

启用前机密 北京大学 2006 年硕士研究生入学考试试题

考试科目: 计算机软件基础

考试时间: 2006 年 1 月 15 日下午

招生专业: 计算机软件与理论、计算机应用 研究方向:

说明: 答题一律写在答题纸上(含填空题、选择题等客观题), 写在此页上无效。

一、填空 (30 分)

1. n 表示线性表中当前元素的数目, P 表示指针的存储单元大小, E 表示数据元素的存储单元大小, D 表示可以在数组中存储的线性表元素的最大数目 ($D \geq n$), 那么使用顺序表所需的空间大小为 _____; 使用双向链表 (不考虑头结点) 所需的空间大小为 _____。
2. 已知一棵树的边集合为 $\{ \langle A, B \rangle, \langle A, C \rangle, \langle B, D \rangle, \langle B, E \rangle, \langle B, F \rangle, \langle C, G \rangle, \langle C, H \rangle, \langle E, I \rangle, \langle E, J \rangle \}$, 则与 E 同层 (到根的路径长度相同) 的其他结点是 _____。
3. 设图 G 的邻接矩阵 (顶点依次是 v_1, v_2, v_3, v_4, v_5 , 矩阵元素 0 表示无连接) 如图示, 从顶点 v_1 出发的深度优先遍历序列是 _____, 从顶点 v_1 出发的广度优先遍历序列是 _____。

$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \end{bmatrix}$

$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 \end{bmatrix}$

$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

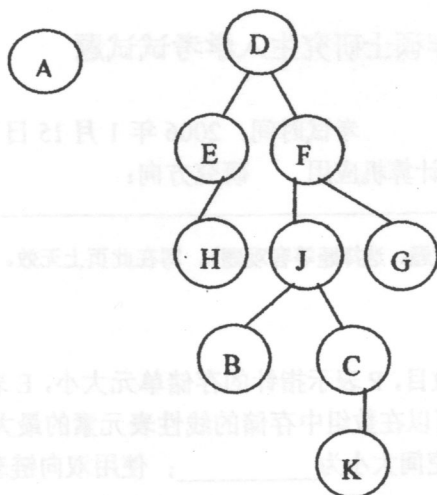
$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 \end{bmatrix}$

4. 对 10000 个记录的文件进行内排序产生了 100 个初始顺串, 进行 5 路归并排序, 共需进行 _____ 趟 (一趟指归并树的一层) 归并完成排序。假设一个记录占一个磁盘块, 以单记录的读或写为访外操作单元 (一次访外), 共计要进行 _____ 次访外操作。
5. 对关键码依次为 $\{8, 32, 46, 49, 52, 68, 70, 75, 88, 90, 92, 95, 100\}$ 的查找表, 采用二分法查找, 查找关键码为 75 的记录时, 经过 _____ 次关键码比较后查找成功, 当查找关键码为 66 的记录时, 经过 _____ 次与查找表中的关键码比较, 可以确认该记录不存在。

二、简答 (15 分)

1. 下图是父结点表示法表示的树, 采用带路径压缩的 Find 算法, 请画出查找 C 之后的树。



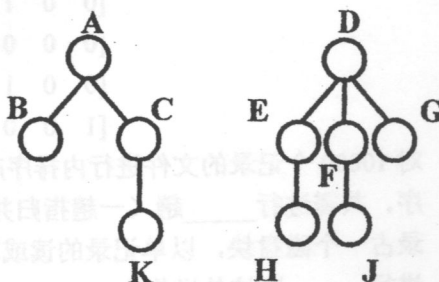
2. (a, b, c, d)任意加入括号可以组成多少个不同的广义表? 说明你的计算过程。
 注: 规定此广义表中的元素不能为空表(), 也不包括多余的括号。例如((a), (b), (c), (d))中 4 对内层括号是多余的, (a, b, c, d, ())中的空表是非法的。

三、算法辨析题 (10 分)

下面是一个对树/森林结构进行处理的一个算法。

1. 对于右图的简单实例, 写出该算法的运行结果。

2. 传入森林的第一棵树的根结点 root, 调用 Print(root), 是否可以打印森林的所有结点? 如果你认为可以, 请简要说明理由。如果你认为不可以, 请在此算法框架的基础上改写出一个可以打印森林的新算法。



Template <class T>

```

void Tree<Elem>::Print(TreeNode<T>* rt) {
    if(rt == NULL)
        return;
    Visit(rt);           // 例如, cout << rt->Value() << "\n";
    for(TreeNode<T>*tmp=rt->LeftMostChild();
        tmp!=NULL; tmp=tmp->RightSibling())
        Print(tmp);
}
  
```

四、算法填空 (15 分)

下面是一个找有向无环图 (DAG) 每对顶点间“最长简单路径”的算法。所谓“最长简单路径”，是指该简单路径包含的边数最多。

void Floyd_LongestPath(Graph& G, Dist ** & D)

```
{
    int i, j, v;           //i, j 是计数器, v 记录相应顶点
    D = new Dist *[G.VerticesNum()];           //为 D 数组申请空间

    for( i = 0; i < G.VerticesNum(); i++)
        D[i] = new Dist[G.VerticesNum()];
    //初始化 D 数组
    for(i=0; i<G.VerticesNum(); i++) {
        for(j=0; j<G.VerticesNum(); j++) {
            if(i == j) {
                D[i][j].length = 0;
                D[i][j].pre = i;
            }
            else {
                D[i][j].length = MinusINFINITY;
                D[i][j].pre = -1;
            }
        }
    }
    //D=adj[0], 将边的值填入 D 数组
    for(v=0; v<G.VerticesNum(); v++)
    {
        for(Edge e=G.FirstEdge(v); G.IsEdge(e); e=G.NextEdge(e))
        {
            _____ //空缺 1
            D[v][G.ToVertex(e)].pre = v;
        }
    }
    for(v=0; v<G.VerticesNum(); v++)
        for(i=0; i<G.VerticesNum(); i++)
            for(j=0; j<G.VerticesNum(); j++)
            {
                if (D[i][v].length == MinusINFINITY
```

```

        || D[v][j].length == MinusINFINITY)
        _____ //空缺 2
    if( D[i][j].length == MinusINFINITY)
    {
        _____ //空缺 3
        D[i][j].pre = D[v][j].pre;
    }
    else if ( _____ ) //空缺 4
    {
        _____ //空缺 5
        D[i][j].pre = D[v][j].pre;
    }
}
}

```

五、算法设计题 (10 分)

设 A, B 是长为 n 的数表, 已经按照非降顺序排好。如果将这 2n 个数全体排序, 处于第 n 个位置的数称为中位数。设计一个最坏情况下 $O(\log n)$ 时间的算法求 A 和 B 的中位数。

1. 描述算法的设计思想;
2. 证明算法的时间复杂性。

六、操作系统问答题 (共 40 分)

1. (10 分) 请描述中断响应过程, 并说明操作系统对该过程提供什么支持。
2. (每小题 8 分, 共 16 分) 关于存储管理:
 - (1) 地址转换过程中快表 (TLB) 的作用、特点和内容。
 - (2) 提出工作集模型是为了解决什么问题? 举例说明该模型对软件编程人员的影响。
3. (14 分) 试设计一个多级目录结构, 要求目录检索速度快。请详细描述你的方案。

七、操作系统应用题 (共 16 分)

1. (10 分) 在一个多道程序系统中, 采用最高响应比优先算法调度作业。现有如下表所示的作业序列, 请列出各个作业的开始时间、完成时间和周转时间。注意: 忽略系统开销。

作业名	进入输入井的时间	估计运行时间
JOB1	8:00	70 分钟
JOB2	8:20	20 分钟
JOB3	8:40	40 分钟
JOB4	8:50	30 分钟
JOB5	9:00	10 分钟

2. (6 分) 假设一个活动头磁盘有 200 道, 编号从 0-199。当前磁头正在 155 道上服务, 并且在此之前完成的是 173 道的访盘请求。现有如下访盘请求序列 (磁道号):

75, 168, 81, 138, 87, 143, 187, 129, 198, 44

试给出采用下列算法后磁头移动的顺序和移动总量 (总磁道数)。

(1) 最短寻道时间优先 (SSTF) 磁盘调度算法。

(2) 扫描法 (SCAN) 磁盘调度算法 (假设沿磁头移动方向不再有访问请求时, 磁头沿相反方向移动。)

八、PV 操作题 (14 分)

一个经典的 IPC 问题发生在牙科诊所。诊所里有三位牙医, 三张诊椅和 10 把供等候就诊病人坐的椅子。如果没有病人, 牙医们就坐在诊椅上聊天或休息。病人到来时, 选择一名牙医为他治疗; 如果牙医们都在看病人, 他就坐下来等候; 如果诊所病人已满, 没有空椅子, 他就离开。请编写牙医和病人的进程, 要求正确实现各进程的同步互斥问题。