上海工程技术大学

（勤奋、求是、创新、奉献）

2019～ 2020 学年第 2 学期考试试卷

主考教师：施一萍，张娟

学院 电子电气工程学院 班级 _____ 姓名 _____学号 _____

《 算法与数据结构》课程试卷 A 参考答案

（本卷考试时间 120 分钟）

题号	一	二	三	四	五	六	七	八	九	十	总得分
题分	20	20	32	10	8	10					100
得分											

一、 选择题（本题共 10 小题，每小题 2 分，共 10 分）

1、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

2、栈和队列的共同特点是 (D)。

A. 都是先进先出 B. 都是后进先出

C. 没有共同点 D. 只允许在端点操作元素

3、树最适合表示(C)。

A.有序的数据元素 B.无序的数据元素

C.元素间具有层次关系的数据 D.元素间无联系的数据

4、 设有序表中有 1000 个元素，则用折半查找元素 X 最多需要比较 (**B**)次。

A．1 B. 10 C. 20 D. 500

5、平衡二叉树上一个结点的平衡因子的绝对值最大为 (**B**)。

A. 0 B. 1 C. 2 D. 3

6、图的深度优先遍历的思想实际上是二叉树(A)遍历的推广。

A. 先根 B. 中根 C. 后根 D. 层次

7、在有向图中，所有结点的入度之和等于所有结点出度之和的(B)倍。

A. 1/2 B. 1 C. 2 D. 4

8、解决哈希表冲突的方法不包括(C)。

A. 线性探测再散列 B. 链地址法

C. 除留余数法　　D. 双重散列法

9、快速排序是一种(　C　)。

A. 插入排序　B. 选择排序　C. 交换排序　D. 归并排序

10、在一棵具有 5 层的完全二叉树中，结点数最多为(　**A**　)。

A.31　B.32　C.33　D.16

二、填空题（本题共 10 空，每空 2 分，共 20 分）

1、线性结构中元素之间存在（一对一）的关系，树形结构中元素之间存在(　一对多　)关系，图状结构中元素之间存在(　多对多　)关系。

2、一个算法的时间复杂度为$(n^2+n\log_2 n+14)$，其数量级表示为(　O（n^2）　)。

3、向一个长度为 n 的顺序表的第 i 个元素（$1\le i\le n+1$）之前插入一个元素时，需移动(　$n-i+1$　)个元素。

4、对一棵完全二叉树从 1 开始按从上到下、从左到右的顺序对结点编号，那么，对编号为 i 的结点，它的左孩子（如果有的话）结点编号为(　2i　)，它的右孩子（如果有的话）结点编号为(　$2i+1$　)，它的双亲结点（如果有的话）的编号为(　i/2　)。

5、对一个二叉排序树进行(　中根　)遍历时，得到的结点序列是一个有序序列。

6、基数排序就是一种借助（ 多关键字排序 ）的思想来实现“单关键字排序”的算法。

三、算法理解题（本题共 4 小题，每小题 8 分，共 32 分）

1、给出如图 1 所示的二叉树，求出先序、中序、后序和层序遍历序列。

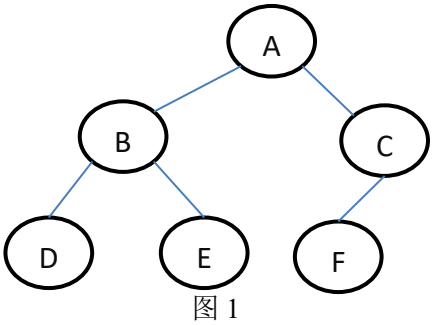


图 1

答：先序遍历：ABDECF　　后序遍历：DEBFCA

中序遍历：DBEAFC　　层序遍历：ABCDEF

2、请画出如图 2 所示有向图的邻接矩阵和邻接表。

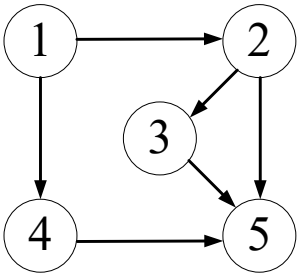
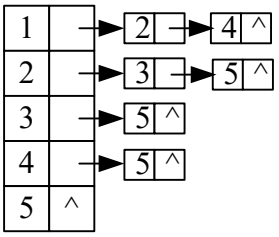


图 2

答：

邻接矩阵为：
$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

邻接表为：



3、 设哈希函数 H(k) = k % 13，设关键字序列为{22,12,24,6,45,7,8,13,21}，散列表长 m=13，使用线性探测再散列处理冲突，构造哈希表，并给出查找成功时的平均查找长度。

答：H(22)= 22 % 13= 9,H(12)= 12% 13=12,H(24)= 24% 13=11,H(6)=6 % 13=6,

H(45)=45 % 13=6，冲突，所以，由线性探测，45 存储在编号 7 的位置。

H(7)=7 % 13=7，冲突，所以，由线性探测，7 存储在编号 8 的位置。

H(8)=8 % 13=8，冲突，所以，由线性探测，8 存储在编号 10 的位置。

H(13)=13 % 13=0

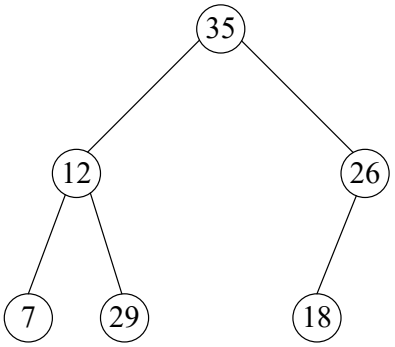
H(21)=21%13=8，冲突，所以，由线性探测，21 存储在编号 1 的位置。

最终得到的 Hash 表为：

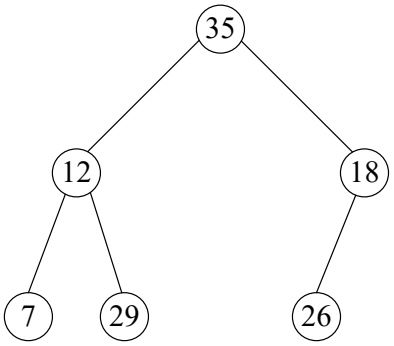
地址	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字	13	21					6	45	7	22	8	24	12
探测次数	1	7					1	2	2	1	3	1	1

ASL = （1+7+1+2+2+1+3+1+1）/ 9 = 2.1

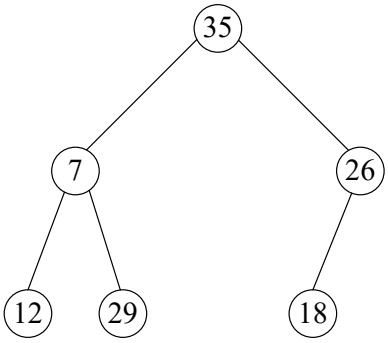
4、 假设待排序的关键字序列为{35，12，26，7，29，18}，试写出使用堆排序建小顶堆的过程和第一次输出堆顶调堆后的结果。



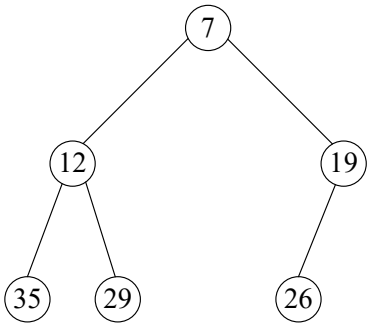
无序序列



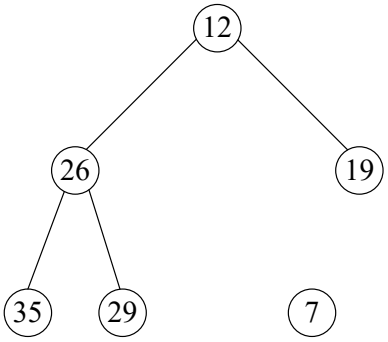
26 与 18 筛选



12 与 7、29 筛选



35 与 7、18 筛选得初始堆



输出 7 调整后的新堆

四、程序填空题：（本题共 5 空，每空 2 分，共 10 分）

已知单链表的各结点数据为互不相等的整数。下面是创建单链表，并求单链表中元素值最大

的结点序号的算法。请将该算法填写完整。

```
typedef int ElemType;
typedef struct node
    {
       ElemType    data;
       struct node    *next;
    }LNode,*LinkList;
void CreateList(LinkList &h,int n)
       {
           LNode    *s,*r; int i;
           h=( LNode *)malloc(sizeof(LNode));
           r=h;
           for (i=0;i<n;i++)
               {
                   s=( LNode *)malloc(sizeof(LNode));
                   scanf("%d",&s->data);
                   r->next=   (1)    ;
                     r=s;
               }
           r->next=    (2)        ;
       }
int MaxNode(LinkList sl)
      {     int j,k;
           LNode *p,*q;
           if (sl->next ==NULL) return 0;
           q=sl->next;
           p=q->next;
           k=1;      j=   (3)      ;
           while (p!=NULL)
               { if (p->data>q->data) {q=p;k=      (4)          ;}
                           p=   (5)      ;    j++;
               }
           return k;
       }
```

答：（1）s　(2) null　(3)2　(4)j　(5) p->next

五、算法设计题：（本题 8 分）

已知二叉树的结点结构如下，设计算法int LeafCount(BTree bt)，求二叉树BT的叶子结点数。

```
typedef struct   node
{
ElemType  data；
struct node    *lchild，*rchild；
} Bnode, *BTree;
```

```
int LeafCount(BTree bt)
  {   int num1,num2;
      if (bt！=NULL)
        { if (bt->lchild==NULL && bt->rchild==NULL)
              return 1;
          else
              {   num1=LeafCount(bt->lchild);
                  num2=LeafCount(bt->rchild);
                  return(num1+num2);
              }
      return 0;
    }
```

六．分析题（本题10分）

已知一个公路网，该网内有n个城市，以及这些城市之间相连的公路，将城市用结点表示，城市间的公路用边表示。要求从某一城市出发，对公路网内n个城市都访问一次。试根据题目的含义分析解决问题采用的数据结构是什么？采用什么算法来完成任务?算法要求用文字描述。

答：采用的数据结构为图；

用深度优先搜索来完成任务。算法如下：

从图中某个顶点v出发，访问此顶点，然后依次从v的未被访问的邻接点出发深度优先遍历图，直至图中所有和v有路径相通的顶点都被访问到；若此时图中尚有顶点未被访问，则另选图中一个未曾被访问的顶点作起始点，重复上述过程，直至图中所有顶点都被访问到为止。

或用广度优先搜索来完成。算法如下：

从图中某顶点v出发，在访问了v之后依次访问v的各个未曾访问过的邻接点，然后分别从这些邻接点出发依次访问它们的邻接点，并使“先被访问的顶点的邻接点”先于“后被访问的顶点的邻接点”被访问，直至图中所有已被访问的顶点的邻接点都被访问到。若此时图中尚有顶点未被访问，则另选图中一个未曾被访问的顶点作起始点，重复上述过程，直至图中所有顶点都被访问到为止。