上 海 交 通 大 学 试 卷（偶卷）

（ 2022 至 2023 学年 第 1 学期 ）

班级号_____ 学号_____ 姓名 _____

课程名称 数据结构 成绩 _____

一、 选择题（每题 1 分，共 15 分，请将答案填在以下表格里）

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

1. 如果一棵二叉树有 85 个结点，则这棵二叉树的高度最少是？
   A. 5 B. 6 C. 7 D. 8

2. 如果一个二叉树的前序遍历为 ABDCEFGH，中序遍历为 BDAFEGCH，则其后序遍历为?
   A.BDFGEHCA B. DBFGEHAC C. BDFGAEHC D. DBFGEHCA

3. 通过调整数组而非逐一插入，创建一个堆的时间复杂度是?
   A. O(N) B. O(logN) C. O(NlogN) D. $O(N^2)$

4. 已知关键字序列{5，8，12，19，28，20，15，22}是最小堆（小根堆、小顶堆），插入关键字 3，调整后得到的最小堆是?
   A. 3, 5, 12, 8, 28, 20, 15, 22, 19 B. 3, 5, 12, 19, 20, 15, 22, 8, 28
   C. 3, 8, 12, 5, 20, 15, 22, 28, 19 D. 3, 12, 5, 8, 28, 20, 15, 22, 19

5. 在下列方法中，平均查找时间与结点个数 n 无关的是?
   A．顺序查找 B．折半查找 C．分块查找 D．哈希查找

6. 某线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用哪种存储方式最节省运算时间和空间？
   A．单链表 B．仅有头指针的单循环链表 C．双链表 D．仅有尾指针的单循环链表

7. 设用一维数组 a[n]实现顺序栈，以 a[n-1]为栈底，用整型变量 t 指示当前栈顶位置，a[t]为栈顶元素。当从栈中成功弹出一个元素时，变量 t 的变化是？
   A. t = t-1 B. t = t+1 C. t 保持不变 D. t = n

8. 利用栈，求后缀表达式 ab+cd*+的值，问操作数栈至少有几个存储单元，以避免发生上溢？
   A. 2 B. 3 C. 4 D.以上都不对

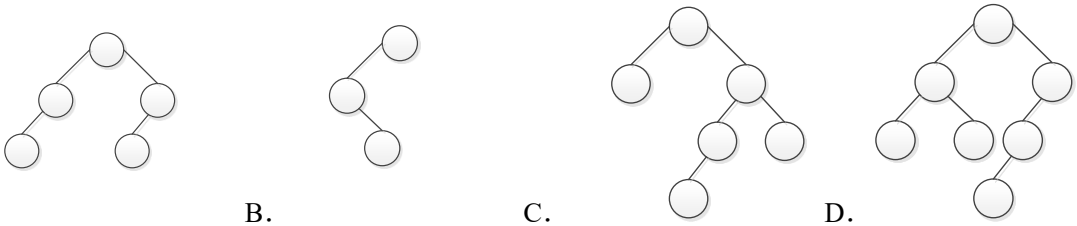
9. 若用数组 A[0…5]来实现循环队列，且当前 rear 和 front 的值分别为 1 和 5，当从队列中出队一个元素，再进队两个元素后，rear 和 front 的值分别为？
   A. 3,4 B. 5,1 C. 5,0 D. 3,0

我承诺，我将严格遵守考试纪律。

承诺人：_____

题号	一	二	三	四
得分				
批阅人(流水阅卷教师签名处)				

10. 对下列二叉查找树，AVL 树是?



A. B. C. D.

11. 若外部存储上有 32400 个记录，做 3 路平衡归并排序，计算机内存工作区能容纳 400 个记录，则排序好所有记录，需要作多少趟归并排序？

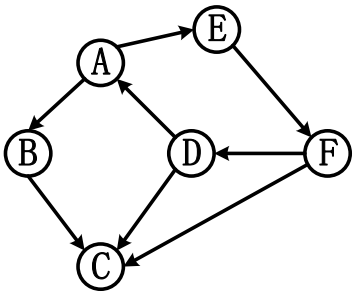
A.2 B.4 C. 5 D. 3

12. 在有 8 个顶点的无向图中，至少需要多少条边它才可能是连通的？

A. 3 B. 6 C. 7 D. 9

13. 在如下所示的有向图中，下列哪项不是该图的广度优先搜索结果？

A. ABECFD B. BCAEFD C. CBAEFD D. AEFDCB



14. 下列内部排序算法中，最坏情况下时间复杂度为 O(n logn) 是？

A．冒泡排序 B. 插入排序 C. 堆排序 D. 快速排序

15. 已知序列{515，99，524，73，920，182，909，287，665，474}，采用 2 路归并排序法对该序列做升序排序时需要几趟排序？

A. 2 B. 3 C. 4 D. 5

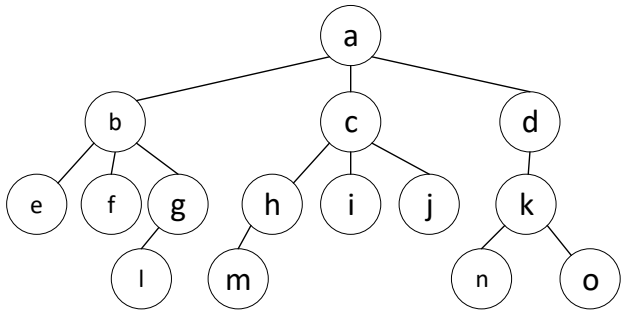
二、 简答题（每题 5 分，共 40 分）

1. 假设需要编码的字符及其出现的频度为：
   a：16，b：30，c：1，d：9，e：8，f：12，g：14，h：10。
   请问在按照哈夫曼算法构造的哈夫曼树中，f 在第几层（假设根为第一层）？如果哈夫曼树中左分支为 1，右分支为 0，那么 b 的编码长度是多少？

2. {28, 15, 42, 18, 22, 5, 40}逐个按顺序插入到初始为空的最小化堆中，则该二叉树的前序遍历结果为：

3. 设入栈序列为 A、B、C，且在入栈的过程中允许出栈，试写出所有可能得到的出栈序列，并写出每种序列对应的操作。例如：A 入 A 出,B 入 B 出,C 入 C 出， 即 ABC。

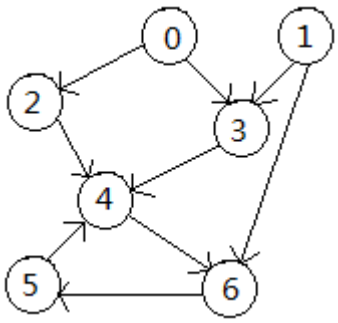
4. 利用孩子兄弟链的方法将下图中的树转换成二叉树的形式。



5. 若哈希表的表长为 16，采用除留取余法确定哈希函数，并用线性探测法处理冲突。
   对于关键字序列：（10，24，58，17，31，30，47，40，63，13，49，6），请给出哈希函数和哈希表的存储示意图。

6. 试分析简单插入排序的最优情况与最差情况的时间复杂度，并判断该排序算法的稳定性。

7. 试问，以下有向图是否存在拓扑序列？如果存在，给出一个拓扑序列； 如果不存在，分析其原因。



8. 如果一个外存储器上的结构型数据文件，使用一棵高度为 3（即不算空叶子层的索引结点层数为 3）的 5 阶 B 树作为数据记录关键字索引，该数据文件中最少允许多少个记录？写出分析过程。

三、 程序填空（每空 2 分，共 20 分）

1. 下面的程序用于删除二叉树以 t 为根节点一个子树，请补充完整其中确实的代码。

```
template<class T>
void binaryTree<T>::clear(binaryTree<T>::Node  _____ )
```

```
{
    if (_____)    return;
    clear(t->left);
    clear(t->right);
    _____
    t = NULL;
}
```

2. 以下程序段为链式栈弹出栈顶元素的算法实现，请在空白处补充语句。

```
template <class elemType>
elemType linkStack<elemType>::pop()
{   if(_____) throw outOfBound();
    node *tmp = top_p;
    elemType x = tmp->data;
    _____
    delete tmp;
    _____
 }
```

3. 以下为在采用邻接表存储结构的图中进行非递归深度优先搜索的算法实现，请在空白处补充语句。

```
template <class TypeOfVer, class TypeOfEdge>
void adjListGraph<TypeOfVer, TypeOfEdge>::DepthFirstSearch() const
{
   int i;
   linkStack<edgeNode*> stkNode;
   edgeNode* p;
   bool* visited = new bool[Vers];

   for (i = 0; i < Vers; i++)  _____

   for (i = 0; i < Vers; i++) {
      if (visited[i]) continue;
      std::cout << verList[i].ver << " ";
      visited[i] = true;
      stkNode.push(verList[i].head);
      do {
         _____
         while (p) {
           if (!visited[p->end]) break;
           p = p->next;
         }
         if (p) {
           _____
           std::cout << verList[p->end].ver << " ";
           visited[p->end] = true;
           stkNode.push(verList[p->end].head);
```

```
        }
      } while (!stkNode.isEmpty());
   }
```

```
   std::cout << endl;
}
```

四、 编程题（共 25 分）

1. 编写函数 bool judge(node *t) const，判断以 t 为根的二叉树是否是 AVL 树(假设结点中含有高度 height 字段) (5 分)。

```
int height(node *t) const
{ if (!t) return 0; return t->height; }
```

2. 请写出二叉树层次遍历的实现代码 void binaryTree<T>::levelOrder() const，要求遍历按照从上层到下层，每一层从右到左的方式访问。（10 分）

3. 一个无向图（不带权值）的邻接表类定义如下：

```
template <class TypeOfVer, class TypeOfEdge>
struct edgeNode {
      int end;
      edgeNode* next;
      edgeNode(int e, edgeNode* n = NULL) : end(e), next(n) {}
   };

   template <class TypeOfVer, class TypeOfEdge>
   struct verNode {
      TypeOfVer ver;
      edgeNode* head;
      verNode(edgeNode *h = NULL) : head(h) {}
   };

template <class TypeOfVer, class TypeOfEdge>
class adjListGraph
{
private:
   verNode* verList; //图中结点组成的动态数组
   int Vers, Edges; //图中结点数及边数
   int find(TypeOfVer v) const; //查找 v 在图中的内部序号
public:
   void allVetexes(TypeOfVer v) const;
};
```

请编码实现函数 void allAdjVetexes(TypeOfVer v) const，要求利用非递归算法，输出和顶点 v 没有路径相连的所有顶点的值及个数。（10 分）