一、单项选择题（每小题2分，共30分）

1. **设栈S和队列Q的初始状态为空，元素e1、e2、e3、e4、e5和e6依次进入栈S ,一个元素出栈后即进入Q，若6个元素出队的序列是e2、e4、e3、e6、e5和e1，则栈S的容量至少是（ ）个。**

**A．3 B．4 C．5 D．6**

1. **银行业务叫号系统采用了（ ）数据结构。**

**A．栈 B．广义表 C．图 D．队列**

1. **按照二叉树的定义，具有3个结点的不同形状的二叉树有（ ）种。**

**A．3 B．4 C．5 D．6**

1. **在数据结构中，从逻辑上可以把数据结构分为（ ） 。**

**A．动态结构和静态结构　　　 B．线性结构和非线性结构**

**C．紧凑结构和非紧凑结构　　 D．内部结构和外部结构**

1. **非空的循环单链表head的尾结点（由p所指向）满足（ ）。**

**A．p->next == NULL B．p == NULL**

**C．p->next == head D．p == head**

1. **栈和队列的共同点是（ ）。**

**A．都是先进后出　　　　　　　　　　　 B．都是先进先出**

**C．只允许在端点处插入和删除元素　　　 D．没有共同点**

1. **一个队列的入队序列是1，2，3，4，则队列的输出序列是（ ）。**

**A．4，3，2，1　　　 B．1，2，3，4**

**C．1，4，3，2　　　 D．3，2，4，1**

1. **串的长度是指（ ）。**

**A．串中所含字符的个数　　　 B．串中所含不同字母的个数**

**C．串中所含不同字符的个数　　　 D．串中所含非空格字符的个数**

1. **具有10个叶子结点的二叉树中有（ ）个度为2的结点。**

**A．8　　 B．9　　 C．10　 D．11**

1. **某二叉树结点的中序序列为ABCDEFG，后序序列为BDCAFGE，则其左子树中结点数目为（ ）**

**A．5　　　　　　B．2　　　　　 C．3　　　　　　D．4**

1. **设森林F对应的二叉树B有m个结点，B的右子树结点个数为n，森林F中第一棵树的结点个数是（ ）**

**A．m-n B．m-n-1 C．n+1 D．m+n**

1. **在一个无向图中，所有顶点的度数之和等于所有边数的（ ）倍。**

**A．1/2 B． 1 C．2 D．4**

1. **堆是一种有用的数据结构。下列关键码序列（ ）是一个堆。**

**A．91,31,56,23,15,68 B．91,56,31,68,15,23**

**C．15,56,23,91,31,68 D．15,31,23,91,56,68**

1. **以下内部排序方法中，平均时间复杂度为O(nlogn)，最坏情况下时间复杂度为O(n2)的是（ ）。**

**A．快速排序 B．堆排序 C．直接选择排序 D．插入排序**

1. **对一组数据（84，47，25，15，21）排序，数据的排列次序在排序过程中变化如下：（1）84 47 25 15 21 （2）15 47 25 84 21 （3）15 21 25 84 47（4）15 21 25 47 84则采用的排序方法是（ ）**

**A．选择排序 B．起泡排序 C．快速排序 D．插入排序**

**二、判断题（正确的划“√”，错误的划“×”，每小题1分，共15分）**

1. **算法独立于具体的程序设计语言，与具体的计算机无关。（ ）**
2. **线性表采用链式存储时，结点内部的存储空间可以是不连续的。（ ）**
3. **栈和队列的存储方式，既可以是顺序方式，又可以是链式方式。（ ）**
4. **哈夫曼树的结点总个数一定是偶数。（ ）**
5. **已知二叉树的先序遍历序列和中序遍历序列，可以画出这棵二叉树（ ）。**
6. **有e条边的无向图，在其对应的邻接表中有e个结点。（ ）**
7. **连通分量指的是无向图的极大连通子图。（ ）**
8. **起泡排序算法在最好情况下的时间复杂度为O（n）。（ ）**
9. **在哈希表的查找过程中的“比较”操作是无法避免的。（ ）**
10. **不论是入队列操作还是入栈操作，在顺序存储结构上都需要考虑“溢出”情况。（ ）**
11. **完全二叉树不一定是平衡二叉树。（ ）**
12. **双向链表可随机访问任一结点。（ ）**
13. **空串和空白串是相同的。（ ）**
14. **抽象数据类型（ADT）包括定义和实现两方面，其中定义是独立于实现的，定义仅给出一个ADT的逻辑特性，不必考虑如何在计算机中实现。（ ）**
15. **堆是完全二叉树，完全二叉树不一定是堆。（ ）**

**三、填空题（每空1分，共15分）**

1. **n个记录的折半查找，若查找失败，进行了 （1） 次比较。**
2. **在单链表中，指针p所指结点有后继的条件是 （2） 。（结点构成：data和next）**
3. **栈的特点是 （3） 。**
4. **判断循环队列是否队满的条件表达式是 （4） 。（Q.rear和Q.front的关系）**
5. **完全二叉树中的结点个数为n，则编号最大的分支结点的编号为 （5） 。**
6. **如果二叉树中有10个叶结点，12个度为1的结点，则该二叉树的总结点数为 （6） 。**
7. **如果A有7个兄弟，而B是A的双亲，则B的度是 （7） 。**
8. **若无向图中有n个顶点，则其完全无向图恰有 （8） 条边。**
9. **如果具有n个顶点的图是一个环，则它有 （9） 棵生成树。**
10. **普里姆（Prim）算法的时间复杂度是 （10） ，与图中的边数无关，它适合求 （11） 图的最小生成树。**
11. **顺序查找有n个元素的线性表，若查找成功时的平均查找长度为 （12） 。**
12. **高度为5的二叉树，其结点最少有 （13） 个，最多有 （14） 个。**
13. **（15） 操作是是二叉树各种操作的基础。**

**四、算法填空（每空2分，共10分）**

1. **在顺序表L的第 i 个元素之前插入新的元素e**

**Status ListInsert(SqList &L, int i, ElemType e) {**

**if (i < 1 || i >** **L.length+1) return ERROR; // 插入位置不合法**

**for ( j= L.length ; j>i ; j-- )**

**(1) ; // 将第i个及后面的元素后移**

**L.elem[i-1] = e ; // 插入e**

**(2) ; // 表的长度增加1**

**return OK;**

**}**

1. **下面程序段的功能是实现一趟快速排序，请在下划线处填上正确的语句。**

**int Partition(Sqlist &L, int low, int high){**

**//以L.r[low]为主记录，对子系列L.r[low...high]的一趟划分**

**temp= L.r[low];**

**while (low<high){ //进行一趟划分**

**while (low<high &&L.r[high].key>=temp.key) --high;**

**L.r[low]=L.r[high];**

**while (low<high &&L.r[low].key<=temp.key) ++low;**

**L.r[high]=L.r[low];**

**}**

**(3) ; //找到主记录的位置low**

**return low;**

**}**

**void Qsort(Sqlist &L, int low, int high) { //递归算法实现**

**if ( (4) ) { //长度大于1**

**loc=Partition(L, low, high); //将L.r[low...high]一分为二**

**Qsort(L, low, loc-1); //对低子表递归排序**

**(5) ; //对高子表递归排序**

**}**

**}**

**void QuickSort(SqList L) { //对顺序表L做快速排序**

**Qsort(L, 1, L.length);**

**}**

**五、综合题（每小题6分，共30分）**

1. **给定叶结点(a,b,c,d,e,f,g)，权值分别为{24,12,17,10,27,2,8}，画出对应的哈夫曼树，并写出各叶结点的哈夫曼编码。（6分）**
2. **已知一棵二叉树的后序遍历序列为EICBGAHDF，中序遍历序列为CEIFGBADH，请画出这棵二叉树，写出其前序，并把这棵二叉树转换成相应的树（或森林）。（6分）**
3. **假设一个有向图的顶点集为{1,2,3,4,5,6}，其邻接矩阵如图1所示。（6分）**

**（1）请根据邻接矩阵画出该图（图中顶点的排列位置如 图2）；**

**（2）分别求每一个顶点的入度和出度；**

图1

**（3）请写出一个从顶点3出发进行广度优先搜索的遍历序列。**

1

2

4

3

5

6

图2

1. **对于非连通图，每个连通分量中的顶点和遍历时走过的边一起构成若干棵生成树，这些连通分量的生成树组成非连通图的生成森林。画出图3从顶点A出发，按深度优先遍历得到的生成森林。（6分）**

**A**

**B**

**C**

**D**

**F**

**L**

**M**

**E**

**G**

**H**

**K**

**I**

**J**

图3

1. **一个带权网络如图4所示。从顶点1开始，用克鲁斯卡尔(Kruskal)算法求出该网的最小生成树。要求在答题卷上画出最小生成树的产生过程，并用< >括起来的数字标号反映最小生成树中各条边的求取次序。（6分）**

**5**

**1**

**4**

**2**

**3**

**5**

**6**

**1**

**5**

**4**

**3**

**4**

**8**

**4**

**2**

**3**

图4

**一、单项选择题（每小题2分，共30分）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1~5** | **ADCBC** | **6~10** | **CBABD** | **11~15** | **ACDAA** |

**二、判断题（正确的划“√”，错误的划“×”，每小题1分，共15分）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1~5** | **√√√×√** | **6~10** | **×√√√√** | **11~15** | **√××√√** |

**三、填空题（每空1分，共15分）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | [log2n]+1 | **2** | p->next!=NULL | **3** | FILO或LIFO |
| **4** | Q.rear==Q.front | **5** |  | **6** | **31** |
| **7** | **8** | **8** | n(n-1)/2 | **9** | **n** |
| **10** | O(n2) | **11** | 稠密图 | **12** | n/2 |
| **13** | **5** | **14** | **31** | **15** | 遍历 |

**四、算法填空（每空2分，共10分）**

**（1）L.elem[j] = L.elem[j-1] （2）L.length++或L.length= L.length+1**

**（3）L.r[low]=temp （4）low<high**

**（5）Qsort(L, loc+1, high)**

**五、综合题（每小题6分，共30分）**

1. 哈夫曼树为（2.5分）

2

10

8

10

12

17

20

27

24

29

44

56

100

0

1

1

1

1

1

1

0

0

0

0

0

f

g

d

b

c

e

a

哈夫曼编码（3.5分，每个编码0.5分）a:01 b:110 c:111 d:001 e:10 f:0000 g:0001

哈夫曼树和哈夫曼编码都不唯一，只要满足条件都可以。

1. 已知一棵二叉树的后序遍历序列为EICBGAHDF，中序遍历序列为CEIFGBADH，请画出这棵二叉树，写出其前序，并把这棵二叉树转换成相应的树（或森林）。（6分）

这颗二叉树的形态为：（2分） 二叉树转换成的森林为：（2分）

**A**

**C**

**D**

**F**

**H**

**I**

**E**

**G**

**B**

**A**

**C**

**D**

**F**

**H**

**I**

**E**

**G**

**B**

其前序遍历序列为：FCIEDAGBH（2分）

3.(1)根据邻接矩阵，该有向图结构如下：（2分）

1

2

4

3

5

6

（2）有向图中每一个顶点的度为该顶点的入度和出度之和，即与该顶点相关联的边的条数，用D（i）表示顶点i的度，则结果如下：（2分）

D（1）入度为1，出度为3，D（2）入席为1，出度为1，D（3）入度为0，出度为2，D（4）入度为2，出度为1，D（5）入度为3，出度为0，D（6）入度为1，出度为1。

（3）从顶点3出发进行广度优先搜索的遍历序列：3,1,4,5,2,6。（2分）

4. 按深度优先遍历得到的生成森林如下：（每颗树2分）

G

K

I

H

D

E

A

L

C

F

M

J

B

5. 用克鲁斯卡尔(Kruskal)算法求出该网的最小生成树的过程如下：

（因为有多个边权值相同，因此本题答案不唯一。只要满足克鲁斯卡尔算法的生成树都是可以的）最小生成树各边的求取的可能顺序为：<1,2>，<5,6>，<3,4>，<2,3>，<2,5>。（评分标准：按克鲁斯卡尔算法产生最小生成枝的各条边，对相同值可取任一符合要求的边，每条边1分，写出对应的求取顺序1分。）

