

2000 年硕士研究生入学考试试题

考试科目:

软件基础

注:

数据结构部分(共 50 分)

一、简答题(每小题 5 分, 共 30 分)

1. 在利用某算法能解决某应用问题的相应表格中打上√。

	求最短路径	判定有向图是否存在回路	求连通分量个数
拓扑排序算法		✓	
深度优先搜索算法		✓	✓
迪杰斯特拉算法	✓		
弗洛伊德算法	✓		

2. 对插入、选择、快速和归并四种排序算法, 回答下列问题:

- (1) 在待排序的元素序列基本有序时, 效率最高的排序方法是哪一种? 插入
- (2) 排序要求内存量最大的排序方法是哪一种? 归并
- (3) 关键字比较次数与元素的初始排列次序无关的排序方法是哪一种? 选择
- (4) 写出其中排序不稳定的方法。选择、快速

3. 设循环队列 cq 的队首指针为 $front$, 队尾指针为 $rear$, 队列可以容纳的最大元素个数为 max , 分别用下列三种方法来区分队满或队空, 试在表中写入相应的处理。

	用计数变量 C 记载元素个数	用标志位 $flag$	牺牲一个元素的存储单元
初始空队列各变量初值	$C := 0$ $front = rear = \text{任意值}$	$flag := \text{true} \text{ false};$ $front = rear = \text{任意值}$	$front = rear = \text{任意值}$
出队前判队空条件	$C = 0$	$front = rear \text{ AND } \text{not } flag$	$front = rear$
入队前判队满条件	$C = max$	$front = rear \text{ AND } flag$	$(rear + 1) \bmod max = front$
出队时该方法的特殊处理	$C := C - 1$	出队后, 则令 $flag := \text{false}$	
入队时该方法的特殊处理	$C := C + 1$	入队后, 使 $front = rear$ 则令 $flag := \text{true}$	

4. 在线性表的顺序和链式存储结构下, 试分析下表各种基本运算的时间复杂度, 并填入相应表格中。

运算	求表长	取元素	取前趋	取后继	插入
顺序存储结构	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(n)$
链式存储结构	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)/O(1)$

5. 对二叉树回答下列问题:

- (1) 在先序、中序和后序遍历结果中, 哪些结点的相对次序不发生改变? **叶结点**
 (2) 若中序遍历某二叉树得到一个结点值递增的有序序列, 则该二叉树为二叉排序树。

该判断是否正确? 为什么? **正确**

- (3) 设二叉树中无度为 1 的结点, 试用叶结点数表示二叉树的结点数。 $n = 2n_0 - 1$

设叶结点数为 n_0

$$\begin{aligned} n_0 = n_2 + 1 \Rightarrow n_2 = n_0 - 1 \\ n = n_0 + n_2 \Rightarrow n = 2n_0 - 1 \end{aligned}$$

6. 对折半查找(二分查找)回答下列问题:

顺序存储结构 表元素有序

- (1) 折半查找对查找表的存储结构和表中元素有何特殊要求?
 (2) 在满足折半查找要求的、具有 20 个元素的查找表上进行折半查找, 比较四次查找成功的元素个数是多少? 比较五次查找成功的元素个数又是多少? 平均查找长度是多少?
 $ASL = (1 \times 1 + 2 \times 2 + 3 \times 4 + 4 \times 8 + 5 \times 5) / 20 = 3.7$

二、算法题(每小题 10 分, 共 20 分)

1. 设广义表 LS 用如下表头表尾链结构存储, 试编写分别实现取表头 HEAD() 和取表尾 TAIL() 的过程。

表结点:

Tag = 1	hp	tp
---------	----	----

元素结点:

Tag = 0	data
---------	------

PROC HEAD(ls: glist; VAR h: glist; x: elemtp);

{表头为元素时: h 为 NIL, x 为元素; 表头为表时: h 指向该表, x 为 NULL}

ENDP; {HEAD}

PROC TAIL(ls: glist; VAR h: glist);

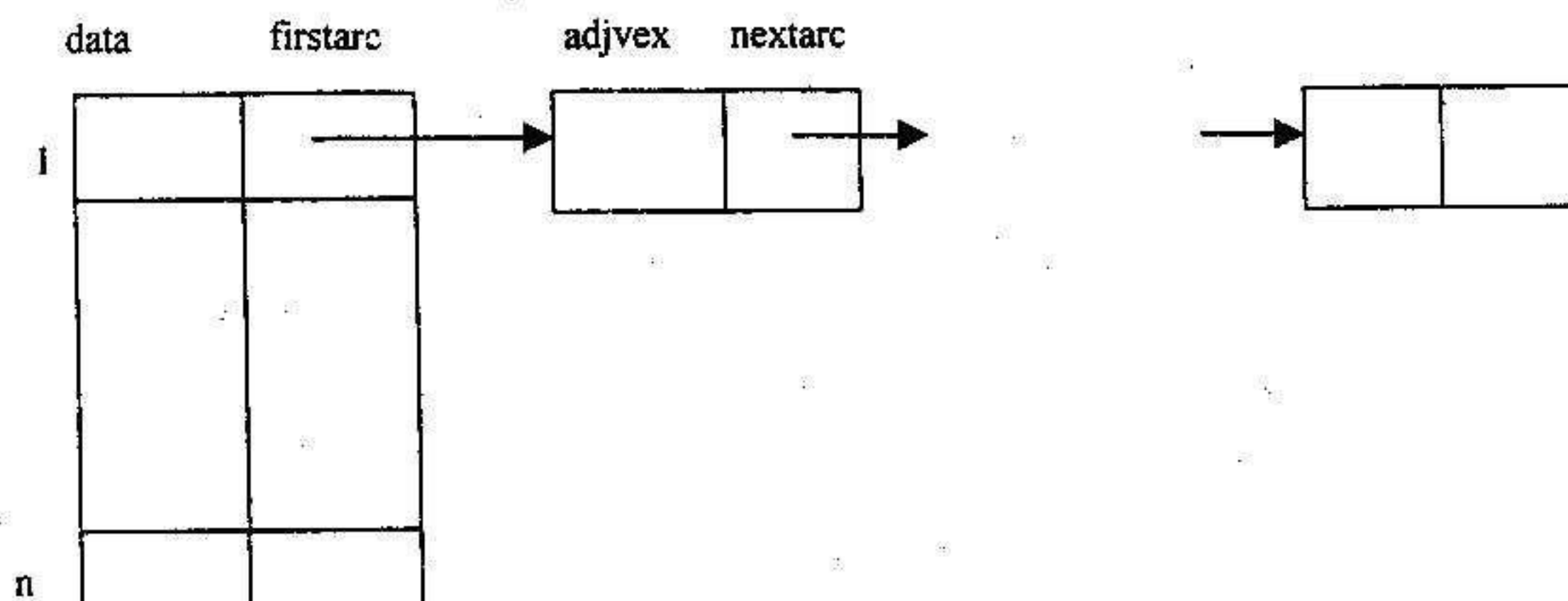
{h 指向表尾}

ENDP; {TAIL}

2. 试编写将用二叉链表表示的具有 n 个结点的二叉树转换成用邻接表存储二叉树的算法。
 设二叉链表的结点结构为：

lchild	data	rchild
--------	------	--------

邻接表的头结点数组为 $\text{adjlist}(1:n)$ ，数据结构为：



并假设算法中可以直接使用以下队列操作：

初始化操作： $\text{INIQUEUE}(Q)$

判队空函数： $\text{EMPTY}(Q)$

入队操作： $\text{ENQUEUE}(Q, x)$

出队函数： $\text{DLQUEUE}(Q)$

$\text{PROC bt_to_adj}(\text{bt: bitreptr}; n: \text{integer}; \text{adjlist: ARRAY}[n] \text{ OF vexnode});$

{bt 为二叉树的根指针, n 为结点数, adjlist 为邻接表头结点数组}

$\text{ENDP}; \{ \text{bt_to_adj} \}$

操作系统 (共 50 分)

一、单选题 (请选出一个正确的编号填入括号中, 每小题 1 分, 共 7 分)

1. 线程是进程的实体, 意味着 (2)
 ① 线程在进程中是唯一的
 ② 线程可以使用进程中的资源
 ③ 线程在运行中不能中断
 ④ 在同一进程中的多个线程具有不同的地址空间
2. 检测死锁的算法是在 (4)
 ① 程序中申请资源时使用
 ② 死锁出现之后使用
 ③ 死锁即将出现时使用
 ④ 定时检查系统状态时使用
3. 在下列问题中, 哪一个不是设备中应考虑的问题 (4)
 ① 设备的固有属性
 ② 与设备无关性
 ③ 安全性
 ④ 及时性
4. 在下列哪一个不是外存分配方式 (3)
 ① 连续分配
 ② 链接分配
 ③ 互斥分配
 ④ 索引分配
5. 联想存储器就是 (1)
 ① 快表
 ② 页表
 ③ 段表
 ④ 内存
6. 磁盘为共享设备的主要原因是 (2)
 ① 多个用户可同时访问磁盘
 ② 磁盘空间可让多个用户共享
 ③ 磁盘可支持 SPPOOLING 技术
 ④ 磁盘有多个磁头
7. 指出以下非临界资源 (4)
 ① 变量.
 ② 数据结构.
 ③ 队列.
 ④ 纯代码.

二、填空题 (每小题 1 分, 共 6 分)

1. 用户与操作系统的接口是: 命令接口 和 程序接口.
2. 多处理机有两种结构: 紧密耦合MPS 和 松散耦合.
3. I/O 控制方式有: 程序 I/O 方式, 中断驱动方式 和 DMA方式.

4、产生死锁的原因：竞争资源 和 进程推进顺序不当。

5、文件保护的方法有：访问矩阵，访问控制表 和 访问权限表。

用于磁盘的主要调度算法有：先来先服务 FCFS 最短寻道时间优先 SSTF 和 扫描法 SCAN。

三、判断改错题（正确的括号中填 V，否则填 X，每小题 2 分，共 16 分）

1、(X) 缓冲技术是以空间换时间，而且只能在设备使用均衡时起到平滑作用。
不均衡时

2、(X) 动态重定位与装入时动态链接在概念上是相同的。

3、(V) 在分时系统中采用虚拟存储技术可以改善响应时间。

4、(X) 在现代的分时系统中，逻辑处理机隐含了虚拟处理机的功能。
↓
数量上扩展 功能上扩展

[把"V"也对]

5、(V) 独享设备与共享设备的属性不同，其共享方式也不同。

6、(V) 采用 AND 型信号量机制是为了防止系统的不安全。

7、(X) 如果一个站点既可以作为客户，又可以作为服务器向其它站点提供服务，称为客
户/服务器模式。

8、(V) 设备处理程序是 I/O 进程与设备控制器之间的通信程序。

在页式存储管理系统中, 实现程序共享时, 必须对共享程序给出相同的页号. 因为页式存储管理中, 程序执行前必须实现链接, 以后就固定下来不能改变. 其它程序要共享某页, 必须使用相同页号. 而段式管理中实现程序共享时共享段的段号可以不同, 因为段式管理支持动态链接, 通过地址变换就可实现程序共享.

四、问答题 (每小题 7 分, 共 21 分)

为什么在页式存储管理中实现程序共享时, 必须对共享程序给出相同的页号, 而段式存储管理系统实现程序共享时, 共享段的段号是否一定要相同? 如相同, 为什么相同? 如不相同, 为什么不相同?

- 1、假定一个操作系统的进程调度采用抢占式短进程优先调度策略 (单 CPU), 系统中各进程到达的时间如下表所示. 请给出各进程的调度次序, 并计算平均周转时间和平均加权周转时间. 1-796

进程	到达就绪队列的时间	执行时间
P1	1	8
P2	2	4
P3	3	9
P4	4	5

注: 表中的时间均为基本单位时间.

调度次序: P1 P2 P4 P3 P1

- 2、记录型信号量的值为 N , 最少可由多少个经典信号量来实现记录型信号量的功能, 为什么?

答: 最少 2 个信号量. 记录型信号量值为 N , 最少可由 2 个经典信号量实现, 其中一个作互斥信号量, 另一个作资源信号量. 设变量 m 为资源数 N , 设有 2 个信号量 S_1, S_2 . 则算法如下.

```

S1 = 1
m = N
S2 = 0
P(S1)  // 保证对 m 操作互斥
if (m > 0)
{
    m = m - 1;
    V(S1); // 释放 m 变量
}
else P(S2) // 当 m < 0 进程等待
P(S1) // 互斥使用 m 变量
m = m + 1
if (m == 0)
{
    m = 1;
    V(S2);
}
V(S1)
    
```

作为资源信号量. 这样, 既可互斥访问临界资源, 又能记录申请资源个数.

else V(S1); 隐含 m > 1 释放 m 变量