

上海大学 2012~2013 学年度春季学期试卷 A

成绩

课程名: 数据结构(二) 课程号: 08305010 学分: 4

应试人声明:

我保证遵守《上海大学学生手册》中的《上海大学考场规则》, 如有考试违纪、作弊行为, 愿意接受《上海大学学生考试违纪、作弊行为界定及处分规定》的纪律处分。

应试人 应试人学号 应试人所在院系

题号(分值)	一(10)	二(15)	三(15)	四(24)	五(26)	六(10)
得分						

一、判断题, 叙述正确的标记 T, 错误的标记 F (每题 1 分, 共 10 分)

得分

- 任意一棵二叉树都可以转换为树来表示 ()
- 折半查找进行时间性能分析的判定树不一定是完全二叉树。 ()
- 散列表的平均查找长度只与采用的散列函数及处理冲突的方法有关。 ()
- 对 B 树删除某一关键字值时, 可能会引起结点的分裂。 ()
- 有 e 条边的无向图, 在邻接表中有 e 个结点。 ()
- 十字链表是有向图的一种存储结构。 ()
- 不同的求最小生成树的方法最后得到的生成树是相同的。 ()
- 若一个有向图的邻接矩阵对角线以下元素均为零, 则该图的拓扑有序序列必定存在。 ()
- 顺序表上的直接选择排序是一种稳定的排序方法。 ()
- 对长度为 n 的表作快速排序, 最坏情况下, 算法时间复杂度为 $O(n^2)$ 。 ()

二、选择题 (每题 1 分, 共 15 分)

得分

- 如果要求一个线性表既能较快的查找, 又能适应动态变化的要求, 则可采用 () 法。
A. 分块查找 B. 顺序查找 C. 折半查找 D. 基于属性的查找
- 在一个无向图中, 所有顶点的度数之和等于所有边数 () 倍。
A. $1/2$ B. 2 C. 1 D. 4
- 用 DFS 遍历一个无环有向图, 并在 DFS 算法退栈返回时打印相应的顶点, 则输出的顶点序列是 ()。
A. 逆拓扑有序 B. 拓扑有序 C. 无序的

注: 教师应使用计算机处理试题的文字、公式、图表等; 学生应使用水笔或圆珠笔答题。答题时不得用铅笔, 否则查卷有误, 阅卷者概不负责。

4. 下列哪一种图的邻接矩阵是对称矩阵? ()
 A. 有向图 B. 无向图 C. AOV 网 D. AOE 网
5. 用邻接矩阵 A 表示图, 判定任意两个顶点 V_i 和 V_j 之间是否有长度为 m 的路径相连, 则只要检查 () 的第 i 行第 j 列的元素是否为零即可。
 A. mA B. A C. A^m D. A^{m-1}
6. 下面哪一方法可以判断出一个有向图是否有环 (回路)
 A. 深度优先遍历 B. 拓扑排序 C. 求最短路径 D. 求关键路径
7. 在图采用邻接表存储时, 求最小生成树的 Prim 算法的时间复杂度为 ()。
 A. $O(n)$ B. $O(n+e)$ C. $O(n^2)$ D. $O(n^3)$
8. 下列关于 AOE 网的叙述中, 不正确的是 ()。
 A. 关键活动不按期完成就会影响整个工程的完成时间
 B. 任何一个关键活动提前完成, 那么整个工程将会提前完成
 C. 所有的关键活动提前完成, 那么整个工程将会提前完成
 D. 某些关键活动提前完成, 那么整个工程将会提前完成
9. 二叉查找树在 () 时其查找效率最低。
 A. 结点太多 B. 完全二叉树 C. 呈单枝树 D. 结点太复杂
10. 设有一个用线性探测法解决冲突得到的散列表:

0	1	2	3	4	5	6	7	8	9	10
		13	25	80	16	17	6	14		

散列函数为 $H(k)=k \bmod 11$, 若要查找元素 14, 探测的次数是 ()。

- A. 18 B. 9 C. 3 D. 6
11. 下列排序方法中, () 是稳定的排序方法
 A. 直接选择排序 B. 折半插入排序
 C. 希尔排序 D. 快速排序
12. 下述排序方法中, 比较次数与待排序记录的初始状态无关的是 ()。
 A. 插入排序和快速排序 B. 归并排序和快速排序
 C. 选择排序和基数排序 D. 插入排序和归并排序
13. 设有 5000 个元素, 希望用最快的速度挑选出前 10 个最大的, 下列排序方法中采用 () 方法最好。
 A. 快速排序 B. 堆排序 C. 希尔排序 D. 归并排序
14. 并查集的结构是 ()
 A. 二叉链表存储的二叉树 B. 双亲表示法存储的树
 C. 三叉链表存储的二叉树 D. 线性链表
15. 下列哪一个关键码序列不符合堆的定义? ()
 A. A, C, D, G, H, M, P, Q, R, X B. A, C, M, D, H, P, X, G, Q, R
 C. A, D, P, R, C, Q, X, M, H, G D. A, D, C, M, P, G, H, X, R, Q

三、填空题 (每空1分, 共15分)

得分

1. G 是一个非连通无向图, 共有 28 条边, 则该图至少有_____个顶点。
2. 已知一无向图 $G=(V, E)$, 其中 $V=\{a, b, c, d, e\}$, $E=\{(a, b), (a, d), (a, c), (d, c), (b, e)\}$ 现用某一种图遍历方法从顶点 a 开始遍历图, 得到的序列为 $abecd$, 则采用的是_____遍历方法。
3. 求图的最小生成树有两种算法, 其中_____算法适合于求稀疏图的最小生成树。
4. 求从某源点到其余各顶点的 Dijkstra 算法在图的顶点数为 10, 用邻接矩阵表示图时计算时间约为 10ms, 则在图的顶点数为 40, 计算时间约为_____ms。
5. 设有向图有 n 个顶点和 e 条边, 进行拓扑排序时, 总的计算时间复杂度为_____。
6. 设线性表 $(a_1, a_2, \dots, a_{500})$ 元素的值由小到大排列。对一个给定的 k 值, 在等概率情况下, 用顺序查找法查找一个记录, 查找成功的平均查找长度 ASL 为_____; 用二分法检索查找表中与 k 相等的元素, 在查找不成功的情况下至多比较_____次。用分块查找 (索引表和各块内均用顺序查找), 若分成 25 块, 查找成功的其平均查找长度为_____。
7. 在顺序表 $(8, 11, 15, 19, 25, 26, 30, 33, 42, 48, 50)$ 中, 用折半法查找关键码值 8, 需做的关键码比较次数为_____, 查找关键码值 20, 需做的关键码比较次数为_____。
8. 对于一个高度为 h (空树的高度为 -1) 的 AVL 树, 其最少结点数是_____。反之, 对于一个有 n 个结点的 AVL 树, 其最大高度是_____, 最小高度是_____。
9. 设用希尔排序对数组 $\{98, 36, 19, 5, 47, 23, 1, 8, 10, 7\}$ 进行排序, 给出的步长 (也称增量序列) 依次是 5、3、1, 则写出第一趟结束后, 数组中数据的排列第三个元素是 (从 0 开始计数) _____。
10. 对一组记录 $(54, 38, 106, 21, 15, 72, 60, 45, 83)$ 进行直接插入排序, 当把元素 60 插入到有序表时, 为寻找插入位置需比较_____次。

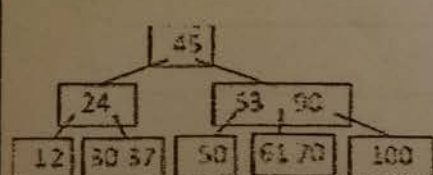
四、简答题 (共 24 分)

得分

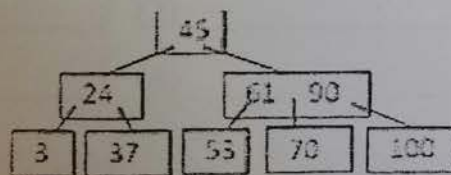
1. (8分) 已知 2 棵 2-3 树 (3 阶 B-树) 如下:

(1) 对树 (a), 请分别画出先后插入 26, 85 两个新结点后的树形;

(2) 对树 (b), 请分别画出先后删除 53, 37 两个结点后的树形。



(a)



(b)

注: 教师应使用计算机处理试题的文字、公式、图表等; 学生应使用水笔或圆珠笔答题。答题时不得用铅笔, 否则查卷有误, 阅卷者概不负责。

【解答】:

(1)

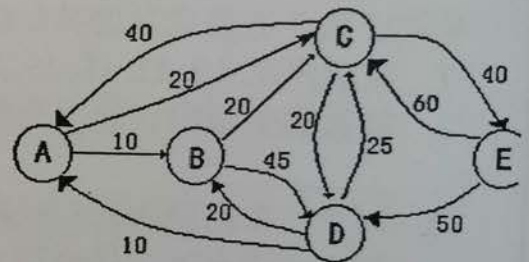
(2)

2. (8 分) 给定 5 个村庄(A、B、C、D、E)之间的交通图如下所示, 若村庄 i 到 j 有道路, 则将顶点 i 到 j 用有向边连接, 边上的 W_{ij} 表示这条道路的长度。现在请回答以下问题:

(1) 画出该有向图的邻接表存储结构

(2) 求其它各村庄到村庄 B 的最短路径和最短路径长度。

(3) 要从这 5 个村庄中选择一个村庄建一所医院, 问这所医院应建在哪个村庄, 才能各村到医院的来回路程最短? 说明解答上述问题的算法。



【解答】:

(1)

(2)

(3)

3. (8分) 对初始序列(58, 85, 47, 39, 70, 47*, 101, 68, 10, 66, 34)按递增方式进行排序。

(1) 给出快速排序的第一趟排序结果(以第1个元素58为基准元素)。

(2) 选取 $d = \{5, 3, 1\}$, 给出希尔排序的第一趟排序结果。

(3) 写出二路归并的第一趟排序结果。

(4) 给出基数排序的第一趟的回收结果。

【解答】: (1) 快速排序的第一趟排序结果为:

(2) 希尔排序的第一趟排序结果为:

(3) 二路归并的第一趟排序结果为:

(4) 基数排序的第一趟回收结果是:

五、算法分析题(每空2分, 共26分)

得分	
----	--

1. (8分) 下列递归算法的功能是: 从大到小输出给定二叉排序树中所有关键字不小于 x 的数据元素。该算法的时间复杂度为 $O(\log_2 n + m)$, 其中 n 为二叉排序树中的结点数, m 为输出的数据元素个数。请完善该算法。

提示: 算法可以借助逆中序遍历二叉排序树来实现。所谓逆中序遍历二叉树是指: 如果当前结点 p 非空, 则先逆中序遍历 p 的右二叉树; 然后访问 p 结点; 最后再逆中序遍历 p 的左二叉树。在本算法中访问 p 结点时, 如果 p 的值小于 x , 则算法结束, 否则输出 p 的值。

```
void PrintNLT (BSTreeNode<Type>* p, const Type x) {
    if (p) {
        (1) _____;
        if (p->data < x) //当遇到小于x的元素时立即结束运行
            (2) _____; ;
        cout << (3) _____ << endl;
        (4) _____;
    }
}
```

}//PrintNLT

(1) _____ (2) _____

(3) _____ (4) _____

注: 教师应使用计算机处理试题的文字、公式、图表等; 学生应使用水笔或圆珠笔答题。答题时不得用铅笔, 否则查卷有误, 阅卷者概不负责。

2. (10 分) 在有向图的邻接表存储结构中, 为每个顶点 v 增加一个 MPL 域, 其定义为图中所有顶点到达顶点 v 的最长路径长度(路径上的边数)。下面算法完成有向无环图 G 中每个顶点的 MPL 值, 如果 g 无回路, 则求出各顶点的 MPL, 并返回 SUCCESS; 否则返回 FAIL。

```
template <class ElemType>
Status FillMPL(const AdjListDirGraph<ElemType> &g) // 初始条件: 存在有向图 g
// 操作结果: 如 g 无回路, 则求出 g 中每个顶点的 MPL, 并返回 SUCCESS, 否则返回 FAIL
{
    int *indegree = new int[g.GetVexNum()]; // 顶点入度数组
    int v, u, mpl, count = 0, top = -1;
    for (v = 0; v < g.GetVexNum(); v++) {
        indegree[v] = 0; // 初始化顶点 v 的入度为 0
        g.SetMPL(v, 0); // 初始化 v 顶点的 MPL 为 0
    }
    for (v = 0; v < g.GetVexNum(); v++) // 统计图 g 各顶点的入度
        for (u = g.FirstAdjVex(v); u != -1; u = g.NextAdjVex(v, u))
            indegree[u]++;
    for (v = 0; v < g.GetVexNum(); v++)
        if (indegree[v] == 0) { // 入度为 0 的顶点入栈
            (1) _____; top = v; }
        while ( (2) _____ ) { // 栈非空
            v = top; (3) _____; mpl = g.GetMPL(v) + 1;
            count++; // 已求出 MPL 的顶点数累加
            for (u = g.FirstAdjVex(v); u != -1; u = g.NextAdjVex(v, u)) {
                // 对 v 的每个后继顶点 u 入度减 1, 并修改其 MPL
                if (mpl > g.GetMPL(u)) (4) _____;
                if (--indegree[u] == 0) { // u 入度为 0, 将 u 入栈
                    indegree[u] = top; top = u; }
            } // end for
        } // end while
    delete []indegree; // 释放 indegree 所占用的存储空间
    if ( (5) _____ ) return FAIL; // 图 g 有回路
    else return SUCCESS; // 求 MPL 成功
}
```

(1) _____ (2) _____ (3) _____

(4) _____ (5) _____

3. (8 分) 折半插入排序的算法基本思想是: 设在排序表中有 n 个数据元素 $Arr[0], Arr[1], \dots, Arr[n-1]$ 。其中, $Arr[0], Arr[1], \dots, Arr[i-1]$ 是已经排好序的部分数据元素

列。在插入 $Arr[i]$ 时, 利用折半查找方法寻找 $Arr[i]$ 的插入位置。在下面算法的处, 填上适当语句, 实现上面的算法。

【注】: 关键字用成员函数 $getKey()$ 获取。

```
template <class Type>
void BinaryInsertSort ( sortlist<Type> & table )
{
    element<Type> temp;
    int low , high, mid ;
    for ( int i = 1; i < table.CurrentSize; i++)
    {
        low = 0;  high = i-1; temp = table.Arr[i];
        while ( low <= high )
        {
            (1) _____;
            if ((2) _____)
                high = mid - 1;
            else low = mid + 1;
        }
        for (int k = i-1; k >= low; k--)
            (3) _____;
        (4) _____;
    }
}
```

(1) _____ (2) _____
(3) _____ (4) _____

得分	
----	--

六、算法设计题(10 分)

在有向图中顶点的度等于其入度与出度之和, 现定义有向图的度为其所有顶点度的最大值。试编写算法 $CountDegree(g)$, 在有向图的邻接表存储结构上求有向图 g 的度。

下面是有向图的邻接表存储结构类模板的部分定义:

```
template <class ElemType>
class AdjListDirGraph
{
protected:
```

// 邻接表的数据成员:

```
int vexNum, vexMaxNum, arcNum;
```

// 顶点数目、允许的顶点最大数目和边数

注: 教师应使用计算机处理试题的文字、公式、图表等; 学生应使用水笔或圆珠笔答题。答题时不得用铅笔, 否则查卷有误, 阅卷者概不负责。

```

AdjListGraphVex<ElemType> *vexTable;    // 顶点表
public:
// 抽象数据类型方法声明及重载编译系统默认方法声明:
AdjListDirGraph(ElemType es[], int vertexNum, int vertexMaxNum = DEFAULT_SIZE);
    // 构造函数
AdjListDirGraph(int vertexMaxNum = DEFAULT_SIZE);
    //构造函数
~AdjListDirGraph();                // 析构函数
.....
int GetVexNum() const;              // 求有向网的顶点个数
int GetArcNum() const;              // 求有向网的边数个数
int FirstAdjVex(int v) const;        // 求有向网中顶点 v 的第一个邻接点序号
int NextAdjVex(int v1, int v2);      // 求顶点 v1 的相对于 v2 的下一个邻接点序号
.....
};

```

编写的算法:

```

template <class ElemType> int CountDegree(const AdjListDirGraph<ElemType> &g)
// 返回有向图 g 的度数

```