

电子科技大学
2010 年攻读硕士学位研究生入学试题
科目名称：计算机专业基础
操作系统参考答案

请注意：所有答案必须写在答题纸上，做在试卷或草稿纸上无效。

一、单项选择题（在每小题 2 分，共 20 分）

1. C 2. C 3. C 4. D 5. B 6. D 7. A 8. C 9. B 10. B

二、多项选择题（在每小题的五个备选答案中，选出二个至五个正确的答案，并将其号码分别填在题干的括号内，多选，少选、错选，均无分。每小题 2 分，共 10 分）

1. CDE 2. BC 3. DE 4. ABCD 5. ABE

三、判断并改错（每小题 2 分，共 14 分）

1. × 2. × 3. × 4. × 5. √ 6. √ 7. √

四、简答题（共 31 分）

1. 答：平均访问时间= $0.9 \times 20 + (1-0.9) \times 0.6 \times (60+20) + (1-0.9) \times (1-0.6) \times (12 \times 10^6 + 60+20)$
 $= 4.8 \times 10^5 \text{ (ns)}$

2. 答：调度次序：P1, P2, P3, P2, P1, P4
平均周转时间=8.5
平均带权周转时间=1.625

3. 答：每一块中能记录的数据块数= $512/4 \approx 128$
一级索引时文件最大长度的字节数= 128×512
二级索引时文件最大长度的字节数= $128 \times 128 \times 512$
三级索引时文件最大长度的字节数= $128 \times 128 \times 128 \times 512$
最大搜索文件的长度= $(10+128+128 \times 128+128 \times 128 \times 128) \times 512$

电子科技大学
2009 年硕士学位研究生入学试题
科目名称：计算机专业基础

数据结构 (75 分)

一、单项选择题：从备选答案中选择一个正确的答案（每小题 1 分，共 10 分）

1. 线性表是一个(①)。

① 有限序列，可以为空	② 有限序列，不能为空
③ 无限序列，可以为空	④ 无序序列，不能为空
2. 在下列 4 种排序算法中，不能保证每趟排序至少能将一个元素放到其最终位置上的排序方法是(③)。

① 快速排序	② 冒泡排序排序
③ 希尔排序	④ 堆排序
3. 单链表中各结点之间的地址(④)。

① 必须连续	② 一定不连续
③ 部分地址必须连续	④ 连续与否无所谓
4. 能正确完成删除单链表中 p 所指结点的后继的操作是(②)。

① $p=p \rightarrow next;$	② $p \rightarrow next=p \rightarrow next \rightarrow next;$
③ $p \rightarrow next=p;$	④ $p=p \rightarrow next \rightarrow next;$
5. 与 Hash 查找效率无关的因素是(④)。

① 哈希函数是否均匀	② 处理冲突的方法
③ 哈希表的装填因子	④ 缩小查找范围的大小
6. 在下列关于平衡二叉树的叙述中, 不正确的是(①)。

① 二叉树中所有结点的左右子树的深度之差小于等于 1	② 二叉树中所有左右子树的深度之差的绝对值小于等于 1
③ 二叉树中每个结点的平衡因子= $\{-1, 0, 1\}$	④ 二叉树中度为 0 的叶结点比度为 2 的结点多
7. 用广义表可以表示一棵二叉树。在下列广义表中，不能唯一对应一棵二叉树的是(④)。

① $(A(B(d, e), c))$	② (a)
③ $()$	④ $(A(B(C, D(e))))$
8. 在下列关于单枝二叉排序树的叙述中, 不正确的是(③)。

① 非叶结点的度均为 1	② 查找性能退化为顺序查找
③ 顺序存储时占用空间最大	④ 树的深度与等于结点个数
9. 给定结点个数 n, 在下面二叉树中, 叶结点个数不能确定的是(④)。

① 满二叉树中	② 完全二叉树
③ 哈夫曼树	④ 二叉排序树
10. 关键路径是 AOE 网中(①)。

① 从源点到汇点的最长路径	② 从源点到汇点的最短路径
③ 最长的回路	④ 最短的回路

二、填空题（每空 1 分，共 10 分）

1. 读取数组给定下标的数据元素的操作，称为取值操作；存储或修改数组给定下标的数据元素的操作，称为赋值操作。

2. 若能得到拓扑有序序列, 则有向图 无 回路。
3. 连通分量是无向图中的 极大 连通子图。
4. 实现二叉树按层次遍历算法时, 最适合的数据结构是 队列。
5. 对查找表除进行查找操作外, 可能还要进行向表中插入数据元素, 或删除表中数据元素的表, 称为 动态查找表。
6. 对广义表进行操作, 结果总是表的基本操作是 取表尾 操作。
7. 实现基数排序算法时, 最适合的数据结构是 链队列。
8. 广义表 $L=(a, (b, c), e, (d), ())$ 的长度是 5、深度是 2。

三、简答题 (每小题 5 分, 共 35 分)

1. 队列是一个表头和表尾, 既能插入又能删除的线性表。该说法是否正确? 为什么?
答: 不正确。队列是一个限制只能在表头删除和只能在表尾插入的线性表。上述说法只说明了队列有表头和表尾, 而插入和删除位置没有具体限定。
2. 什么样的连通图其最小生成树是唯一的?
答: 具有 n 的顶点, $n-1$ 条边的连通图其生成树是唯一的。
3. 已知一棵二叉排序树 BST 和中序遍历算法 inorder, 如何能得到从大到小的结点序列。
答: 方法 1: 修改中序遍历算法为 RNL, 即先递归遍历右子树, 输出根结点, 然后递归遍历左子树;
方法 2: 将二叉排序树的所有左右子树交换, 然后用中序遍历算法遍历, 所输出的结点序列就是从大到小的有序序列;
4. 已知 $\{4, 7, 14, 20, 48, 25, 15\}$ 是小顶堆按层次遍历输出的结点序列, 插入 2 经过调整形成新的堆, 给出新堆按层次遍历输出的结点序列。

答: 2, 4, 14, 7, 48, 25, 15, 20

5. 对一个连通网, 用迪杰斯特拉算法求得指定顶点到其他 $n-1$ 个顶点之间的最短路径, 由这些最短路径边构成的生成树也一定是一棵最小生成树。上述说法是否正确? 并举例说明。
答: 不对。得到生成树不一定是最小生成树。如:

V1 0 1 3 5

V2 1 0 ∞ ∞

V3 3 ∞ 0 4

V4 5 ∞ 4 0

以 V1 为源点用最短路径求得的生成树是 (邻接矩阵的第一行), 生成树代价为 $1+3+5=9$; 而用最小生成树求得的最小生成树为 $\{(V1, V2), (V1, V3), (V3, V4)\}$, 生成树代价为 $1+3+4=8$, 所以说法是错误的。即所谓加权无向图 G 的最小生成树 T 就是将 G 中各结点间的最短路径作为边所构造出来的 G 的子图说法是不对的。

三、 算题 (每小题 10 分, 共 20 分)

1. 设 p 为指向循环单链表 L 中某一结点的指针, 循环单链表 L 既无头结点也无头指针, 长度大于 1, 结点指针域为 next, 试编写算法, 完成删除 p 所指结点的前趋结点。

详解:

- 正确性:
1. 不考虑空表和只有一个结点的情形, 所以总可以删;
 2. 关键在定位, 必须是 p 的前趋的前趋;

算法框架为:

初始化部分;

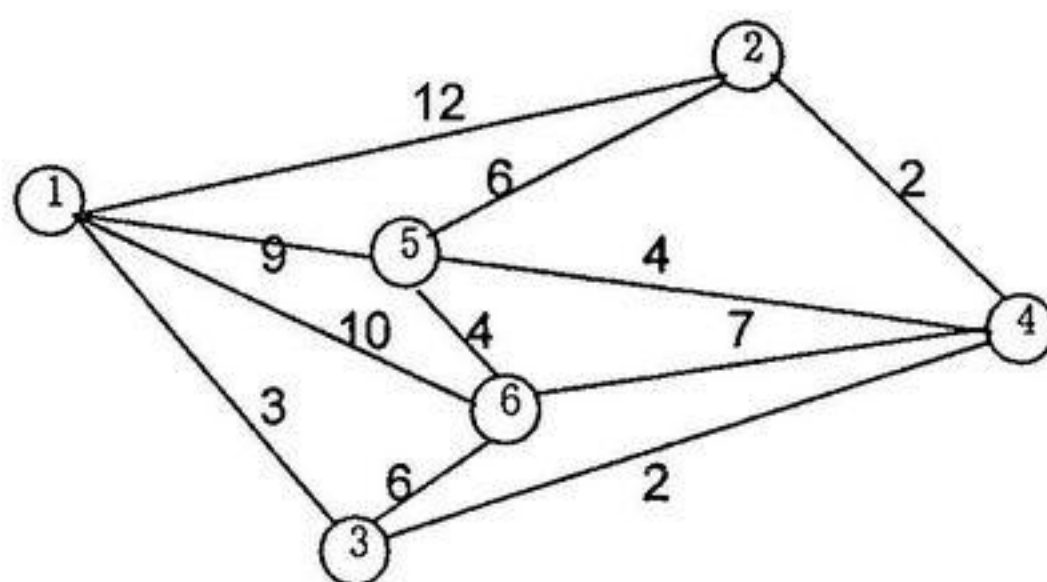
循环控制条件;

循环体语句;
出循环后的处理。

参考答案:

```
DelPrior (LinkList p) {
    q=p; // 初始化: 设置移动指针 q
    while (q->next->next != p) // 循环控制条件, 定位到 p 的前趋的前趋
        q=q->next;           // 移动 q 指针
    dispose (q->next);        // 释放删除结点
    q->next=p;                // 链接好链表, 即原 p 的前趋的前趋成为了 p 现在的 p 前趋
}
```

2. 给定 n 个村庄之间的交通图, 顶点表示村庄, 边表示村庄之间有道路, 边上的权表示道路的长度。现在要从这 n 个村庄中选择一个村庄建一所医院, 在 n 个可选的方案中, 选择其中使得离医院最远的村庄到医院的路程最短的方案建医院。试设计求解该问题的算法, 并应用该算法对下图所示的实例进行解答, 回答: 医院应建的村庄和最远村庄到医院的最短路程。



解答:

求出每对顶点的最短路径, 得到 $n \times n$ 的最短路径邻接矩阵, 即矩阵元素值为顶点的最短路径, 然后将每行中的最大值进行比较, 选取其中最小值所对应的村庄 (行、顶点) 建医院。

该例有:

行 1: 0 7 3 5 9 9

行 2: 7 0 4 2 6 9

行 3: 3 4 0 2 6 6

行 4: 5 2 2 0 4 7

行 5: 9 6 6 4 0 4

行 6: 9 9 6 7 4 0

选取村庄 3 建医院, 最远的是村庄 5 和 6, 但是路径为 6; 而其他的路径都大于 6, 村庄 1、2、5、6 均为 9, 村庄 4 为 7。

还可以优化, 不必求出每对顶点的最短路径。

初始化 Maxmin=极大值; Node=V1;

For i=1 to n {

```
If Vi 没有被处理
{ 以 Vi 为源点求到其他顶点的最短路径（调迪杰斯特拉算法）；
  将 Vi 置为被处理； S 置为空；
  选出 本次求得的最短路径的最大值 Max，及相应的顶点集 S；
  If Max>MaxMin 将 S 集中的顶点置为被处理；
    Else { MaxMin=Max; Node=Vi; }
}
```