=time2;
    s->next=lend->next;
    back->next=s;
    numBook=numBook-1;
}

```

(3) 借书的时间复杂度  $O(1)$ ，空间复杂度  $O(1)$ 。还书的时间复杂度  $O(n)$ ，空间复杂度  $O(1)$

42

(1)

**解: (1) 算法思想:** 当寻找一个结点的祖先时可以采用递归的方式进行寻找。如果根节点的左孩子或右孩子是需要寻找的顶点, 则直接输出根节点, 否则递归调用该函数。该函数输入的是以根节点的左孩子为根节点的子树和根节点的右孩子为根节点的子树。

寻找两个节点的最近公共祖先可以采用递归的方式。如果两个节点中有一个节点为根节点的左孩子或者右孩子则直接返回根节点, 否则递归调用该函数。

寻找到两个节点的最近公共祖先, 找到这个公共祖先的所有祖先结点, 即是两个节点的所有公共祖先。

(2)

bool allAncestor(BTNode \*b, ElemType t)//求所有祖先节点

```

{
    if(b==NULL)return false;
    else if((b->lchild != NULL && b->lchild->data == t) || (b->rchild != NULL &&
b->rchild->data == t))
    {

```

```

        cout<< b->data << " ";
        return true;
    }
    else if(allAncestor(b->lchild,t)||allAncestor(b->rchild,t))
    {
        cout<< b->data << " ";
        return true;
    }
    else return false;
}

```

BTNode \*nearestAncestor(BTNode \*b,ElemType x,ElemType y)//求 x 和 y 的最近公共祖先

```

{
    if(b==NULL || b->data==x || b->data==y)return b;
    BTNode *left=nearestAncestor(b->lchild,x,y);
    BTNode *right=nearestAncestor(b->rchild,x,y);
    if(left!=NULL && right!=NULL)return b;
    else if(left!=NULL)return left;
    else return right;
}

```

bool nearestAllAncestor(BTNode \*b,ElemType x,ElemType y)//求 x,y 的所有公共祖先节点

```

{
    BTNode *ancestor;
    if(b==NULL)return false;
    else
    {
        ancestor=nearestAncestor(b,x,y);
        return allAncestor(b,ancestor);
    }
}

```

43.

```
#include<iostream>
```

```
typedef struct node
```

```

{
    int data;
    struct node *next;
}

```

```
typedef struct queue
```

```

{
    node *first,*rear;
}

```



```
}
```

```
typedef struct
```

```
{
```

```
    int value;//记录了这个信号量的值
```

```
    struct process *list;//存储正在等待这个信号量的进程
```

```
}semaphore;
```

```
wait(semaphore *S)
```

```
{
```

```
    S->value--;
```

```
    if(S->value<0)
```

```
    {
```

```
        add this process to S->list;
```

```
        block();
```

```
    }
```

```
}
```

```
signal(semaphore *S)
```

```
{
```

```
    S->value++;
```

```
    if(S->value<=0)
```

```
    {
```

```
        remove a process P from S->list;
```

```
        weakup(P);
```

```
    }
```

```
}
```

```
semaphore *mutex=(semaphore *)malloc(sizeof(semaphore));
```

```
mutex->value=1;
```

```
queue *insert(queue *Q,int x)
```

```
{
```

```
    node *i;
```

```
    i=(node *)malloc(sizeof(node));
```

```
    i->data = x;
```

```
    wait(mutex);
```

```
    if(Q==NULL)
```

```
    {
```

```
        //如果队列结构体为空（还未插入值），则创建队列结构体。
```

```
        Q=(queue *)malloc(sizeof(queue));
```

```
        Q->first=Q->rear=i;
```

```
        Q->rear->next=NULL;
```

```
    }
```

```

else
{
    //按顺序将节点入队
    Q->rear->next=i;
    Q->rear=i;
    i->next=NULL;
}
signal(mutex);
return Q;
}

queue *del(queue *Q)
{
    node *temp;//先定义一个中间节点，以便将队头处理完将原队头空间释放
    wait(mutex);
    if(Q->first=NULL)
    {
        //如果队列为空，则报错退出。
        printf("Queue is null!");
        return NULL;
    }
    else
    {
        //和单链表删除头结点的处理一样
        temp=Q->first;
        Q->first=Q->first->next;
        free(temp);
    }
    signal(mutex);
    return Q;
}

```

44

答：（1）文件打开，将文件控制块复制到内存的文件打开表中，根据文件控制块中的文件物理地址找到文件的起始盘块号，然后寻找读取字节所在的盘块（如果文件采用连续结构可以直接计算得出盘块号进而访问，如果采用非显式链接结构，需要逐个访问找到盘块然后访问，如果采用索引结构，需要查询索引表找到盘块号）。找到对应的盘块后读出该字节（一次拷贝），读入到内存中，再从内存中读入到CPU的寄存器中（二次拷贝）。

（2）数据最多进行两次拷贝。

45

答：(1) 由于该指令格式的OP字段长度为4位，则该指令系统最多可以有 $2^4=16$ 种指令。

(2) 由于该计算机按字节编址，即每个存储单元大小为8bit，则MDR至少为8位；又因为存储器容量为 $64K \times 16\text{位} = 128K \times 8\text{位}$ ，即共有 $128K = 2^7 \times 2^{10} = 2^{17}$ 个存储单元，则MAR至少为17位。

(3) 由于相对偏移用补码表示，长度为8位，则其范围为-128~127，且转移目标地址为 $(PC) + 2 \times A$ ，则转移指令的目标地址范围为00000H~1FFFFH。

(4) 由于该转移指令的地址为2009H，则执行到该指令时的PC寄存器的值应当为200BH，要求转移成功后的目标地址为2000H，则有 $2000H = 200BH + 2 \times A$ ，所以 $A = -6H = [1000\ 0110]_{\text{原}} = [1111\ 1010]_{\text{补}}$ 。

46

答：(1)  $\text{MIPS} = \frac{\text{指令条数}}{\text{执行时间} \times 10^6} = \frac{\text{主频}}{\text{CPI}} = \frac{80}{4} = 20$ 。

(2) 由于cache容量为16KB，且cache块大小为32B，则cache共有 $16KB / 32B = 2^9$ 个块，且每个cache块可以存放 $32B / 32\text{bit} = 8$ 个字。由于cache采用四路组相联结构，即 $2^9$ 个cache块分为 $2^9 / 4 = 2^7$ 个组，每组内有4个cache块。

由于cache块大小为32B，且每个字32bit=4B，则每个cache块可以存放 $32B / 4B = 8$ 个字。由于题目要求从第0号单元读出100个字，则这100个字应当存放在连续的13个内存块中，由于cache共有 $2^7$ 个组，则这13个内存块映射到cache不同的13个组中，即第一次访问完100个字后，这100个字会同时存放在cache中。则按照题意依次访问100个字共重复10次相当于访问内存 $100 \times 10 = 1000$ 次，其中在第一次访问每个内存块第一个存储单元时会产生cache缺失，即产生了13次缺失，则命中率 $= 1 - 13 / 1000 = 98.7\%$ 。

(3) 设无cache的情况下速度为S，即主存速度为S。则有cache情况下的速度 $s_1 = 98.7\% \times 6 \times S + 1.3\% \times S = 5.935 \times S$ 。则速度提升了5.935倍。

(4) 由于平均每条指令访存1.5次，且MIPS=20，即每秒执行的 $20 \times 10^6$ 条指令需要访存 $20 \times 10^6 \times 1.5 = 30 \times 10^6$ 次。由于cache的命中率为98.7%，则其中 $30 \times 10^6 \times (1 - 98.7\%) = 0.39 \times 10^6$ 次需要从主存中读取一个内存块的数据，则主存带宽至少 $= 0.39 \times 10^6 \times 32B / s = 12.48MB / s$ 。

47

答案：分配网络前缀应该分配地址数目较多的前缀。

注意：题目中的LAN1共有3个路由器，再加一个网关地址，共4个IP地址

LAN2 LAN3 LAN4 LAN5分别有1台路由器，一个路由器的端口也要占用一个ip地址

(1) LAN3：主机数=147， $2^7 - 2 < 147 + 1 < 2^8 - 2$ ，主机号占8Bit，网络前缀占24bit

取第24bit为0，分配地址块是130.138.118.0/24。可用的有效IP地址范围是：

130.138.118.1-130.138.118.254。147台主机的ip在这个范围内可以随意选择。

(2) LAN2: 主机数=89,  $2^6-2 < 89+1 < 2^7-2$ , 主机号占7Bit, 网络前缀占25bit

取第24bit为1, 第25bit为0, 分配地址块是 130.138.119.0/25. 可用的有效IP地址范围是: 130.138.119.1-130.138.119.126. 89台主机的ip在这个范围内可以随意选择。

(3) LAN5: 主机数=13,  $2^3-2 < 13+1 < 2^4-2$ , 主机号占4Bit, 网络前缀占28bit

取第24bit为1, 第25bit为1, 第26bit为1. 第27bit为1, 第28bit为0。

分配地址块是 130.138.119.224/28. 可用的有效IP地址范围是:

130.138.119.225-130.138.119.238, 共14个有效IP地址。

~~(4) LAN1: 共有3个路由器, 再加一个网关地址, 共4个IP地址。~~

~~$2^2-2 < 3+1 < 2^3-2$ , 主机号占3Bit, 网络前缀占29bit。取第24bit为1, 第25bit为1, 第26bit为1. 第27bit为1, 第28bit为1, 第29bit为0。分配地址块是 130.138.119.240/29. 可用的有效IP地址范围是:~~

~~130.138.119.241-130.138.119.246, 共6个有效IP地址。~~

(5) LAN4: 主机数=5,  $2^2-2 < 5+1 < 2^3-2$ , 主机号占3Bit, 网络前缀占29bit。取第24bit为1, 第25bit为1, 第26bit为1. 第27bit为1, 第28bit为1, 第29bit为1。分配地址块是 130.138.119.248/29. 可用的有效IP地址范围是: 130.138.119.249-130.138.119.254, 共6个有效IP地址。