考试科目名称 数据结构和算法 （期中试卷）

2023—2024 学年第一学期 教师 姜远 考试方式：闭卷

系（专业） 年级 班级

学号 姓名 成绩

题号	一	二	三	四	总分
分数					

得分 **一、单选题（每小题 2 分，本题满分 20 分）**

一、选择题。(每小题 2 分，共 20 分)

1. 以下算法的时间复杂度为（ ）。

```
int i = 1, j = 1;
while ( i+j <= n ) {
        if ( i > j )    j++;
        else            i=i*2;
}
```

A. O(n) B. $O(n^2)$ C. $O(n^{1/2})$ D. $O(\log_2 n)$

答案：A

2. 在双向链表指针 p 的结点前插入一个指针 q 的结点操作是（ ）。

A. p->Llink=q; q->Rlink=p; p->Llink->Rlink=q; q->Llink=q;

B. p->Llink=q; p->Llink->Rlink=q; q->Rlink=p; q->Llink=p->Llink;

C. q->Rlink=p; q->Llink=p->Llink; p->Llink->Rlink=q; p->Llink=q;

D. q->Llink=p->Llink; q->Rlink=q; p->Llink=q; p->Llink=q;

答案：C

3. 循环队列 A[0..m-1]存放其元素值，用 front 和 rear 分别表示队头和队尾，请问若 rear<front，队列中的元素个数是（ ）。

A. rear-front+m B. (rear-front -1+m) % m

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

答案：A

4. 一棵右子树为空的二叉树在后序线索化后，其中值为 NULL 的指针域的个数是（ ）。

A. 不确定 B. 0 C. 1 D. 2

答案：D

5. 某二叉树的前序遍历结点访问顺序是 ALBECDWX，中序遍历的结点访问顺序是

BLEACWXD，则其后序遍历的结点访问顺序是（　　）。

A. BELXWDCA

B. BEXLDWCA

C. LACDBEWX

D. LBEAWXDC

答案：A

6. 由分别带权为 9、2、5、7 的四个叶子结点构成一棵哈夫曼树，该树的带权路径长度为（　　）。

A.23

B.37

C.44

D.46

答案：C

7. 四叉树中，度为 1、2、3、4 的结点个数分别为 4、2、1、1。则树中有（　　）个叶子结点。

A.4

B.6

C.8

D.10

答案：C

8. 下面关于哈夫曼树的叙述中，错误的是（　　）。

A. 用 n 个结点构造的哈夫曼树是唯一的

B. 哈夫曼树中只有度为 0 和度为 2 的结点

C. 树中两个权值最小的结点是兄弟结点

D. 同一结点集构造的二叉树中，哈夫曼树的带权路径长度最小

答案：A

9. 如果采用树的双亲表示法作为树的存储结构，操作 Find (int x)实现对 x 结点所在树根的查找，则完成一次 Find 操作的最坏情况下的时间复杂度为（　　）。

```
int UFSets :: Find ( int x ) {
    if ( parent [x] < 0 ) return x;
   else return Find ( parent [x] );
}
```

A. O(1)

B. O(n)

C. $O(n\log_2 n)$

D. $O(\log_2 n)$

答案：B

10. 若对 n 阶对称矩阵 A 以行优先方式将其下三角形的元素(包括主对角线上所有元素)依次存放于一维数组 B 中，则在 B 中确定 A_{ij}（0<=i<j<=n-1）的位置为(　　)。（数组下标都从 0 开始）

A. i*(i-1)/2+j　　B. j*(j-1)/2+i　　C. i*(i+1)/2+j　　D. j*(j+1)/2+i

答案：D

得分	

二、填空题。（每空 2 分，共 20 分）

1. 利用 head 与 tail 操作从广义表 L=((a,b,c)，(d，(e，f，g)))中取出原子项 e 的运算是_____。

答案：head (head(tail(head(tail(L)))))

2. （根结点的层次为 1）已知完全二叉树的第 7 层有 20 个结点，则整个完全二叉树的叶子结点数是_____。

答案：42

3. 若一个森林以广义表的形式表示为（A（B（E，K），F（C，H，I）），D（J），L（M（N））），则将该森林转换为二叉树，转换后二叉树的高度为 _____（根结点的层次为 1）。

答案：6

4. 一棵完全二叉树上有 2021 个结点，其中叶子结点的个数是_____。

答案：1011

5. 一个有序顺序表有 155 个元素，采用顺序搜索法查表，成功时平均搜索长度为 _____。

答案：78

6．后缀表达式 A B * C + D E - F / * 对应的中缀表达式为_____，前缀表达式是_____ 。

答案：（A*B+C）* (D-E) / F　　和　　* + * A B C / - D E F

7. 设有一个顺序栈 A，元素 a1, a2, a3, a4, a5，依次进栈，如果 5 个元素的出栈顺序为 a2, a4, a3, a1, a5，则顺序栈的容量至少为_____。

答案：　3

8. 广义表 L（((b,c) ,d) ,(a),((a),((b,c),d)),e,( )）的深度为_____。

答案：4

9. 有一个 20*20 的稀疏矩阵（元素类型为整型），非零元素有 10 个，设每个整型数占 4 字节，则用三元组表示该矩阵时，所需的字节数是_____。

答案：120

得分	

三、解答题：（每小题 5 分，共 30 分）

1. 算法 Print 及所引用的数组 A 的值如下，写出调用 Print(0)的运行结果（其中 n = 14）。

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	B	C	D	E	F	G	0	0	H	0	I	J	K	L

```
void    print(int w);
    {
        if ( (w<=n) && (A[w]!='0') )
          {         Print(2*w+1);
                    Print(2*w+2);
                    cout<<A[w]<<' ';
           }
    }
```

答案： D H E B I J F K L G C A

2. 已知输入序列为 N、O、P、Q 要放入栈中进行存取，在输入过程中随时可以进行入栈和出栈。请写出所有可能的输出序列。

答案：共 14 种输出序列，

QPON / PQON / OQPN / OPQN / PONQ / OPNQ

ONPQ / ONQP / POQN / NPOQ / NOPQ / NOQP

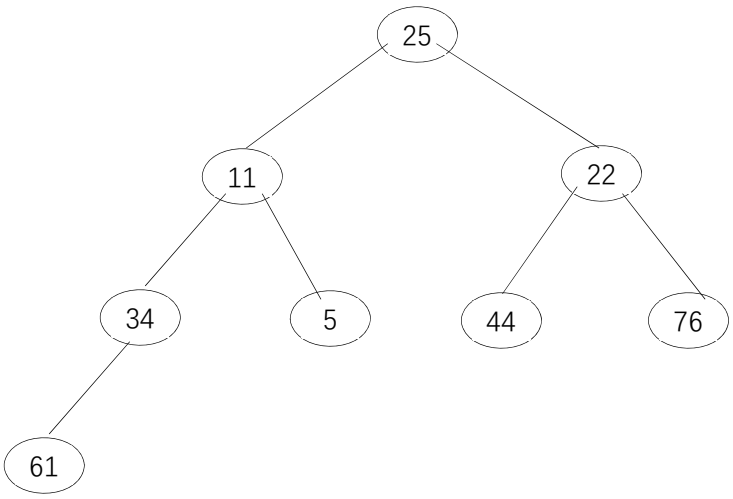
NPQO / NQPO

3. 用下面数据调整成最大堆。要求给出调整每个关键码的起始和结束的图。

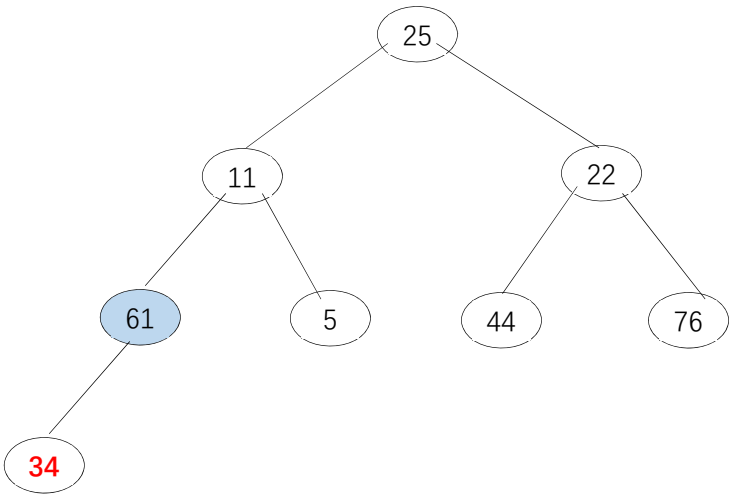
（25 11 22 34 5 44 76 61 ）

答案：

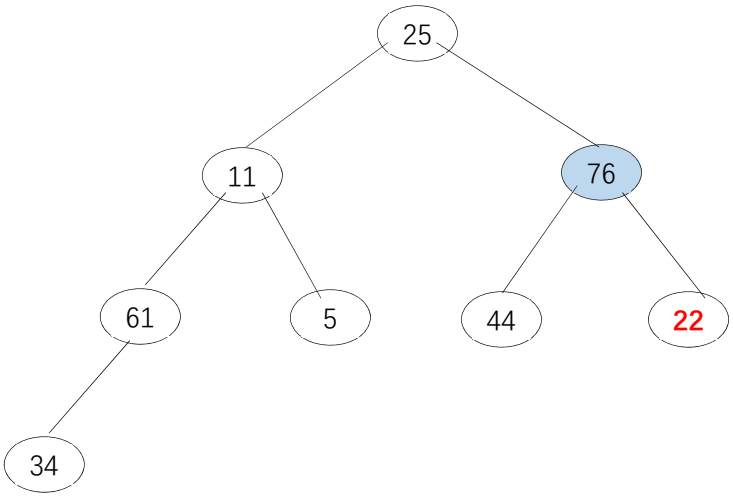
初始：



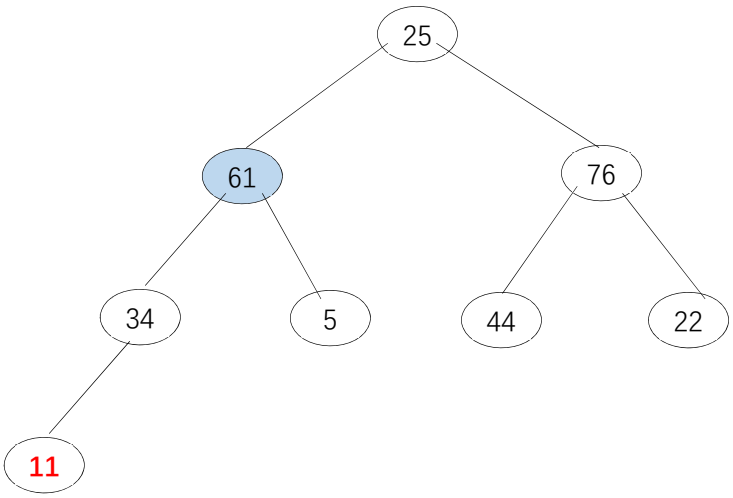
(1) 调整 34



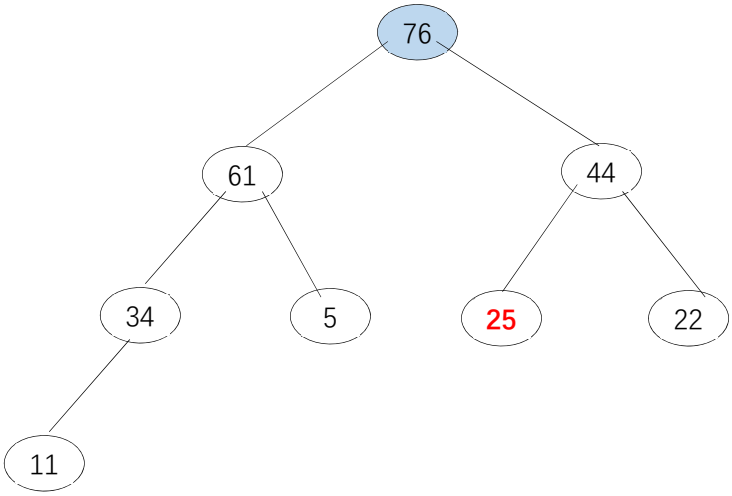
(2) 调整 22



(3) 调整 11



（4） 调整 25



4. 设有一个三对角矩阵（a_{ij}）$_{n\times n}$，将其三条对角线上的元素逐行的存放在数组 B[ ]中，使得 B[k]= a_{ij}，求：用 i,j 表示 k 的下标变换公式。（数组下标从 0 开始）

$$\begin{bmatrix} a_{11} & a_{12} & & & & & \\ a_{21} & a_{22} & a_{23} & & & & \\ & a_{32} & a_{33} & a_{34} & & & \\ & & \cdots & \cdots & \cdots & & \\ & & & \cdots & \cdots & \cdots & \\ & & & & a_{n-1n-2} & a_{n-1n-1} & a_{n-1n} \\ & & & & & a_{nn-1} & a_{nn} \end{bmatrix}$$

答案：(和书上不同的是，该矩阵从 1 开始标下标)

K = (i-1)*3-1+(j-i+1) = 2*i + j - 3

5. 以字符集合{ F, I, B, S, U }组成一份报文，各个字符出现的频率分别是{ 3, 4, 2, 7, 8 }，请对组成报文的字符进行 Huffman 编码，并计算该报文的编码总长度。

答案：（答案不唯一）

F: 0 0 0

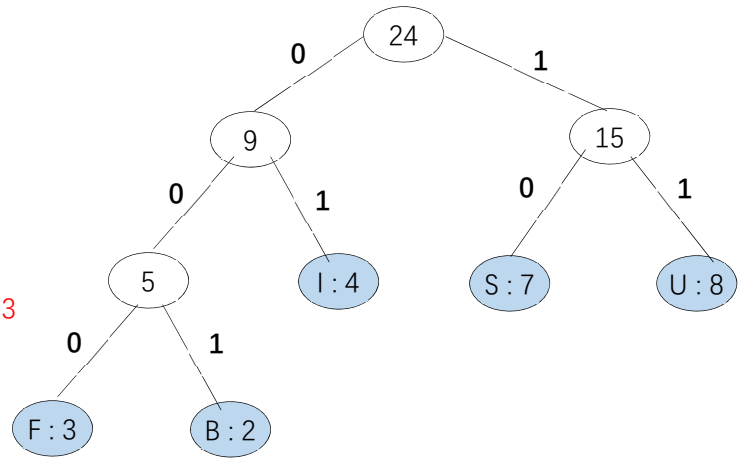
B: 0 0 1

I: 0 1

S: 1 0

U: 1 1

编码总长度：（3+2）*3 + （4 +7+8）*2 = 53



6. 设二叉树 T 的静态链表存储结构如下:

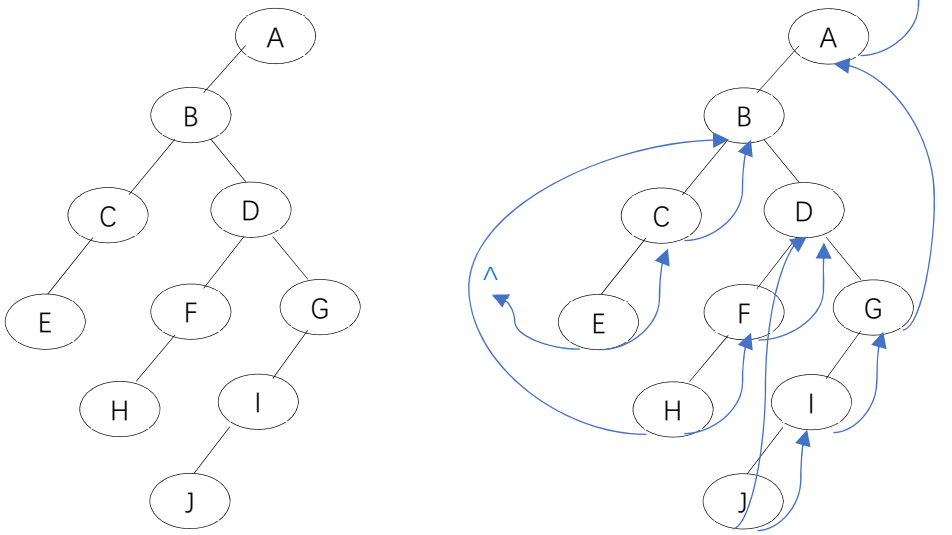
其中，Lchild,Rchild 分别为结点的左、右孩子指针域,data 为结点的数据域。

	1	2	3	4	5	6	7	8	9	10
Lchild	0	0	2	3	7	5	8	0	10	1
Data	J	H	F	D	B	A	C	E	G	I
Rchild	0	0	0	9	4	0	0	0	0	0

（1）请画出该二叉树的结构。

（2）请画出该二叉树的中序线索树。

答案:



得分	

四、算法填空题（本题满分 30 分）

1. 给定 pushed 和 popped 两个序列，每个序列中的值都不重复。以下算法验证对一个空栈**按顺序**压栈 pushed 中的元素或弹出栈中元素，是否能够得到 popped 序列。例如: pushed = [1,2,3,4,5], popped = [4,5,3,2,1]，此时验证结果为 true。因为可以对一个空栈执行: push(1), push(2), push(3), push(4), pop():4, push(5), pop():5, pop():3, pop():2, pop():1，来得到 popped 序列。请填写代码中空缺的部分，使得代码正确实现栈序列验证的功能。（每空 3 分，共 12 分）

```
#define ADefaultSize 100;
bool checkStackSequences(SeqList & pushed, SeqList & popped)
{
    Int pushedSize = pushed.Length( );
    if (pushedSize == 0)
```

```
        return true;
    int *a, top = -1, n = 0;     //利用数组 a 模拟栈
    a = new int[pushSize];
    for (int i = 0; i < pushedSize; i++)
    {
        if (top == -1 || pushed[i] != popped[n])
        {
            ___(1)_____________________________;
        }
        else
        {
            n++;
        }
        while (____(2)_____________________________)
        {    // 循环出栈

            ______(3)_____________________________;
            n++;
        }
    }
    if (___(4)_____________________________)
        return TRUE;
    else
        return FALSE;
}
```

答案:

1. a[++top] = pushed[i]

2. top != -1 && a[top] == popped[n]

3. top--

4. top == -1 || n>= pushedSize

3. 已知二叉树在二叉链表存储下的根结点为 T（结点结构如下图），设计**非递归算法**将该二叉树以完全二叉树的顺序存储方式存放于数组 A 中。（每空 3 分，共 18 分）

LeftChild	Data	RightChild

```
#define DefaultSize 100
int main( )
{
    static int A[DefaultSize];
    for (i=0; i<DefaultSize; i++) A[i] = -1;
    BinTreeNode *T;
    ifstream in;
    in.open("infile");                          //打开文件 infile 用于建二叉树
    CreatBinTree(in, T);
    Trans(T, A, 0);

}

void Trans(BinaryTreeNode * & SubTree; int A[ ]; int loc )
{
     Stack S;
     S.makeEmpty( );
     while (  (5)                                    ){
          while (SubTree!=NULL){
               A[loc] = (6)                    ;
               S. push(SubTree, loc);
               SubTree = SubTree ->leftchild;
               (7)                                  ;
          }
          if   (!S.IsEmpty( )){
               (8)                                  ;
               (9)                                  ;
               (10)                                 ;
          }
     }
}
```

参考答案:

5. (SubTree!=NULL) || (!S.IsEmpty( ))
6. SubTree->Data

7. loc = 2*loc+1
8. S. pop(SubTree, loc)
9. SubTree = SubTree ->Rightchild
10. loc = 2*loc+2