一、选择填空题（20 分，每空 2 分）

1. 从逻辑上可以把数据结构分为 A 两大类。
   A. 线性结构、非线性结构　　B. 顺序结构、链式结构
   C. 动态结构、静态结构　　D. 简单结构、复杂结构

2. 在发生非法操作时，算法能够做出适当处理的特性称为 D。
   A. 正确性　　B. 可读性　　C. 确定性　　D. 健壮性

3. 用链表表示线性表的优点是 A。
   A. 便于插入和删除　　B. 花费的存储空间较顺序存储少
   C. 便于随机存取　　D. 数据元素的物理顺序与逻辑顺序相同

4. 设栈 S 和队列 Q 的初始状态为空，元素 a、b、c、d、e、f 依次通过栈 S，一个元素出栈后即进队列 Q，若 6 个元素出队的序列是 b、d、c、f、e、a，则栈 S 的容量至少应该是 C。
   A. 6　　B. 4　　C. 3　　D. 2

5. 模式串"abaabcac"的 nextval 数组值是 B。
   A. 01122312　　B. 01021302　　C. 00011201　　D. 01011201

6. 数组 A[8][10]中，每个元素占用 3 个存储单元，起始存储地址是 1000，则数组元素 A[5][3]的存储地址是 D。
   A. 1126　　B. 1129　　C. 1156　　D. 1159

7. 广义表（a，(b，(c，d)，e)，f）的深度和元素个数分别是 D。
   A. 2、2　　B. 2、3　　C. 3、2　　D. 3、3

8. 已知一棵树中有 n 个分支结点，其对应的二叉树中右子树为空的结点有 A 个。
   A. n-1　　B. n　　C. n+1　　D. n+2

9. 由三个关键字不相同的结点构成的二叉树共有 A 种。
   A. 30　　B. 20　　C. 10　　D. 5

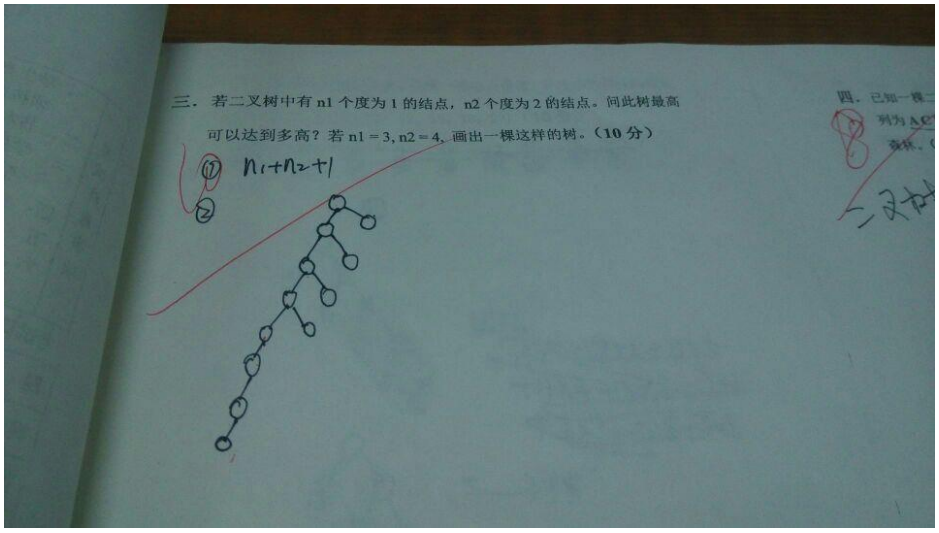
10. 在后序线索树中，某结点的线索不可能是它的 B。
    A. 父母结点　　B. 左子树结点　　C. 右子树结点　　D. 兄弟结点

A. 父母结点　　B. 左子树结点　　C. 右子树结点　　D. 兄弟结点

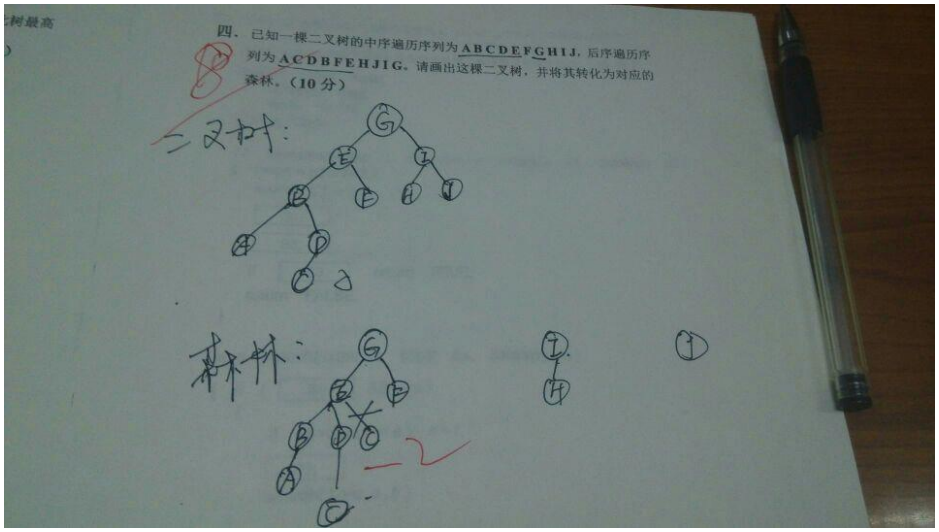
二、判断题（10 分，每空 1 分）

1. （×）数据项是数据的最小单位。
2. （×）数据的逻辑结构是指数据的各数据项之间的逻辑关系。
3. （×）顺序存储插入和删除时效率太低，因此它不如链式存储方式好。
4. （×）线性表的逻辑顺序总与其物理顺序一致。
5. （×）栈在递归时非常重要，因此若整个程序没有递归，则栈也可以不要。
6. （×）若一个广义表的表头为空表，则此广义表也为空表。
7. （√）稀疏矩阵压缩存储后，必会失去随机存取功能。
8. （×）用树的前根序遍历和后根遍历可以导出该树。
9. （√）由树转换为二叉树后，其根结点的右子树总是空的。
10. （√）满二叉树一定是完全二叉树，反之未必。

三．若二叉树中有 n1 个度为 1 的结点，n2 个度为 2 的结点。问此树最高可以达到多高？若 n1 = 3, n2 = 4，画出一棵这样的树。（10 分）

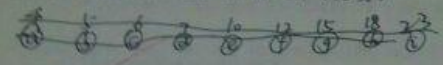


四．已知一棵二叉树的中序遍历序列为 A B C D E F G H I J，后序遍历序列为 A C D B F E H J I G。请画出这棵二叉树，并将其转化为对应的森林。（10 分）

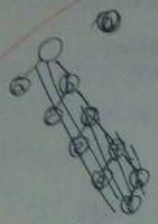


五、 对下面给出的数据序列，构造一棵哈夫曼树，并求出其带权路径长度。(4, 5, 6, 7, 10, 12, 15, 18, 23) (10 分)

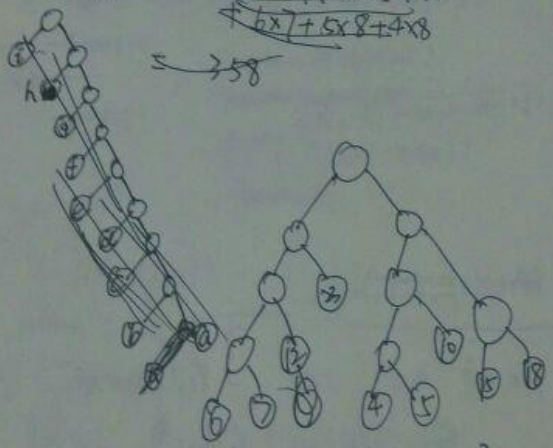
六、



10



WPL
= 23×1 + 18×2 + 15×3
+ 12×4 + 10×5 + 7×6
+ 6×7 + 5×8 + 4×8
= 358



$WPL = 4\times6 + 7\times4 + 12\times3 + 23\times1 + 10\times3 + 4\times4 + 15\times4 + 15\times3 + 18\times3$

$= 299$

六. 已知二叉链表表示的二叉树中有值为 e、e1、e2 三个结点，下面算法是判断 e 是否为 e1 和 e2 的共同祖先，请在空格处填上相应的语句或表达式。(10 分)

```
typedef struct node{
    datatype data;
    node *lc, *rc;
} node, *bitptr;

int forefather( bitptr t, datatype e, datatype e1, datatype e2 )
{   f = p1 = p2 = NULL;
    search( t, f, e );
    [ S1; ]
    [ S2; ]
    if [ S3 ] return TRUE;
    return FALSE;
}

void search( bitptr t, bitptr &s, datatype e )
{ if ( [ S4 ] && !s )
  {
      if ( t->data = = e ) s = t;
      [ S5; ]
      search( t->lc, s, e );
  }
}
```

S1: search(f, p1, e);

S2: search(f, p2, e)

S3: (p1 && p2)

S4: t

S5: search(t->rc, s, e);

七. 已知矩阵 $A_{m\times n}$ 中，从上到下、从左到右元素值从小到大，有元素 x 一定存在于 A 中。给出算法不超过 m+n 次比较找到元素 x 的下标（15 分）

```
void search( int A[][50], int m, int n, int x, int &i, int &j)
// x 的下标由引用类型参数 i 和 j 返回
{
    int i=0, j=n-1;
    while (A[i][j]!=x)
    {
        if (A[i][j]>x)
            j--;
        else
            i++;
    }
}
```

八．试编写算法，对一棵以孩子—兄弟链表表示的树统计叶子节点的个数。（15 分）

```
typedef  struct  node{
     datatype  data;
     node  *child, *sibling;
} node, *bitptr;

int  countleaf ( bitptr  t )  // t 为根节点，函数返回值为叶子结点个数
{   int i=0
    if (!t)
        return 0;
    else{
        if(!t->child==NULL)
            i++;
        else{ count leaf(t->child);
              i+=count leaf(t->sibling);
        }
    }
    return i;
}
```