**数据结构期末模拟题库**

**一、 选择题**

1.在一个长度为n的单链表的第i（0<=i<n）个元素后面插入一个元素时，需要向后移动（　**D** ）个元素。

A．n-i B.n-i+1 C.n-i-1 D.0

2．线性表的顺序存储结构是一种（ **B** ）的存储结构，线性结构的链式存储是一种（　**C**　）的存储结构。

A．随机存取 B.顺序存取 C.索引存取 D.散列存取

3.带头结点head的单向循环链表L为空的判断条件是（　**A**）

A. head==NULL B. head->next==NULL C. head->next==head D. head!=NULL

4.若某线性表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进入表中的最后一个元素，则采用（　**B**）存储方式最节省运算时间和存储空间。

A.单链表 B.仅有头指针的单循环链表

C.双向链表 D.有头尾指针的单循环链表

5.设有一个顺序栈S，元素a b c d e f依次进栈，如果6个元素出栈的顺序是b d c f e a，则栈的容量至少应该是（　**B**　）

A.2 B.3 C.5 D.6

6．向一个栈顶指针为top的带头结点的非空的链栈中删除结点，则其操作步骤是（　**C**）

A.top->next=s; B.s->next=top->next;top->next=s; free(s)

C. s = top;top= top->next;free(s) D. s = top->next;top= top->next;free(s)

7. 以下为平衡二叉排序树的查找算法, 假设表长为N，则算法的时间复杂度为（　**D** ）

A）O（1） B） O（N） C） O（N\*N） D） O（log2 N）

bitree \*search\_fsortree(t, key)

bitree \*t;

keytype key;

{ while(t!=NULL)

{ if(t->data==key)

return(t);

if(t->data>key)

t=t->lchild;

else t=t->rchild;

}

return(NULL);

}/\* SEARCH\_FSORTTREE \*/

8. 在解决计算机主机与打印机间速度不匹配问题时通常设置一个打印数据缓冲区，主机将要输出的数据依次写入该缓冲区，而打印机则从该缓冲区中取出数据打印。该缓冲区应该是一个（**D**　）结构

A、数组　　　B、线性表　　　C、堆栈　　　D、队列

9. 一棵有124个叶结点的完全二叉树，最多有（ **A** ）个结点

A、247　　　　B、248　　　　C、249　　　D、251

10.一棵非空的二叉树的前序遍历序列和后序遍历序列正好相同，则该二叉树一定满足（　**C**　）

A.所有的结点均无左孩子 B.所有的结点均无右孩子

C.只有一个孤立的结点 D.是任意一棵二叉树

11. 已知字符A、B、C、D的使用频率（权值）分别为22，7，9，27。对其进行HUFFMAN编码，各字符对应的编码为（ 　**C**  ）

A（001）B（100）C（110） D（0）

A（100） B（101）C（0） D（11）

A（11） B（100）C（111） D（0）

A（100）B（1011）C（11） D（0）

12.设有1000个基本有序的元素，希望用最快的速度挑选出其中前10个最大的元素，最后选用（　**A**　）排序法。

A.冒泡排序 B.快速排序 C. 直接插入排序 D. 归并排序

13.对序列（15，9，7，8，20，-1，4）进行排序，进行一趟排序后，数据的排列变为（4，9，7，8，-1，15，20），则采用的是（　**B**）排序。

A.选择排序 B.快速排序 C.希尔排序 D.冒泡排序

14.栈和队列的共同特点是(  **A**  )。

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

15.用链接方式存储的队列，在进行插入运算时( **D**  ).

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

16.以下数据结构中哪一个是非线性结构？(  **D**  )

A. 队列 　　 B. 栈 C. 线性表 　　 D. 二叉树

17.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( **C**  ){脚注(10)表示用10进制表示}

A．688 B．678 C．692 D．696

18.树最适合用来表示(  **C**  )。

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

19.二叉树的第k层的结点数最多为(  **D**  ).

A．2k-1 B.2K+1 C.2K-1 　　　D. 2k-1

20.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(  **D** )

A. 1，2，3 B. 9，5，2，3

C. 9，5，3 D. 9，4，2，3

21.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(  **C**  )

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

22.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（ **D** ）个，

A．1 B．2 C．3 D．4

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

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

24．下面关于线性表的叙述错误的是（ **D** ）。

(A) 线性表采用顺序存储必须占用一片连续的存储空间

(B) 线性表采用链式存储不必占用一片连续的存储空间

(C) 线性表采用链式存储便于插入和删除操作的实现

(D) 线性表采用顺序存储便于插入和删除操作的实现

25．设哈夫曼树中的叶子结点总数为m，若用二叉链表作为存储结构，则该哈夫曼树中总共有（ **B** ）个空指针域。

(A) 2m-1 (B) 2m (C) 2m+1 (D) 4m

26．设顺序循环队列Q[0：M-1]的头指针和尾指针分别为F和R，头指针F总是指向队头元素的前一位置，尾指针R总是指向队尾元素的当前位置，则该循环队列中的元素个数为（ **C** ）。

(A) R-F (B) F-R (C) (R-F+M)％M (D) (F-R+M)％M

27．设某棵二叉树的中序遍历序列为ABCD，前序遍历序列为CABD，则后序遍历该二叉树得到序列为（  **A** ）。

(A) BADC (B) BCDA (C) CDAB (D) CBDA

28．设某完全无向图中有n个顶点，则该完全无向图中有（ **A** ）条边。

(A) n(n-1)/2 (B) n(n-1) (C) n2 (D) n2-1

29．设某棵二叉树中有2000个结点，则该二叉树的最小高度为（ **C** ）。

(A) 9 (B) 10 (C) 11 (D) 12

30．设某有向图中有n个顶点，则该有向图对应的邻接表中有（ **B** ）个表头结点。

(A) n-1 (B) n (C) n+1 (D) 2n-1

31．设一组初始记录关键字序列(5，2，6，3，8)，以第一个记录关键字5为基准进行一趟快速排序的结果为（ **C** ）。

(A) 2，3，5，8，6 (B) 3，2，5，8，6

(C) 3，2，5，6，8 (D) 2，3，6，5，8

32．设某数据结构的二元组形式表示为A=(D，R)，D={01，02，03，04，05，06，07，08，09}，R={r}，r={<01，02>，<01，03>，<01，04>，<02，05>，<02，06>，<03，07>，<03，08>，<03，09>}，则数据结构A是（ **B** ）。

(A) 线性结构 (B) 树型结构 (C) 物理结构 (D) 图型结构

33．下面程序的时间复杂为（ **B** ）

for（i=1，s=0； i<=n； i++） {t=1；for(j=1；j<=i；j++) t=t\*j；s=s+t；}

(A) O(n) (B) O(n2) (C) O(n3) (D) O(n4)

34．设指针变量p指向单链表中结点A，若删除单链表中结点A，则需要修改指针的操作序列为（ **A** ）。

(A) q=p->next；p->data=q->data；p->next=q->next；free(q)；

(B) q=p->next；q->data=p->data；p->next=q->next；free(q)；

(C) q=p->next；p->next=q->next；free(q)；

(D) q=p->next；p->data=q->data；free(q)；

35.一组初始关键字记录关键字为(20，15，14，18，21，36，40，10)，则以20为基准记录的一趟快速排序结束后的结果为(  **A**  )。

(A) 10，15，14，18，20，36，40，21

(B) 10，15，14，18，20，40，36，21

(C) 10，15，14，20，18，40，36，2l

(D) 15，10，14，18，20，36，40，21

36.无向图G中有n个顶点e条边，则其对应的邻接表中的表头结点和表结点的个数分别为（ **D** ）。

(A) n，e (B) e，n (C) 2n，e (D) n，2e

37.设某强连通图中有n个顶点，则该强连通图中至少有（ **C** ）条边。

(A) n(n-1) (B) n+1 (C) n (D) n(n+1)

1. **填空题**

1.栈的逻辑特点是\_**后进先出**\_，队列的逻辑特点是 \_**先进先出**\_。

2.将含100个结点的完全二叉树从根开始，每层从左到右依次对结点编号，根结点的编号为1，则编号为31的结点的双亲的编号为\_\_**15**\_\_，其右子的编号为\_\_**63**\_\_。

3.设树F由T1，T2,T3三棵子树组成，与F对应的二叉树为B。已知T1,T2,T3的结点数分别为x，y，z，则该二叉树B的左子树中有\_\_**x-1**\_\_ 个结点，右子树中有\_**y+z**\_个结点。

4.设链队列的队头指针为front，队尾指针为rear，队列为空的条件是\_\_**rear==head**\_\_，队列为满的条件是\_\_\_(**rear+1)%MAXSIZE==head**\_\_\_\_。

5. 在有向图中，以顶点v为终点的边的数目称为v的\_\_**入度**\_\_。

6. 堆是一个键值序列 (k1,k2,…,kn)，对i=1,2,…, test_11，满足\_\_**（ki≤k2i且ki≤k2i+1 (2i+1≤n)**\_\_\_。

7.通常从四个方面评价算法的质量：\_\_**正确性**\_\_、\_\_**易读性**\_\_、\_\_**强壮性**\_\_、\_\_**高效率**\_\_

8.一个算法的时间复杂度为(n3+n2log2n+14n)/n2，其数量级表示为\_\_\_\_**O(n)**\_\_\_\_。

9.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\_\_**2n**\_\_个指针域，其中有\_\_**n-1**\_\_个指针域是存放了地址，有\_\_**n+1**\_\_个指针是空指针。

10.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\_\_**e\_**\_个和\_\_**2e\_**\_个。

11.在一个具有n个顶点的无向完全图中，包含有\_\_**n(n-1)/2**\_\_条边，在一个具有n个顶点的有向完全图中，包含有\_\_**n(n-1)**\_\_条边。

12.在快速排序、堆排序、归并排序中，\_\_**归并**\_\_排序是稳定的。

13.为了能有效地应用HASH查找技术，必须解决的两个问题是\_\_**构造一个好的HASH函数**\_\_和\_\_**确定解决冲突的方法\_**\_。

14.下面程序段的功能实现数据x进栈，要求在下划线处填上正确的语句。

typedef struct {int s[100]; int top;} sqstack;

void push(sqstack &stack,int x)

{

if (stack.top==m-1) printf(“overflow”);

else {\_\_\_**stack.top++**\_\_\_;\_\_\_**stack.s[stack.top]=x**\_\_\_;}

}

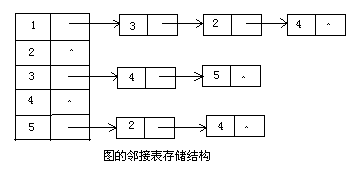
15.中序遍历二叉排序树所得到的序列是\_\_**有序**\_\_序列（填有序或无序）。

16.快速排序的最坏时间复杂度为\_\_**O(n2)**\_\_，平均时间复杂度为\_\_**O(nlog2n)**\_\_。

17.设某棵二叉树中度数为0的结点数为N0，度数为1的结点数为N1，则该二叉树中度数为2的结点数为\_\_**N0-1**\_\_；若采用二叉链表作为该二叉树的存储结构，则该二叉树中共有\_\_**2N0+N1**\_\_个空指针域。

18.设某无向图中顶点数和边数分别为n和e，所有顶点的度数之和为d，则e=\_\_**d/2**\_\_。

19． 已知一有向图的邻接表存储结构如下：从顶点1出发，DFS遍历的输出序列是\_\_ **(1，3，4，5，2)\_\_**，BFS遍历的输出序列是\_\_ **(1，3，2，4，5)\_\_**。



20.数据的物理结构主要包括\_\_**顺序存储结构**\_\_和\_\_**链式存储结构**\_\_两种情况。

21.设一棵完全二叉树中有500个结点，则该二叉树的深度为\_\_**9**\_\_；若用二叉链表作为该完全二叉树的存储结构，则共有\_\_**501**\_\_个空指针域。

22.设输入序列为1、2、3，则经过栈的作用后可以得到\_\_**5**\_\_种不同的输出序列。

23.设哈夫曼树中共有n个结点，则该哈夫曼树中有\_\_**0**\_\_个度数为1的结点。

24.\_\_**中序**\_\_遍历二叉排序树中的结点可以得到一个递增的关键字序列（填先序、中序或后序）。

25.设查找表中有100个元素，如果用二分法查找方法查找数据元素X，则最多需要比较\_\_7\_\_次就可以断定数据元素X是否在查找表中。

26.不论是顺序存储结构的栈还是链式存储结构的栈，其入栈和出栈操作的时间复杂度均为\_\_**O(1)**\_\_。

27.设有n个结点的完全二叉树，如果按照从自上到下、从左到右从1开始顺序编号，则第i个结点的双亲结点编号为\_\_**i/2**\_\_，右孩子结点的编号为\_\_**2i+1**\_\_。

28.设一组初始记录关键字为(72，73，71，23，94，16，5)，则以记录关键字72为基准的一趟快速排序结果为\_\_**(5，16，71，23，72，94，73)**\_\_。

1. 29.设有向图G中有向边的集合E={<1，2>，<2，3>，<1，4>，<4，2>，<4，3>}，则该图的一种拓扑序列为\_\_**(1，4，3，2)**\_\_。

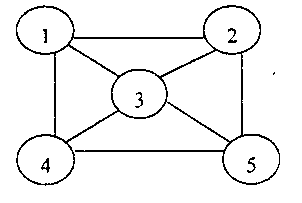
30.假定一棵树的广义表表示为A（C，D（E，F，G），H（I，J）），则树中所含的结点数为­­­­\_\_\_**9**\_\_\_个，树的深度为\_\_\_**3**\_\_\_，树的度为\_\_\_**3**\_\_\_。

31.AOV网是一种\_\_\_**有向无回路**\_\_\_的图。

32.设有向图G中有n个顶点e条有向边，所有的顶点入度数之和为d，则e和d的关系为\_\_**e=d**\_\_。

**三、计算题**

1. 请画出下图的邻接矩阵和邻接表。



1. 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。

3.已知二叉树的前序遍历序列是AEFBGCDHIKJ，中序遍历序列是EFAGBCHKIJD，画出此二叉树，并画出它的后序线索二叉树。

4.已知待散列的线性表为（36，15，40，63，22），散列用的一维地址空间为[0**..**6]，假定选用的散列函数是H（K）= K mod 7，若发生冲突采用线性探查法处理，试：

（1）计算出每一个元素的散列地址并在下图中填写出散列表：

` 0 1 2 3 4 5 6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

（2）求出在查找每一个元素概率相等情况下的平均查找长度。

5．已知序列（10，18，4，3，6，12，1，9，18，8）请用快速排序写出每一趟排序的结果。

**答案**：**1**.**邻接矩阵：**



**邻接表如图11所示：**

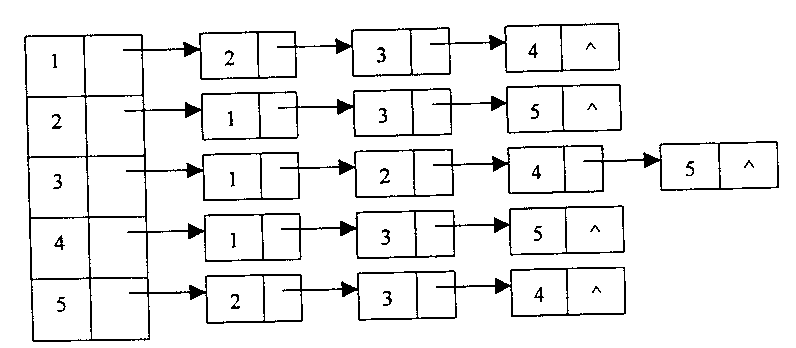


图11

1. **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**

**3．**

****

**4、H(36)=36 mod 7=1; H１(22)=(1+1) mod 7=2; ….冲突**

**H(15)=15 mod 7=1;….冲突 H2(22)=(2+1) mod 7=3;**

**H１(15)=(1+1) mod 7=2;**

**H(40)=40 mod 7=5;**

**H(63)=63 mod 7=0;**

**H(22)=22 mod 7=1; ….冲突**

**（1） 0 1 2 3 4 5 6**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **63** | **36** | **15** | **22** |  | **40** |  |

**（2）ASL=**

**5、(8,9,4,3,6,1),10,(12,18,18)**

**(1,6,4,3),8,(9),10,12,(18,18)**

**1,(3,4,6),8,9,10,12,18,(18)**

**1,3,(4,6),8,9,10,12,18,18**

**1,3, 4,6,8,9,10,12,18,18**

**四、应用题**

1. 设一组初始记录关键字序列为(45，80，48，40，22，78)，则分别给出第4趟简单选择排序和第4趟直接插入排序后的结果。
2. 设指针变量p指向双向链表中结点A，指针变量q指向被插入结点B，要求给出在结点A的后面插入结点B的操作序列（设双向链表中结点的两个指针域分别为llink和rlink）。
3. 设一组有序的记录关键字序列为(13，18，24，35，47，50，62，83，90)，查找方法用二分查找，要求计算出查找关键字62时的比较次数并计算出查找成功时的平均查找长度。
4. 设一棵树T中边的集合为{(A，B)，(A，C)，(A，D)，(B，E)，(C，F)，(C，G)}，要求用孩子兄弟表示法（二叉链表）表示出该树的存储结构并将该树转化成对应的二叉树。

**答案：**

1. **(22，40，45，48，80，78)，(40，45，48，80，22，78)**
2. **q->llink=p; q->rlink=p->rlink; p->rlink->llink=q; p->rlink=q;**
3. **2,ASL=91\*1+2\*2+3\*4+4\*2)=25/9**
4. **树的链式存储结构略，二叉树略**

**五、阅读算法**

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. void ABC(BTNode \* BT)

{

if BT {

ABC (BT->left);

ABC (BT->right);

Printf(“%d”,BT->data);

}

}

该算法的功能是：

**答案：**

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. **递归后序遍历链式存储的二叉树。**
2. 编写算法

1.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)

**答案：**

**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**

1. 设有两个集合A和集合B，要求设计生成集合C=A∩B的算法，其中集合A、B和C用链式存储结构表示。

**答案：**

**typedef struct node {int data; struct node \*next;}lklist;**

**void intersection(lklist \*ha,lklist \*hb,lklist \*&hc)**

**{**

**lklist \*p,\*q,\*t;**

**for(p=ha,hc=0;p!=0;p=p->next)**

**{ for(q=hb;q!=0;q=q->next) if (q->data==p->data) break;**

**if(q!=0){ t=(lklist \*)malloc(sizeof(lklist)); t->data=p->data;t->next=hc; hc=t;}**

**}**

**}**

**3.设计在单链表中删除值相同的多余结点的算法。**

**答案：**

**typedef int datatype;**

**typedef struct node {datatype data; struct node \*next;}lklist;**

**void delredundant(lklist \*&head)**

**{**

**lklist \*p,\*q,\*s;**

**for(p=head;p!=0;p=p->next)**

**{**

**for(q=p->next,s=q;q!=0; )**

**if (q->data==p->data) {s->next=q->next; free(q);q=s->next;}**

**else {s=q,q=q->next;}**

**}**

**}**