

日期: /

—
1.是指一个数学模型以及定义在该模型上的一组操作。形式定义是用一个三元组来表示一个抽象数据类型

— 5.堆排序将无序序列建成一个堆，得到关键字最小（或最大）的记录；输出堆顶的最小（最大）值后，使剩余的 $n-1$ 个元素重复又建成一个堆，则可得到 n 个元素的次小值；重复执行，得到一个有序序列的过程

6.AOE网中路径长度最长的路径

1.是指一个数学模型以及定义在该模型上的一组操作。形式定义是用一个三元组来表示一个抽象数据类型

南京审计大学

2020 年硕士研究生招生考试初试(笔试)试题(A卷)

科目代码: 911

科目名称: 数据结构

2.数据元素之间的逻辑关系的描述,它可以用一个数据元素的集合和定义在此集合上的若干关系来表示(集合,线性结构,树型结构,图形结构)

3.不要求逻辑上相邻的结点物理上也相邻。结点间的逻辑关系是由附加的指针字段表示的

注意: ①认真阅读答题纸上的注意事项; ②所有答案必须写在答题纸上, 写在本试题纸或草稿纸上均无效; ③本试题纸须随答题纸一起装入试题袋中交回!

4.将二叉树中空的指针域利用起来, 用来保存遍历过程(先序中序后序)中前驱结点和后继结点的信息, 这样的信息叫线索, 左指针指向前驱结点, 右指针指向后继结点。

一、 名词解释(共 6 小题, 每题 5 分, 总分 30 分)

1. 抽象数据类型
2. 数据逻辑结构
3. 链式存储
4. 线索二叉树
5. (排序) 堆
6. 关键路径

5.堆排序将无序序列建成一个堆, 得到关键字最小(或最大)的记录; 输出堆顶的最小(最大)值后, 使剩余的 $n-1$ 个元素重复又建成一个堆, 则可得到 n 个元素的次小值; 重复执行, 得到一个有序序列的过程

6.AOE(边表示活动)网中路径长度最长的路径, 即完成工程的最短时间是开始顶点到完成顶点的最长路径长度(没有时间余量的步骤)

二、 简答题(共 4 小题, 每题 10 分, 总分 40 分)

1. 说明哈希表工作原理以及常见的哈希冲突处理方法
2. 说明快速排序的基本原理, 以及其优缺点
3. 说明顺序存储和链式存储的优缺点
4. 结合数据结构课程, 举例说明“空间换时间”

7.将记录的存储位置与它的关键字之间建立一个确定的关系 H , 使每个关键字和唯一的存储位置对应。在查找时, 只需要根据对应关系计算出给定的关键值 $H(K)$, 就可以得到记录的存储位置。(线性探测再散列)(二次探测再散列)(伪随机探测再散列)

三、 综合应用题(共 8 小题, 每题 10 分, 总分 80 分)

1. 一棵二叉树, 先序遍历顺序为: ABDCEFG; 后序遍历顺序为: DBEFGCA。画出所有满足上述条件的二叉树。

2.找一个记录, 以它的关键字作为“枢轴”, 凡其关键字小于枢轴的记录均移动至该记录之前, 反之, 凡关键字大于枢轴

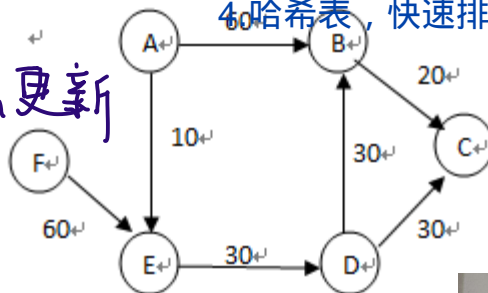
2. 在一份电文中共使用有 A,B,C,D,E,F 6 种字符, 他们出现的频率依次为 3,8,6,17,7,26。画出对应的编码哈夫曼树, 设计每个字符的哈夫曼编码。

轴的左右子序列执行相同操作

3. 已知带权有向图 G 如下图所示, 用 Dijkstra 迪杰斯特拉算法计算最短路径。S 为来源记录求解过程已求得最短路径的终点集合。用图表演示 A 删除操作需移动较多数据元素

3.顺序存储优点: 随机存取, 缺点: 插入删除操作需移动较多数据元素

4.哈希表, 快速排序, 多重链表



只能用最值去更新

4. 已知无向图 G 的邻接矩阵如图所示, 画出其邻接表。

	B	C	D	E	F	S
k=1	60			10		AE
k=2	60		40			AE D
k=3	60	70				AE D B
k=4		70				AE D B C
k=5						

i \ j	0	1	2	3	4
0	0	1	1	0	0
1	1	0	0	1	0
2	1	0	0	1	1
3	0	1	1	0	0
4	0	0	1	0	0

5. 对起始为空的平衡二叉树依次插入关键字序列 10, 20, 30, 60, 50。画出平衡二叉树的生成过程。

6. 写出以下代码的功能，并分析时间复杂度和空间复杂度。

```
void algorithm(Stack S)
{
    → int i, n, T[m];
    → n=1;
    → while(!StackEmpty(S)) Pop(S, T[n++]);
    → for(i=1; i<=n; i++) Push(S, T[i]);
}

#include<stdio.h>
int main()
{
    score temp;
    temp.sno=-1;
    temp.mark=-1;
    for(i=0; i<n; i++)
    {
        typedef struct {
            int sno; //学号
            int mark; //分数
        } score;
        temp.sno=LC[i].sno+LM[i].sno;
        temp.mark=LC[i].mark+LM[i].mark;
        printf("%d\n", temp.sno);
        printf("%d\n", temp.mark);
    }
}
```

通过数组逆置栈中内容，都是O(S)

7. 已知顺序表 LC[m] 和 LM[n] 分别存储有语文成绩和数学成绩，且以学号从低到高排序，个别缺考学生在成绩表中未出现。写时间复杂度最优算法输出总分最高的人。已知成绩单结构：

```
{
    typedef struct {
        int sno; //学号
        int mark; //分数
    } score;
    score LC[m], LM[n]; //语文和数学成绩清单
}
```

8. 完成二叉树按层次遍历算法 void LevelOrder(BiTree T)。已知二叉链表存储结构：

```
typedef struct BiTNode {
```

```
TElemType data;
```

```
struct BiTNode *lchild, *rchild; //左右孩子指针
```

```
} BiTNode, *BiTree;
```

```
QElemType a;
```

```
if(T){
```

```
InitQueue(q);
```

```
EnQueue(q, T);
```

```
while(!QueueEmpty(q)){
```

```
DeQueue(q, a);
```

```
PRINTF();
```

```
if(a->lchild!=NULL)
```

```
EnQueue(q, a->lchild);
```

```
if(a->rchild!=NULL)
```

```
EnQueue(q, a->rchild);
```

```
}
```

```
printf("\n");
```

```
}
```

```
}
```

