

任课教师:

学号:

姓名:

班级:

装订线

装订线

装订线

## 西安电子科技大学

考试时间 120 分钟

## 试 题

题号	一	二	三	四	总分
分数					

1. 考试形式: 闭卷; 2. 本试卷共四大题, 满分 100 分;  
3. 考试日期: 2021 年 12 月 30 日; (答题内容请写在装订线外)

## 一、单选题 (10 小题, 20 分)

1. 在具有  $n$  个元素结点的有序单链表中插入一个新的结点并保持该链表有序的运算时间复杂度是 ( B )。

- A.  $O(1)$       B.  $O(n)$       C.  $O(n^2)$       D.  $O(n\log n)$

2. 栈和队列都是 ( C )。

- A. 顺序存储的线性结构      B. 链式存储的非线性结构  
C. 限制存取点的线性结构      D. 限制存取点的非线性结构

3. 一个循环队列的最大容量为  $\text{maxSize}$ , 队尾指针是  $\text{rear}$ , 队头指针是  $\text{front}$ , 则队空的条件是 ( A )。

- A.  $\text{rear} == \text{front}$       B.  $(\text{rear} + 1) \% \text{maxSize} == \text{front}$   
C.  $\text{rear} + 1 == \text{front}$       D.  $(\text{rear} - 1) \% \text{maxSize} == \text{front}$

4. 模式串 'abbacabcb' 的 next 函数值为 ( D )。

- A. 011232312      B. 012111232  
C. 012345678      D. 011121231

5. 设二维数组  $a[60][70]$  的首地址 (即  $a[0][0]$  的地址) 为 2048, 每个元素占 2 个存储单元, 若以列序为主序顺序存储, 则元素  $a[31][57]$  的存储地址为 ( C )。

- A.  $2048 + (31 * 70 + 57) * 2$       B.  $2048 + (31 * 60 + 57) * 2$   
C.  $2048 + (57 * 60 + 31) * 2$       D.  $2048 + (57 * 70 + 31) * 2$

6. 设广义表  $A = (a, b, (c, d), (e, (f, g)))$ , 则  $\text{Head}(\text{Tail}(\text{Head}(\text{Tail}(\text{Tail}(A))))$  的值为 ( B )。

- A. (f, g)      B. d      C. c      D. (d)

7. 已知一棵度为 3 的树有 2 个度为 1 的结点, 3 个度为 2 的结点, 4 个度为 3 的结点, 则该树有 ( C ) 个叶子结点。

- A. 10      B. 11      C. 12      D. 13

8. 一个无向连通图的生成树是含有该图全部顶点的 ( A )。

A. 极小连通子图

B. 极小子图

C. 极大连通子图

D. 极大子图

9. 下面算法中, ( B ) 最适用于求解一个稀疏图的最小生成树。

A. 普利姆(Prim)算法

B. 克鲁斯卡尔(Kruskal)算法

C. 迪杰斯特拉(Dijkstra)算法

D. 弗洛伊德(Floyd)算法

10. 对于关键字序列{45,72,23,31,94,53,16,67}, 以第一个元素为基准(枢轴), 进行快速排序的第一趟划分结果为 ( D )。

A. 16, 23, 31, 45, 53, 67, 72, 94

B. 16, 23, 31, 45, 72, 94, 53, 67

C. 16, 23, 31, 45, 67, 53, 72, 94

D. 16, 31, 23, 45, 94, 53, 72, 67

## 二、简答题 (7 小题, 52 分)

1. (6 分) 若线性表的长度需动态变化且频繁地进行插入操作, 应采用哪种存储表示, 为什么? 如果线性表长度基本不变, 且很少进行插入和删除, 但要求快速存取表中的元素, 这时应采用哪种存储表示, 为什么?

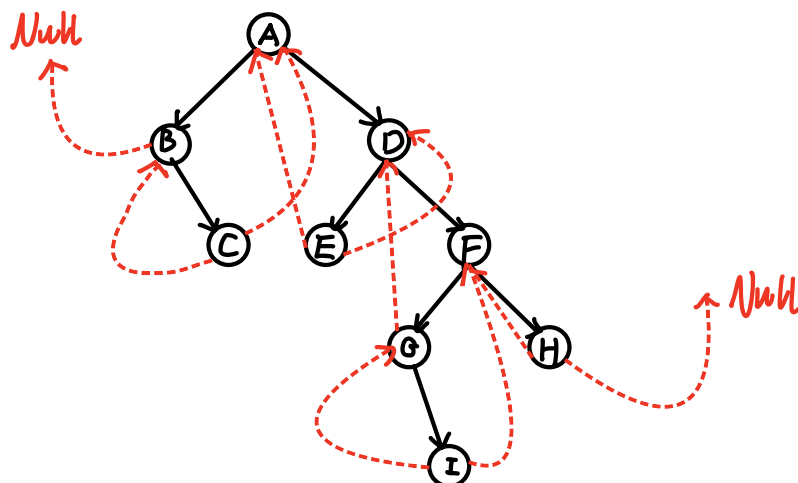
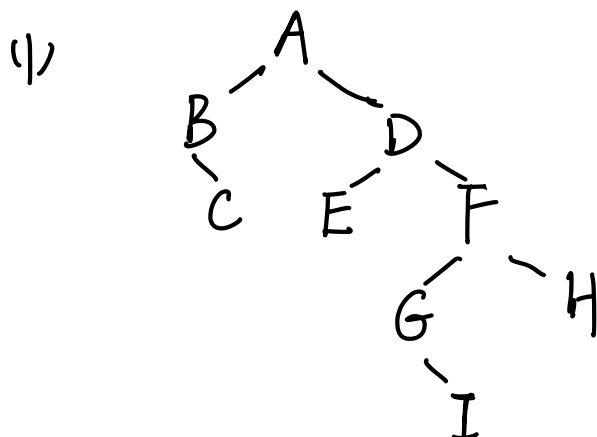
① 链表 链表插入或删除时只需修改指针, 无需移动元素

② 顺序表 顺序表可以随机存取

2. (6 分) 已知某二叉树的先序遍历序列和中序遍历序列分别为 ABCDEFGIH 和 BCAEDGIFH。

(1) 画出该二叉树;

(2) 给出对应的中序线索二叉树。

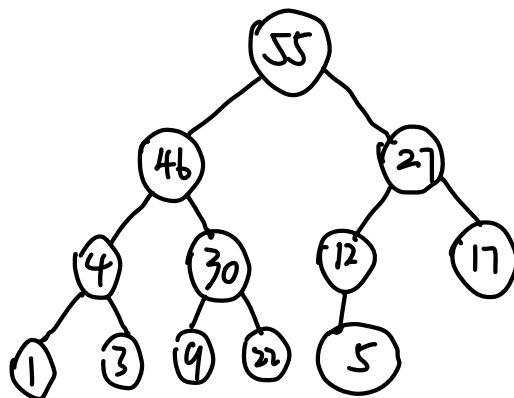
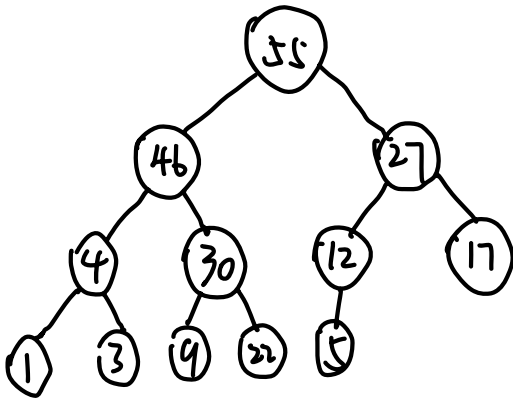


3. (6分) 对关键字序列(30, 22, 27, 4, 55, 12, 17, 1, 3, 9, 46, 5)按递增排列的要求进行排序:

(1) 假设增量  $d=5$ , 给出采用希尔排序方法的第一趟排序结果;

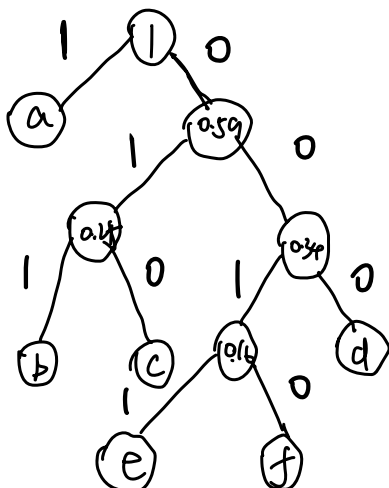
12 5 1 3 9 30 17 27 4 55 46 22

(2) 给出所构建的大顶堆(以完全二叉树表示最终的大顶堆), 以及将堆顶元素与序列末端元素互换后, 再次调整后所得的大顶堆(以完全二叉树表示)



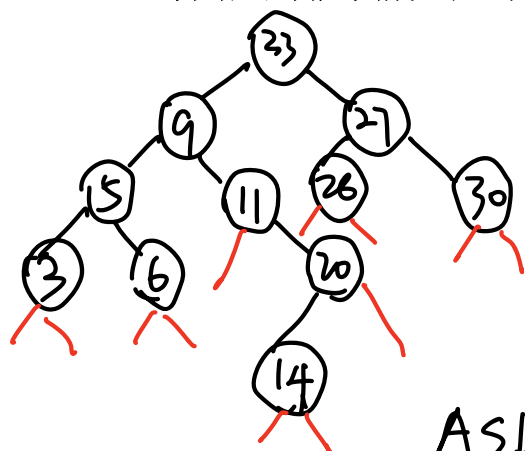
4. (7分) 假设某通信电文由字符集{a,b,c,d,e,f}中的字符构成, 已知这6个字符在电文中出现的概率(百分比)分别为{0.41, 0.11, 0.14, 0.18, 0.06, 0.10}, 现采用哈夫曼编码方案进行编码: (1) 给出所构造的哈夫曼编码树; (2) 给出每个字符的编码; (3) 求该编码树的带权路径长度 WPL.

e 0.06 f 0.10 b 0.11 c 0.14 d 0.18 a 0.41



a: 1  
b: 011  
c: 010  
d: 000  
e: 0011  
f: 0010

5. (8分) 设有关键字序列(23, 27, 9, 11, 5, 30, 6, 20, 14, 3, 26), 构造对应的二叉排序树, 并给出等概率情况下查找成功和不成功的平均查找长度。



ASL(success)

$$= \frac{1}{11} (1 + 2 \times 2 + 3 \times 4 + 3 \times 4 + 5)$$

$$= \frac{34}{11}$$

$$ASL(fail) = \frac{1}{12} (3 \times 5 + 4 \times 5 + 5 \times 2)$$

$$= \frac{45}{12}$$

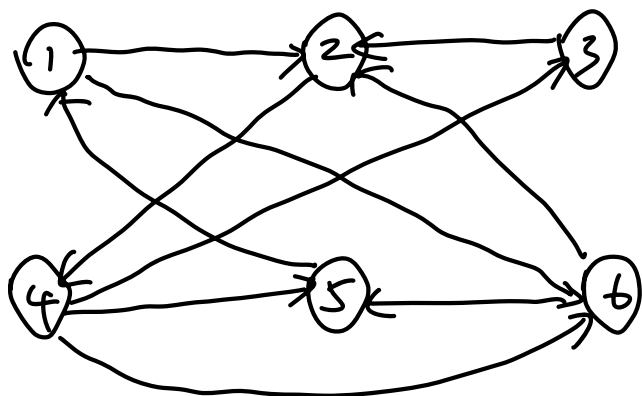
6. (7分) 设有关键字序列(34, 42, 72, 12, 30, 8, 49, 58, 76), 哈希函数为  $h(k) = k \% 7$ , 设哈希表长为 10, 画出用线性探测法处理冲突构造的哈希表, 并计算在等概率情况下查找成功的平均查找长度 (写出算式)。

hash	0	1	2	3	4	5	6	7	8	9
val	42	8	72	30	49	12	34	58	76	
查找次数	1	1	1	2	5	1	1	6	3	

$$ASL(success) = \frac{1}{9} (1 + 1 + 1 + 2 + 5 + 1 + 1 + 6 + 3)$$

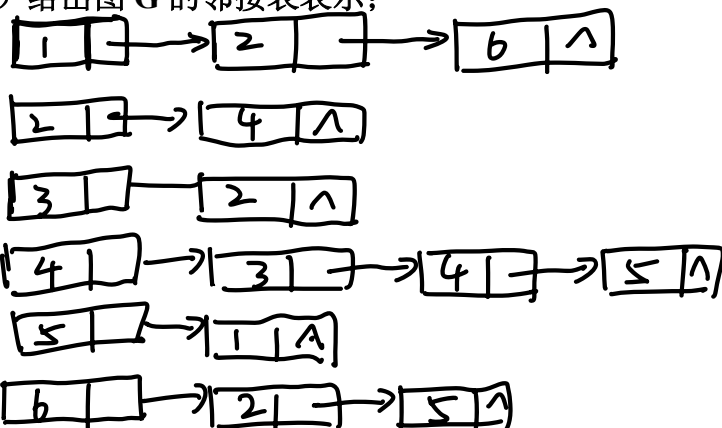
7. (12分) 有向图 G 的邻接矩阵如下所示:

(1) 画出图 G;



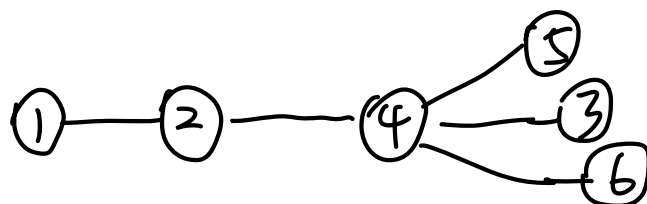
	1	2	3	4	5	6
1	0	1	0	0	0	1
2	0	0	0	1	0	0
3	0	1	0	0	0	0
4	0	0	1	0	1	1
5	1	0	0	0	0	0
6	0	1	0	0	1	0

(2) 给出图 G 的邻接表表示;



(3) 根据 G 的邻接矩阵, 给出从顶点 1 出发的深度优先遍历序列及生成树;

1 2 4 5 3 6



(4) 判断 G 是否强连通, 若否, 画出其最大强连通分量。

是

### 三、完善算法 (2 小题, 18 分)

1. 采用含头结点的单链表实现栈, 将栈顶设在表头, 请完善下面的入栈算法 **push** 和出栈算法 **pop**, 填写算法中空 (1) ~ (5) 处的代码。

```

typedef struct SNode{
    ElemType data;
    struct SNode *next;
}SNode, *LinkStack;

LinkStack initStack() { //创建空栈
    SNode *top_Ptr = (SNode *)malloc(sizeof(SNode));
    if (!top_Ptr) return NULL;
    top_Ptr->next = NULL;
    return top_Ptr;
}
  
```

```

bool push(LinkStack &top_Ptr, ElemType e)  { //入栈
    SNode *p = (SNode *)malloc(sizeof(SNode));
    if (!p) return false;
    p->data = e;
    ____ (1) ____;
    ____ (2) ____;
    return true;
}

```

```

bool pop(LinkStack &top_Ptr, ElemType &e){ //出栈
    if (____ (3) ____ ) return false;    //栈空
    SNode *p = ____ (4) ____;
    e = p->data;
    ____ (5) ____;
    free(p);
    return true;
}

```

- |   |
|---|
| <p>(1) <math>p \rightarrow next = top\_Ptr \rightarrow next</math></p> <p>(2) <math>top\_Ptr \rightarrow next = p</math></p> <p>(3) <math>top\_Ptr \rightarrow next == NULL</math></p> <p>(4) <math>top\_Ptr \rightarrow next</math></p> <p>(5) <math>top\_Ptr \rightarrow next = p \rightarrow next</math></p> |
|---|

2. 下面的算法 insertKey 在递增有序的顺序表 L 中插入一个新元素 x，并保持表 L 的递增有序性。请填写算法中空 (1) ~ (3) 处的代码，并回答问题。

```
typedef struct{
    ElemType *r; //存储空间首地址
    int size;    //表长(即当前元素个数)
    int capacity; //存储容量(及最多能存储的元素个数)
}SqList;

bool insertKey(SqList &L, ElemType x) {
    /*递增有序顺序表 L 的元素存储在 L.r[1]~L.r[L.size]中，插入新元素 x 并保持 L 的有序性*/

    if (L.size>=L.capacity) {
        newaddr = (ElemType *)realloc(L.r,
                                       (L.capacity + L.capacity/2)*sizeof(ElemType));
        if (!newaddr) return false;
        L.r = newaddr;    L.capacity += L.capacity/2;
    }

    L.r[0] = x;    //L.r[0]为监视哨
    for(i=L.size; (1); i--)
        L.r[i+1] = L.r[i];    //@

    (2);
    (3);
    return true;
}
```

- (1)  $i \geq 1 \ \&\& \ L.r[i] > x$
- (2)  $L.r[i+1] = x$
- (3)  $L.size += 1$

问题:

假设上面算法中 L.size 等于 n，若要加入的新元素 x 小于表中所有元素，给出这种情况下 “L.r[i+1] = L.r[i];” (注释为@) 的语句频度。

#### 四、算法设计（1 小题，10 分）

设二叉树采用三叉链表表示（即结点中有左孩子指针 `lchild`、右孩子指针 `rchild` 和父结点指针 `father`），目前链表中每个结点的 `lchild`、`rchild` 已设置，设根结点的指针为 `root`，请设计一个算法，将每个结点的父结点指针 `father` 进行正确设置，需给出结点的类型定义。

```
typedef struct Node {  
    struct Node* lchild;  
    struct Node* rchild;  
    struct Node* father;  
} Node;
```

```
void setf(Node* Father, Node* root) {  
    if (root == NULL) return;  
    root->father = Father;  
    setf(root, root->lchild);  
    setf(root, root->rchild);  
}
```

装

订

线