1. **一、                   单选题（每题 2 分，共20分）**
   1. 1.     栈和队列的共同特点是( )。

A.只允许在端点处插入和删除元素

B.都是先进后出

C.都是先进先出

D.没有共同点

* 1. 2.     用链接方式存储的队列，在进行插入运算时( ).

A. 仅修改头指针 　 B. 头、尾指针都要修改

C. 仅修改尾指针 D.头、尾指针可能都要修改

* 1. 3.     以下数据结构中哪一个是非线性结构？( )

A. 队列 　　 B. 栈 C. 线性表 　　 D. 二叉树

* 1. 4.     设有一个二维数组*A*[*m*][*n*]，假设*A*[0][0]存放位置在644(10)，*A*[2][2]存放位置在676(10)，每个元素占一个空间，问*A*[3][3](10)存放在什么位置？脚注(10)表示用10进制表示。

A．688 B．678 C．692 D．696

* 1. 5.     树最适合用来表示( )。

A.有序数据元素 B.无序数据元素

C.元素之间具有分支层次关系的数据 D.元素之间无联系的数据

* 1. 6.     二叉树的第k层的结点数最多为( ).

A．2k-1 B.2K+1 C.2K-1 　　　D. 2k-1

* 1. 7.     若有18个元素的有序表存放在一维数组A[19]中，第一个元素放A[1]中，现进行二分查找，则查找A［3］的比较序列的下标依次为( )

A. 1，2，3 B. 9，5，2，3

C. 9，5，3 D. 9，4，2，3

* 1. 8.     对n个记录的文件进行快速排序，所需要的辅助存储空间大致为

A. O（1） 　　B. O（n）　　 C. O（1og2n） D. O（n2）

* 1. 9.     对于线性表（7，34，55，25，64，46，20，10）进行散列存储时，若选用H（K）=K %9作为散列函数，则散列地址为1的元素有（ ）个，

A．1 B．2 C．3 D．4

* 1. 10. 设有6个结点的无向图，该图至少应有( )条边才能确保是一个连通图。

A.5 B.6 C.7 D.8

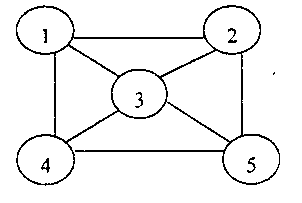
1. **二、                   填空题（每空1分，共26分）**
2. 1.        通常从四个方面评价算法的质量：\_\_\_\_\_\_正确性\_\_\_、\_\_\_\_\_\_\_可执行\_\_、\_\_\_\_健壮性\_\_\_\_\_和\_\_\_\_\_\_\_有穷性\_\_。
3. 2.        一个算法的时间复杂度为(*n*3+*n2*log2*n*+14*n*)/*n2*，其数量级表示为\_\_\_\_\_\_\_\_。
4. 3.        假定一棵树的广义表表示为A（C，D（E，F，G），H（I，J）），则树中所含的结点数为\_­­­­\_\_\_\_\_\_\_\_\_个，树的深度为\_\_\_\_\_\_\_\_\_\_\_，树的度为\_\_\_\_\_\_\_\_\_。
5. 4.        后缀算式9 2 3 +- 10 2 / -的值为\_\_\_\_\_\_\_\_\_\_。中缀算式（3+4X）-2Y/3对应的后缀算式为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
6. 5.        若用链表存储一棵二叉树时，每个结点除数据域外，还有指向左孩子和右孩子的两个指针。在这种存储结构中，n个结点的二叉树共有\_\_\_\_\_\_\_\_个指针域，其中有\_\_\_\_\_\_\_\_个指针域是存放了地址，有\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_个指针是空指针。
7. 6.        对于一个具有n个顶点和e条边的有向图和无向图，在其对应的邻接表中，所含边结点分别有\_\_\_\_\_\_\_个和\_\_\_\_\_\_\_\_个。
8. 7.        AOV网是一种\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_的图。
9. 8.        在一个具有n个顶点的无向完全图中，包含有\_\_\_\_\_\_\_\_条边，在一个具有n个顶点的有向完全图中，包含有\_\_\_\_\_\_\_\_条边。
10. 9.        假定一个线性表为(12,23,74,55,63,40)，若按Key % 4条件进行划分，使得同一余数的元素成为一个子表，则得到的四个子表分别为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
11. 10.    向一棵B\_树插入元素的过程中，若最终引起树根结点的分裂，则新树比原树的高度\_\_\_\_\_\_\_\_\_\_\_。
12. 11.    在堆排序的过程中，对任一分支结点进行筛运算的时间复杂度为\_\_\_\_\_\_\_\_，整个堆排序过程的时间复杂度为\_\_\_\_\_\_\_\_。
13. **12.** 在快速排序、堆排序、归并排序中，\_\_\_\_\_\_\_\_\_排序是稳定的。

1. **三、                   运算题（每题 6 分，共24分）**
2. 1.        在如下数组A中链接存储了一个线性表，表头指针为A [0].next，试写出该线性表。

A 0 1 2 3 4 5 6 7

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| data |  | 60 | 50 | 78 | 90 | 34 |  | 40 |
| next | 3 | 5 | 7 | 2 | 0 | 4 |  | 1 |

1. 2.        请画出图10的邻接矩阵和邻接表。



|  |
| --- |
| 图10 |

1. 3.        已知一个图的顶点集V和边集E分别为：

V={1,2,3,4,5,6,7}; E={(1,2)3,(1,3)5,(1,4)8,(2,5)10,(2,3)6,(3,4)15,

(3,5)12,(3,6)9,(4,6)4,(4,7)20,(5,6)18,(6,7)25};

用克鲁斯卡尔算法得到最小生成树，试写出在最小生成树中依次得到的各条边。

1. 4.        画出向小根堆中加入数据4, 2, 5, 8, 3时，每加入一个数据后堆的变化。

1. **四、                   阅读算法（每题7分，共14分）**
2. 1.     LinkList mynote(LinkList L)

{//L是不带头结点的单链表的头指针

if(L&&L->next){

q=L；L=L－>next；p=L；

S1： while(p－>next) p=p－>next；

S2： p－>next=q；q－>next=NULL；

}

return L；

}

请回答下列问题：

（1）说明语句S1的功能；

（2）说明语句组S2的功能；

（3）设链表表示的线性表为（a1,a2, …,an）,写出算法执行后的返回值所表示的线性表。

1. 1.     （1）查询链表的尾结点

（2）将第一个结点链接到链表的尾部，作为新的尾结点

（3）返回的线性表为（a2,a3,…,an,a1）

1. 2.     void ABC(BTNode \* BT)

{

if BT {

ABC (BT->left);

ABC (BT->right);

cout<<BT->data<<' ';

}

}

该算法的功能是：

1. 2.     递归地后序遍历链式存储的二叉树。

1. **五、                   算法填空（共8分）**

二叉搜索树的查找——递归算法:

bool Find(BTreeNode\* BST,ElemType& item)

{

if (BST==NULL)

return false; //查找失败

else {

if (item==BST->data){

item=BST->data;//查找成功

return ­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_;}

else if(item<BST->data)

return Find(\_\_\_\_\_\_\_\_\_\_\_\_\_\_,item);

else return Find(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,item);

}//if

}

1. **六、                   编写算法（共8分）**

统计出单链表HL中结点的值等于给定值X的结点数。

int CountX(LNode\* HL,ElemType x)

参考答案

1. 一、                      单选题（每题2分，共20分）

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

1. 二、                      填空题（每空1分，共26分）
   1. 1.     正确性 易读性 强壮性 高效率
   2. 2.     O(n)
   3. 3.     9 3 3
   4. 4.     -1 3 4 X \* + 2 Y \* 3 / -
   5. 5.     2n n-1 n+1
   6. 6.     e 2e
   7. 7.     有向无回路
   8. 8.     n(n-1)/2 n(n-1)
   9. 9.     （12，40） （ ） （74） （23,55，63）
   10. 10.  增加1
   11. 11.  O(log2n) O(nlog2n)
   12. 12.  归并
2. 三、                      运算题（每题6分，共24分）
3. 1.     线性表为：（78，50，40，60，34，90）
4. 2.     邻接矩阵：

邻接表如图11所示：

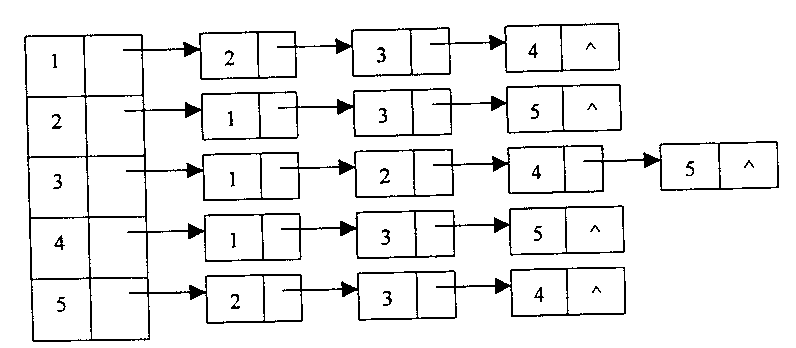


图11

1. 3.     用克鲁斯卡尔算法得到的最小生成树为：

(1,2)3, (4,6)4, (1,3)5, (1,4)8, (2,5)10, (4,7)20

1. 4.     见图12

|  |
| --- |
| 4 |

|  |
| --- |
| 4 |

|  |
| --- |
| 4 |

|  |
| --- |
| 4 |

|  |
| --- |
| 4 |

|  |
| --- |
| 2 |

|  |
| --- |
| 2 |

|  |
| --- |
| 2 |

|  |
| --- |
| 5 |

|  |
| --- |
| 5 |

|  |
| --- |
| 2 |

|  |
| --- |
| 8 |

|  |
| --- |
| 5 |

|  |
| --- |
| 2 |

|  |
| --- |
| 8 |

|  |
| --- |
| 3 |

|  |
| --- |
| 4 |

|  |
| --- |
| 5 |

|  |
| --- |
| 2 |

|  |
| --- |
| 8 |

|  |
| --- |
| 4 |

|  |
| --- |
| 3 |

图12

1. 四、                      阅读算法（每题7分，共14分）
2. 1.     （1）查询链表的尾结点

（2）将第一个结点链接到链表的尾部，作为新的尾结点

（3）返回的线性表为（a2,a3,…,an,a1）

1. 2.     递归地后序遍历链式存储的二叉树。
2. 五、                      算法填空（每空2分，共8 分）

true BST->left BST->right

1. 六、                      编写算法（8分）

int CountX(LNode\* HL,ElemType x)

{ int i=0; LNode\* p=HL;//i为计数器

while(p!=NULL)

{ if (P->data==x) i++;

p=p->next;

}//while, 出循环时i中的值即为x结点个数

return i;

}//CountX