**北 京 交 通 大 学**

**2021 ―2022 学年第 一 学期期末考试试题**

**课程名称： 数据结构（A）**

**出题教师：黄雅平, 李向前，孙永奇，王志海，徐薇，原继东**

**专业:\_\_\_\_\_\_\_\_\_ 班级:\_\_\_\_\_\_\_\_\_\_\_\_ 姓名:\_\_\_\_\_\_\_\_\_ 学号:\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **题号** | **一** | **二** | **三** | **四** | **五** | **六** |  | **总分** |
| **得分** |  |  |  |  |  |  |  |  |
| **阅卷人** |  |  |  |  |  |  |  |  |

备注：请把答案写在答题卡上。

1. **单选题(每题2分，共 10小题，共20分)**
2. 一个单链表没有头结点，它的头指针是L，通过如下（ ）的表示可以指向第二个数据元素。

A. L B. L->next C. L->next->next D. L->next->data

1. 一组字符abcdef按顺序先进入一个栈，然后出栈，再按顺序进入一个队列再出队列，最后得到字符序列为（ ）。

A. abcdef B. aebdcf C. fedcba D. feadbc

1. 已知主串S=‘ababcbaa’ ，模式串T=‘ababcaab’，则next函数值及nextval函数值为（ 　）。

A. 01123112 和01013102 B. 01123122和00011011

C. 01123112和01011001 D. 01123122和01013021

1. 设数组data[m]作为循环队列q的存储空间，front 为队头指针，rear为队尾指针，判定队列q为满的条件是（　）。

A. q.front = q.rear B. (q.front+1)% m = q.rear

C. (q.rear+1)% m = q.front D. (q.rear+1)%（m-1）= q.front

1. 二叉树中度为2的结点有15个，度为1的结点有14个，则结点总数为（ ）。

A. 29 B. 43 C. 45 D. 46

1. 由树转化得到的二叉树称为该树对应的二叉树，则下面（ ）是正确的。

A. 树的先根遍历序列与其对应的二叉树先序遍历序列相同。

B. 树的先根遍历序列与其对应的二叉树中序遍历序列相同。

C. 树的后根遍历序列与其对应的二叉树后序遍历序列相同。

D. 以上都不对。

1. 一棵赫夫曼树共有69个结点，对其进行赫夫曼编码，总共得到（ ）个不同的编码。

A. 34 B. 35 C. 69 D. 70

1. 如下有向带权图，若采用迪杰斯特拉（Dijkstra）算法求源点a到其它各顶点的最短路径，得到的第一条最短路径的目标顶点是b，第二条最短路径的目标顶点是c，后续得到的其余各最短路径的目标顶点依次是（ ）。

A. d, e, f B. f, d, e C. e, d, f D. f, e, d

1. 一棵3阶B-树中含有1023个关键字，包含空的叶子结点层，该树的最大深度为（ ）。

A. 9 B. 10 C. 11 D. 12

1. 对序列(21,15,32,18,28,6,23 )用希尔排序方法排序，经一趟排序后序列变为(21,6,23,18,28,15,32)，则该趟排序采用的增量是（ ）。

A. 2 B. 3 C. 4 D. 5

1. **填空题(每题1分，共15小题，共15分)**
2. 数据结构的形式定义为<D, S>，其中D是数据元素的有限集，S是D上\_\_\_\_\_\_\_\_\_\_的有限集。
3. 若一个算法中的语句频度之和为T(n)=500n+n2，则算法的时间复杂度为\_\_\_\_\_\_\_\_\_\_。
4. 在表长为n的顺序表中，如果要在第i（1≤i≤n）个数据元素之后插入一个新的元素，要移动\_\_\_\_\_\_\_\_\_\_个数据元素。
5. 对串S=‘BEIJINGJIAOTONG’，执行Substring(&Sub,S,10,4)，则Sub=\_\_\_\_\_\_\_\_\_\_。
6. 稀疏矩阵中的元素以三元组顺序表来表示，则矩阵中的元素(7,5,19)在相应转置矩阵中的表示为\_\_\_\_\_\_\_\_\_\_。
7. 设森林F对应的二叉树为B，它有m个结点，B的根为p，p的右子树结点个数为n，森林F中第一棵树的结点个数是\_\_\_\_\_\_\_\_\_\_。
8. 在赫夫曼树中有10个叶子结点，度为1的结点有\_\_\_\_\_\_\_\_\_\_个。
9. n个顶点的无向连通图用邻接矩阵表示时，该矩阵至少有\_\_\_\_\_\_\_\_\_\_个非零元素。
10. 对于有e条边的无向图，所有顶点的度的和为\_\_\_\_\_\_\_\_\_\_。
11. \_\_\_\_\_\_\_\_\_\_算法适用于求边比较稠密的网的最小生成树。
12. 从有序表(17,19,32,43,56,78,80,99)中折半查找元素80时，查找成功时的比较次数为\_\_\_\_\_\_\_\_\_\_次。
13. 一棵深度为K的平衡二叉树，所有结点的平衡因子都为0，则该树共有\_\_\_\_\_\_\_\_\_\_个结点。
14. 采用链地址法解决冲突的哈希表中，查找成功的平均查找长度只与\_\_\_\_\_\_\_\_\_\_有关。
15. 冒泡排序在最好情况下的时间复杂度为\_\_\_\_\_\_\_\_\_\_。
16. 若需要在O(nlogn)的时间内完成对数组的排序，且要求占用辅助空间最少，则应选择\_\_\_\_\_\_\_\_\_\_排序方法。
17. **简答题(每题7分，共5小题，共35分)**
18. 有三个元素的进栈顺序为X、Y、Z，列举出此三个元素可能的出栈序列，并写出相应的进栈和出栈操作序列（假设以push和pop分别表示进栈和出栈操作）。
19. 已知广义表A按表头表尾分析法的存储结构如下图所示, 请写出该广义表，并求其深度和长度。



1. （1）已知一棵二叉树的中序遍历序列为DBACFEGMKLPNS，后序遍历序列为DAFGECBLSNPKM，画出这棵二叉树，并给出层次遍历序列。（2）假设（1）中形态的二叉树是由森林按照孩子兄弟链表表示法转换而来的，画出森林。
2. 已知一组关键字序列为：(35,21,47,60,52,50,73,92)，画出其构成的平衡二叉树，给出主要的平衡过程。
3. 已知关键字序列(45, 27, 20, 32, 19, 58, 12, 25, 38, 16)，用哈希函数H(key)=key mod 13构建哈希表，散列地址空间为0-12，用开放定址法的二次探测再散列方法Hi=(H(key)+di) mod 13 (di=12, -12, 22, -22, 32,…,)处理冲突，请画出所构造的哈希表，并计算等概率情况下查找成功时的平均查找长度。
4. **算法阅读题（共2小题，共10分）**

#include <stdio.h>

#include <stdlib.h>

typedef struct BitNode {

int data;

struct BitNode \* Lchild;

struct BitNode \* Rchild;

}\*BitTree;

BitTree F1(BitTree root) {

int data ;

scanf("%d", &data);

if(data == 0) {

root = NULL;

} else {

root = (BitTree)malloc(sizeof(BitNode));

root->data= data;

if(!root) {

printf("Fail!\n");

exit(1);

}

root->Lchild = F1(root->Lchild);

root->Rchild = F1(root->Rchild);

}

return root;

}

void F2(BitTree root) {

BitTree stack[256];

BitTree p = root;

int tag[256], top=0;

do

{ while(p!=NULL){

top++; stack[top]=p; tag[top]=0; p=p-> Lchild;

}

while((top>0) && tag[top]==1){

printf("%d ",stack[top]->data); top--;

}

p=stack[top];

if((top>0)&&(tag[top]==0)){

p=p->Rchild; tag[top]=1;}

}while(p!=NULL || top!=0);

}

int F3(BitTree root){

int t1=0,t2=0,out=0;

if(!root) out=0;

else{

t1= F3(root->Lchild);

t2= F3(root->Rchild);

out=1+(t1>t2?t1:t2);

}

return out;

}

int main() {

BitTree root = NULL ;

root = F1(root);

F2(root);

printf("\n");

printf("%d \n",F3(root));

}

1. ( 6分) 说明算法中F1，F2和F3的功能。
2. ( 4分) 若F1输入为2 5 6 9 0 0 0 8 0 0 4 1 0 0 3 0 0，其中输入用空格分隔，0表示NULL，请写出程序运行结果。
3. **算法填空题（每空2分，共10分）**

下面算法构造图的最小生成树，请把算法中划横线处缺失的语句补齐。

struct

{ VertexType adjvex; // U集中的顶点序号

VRType lowcost; // 边的权值

} closedge [MAX\_VERTEX\_NUM];

void MiniSpanTree\_PRIM( MGraph G, VertexType u )

{ //用普里姆算法从顶点u出发构造网G的最小生成树

( 33 ) = LocateVex ( G, u );

for ( j = 0; j < G.vexnum; ++j ) // 辅助数组初始化

if( j ( 34 ) k )

closedge[j] = { u, G.arcs[k][j].adj }; // {adjvex, lowcost}

closedge[k].lowcost = 0; // 初始，U＝{u}

for ( i = 1; i < G.vexnum; ++i) //选择其余N-1个顶点

{

k = minimum(closedge); // 求出加入生成树的下一个顶点k

printf(closedge[k].adjvex, G.vexs[k]); // 输出生成树上一条边的两个顶点

closedge[k].lowcost = ( 35 ) ; // 第k顶点并入U集

for ( j = 0; j < G.vexnum; ++j ) // 修改其它顶点的最小边

if ( G.arcs[k][j].adj ( 36 ) closedge[j].lowcost )

// 新顶点并入U后重新选择最小边

( 37 ) = { G.vexs[k], G.arcs[k][j].adj };

}

}

1. **算法设计题（共1题，共10分）**

38、设二维数组为int A[n][n]（n<500），试对该二维数组设计一个排序算法

ArraySort (int A[500][500], int n)，

排序规则如下：

根据二维数组中每一行数字的和，用快速排序算法对行进行非递增排序。排序示例如下图所示，其中(a)为原始的二维数组，(b)为执行排序规则后的结果。

1. (b)

**北 京 交 通 大 学**

**2021 ―2022 学年第 一 学期期末考试试题标准答案**

**课程名称： 数据结构（A）**

**出题教师：黄雅平, 李向前，孙永奇，王志海，徐薇， 原继东**

**一、单选题(每题2分，共10小题，共 20分)**

1~10：BCDCC ABBCC

**二、填空题（每题1分, 共15小题，共 15分）**

1. 关系
2. O(n2)
3. n-i
4. ‘AOTO’
5. (5,7,19)
6. m – n
7. 0
8. 2（n-1）
9. 2e
10. Prim或普里姆
11. 3
12. 2K-1
13. 装填因子或α
14. O(n)
15. 堆

**判分标准：**有微小错误可给0.5分。

**三、简答题(每题7分，共5小题，共35分)**

1. 判分标准：错一个扣1.5分，扣完7分为止。

答案：

出栈顺序 操作序列

X、Y、Z push pop push pop push pop

X、Z、Y push pop push push pop pop

Y、X、Z push push pop pop push pop

Y、Z、X push push pop push pop pop

Z、Y、X push push push pop pop pop

1. **判分标准：**广义表3分，长度2分，深度2分。

答案： 广义表 ((a) , ((b,c),d)) ；

长度2 ；

深度3

1. **判分标准：**二叉树3分，层序1分，森林3分。

答案： (1)二叉树如下图所示



层次遍历序列：MBKDCPAELNFGS

(2)森林如下图所示



1. **判分标准：**错一个扣2分，扣完为止。

答案：



1. **判分标准：**哈希表全对给5分，其中每个关键字位置给0.5分；平均查找长度2分。

答案：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 地址空间 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 关键字 | 25 | 27 | 58 | 16 |  | 32 | 45 | 20 |  |  | 19 | 38 | 12 |
| 比较次数 | 2 | 1 | 5 | 1 |  | 3 | 1 | 1 |  |  | 4 | 3 | 1 |

查找成功时的平均查找长度 ASL =(2+1+5+1+3+1+1+4+3+1)/10 = 2.2

**四、算法阅读题（共2小题，共10分）**

31、F1: 递归创建一个二叉树 （2分）

F2: 后序非递归遍历二叉树（2分）

F3: 递归求二叉树深度 （2分）

32、程序输出结果：

96851342 （2分）

4 （2分）

**五、算法填空题（每空2分，共10分）**

33、k

34、 !=

35、 0

36、 <

37、 closedge[j]

**六、算法设计题（共1题，共10分）**

38、

void CopyArray(int a[], int b[], int n)

{

for (int i=1; i<=n; i++) {

a[i] = b[i];

}

}

//一趟快速排序

int Partition(int S[], int low, int high, int A[][500], int n)

{

int pivotkey;

S[0] = S[low];

CopyArray(A[0], A[low], n);

pivotkey = S[low];

while (low < high) { (1分)

while (low < high && S[high] <= pivotkey) { (1分)

--high;

}

S[low] = S[high]; (1分)

CopyArray(A[low], A[high], n); (2分)

while (low < high && S[low] >= pivotkey) { (1分)

++low;

}

S[high] = S[low];

CopyArray(A[high], A[low], n);

}

S[low] = S[0];

CopyArray(A[low], A[0], n);

return low;

}

//快速排序

void QSort(int S[], int low, int high, int A[][500], int n)

{

int pivotloc;

if (low < high) {

pivotloc = Partition(S, low, high, A, n); (1分)

QSort(S, low, pivotloc - 1, A, n); (1分)

QSort(S, pivotloc + 1, high, A, n);

}

}

//对二维数组的排序算法

void ArraySort(int A[500][500], int n)

{

//对二维数组中每一行数字求和并根据二维数组中每一行数字的和，用快速排序算法对行进行非递增排序

int S[500];

for (int i=1; i<=n; i++) { (2分)

S[i] = 0;

for (int j=1; j<=n; j++) {

S[i] += A[i][j];

}

}

QSort(S, 1, n, A, n);

}

判分标准：

快速排序6分，数组移动2分，排序函数ArraySort 2分。