

# 安徽师范大学

## 2020 年硕士研究生招生考试初试试题

科目代码: 896

科目名称: 计算机理论基础

### 第一部分 数据结构 (80 分)

#### 一、简答题 (每小题 4 分, 共 20 分)

1. 简述数据结构的逻辑结构与存储结构的含义及其相互关系。
2. 线性表有两种存储结构, 在什么情况下选用顺序表比链表更好。
3. 栈底元素是不能删除的元素, 此说法是否正确? 请给出理由。
4. 任意一个具有  $n$  个结点的二叉树, 已知它有  $m$  个叶子结点, 试证明它有  $n-2m+1$  个度数为 1 的结点。
5. 具有  $n$  个顶点的有向完全图中有多少条边?

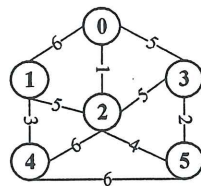
#### 二、应用题 (每小题 8 分, 共 40 分)

1. 给出下列算法被调用后得到的输出结果。

```
char x='e';
initqueue(q); //初始化队列 q
char y='c';
enqueue(q, 'h'); enqueue(q, 'r'); enqueue(q, y); dequeue(q, x);
enqueue(q, x); dequeue(q, x); enqueue(q, 'a');
while(!queueempty(q)) { dequeue(q, y); printf(y); }
printf(x);
```

注意: enqueue(q, 'h')是将 h 入队, dequeue(q, x)是出队, 并将出队元素赋给 x, queueempty(q)用于判断队列是否为空, 为空时, 其值为真。

2. 设一棵二叉树的先序序列为 ABDEGCF, 中序序列为 DBGEACF。请画出这棵二叉树, 并给出其后序序列。
3. 请采用 Prim 算法为下图构造最小生成树, 并给出构造过程。



4. 将关键字序列 {13, 5, 20, 19, 15, 8} 散列存储到一个散列表中, 该散列表的地址范围为 0-7, 散列函数为:  $H(\text{key}) = \text{key} \text{ MOD } 7$ , 采用线性探测再散列法处理冲突。请构造该散列表, 并给出每个关键字的探测次数。
5. 设待排序的关键字序列为 {15, 5, 16, 2, 25, 8, 20, 9, 18, 12}, 试给出采用快速排序算法对该序列进行升序排序时的每一趟排序结果。

三、算法设计题（每小题 10 分，共 20 分）

1. 设计一个算法，用头插法创建包含  $n$  个数据元素的带头结点的单链表  $L$ 。
2. 假设二叉树采用二叉链表存储结构存储，设计算法输出其先序遍历序列中第  $k$  个结点的值，假设  $k$  不大于二叉树的总结点数（结点数据域类型为字符型）。

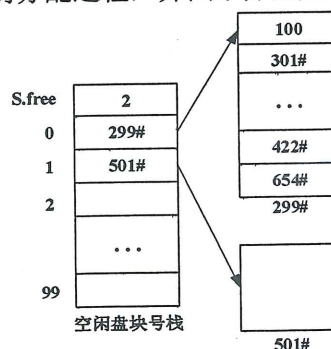
第二部分 操作系统（70 分）

一、简答题（每小题 5 分，共 30 分）

1. 什么是操作系统？
2. 说明进程的三种基本状态。
3. 产生死锁的四个必要条件是什么？
4. 某分页系统的逻辑地址为 16 位，其中高 6 位为页号，低 10 位为页内偏移量，则每页的字节数是多少？最多可有多少页？
5. 按资源分配方式可将外部设备分为哪几类？
6. 分析文件的外存分配中连续分配方式的优缺点。

二、应用题（每小题 10 分，共 40 分）

1. 现有某类资源 12 个，供 3 个进程共享。假定进程 A 已占 1 个资源，其最大需求 4 个，进程 B 已占 4 个资源，其最大需求 6 个，进程 C 已占 5 个资源，其最大需求 8 个。当 3 个进程同时请求尚需的资源时，系统应按怎样的次序为它们分配资源以保证不发生死锁，并解释之。
2. 在采用请求分页存储管理的计算机系统中，运行一个共有 8 页的作业，且该作业在主存中分配到 4 块主存空间，作业执行时访问页面顺序为 7, 0, 1, 2, 3, 0, 4, 3, 2, 3, 6, 7, 3, 1, 5, 7, 6, 2, 6, 7。请问用 LRU 页面调度算法时的缺页率是多少？并给出分析过程。
3. 在 UNIX 文件系统中，盘块的分配与释放是借助于成组链接法中的空闲盘块栈进行的。假定某时刻空闲盘块栈如右图所示，此时要为某文件分配 3 个盘块。请说明分配过程，并图示分配完毕后有关数据及表目的更改情况。



4. 某博物馆最多可容纳 500 人同时参观，只有一个出入口，该出入口一次仅允许一个人通过。当博物馆内少于 500 人时，参观者可进入，否则需在入口外等待。若把一个参观者看作一个进程，用 P、V 操作管理这些并发进程时，请回答如下问题：（1）定义信号量，写出信号量的初值及其含义。（2）插入应执行的 P、V 操作以保证进程能够正确地并发执行。

```

cobegin
    参观者进程 i:
    {
        进门;
        参观;
        出门;
    }
coend
    
```