**计算机2000数据结构考试试题**

（考试时间120分钟）

姓名：—————————————— 考号：—————————————— 班级：——————————

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 总 分 |  | 题 号 | 一 | 二 | 三 | 四 | 五 | 平时成绩 |
| 核分人 |  | 题 分 | 20 | 10 | 20 | 36 | 14 |  |
| 复查人 |  | 得 分 |  |  |  |  |  |  |

1. **单项选择题**（在每小题的四个备选答案中，选出一个正确的答案，并将其号码填在题干的括号内。每小题2分，共20分）

1．向一个有127个元素的顺序表中插入一个新元素并保持原来顺序不变，平均要移动（ ）个元素。

A、8 B、63.5

C、63 D、7

2．要进行二分查找，则线性表（ ）

A．必须以顺序方式存储 B．必须以顺序方式存储，且数据元素按键值有序

C．既可用顺序方式存储，也可用链式方式存储

D．必须以链式方式存储，且数据元素按键值有序

3．用数组 data[0..m]作为循环队列的存储空间，设front为队头指针，指示队头元素在数组中实际位置的前一位置，且约定队头指针指示的位置不用于存储队列元素，rear为对尾指针，指向队尾元素在数组中的实际位置。则该循环队列为空的条件为（ ）

A．rear=0 B．rear=front

C．(rear+1) MOD m =front D. (rear+1) MOD(m+1) =front

4．一个有序顺序表有255个对象，采用顺序搜索法查表，平均搜索长度为（ ）。

A、128 B、127

C、126 D、255

5．如果从无向图的任一顶点出发进行一次深度优先搜索即可访问所有顶点，则该图一定是（ ）。

A．连通图 B．完全图 C．有回路（环） D．一棵树

6．顺序栈的上溢指（ ）

A．栈满时作退栈运算 B．栈满时作进栈运算

C．栈空时作退栈运算 D．栈空时作进栈运算

7．已知一哈希表，采用链地址法处理冲突，在这种表上查找某一键值，可能要查找多次，所有被查找的键值（ ）

A．一定都是同义词 B．均不是同义词

C．不一定都是同义词 D．都相同

8．一个二叉树按顺序方式存储在一个一维数组中，如图

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| A | B | C | D |  | E | F |  | G |  |  | H |  | I | J |

则结点E在二叉树的第（ ）层。

A、1 B、2

C、3 D、4

9．串是任意有限个（ ）

A．符号构成的序列 B．符号构成的集合

C．字符构成的序列 D．字符构成的集合

10．带权有向图G用邻接矩阵A存储，则顶点i的入度等于A中（ ）

A.第i行非∞元素之和 B. 第i列非∞元素之和

C. 第i行非零且非∞元素个数 D. 第i列非零且非∞元素个数

二、**判断题**（判断下列各题是否正确，正确在括号内打“√”，错的打“×”。每小题1分，共10分）

1．如果两个串的长度相等，且含有相同的字符，则说两个串相等。（ ）

2．链栈的初始化是开辟足够多的结点，然后置栈顶指针为NULL。（ ）

3．将一棵树转换成二叉树后，根结点没有右子树。 （ ）

4．在一个有向图的邻接表或逆邻接表中，如果某个顶点的链表为空，则该顶点的度为零。（ ）

5．对一个连通图进行广度优先搜索，若顶点X在顶点Y之前被访问，则X的所有未被访问的邻接点一定在Y的所有未被访问的邻接点之前被访问。（ ）

6．单循环链表中，任一结点的后继指针均指向其逻辑后继。（ ）

7．在一个有向图的拓扑序列中，若顶点a在顶点b之前，则图中必有一条弧<a,b>（ ）

8．用二分查找的查找速度，肯定比顺序查找的查找速度快。（ ）

9．图的深度优先遍历和广度优先遍历均可借助于栈进行。（ ）

10．用链地址法构造的哈希表中各关键字不会发生冲突。（ ）。

三、**填空题**（每题2分，共20分）

1．在一个带头结点的单循环链表中，p指向尾结点的直接前驱，则指向头结点的指针head可用p表示为head= 。

2．在有m 个叶子的赫夫曼树中，其总结点个数为（ ）。

3．设有满足二分查找法要求的查找表R（键值按递增顺序排列），查找区间为[l，h],要查找的键值为K，首先被比较元素的位置为mid=(l+h) DIV 2，若R[MID].key >K，则h改为（ ）；二分查找的结束条件是（ ）。

4．在仅有尾指针R且带有头结点的单循环链表中，指针P所指结点是第一个元素结点的条件是P=（ ）。

5．对有10个元素的有序表，采用二分查找，需要比较4次方可找到的元素个数为（ ）

6．设某双链表的结点形式为 ，若要在指针q所指结点（中间结点）

|  |  |  |
| --- | --- | --- |
| Prior | data | next |

的后面插入一个新结点，则需执行下述语句段：

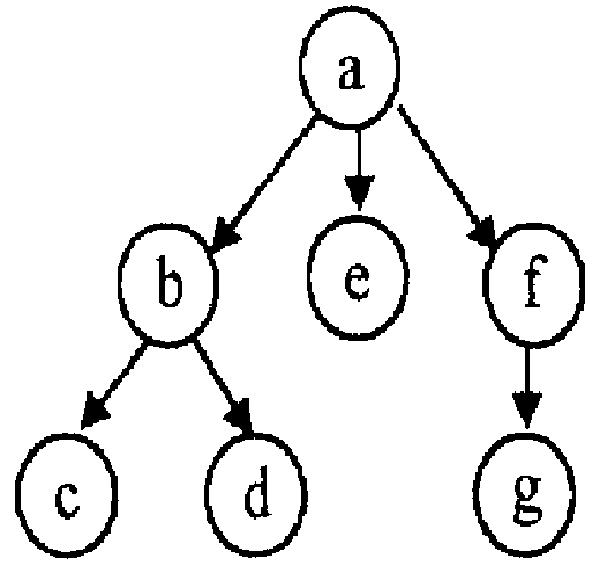
s->prior=q; s->next=q->next ; ( ); q->next=s;

7．设有向图G有n个顶点v1,v2,v3,…vn，它的邻接矩阵为A，顶点vi的入度ID（vi）为（ ）；顶点vi的出度OD（vi）为( ).

8．设一个链栈的栈顶指针为ls，栈中结点格式为 栈空条件是

|  |  |
| --- | --- |
| info | link |

（ ），如果栈不空，则退栈操作为p=ls； （ ）；free(p)。

9．设链队列lq中结点的格式为 。头指针为lq->front，尾指针为lq->rear，队列为空的条件（ ）。

|  |  |
| --- | --- |
| data | next |

10．已知一个图的广度优先生成树如右图所示，则与此相

应的广度优先遍历序列为 。

1. 应用题（共36分）

1．已知一个散列表如下图所示：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 35 |  | 20 |  |  | 33 |  | 48 |  |  | 59 |

0 1 2 3 4 5 6 7 8 9 10 11 12

其散列函数为h(key)=key%13, 处理冲突的方法为双重散列法，探查序列为：

hi=(h(key)+\*h1(key))%m =0,1,…，m－1

其中

h1(key)=key%11+1

回答下列问题：

（1）对表中关键字35，20，33和48进行查找时，所需进行的比较次数各为多少？

（2）该散列表在等概率查找时查找成功的平均查找长度为多少？

2．试分别找出满足下列条件的所有二叉树：

* + 1. 先根序列和中根序列相同；
    2. 中根序列和后根序列相同；

（3） 先根序列和后根序列相同；

3．已知连通网的邻接矩阵如下，试画出它所表示的连通网及该连通网的最小生成树。（用PRIM算法）。

∞ 1 12 5 10

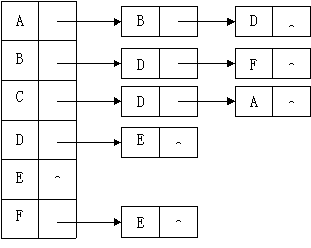
1 ∞ 8 9 ∞

12 8 ∞ ∞ 2

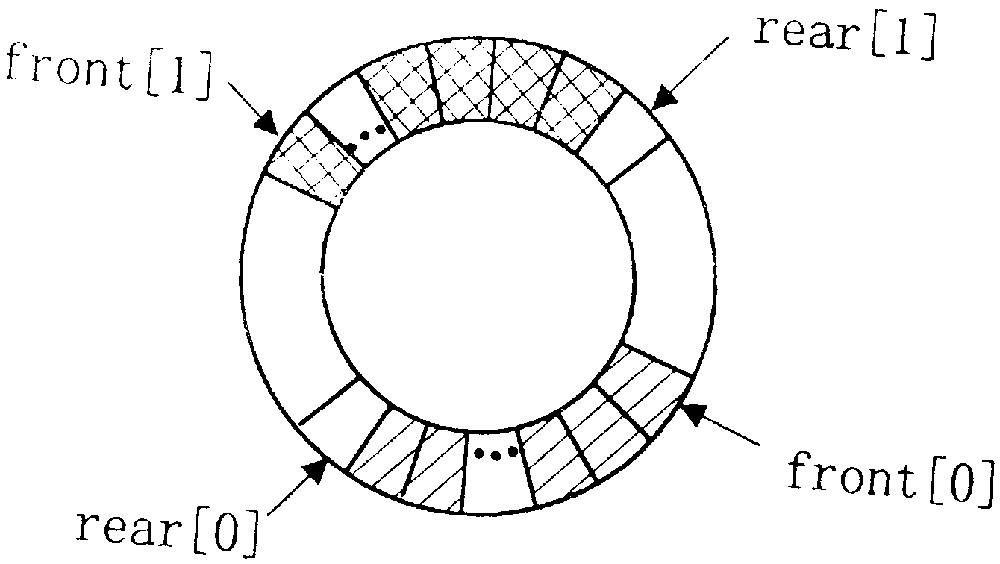
5 9 ∞ ∞ 4

10 ∞ 2 4 ∞

4．图1是用邻接表存储的图，画出此图，并写出从C点开始按深度优先遍历该图的结果。



5．假设两个队列共享一个循环向量空间（参见右下图），

 其类型Queue2定义如下：

typedef struct{

DateType data[MaxSize]；

int front[2],rear[2]；

}Queue2；

对于i=0或1，front[i]和rear[i]分别为第i个队列的头指针和尾指针。请对以下算法填空，实现第i个队列的入队操作。

int EnQueue (Queue2\*Q,int i,DateType x)

{//若第 i个队列不满，则元素x入队列，并返回1；否则返回0

if(i<0||i>1)return 0；

if(Q－>rear[i]==Q－>front[ ① ]return0；

Q－>data[ ② ]=x；

Q－>rear[i]=[ ③ ];

return1；

}

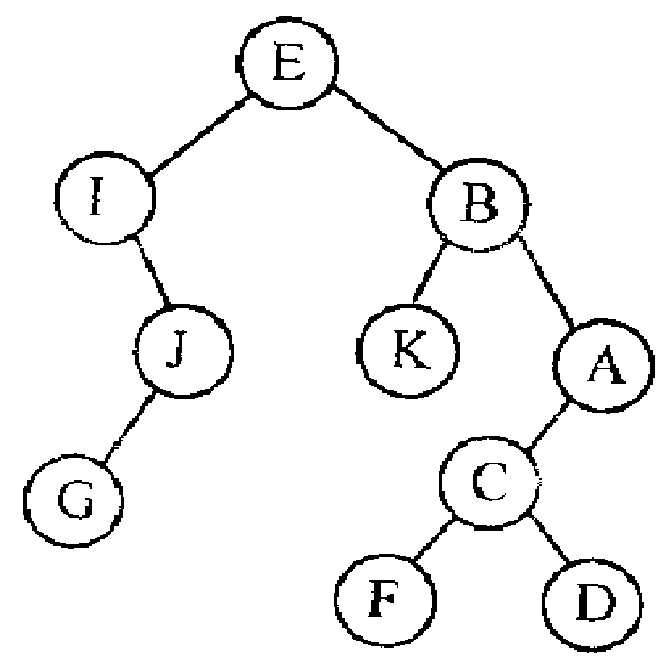
①

②

③

    6．用序列（46，88，45，39，70，58，101，10，66，34）建立一个排序二叉树，画出该树和所有关键字都插入完的AVL树。并求在等概率情况下查找成功的平均查找长度。

7．请画出与下列二叉树对应的森林。



1. **设计题**（共14分）
   1. 设有两个单链表L和L1，试画出单链表L的结构图（一个数据域，一个指针域），并编写算法判断链表L1中的各元素是否都是链表L中的元素。（8分）
   2. 设一棵二叉树以二叉链表为存储结构。设计一个算法求此二叉树树上数据域的值最大的结点。（6分）

**计算机2000-1、2、3、4、5数据结构考试试题参考答案**

1. **单项选择题**（在每小题的四个备选答案中，选出一个正确的答案，并将其号码填在题干的括号内。每小题2分，共20分）

**1．B 2．B 3．B 4．A 5．A**

**6．B 7．A 8．C 9．C 10. D**

二、**判断题**（判断下列各题是否正确，正确在括号内打“√”，错的打“×”。每小题1分，共10分）

1．**×** 2．**×** 3．**√** 4．**×** 5．**√**

6．**×** 7．**×** 8．**×** 9．**×** 10．**×**

**三、填空题**（每题2分，共20分）

1．P->next->next

2．2m-1

3．mid-1 R[mid].KEY==K或 l>h

4．R->next->next

5．3

6．q->next->prior=S

7．第i列非零元素之和（第i列非零元素个数， **）**

第i行非零元素之和（第i行非零元素个数， **）**

8．Ls==NULL ls=ls->link

9．Lq->front==lq->rear

10．a b e f c d g

**四、应用题（共36分）**

1．（6分）

（1）(4分)

解：由题义知 m=13

a)因为 h(35)=35 % 13 =9 但位置9不是35，所以又双重散列得

h1(35)=35 % 11 +1=3

所以 h1=(9+1\*3) % 13=12 但位置12 不是 35，所以 又双重散列得

h2=(9+2\*3) % 13 =2 可知，位置2 是 35 ，所以 查找35共比较了3次。

b)因为 h(20)=20 % 13 = 7 但位置7不是20，所以又双重散列得

h1(20)=20 % 11 +1=10

h1=(7+1\*10) % 13 =4 可知，位置4 是 20 ，所以 查找20共比较了2次。

c)因为 h(33)=33 % 13 = 7 但位置7 是33，所以 查找33共比较了1次。

d)因为 h(48)=48 % 13 = 9 但位置9 是48，所以 查找48共比较了1次。

e)同理，可知查找59需比较2次。

(2)（2分）查询成功的平均查找长度为：

1/5\*（1+1+2+2+3）=9/5=1.8

2．（5分）

1. 其中任意一个结点都无左孩子；
2. 其中任意一个结点都无右孩子；
3. 至多只有一个结点；

3．（6分）

连通网为： 4

5 10

9 2

1 12

8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| i  closedg | 2 | 3 | 4 | 5 | **U** | **V-U** | K |
| Adjvex  lowcost | **1**  **1** | 1  12 | 1  5 | 1  10 | {1} | {2，3，4，5} | 2 |
| Adjvex  lowcost | **0** | 2  8 | **1**  **5** | 1  10 | {1，2} | {3，4，5} | 4 |
| Adjvex  lowcost | **0** | 2  8 | 0 | **4**  **4** | {1，2，4} | {3，5} | 5 |
| Adjvex  lowcost | **0** | **5**  **2** | **0** | **0** | {1，2，4，5} | {3} | 3 |
| Adjvex  lowcost | **0** | **0** | **0** | **0** | {1，2，4，5，3} | { } |  |

所以最小生成树为：

4

5 2

1

4．（5分）由存储结构知此图为如右图所示：

其深度优先搜索序列为

C，D，E，A，B，F

5．（5分）

1. （I+1） % 2
2. Q→rear[I]
3. (Q→rear[I]+1) % maxsize

6．（5分）

二叉排序树为：

（2）平衡二叉树为：

为LL型，转换为：

为LL型，转换为：

7．对应的森林为：

**五、设计题**（共14分）

1.typedef struct node L

^ ……

{ datatype data;

struct node \* next; } node,\*lklist;

int match(lklist L, lklist L1)

^ ……

{ p1=L1->next; L1

flag1=0;

flag2=1;

if(p1==NULL) flag2=1;

else while(p1!=NULL)

{ p=L->next;

while(p1!=NULL && !flag1)

if(p->data==p1->data)flag1=1;

else p=p->next;

if(p==NULL)return(flag1);

else{

flag2=flag2 && flag1;

flag1=0;

}

p1=p1->next;

}

return flag2;

}

2．typedef struct node

{ datatype data;

struct node \* **lchild,\***rchild;

}node,\*bitreptr;

bitreptr ex1(bitreptr r)//m初值为无穷小

{ if(r!=NULL){

if(m>r->data)

{ m=r->data;

p=r; }

ex1(r->lchild);

ex1(r->rchild);

}

return (p);

}