**2014年招收攻读硕士学位研究生入学考试试题( A卷)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

招生专业与代码：计算机系统结构081201，计算机软件与理论081202，计算机应用技术081203，软件工程083500，计算机技术(专业学位) 085211，软件工程(专业学位) 085212

考试科目名称及代码：数据结构830

**一．选择题(每题2分，共30分)**

1.数据结构是研究数据的( **C** )以及它们之间的相互关系.

A. 理想结构,物理结构 B.理想结构,抽象结构

C. 物理结构,逻辑结构 D.抽象结构,逻辑结构

2.线性表的链接实现有利于(  **A**  )运算

A.插入 B.读表元素 C. 查找 D.定位

3．从一个长度为n的顺序表中删除第i个元素（1≤i≤n）时，需向前移动（ **A** ）个元素.

A. n-i B. n-i+1 C. n-i-1 D. i

4.具有n个顶点的完全有向图的边数为(  **B** ).

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

5.快速排序在( **D**  )情况下最不利于发挥其长处.

A. 被排序的数据量太大. B. 被排序数据中含有多个相同的关键字.

C. 被排序的数据完全无序 D. 被排序的数据已基本有序

6. 线性表采用链式存储时,其地址(  **D**  ).

A.必须是连续的 B.部分地址必须是连续的

C.一定是不连续的 D.连续与否均可以

7.一个栈的进栈序列是a, b, c, d, e, 则栈的不可能的输出序列是( **C**  )

A. edcba B. decba C. dceab D. abcde

8. 采用顺序查找法查找长度为n的线性表时,每个元素的平均查找长度为 (  **C**  )

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

9.下列哪种排序需要的附加存储开销最大( **C**  ).

A快速排序 B堆排序 C 归并排序 D插入排序

10.具有6个顶点的无向图至少应有（ **A**  ）条边才能确保是一个连通图.

A．5 B．6 C．7 D．8

11.对具有n个结点的有序表中折半查找时，其时间复杂度是 ( **A**  ) .

A. O(log2n） B. O(nlog2n） C. O(n） D. O(n2）

12. 通过一趟排序就能从整个记录序列中选择出具有最大（或最小）关键字的记录，这种排序方法是(  **D**  ) .

A. 归并排序 B. 快速排序 C. 直接插入排序 D. 堆排序

13. 在AOE网中，完成工程的最短时间是（  **B**  ）.

A．从源点到汇点的最短路径的长度 B．从源点到汇点的最长路径的长度

C．最长的回路的长度 D．最短的回路的长度

14. 设单链表中指针p指着结点A,若要删除A之后的结点(若存在),则需要修改指针的操作为(  **A**  ).

A． p->next=p->next->next B． p=p->next

C． p=p->next->next D． p->next=p

15.下面的序列中，(  **A**  )是堆.

A. 1，2，8，4，3，9，10，5 B.1，5，10，6，7，8，9，2

C. 9，8，7，6，4，8，2，1 D.9，8，7，6，5，4，3，7

**二．填空题(每空2分，共20分)**

1. 线性结构中元素之间存在一对一关系，树型结构中元素之间存在 **一对多**  关系，图型结构中元素之间存在  **多对多** 关系.

2. 单链表中设置头结点的作用是 **使得不需要对表头元素的操作进行特殊化处理，空表和非空表插入元素时的处理得到统一**  .

3.由n个权值构成的哈夫曼树共有  **2\*n-1**  个结点.

4.已知一个图的邻接矩阵表示，删除所有从第i个结点出发的边的方法是 **将第i行的所有元素值（代表了i到j的距离）设置成无穷大**  .

5. 队列只允许在表的一端插入，在另一端删除；插入的一端叫 **队尾** ，删除的一端叫 **队头** ；对队列的访问是按照  **先进先出**  的原则进行的.

6. 在哈希查找方法中，要解决两方面的问题，它们分别是 **设置哈希函数**  及 **解决碰撞冲突**  .

**三．判断题（每题1分，共10分，正确的选t，错误的选f）**

1.已知一颗树的先序序列和后序序列，一定能构造出该树. （ **×** ）

2.双循环链表中，任一结点的前驱指针均为不空. （ **√** ）

3.对于n个记录的集合进行冒泡排序，在最坏情况下的时间复杂度是O(n2). ( **√** )

4. 快速排序是排序算法中最快的一种. （ **×** ）

5. 设有序的关键字序列是（2，5，8，9，12，14，16，18，20，22，25），当用折半查找方法查找关键字22时，需经3次比较运算. （ **√** ）

6.向二叉排序树中插入一个新结点，需要比较的次数可能大于此二叉树的高度h. （ **×** ）

7.散列法存储的思想是由关键字值决定数据的存储地址。 （ **√** ）

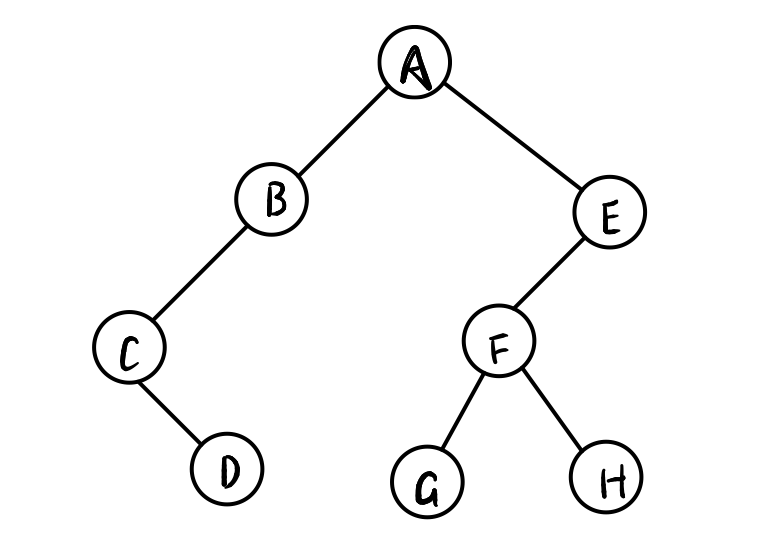
8.连通图的广度优先搜索中可以采用队列来暂存刚访问过的顶点. （ **√** ）

9. 一棵m阶B-树中每个结点最多有m棵子树，非终端结点最少有2棵子树. ( **×** )

10. 冒泡排序是稳定的. （ **√** ）

**四．简答题（共45分）**

1.已知一棵二叉树的中序为CDBAGFHE, 后序为DCBGHFEA,画出这棵二叉树.(6分)



2.如图1所示的AOE网(V1表示工程的开始,V8表示工程的结束), 假设工程从时间0开始,求出所有事件和活动允许发生的最早及最晚时间,并给出关键路径.( 14分)

a2=3

a10=2

a5=4

a1=5

a3=2

a4=12

a6=9

a7=7

a8=3

a9=6

**图1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **V1** | **V2** | **V3** | **V4** | **V5** | **V6** | **V7** | **V8** |
| **ve(i)** | **0** | **3** | **5** | **9** | **21** | **18** | **28** | **30** |
| **Vl(i)** | **0** | **7** | **5** | **9** | **21** | **18** | **28** | **30** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **a1** | **a2** | **a3** | **a4** | **a5** | **a6** | **a7** | **a8** | **a9** | **a10** |
| **e(i)** | **0** | **0** | **3** | **5** | **5** | **9** | **21** | **18** | **18** | **28** |
| **l(i)** | **0** | **4** | **7** | **9** | **5** | **9** | **21** | **18** | **22** | **28** |
| **l(i)-e(i)** | **0** | **4** | **4** | **4** | **0** | **0** | **0** | **0** | **4** | **0** |

**关键路径：V1->V3->V4->V6->V5->V7->V8**

3.简述下列算法的功能.（6分）

void process( Sqlist &L) //L为线性表,用顺序存储结构表示

{ int i=0, j;

While （i<L.length && L.elem[i]!=X）

i++;

for (j=i+1;j<L.length; j++)

if (L.elem[j]!=X)

{ L.elem[i]=L.elem[j];

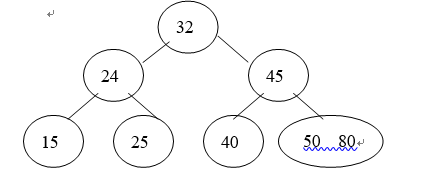
i++; }

L.length=i;

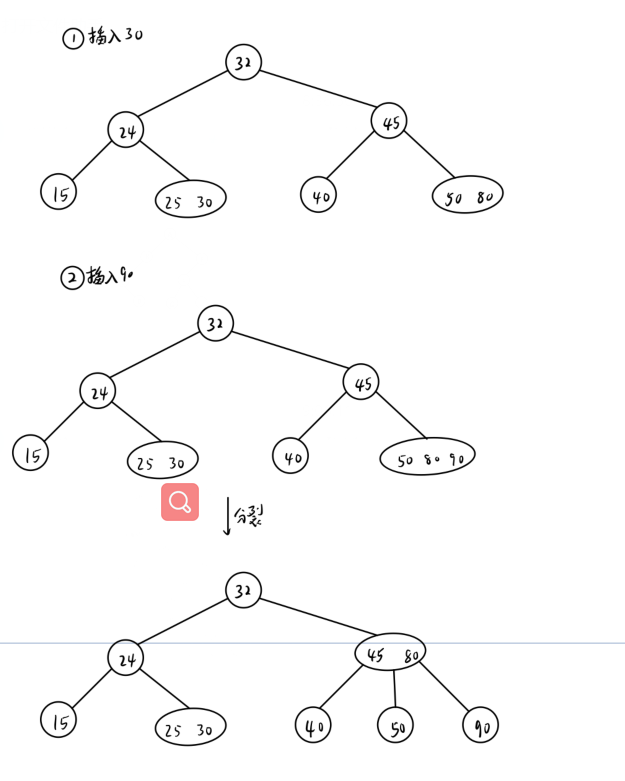
}

**答：该函数所实现的功能是删除线性表L中所有值为X的元素。**

4.已知一棵3阶的B-树如图2所示,依次插入关键字30 及90,分别画出每插入一个关键字后所生成的B-树. (7 分)



**图2**



5.已知序列(12, 178, 200,530,765,149, 52,6),请采用链式基数排序方法对该序列作升序排序, 给出排序过程.(12分)

**答：第一趟：200，530，12，52，765，6，178，149**

**第二趟：200，6，12，530，149，52，765，178**

**第三趟：6，12，52，149，178，200，530，765**

**五．算法填空，（每空2分,共20分）**

1. 以下算法功能是:插入元素e为新的栈顶元素,完成算法的空格部分.

Status Push(SqStack &S, ElemType e) {

if (S.top-S.base >= S.Stacksize) {

S.base=(ElemType \*) realloc(S.base,

(S.Stacksize+STACKINCREME) \* **①sizeof(ElemType)** );

if (② **!S.Base** ) exit (OVERFLOW);

S.top=S.base+ **③S.Stacksize** ;

S.Stacksize=S.Stacksize+STACKINCREMENT;

}

\*S.top= ④**e** ;

top= ⑤**top+1**  ;

return OK;

}

2.以下是图的广度遍历算法, 完成算法的空格部分.

Void BFSTraverse( Graph G, Status(\*visit)(int v)) {

for (v=0;v<G.vexnum;++v) visited[v]=False;

initQueue(Q);

for (v=0; **⑥V.Gvexnum** ;++v)

if (!visited[v]) {

visited[v]=True; Visit(v);

EnQueue(Q,v);

while (!QueueEmpty(Q)) {

**⑦DeQueue(Q,u)** ;

for (w=FirstAdjVex(G,u); w>=0 ; w=NextAdjVex(G,u,w ))

if (  **⑧!visited[w]** ) {

Visited[w]=**⑨True** ; Visit(w);

**⑩Enqueue(Q,w)** ;

}

}

}

}

**六．编写算法（25分）**

1. 设计将两个有序链表合并为一个有序链表的算法. 假设有序链表的元素按照非递减排列.(10 分)

**typedef struct ListLink{**

**elemType data;**

**ListLink \*next;**

**};**

**ListLink \*merge(ListLink \*L1, ListLink \*L2){**

**ListLink \*L = (ListLink \*)malloc(sizeof(ListLink));**

**ListLink \*tail = L;**

**ListLink \*p1 = L1;**

**ListLink \*p2 = L2;**

**while(p1 && p2){**

**if(p1.data < p2.data){**

**tail->next = p1;**

**p1 = p1->next;**

**}**

**else{**

**tail->next = p2;**

**p2 = p2->next;**

**}**

**tail = tail->next;**

**}**

**if(p1) tail->next = p1;**

**if(p2) tail->next = p2;**

**return L;**

**}**

2. 给定带权有向图G和源点V0, 设计V0到其余顶点的最短路径.(15 分)

**#define INF 0x3f3f3f3f**

**void solve(int G[][n], int v0, int n, int D[]){**

**bool vis[n];**

**for(int i = 0; i < n; i++){**

**D[i] = INF;**

**vis[i] = false;**

**}**

**D[v0] = 0;**

**for(int i = 0;i < n; i++){**

**int min = INF, u = -1;**

**for(int j = 0; j < n; j++){**

**if(!vis[j] && min > D[j]){**

**min = D[j];**

**u = j;**

**}**

**}**

**vis[u] = true;**

**for(int v = 0; v < n; v++){**

**if(!vis[v] && D[v] > D[u]+G[u][v]){**

**D[v] = D[u]+G[u][v];**

**}**

**}**

**}**

**}**