

参考答案

(A 卷)

姓名

学号

专业班级

密

一、 简答题 (本题满分 20 分, 每小题 5 分)

1、 简述逻辑结构的四种基本关系 (集合结构、线性结构、树结构、图结构) 并画出他们的关系图。

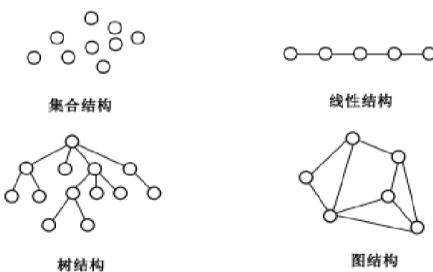
答案:

(1) 集合结构: 数据元素之间除了“属于同一集合”的关系外, 别无其他关系。

(2) 线性结构: 数据元素之间存在一对一的关系。

(3) 树结构: 数据元素之间存在一对多的关系。

(4) 图结构: 数据元素之间存在多对多的关系。



2、 简述什么是栈和队列, 并说明其各自特点。

队列 (Queue): 是限定只能在表的一端进行插入和在另一端进行删除操作的线性表;

栈 (Stack): 是限定只能在表的一端进行插入和删除操作的线性表。

栈: 先进后出 (First In Last Out) FILO

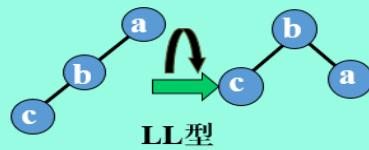
队列: 先进先出 (First In First Out) FIFO

3、 什么是平衡二叉树? 在构造平衡二叉树过程中, 会遇到失衡情况, 简述四种失衡情况 (LL、RR、LR、RL) 下如何进行平衡性调整?

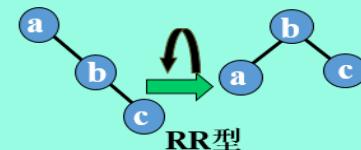
平衡二叉树的性质：

- 左子树和右子树的高度之差的绝对值小于等于 1
- 左子树和右子树也是平衡二叉树

1) LL平衡旋转:



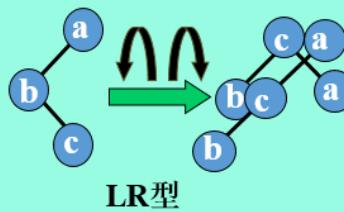
2) RR平衡旋转:



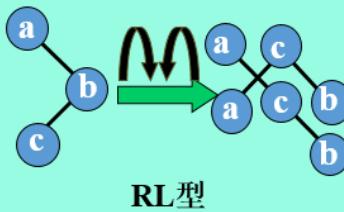
若在a的左子树的左子树上插入节点，使a的平衡因子从1增加至2，需要进行一次顺时针旋转。
(以b为旋转轴)

若在a的右子树的右子树上插入节点，使a的平衡因子从-1增加至-2，需要进行一次逆时针旋转。
(以b为旋转轴)

3) LR平衡旋转:



4) RL平衡旋转:



若在a的左子树的右子树上插入节点，使a的平衡因子从1增加至2，需要先进行逆时针旋转，再顺时针旋转。
(以插入的节点c为旋转轴)

若在a的右子树的左子树上插入节点，使a的平衡因子从-1增加至-2，需要先进行顺时针旋转，再逆时针旋转。
(以插入的节点c为旋转轴)

4、简述排序的核心思想？排序中什么是稳定的，什么是不稳定的？

答：排序就是使一组记录，按照记录中某个或某些关键字的大小，递增或递减排列起来的操作，排序的最终目的是实现快速查找。经过排序后这些具有相同关键字的记录之间的相对次序保持不变，该排序方法是稳定的，若关键字记录之间的相对次序发生变化，则称这种方法是不稳定的。

姓名

学号

专业班级

学院、系

线
卷

二、分析题（本题满分 20 分，每小题 10 分）

1、试证明二叉树的性质：对于任何一棵二叉树 T，如果其终端节点数为 n_0 ，度为 2 的节点数为 n_2 ，则 $n_0=n_2+1$ 。

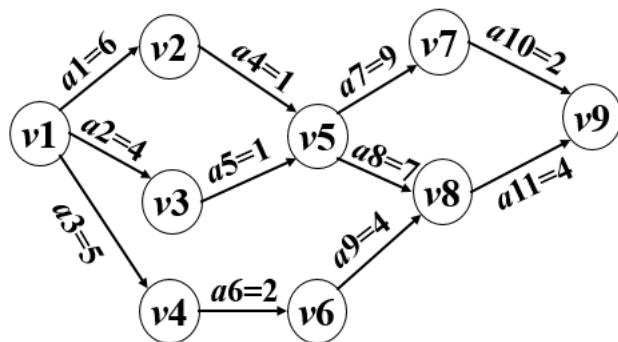
证明：若度为 1 的结点有 n_1 个，总结点个数为 n ，总边数为 e ，则根据二叉树的定义，
 $n = n_0 + n_1 + n_2$

$$e = 2n_2 + n_1 = n - 1 \text{ (除了根节点, 每个节点对应一条边)}$$

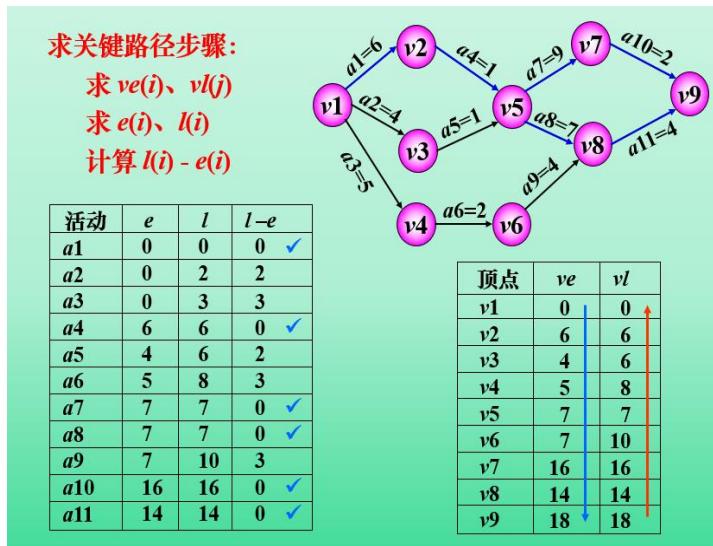
$$\text{因此, 有 } 2n_2 + n_1 = n_0 + n_1 + n_2 - 1$$

$$n_2 = n_0 - 1 \Rightarrow n_0 = n_2 + 1$$

2、试分析下图所示的有向网中哪些活动为关键活动，并画出关键路径。

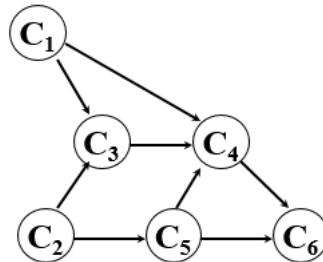


答案



三、应用题（本题满分 40 分，每小题 10 分）

1、写出下图所示的有向图的所有拓扑序列。



答案：C1 C2 C3 C5 C4 C6

C1 C2 C5 C3 C4 C6

C2 C1 C3 C5 C4 C6

C2 C1 C5 C3 C4 C6

C2 C5 C1 C3 C4 C6

2、请用 Prim 或 Kruskal 算法构造下图的最小生成树。

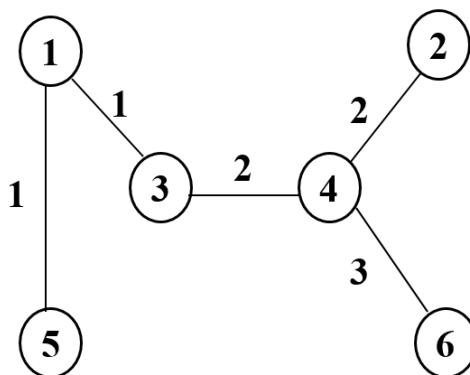
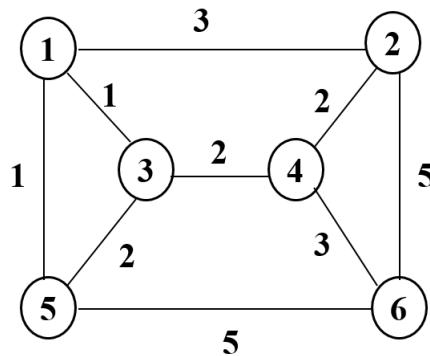
线

姓名

学号

专业班级

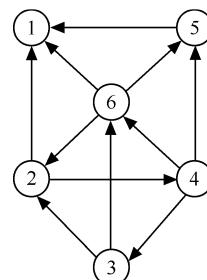
学院、系



答案：

3、已知如右图所示的有向图，请给出：

- ① 每个顶点的入度和出度；
- ② 邻接矩阵；
- ③ 邻接表；

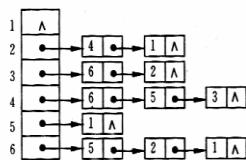


(1) (2) 邻接矩阵

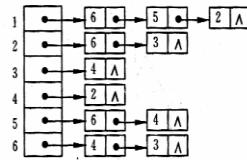
顶点	1	2	3	4	5	6
入度	3	2	1	1	2	2
出度	0	2	2	3	1	3

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

(3) 邻接表



(4) 逆邻接表



4、设哈希函数 $H(K) = 3 K \bmod 11$ ，哈希地址空间为 $0 \sim 10$ ，对关键字序列 $(32, 13, 49, 24, 38, 21, 4, 12)$ ，按线性探测法解决冲突的方法构造哈希表，求出等概率下查找成功时和查找失败时的平均查找长度 ASL_{succ} 和 ASL_{unsucc} 。

答案：

散列地址	0	1	2	3	4	5	6	7	8	9	10
关键字		4		12	49	38	13	24	32	21	
比较次数		1		1	1	2	1	2	1	2	

$$ASL_{succ} = (1+1+1+2+1+2+1+2)/8=11/8$$

$$ASL_{unsucc} = (1+2+1+8+7+6+5+4+3+2+1)/11=40/11$$

四、算法题（本题满分 20 分，每题 20 分）

1、简要说明折半插入排序算法的算法步骤，并用 C 语言实现其算法描述。

```

void BinsertSort(SqList &L)

{
    // 对顺序表 L 做折半插入排序

    for (i=2; i < =L.length; ++i)

    {

        L.r[0]=L.r[i];           // 将待插入的记录暂存到监视哨中

        low=1;high=i-1;          // 置查找区间初值

        while (low<=high) // 在 r[low .. high] 中折半查找插入的位置
  
```

姓名

学号

专业班级

密

学院、系

```

    {
        m=(low+high)/2; // 折半
        if (L.r[0].key<L.r[m].key) high=m-1; // 插入点在前一子表
        else low=m+1; // 插入点在后一子表
    }

    for (j=i-1;j>=high+1; --j) L.r[j+1]=L.r[j]; //记录后移
    L.r[high+1]=L.r[0]; // 将 r[0]即原 r[i], 插入到正确位置
}
// for
}

```

2 设计判断两个二叉树是否相同的算法, 写出算法思想和 C 语言描述。

```

typedef struct node {datatype data; struct node *lchild,*rchild;} bitree;

int judgebitree(bitree *bt1,bitree *bt2)

{
    if (bt1==0 && bt2==0) return(1);

    else if (bt1==0 || bt2==0 || bt1->data!=bt2->data) return(0);

    else
        return(judgebitree(bt1->lchild,bt2->lchild)*judgebitree(bt1->rchild,bt2->rchild));
}

```