机密★启用前

西南交通大学 2015 年全日制硕士研究生

招生入学考试试卷

试题代码： 840

试题名称： 数据结构与程序设计

考试时间： 2014 年 12 月

考生请注意：

1.本试题共 57 题，共 9 页，满分 150 分，请认真检查；

2.答题时，直接将答题内容写在考场提供的答题纸上，答在试卷上的内容无效；

3.请在答题纸上按要求填写试题代码和试题名称

4 试卷不得拆开、否则遗失后果自负。

一单项选择题(50 分。管题 2 分，共 25 小题)(答在试卷上的内容无效)

1.以下数据结构属于非线性结构的是(D)

A.链表　　B 栈

C.队列　　D.二叉树

2.以下对线性表描述错误的是(D)

A.顺序表是随机存储结构，存储的时间复杂度为 O(1)

B.链表是顺序存储结构，存储的时间复杂度为 O(n)

C.顺序表的插入和删除可能要移动大量的元素

D.链表的插入和剩除可能要移动大量的元素

3.已知 p 为指向带头结点的单链表的头指针，以下表达式说明该单链表为空的是(B)

A.p== NULL　　B.p->next== NULL

C.p!= NULL　D.p==p->next

4.设堆栈 S 和队列 Q 的初始状态为空，元素 1,2,3,4,5,6 依次入栈，一个元素退栈后即进入队列，若这6个元素出队列的顺停是3,2,6,5,4,1，则栈 s 的容量至少应该是(C)

A2　　B.3　　C.4　　D.5

5.若从二叉树的根结点出发到达任一叶子结点的路径上所经过的结点序列按其关键字有序，则该二叉树一定是(D)

A.二叉排序树　　B.哈夫曼树

C.B-树　　D.堆

解析：Huffman 树的权值全部在叶子上，那些分支的权值仅仅供构造算法使用的，其实是没有权值的。至于二叉排序树和 B-树肯定不能满足这个条件，　至于堆有两种，最小堆和最大堆，最小堆就是递增有序，最大堆就是递减有序，问题只是问有序，不一定是

降序，所以选 D。

至于二叉排序树和 B-树肯定不能满足这个条件，只是满足中序有序

6.已知在一棵度为 3 的树中，有 4 个度为 1 的结点，3 个度为 2 的结点,2 个度为 3 的结点，那么该树中有叶子节点个数(A)

A.8　　B.9

C.10　　D.11

解析：节点数比分支数多一，有 4×1+3×2+2×3=16 个分支,所以有 17 个结点，17-4-3-2=8 个叶子结点

7.某哈弗曼树的结点总数为 2n-1 若用二叉链表作为存储结构，则该哈弗曼树中一共有多少个空指针域（B）

A.2n-1　　B.2n

C.2n+1　　D.4n

解析：分支数比节点数少一，所以分支数为

2n-2, 2（2n-1）-（2n-2）=2n，所以空指针域为 2n

8.n 个顶点组成的有向图至少需要多少条弧才能构成强连通图（A）

A.n　　B.2n

C. n(n-1)　　D.n-1

9.设用邻接矩阵 M 来表示有向图 G 的存储结构，那么有向图 G 的第 i 个顶点的入度为（B）

A .第 i 行非 0 元素的个数

B .第 i 列非 0 元素的个数

C .第 i 行非 0 元素的个数除以 2

D .第 i 列非 0 元素的个数除以 2

10.对于哈希表查找，若装填因子越大，则（B）

A.平均查找长度越小

B.平均查找长度越大

C.冲突模率越小

D.以上说法均正确

11.以下内部排序方法中，稳定的排序方法是(A)

A 冒泡排序　　B.堆排序

C.希尔排序　　D.快速持序

12. 设一组有序的初始关健字序列为（13,29,36,42,54,67,70,81,89）,则利用二分直找(折半查找)找到关键字 36 需要比较的关键字个数为(C)

A.1　　B.2

C.3　　D.4

解析：查找的关键字是 54,29,36.

13.在 VC++6.0 中，C 语言源程序的扩展名是（B）

A. .c　　B. cpp

C. prj　　D. .obj

14.下列选项中，合法的 C 语言关建字是（C）

A.Float　　B.integer

C.extern　　D.var

15.以下那组含有不合法的 C 语言常量(每个选项中有两个数据常量)（C）

A.'\\'　, '\"'　　B.089　, 123F

C.'\xaa'　, .123　　D.'\101' , 1.5E-10

解析：以 0 打头的数字是八进制，089 是八进制形式，但后面跟的 8 和 9 不在 0～7 之内，所以非法

16.若变量均已正确定义并赋值，以下合法的 C 语言赋值语句是(D)

A .x*= 5= 4 + 1；　　B.x=n%1.5；

C .x+n=i；　　D.x =y ==10；

解析：A 选项中，=是赋值符号，不可将 4+1 的值赋值给常量 5.B 中的%是模除符号,就是求余数用的,用这个符号,两边必须是整型。D

正确，x=y==10 等价于 x=(y==10), ==号是判断两端是否相等,若右边等于左边,则 y==10 表达式的值为 1,若不等,则为 0,假如相等,就等于说把 1 赋值给了 x,若不等就是说把 0 赋值给 x。

17. 若执行以下程序段： int x=10,y=20,a,b=12； a=(x>y)&&b++；执行后 b 的值是(C)

A.10　　B.11

C.12　　D.13

18.定义 C 语言变量 int x； 不能正确表示 100≥x≥0 的 c 语言自关系表达式的是(D)

A.x<=100 && x>=0

B.!(x>100) && !(x<0)

C.100>=x & 0<=x

D.x<=100||x>=0

19.若定义 int a[3][4]，则对 a 的正确引用是

(D)

A.a[2][4]　　B.a[1,3]

C.a (2)(1)　　D.a[1+1][0]

20 有以下程序段：

```
int k=0;
while(k=1)
    k++;
```

则 while 循环执行的次数是(A)

A.陷入死循环

B.无法通过编译

C.一次也不执行

D.执行 1 次

解析： while(k=1)中 k=1 是赋值语句，条件一直成立，将陷入死循环。

21.以下叙述中错误的是(C)

A 不能利用 typedef 定义新的数据类型

B.利用 typedef 为已有的数据类型定义一个新名字

C.用 typedef 定义新的类型名后，原有类型名无效

D.用 typedef 可以为各种类型起别名，但不能为变量起别名

22.以下关于 C 语言的描述指误的是(C)

A 不同的函教中可以使用相同名字的局部受量，互不干扰

B.函数中的形式参数都是局部交量

C.函数定义可以嵌套

D.一个函数中可以使用多个 return 语句

23.以下说法正确的是(C)

```
#include"stdio.h"
#define PI 3.14;
const double CPI=3.14;
void main()
```

```
{     int r=10;
      double area1,area2;
      area1= PI*r*r;
      area2= CPI*r*r;
      printf("area1=%f",area1);
      printf("area2=%f",area2 );
}
```

A.程序能通过编译并输出正确的运行结果

B.程序能通过编译但输出结果不正确

C.程序无法通过编译

D.P1 是符号常量，它的类型是浮点型

解析：#define PI 3.14；若没有分号，可以编译通过，并输出正确的运行结果

24.以下程序段的输出结果为(B)

```
Char a[ ]="Nice to see you",*ptr;        //字符串
a 的长度为 15
ptr =a+7;
```

```
for (;   ptr<a+ 15;   ptr++)
    putchar(*ptr);
```

A Nice to see you

B.see you

C.无法通过编译；

D.Nice to

注解：可与试卷下的简答题第六题对比分析

25.执行 fopen(函数发生错误时的返回值NUUL、EOF 在 stdio.h 中分别被定义为 0、-1)是(C)

A.地址值

B.1

C.NULL

D.EOF

二、填空题(30 分，每空 1 分，共 20 小题)

(答在试卷上的内容无效)

1.根据数据元素之间关系的不同特性，通常有四类基本结构，即： 集合、线性结构、树形结构和__网状__结构。

2.已知一个元素有序的单链表，其长度为 n 那么插入一个元素使得插入后的单链表仍然有序，插入操作的平均时间复杂度为__O（n）__.

3.用 C 语言实现 KMP 算法，若子串为"cacaba"，则 next 数组元素值是__ -1 -1 -1 1 2 -1 __.

解析：

next[1] = -1,代表着除了第一个元素，之前前缀后缀最长的重复子串，这里是空 ,即""，没有，我们记为-1，代表空。（0 代表 1 位相同，1 代表两位相同，依次累加）。

next[2] = -1，即“c”，没有前缀与后缀，故最长重复的子串是空，值为-1；

next[3] = -1，即“ca”，前缀是“c”，后缀是“a”，最长重复的子串“”；

next[4] = 1，即"cac"，前缀是“ca”，后缀是“ac”，最长重复的子串“c”；next 数组里面就是最长重复子串字符串的个数

next[5] = 2，即"caca"，前缀是“cac”，后缀是“aca”，最长重复的子串“ca”；

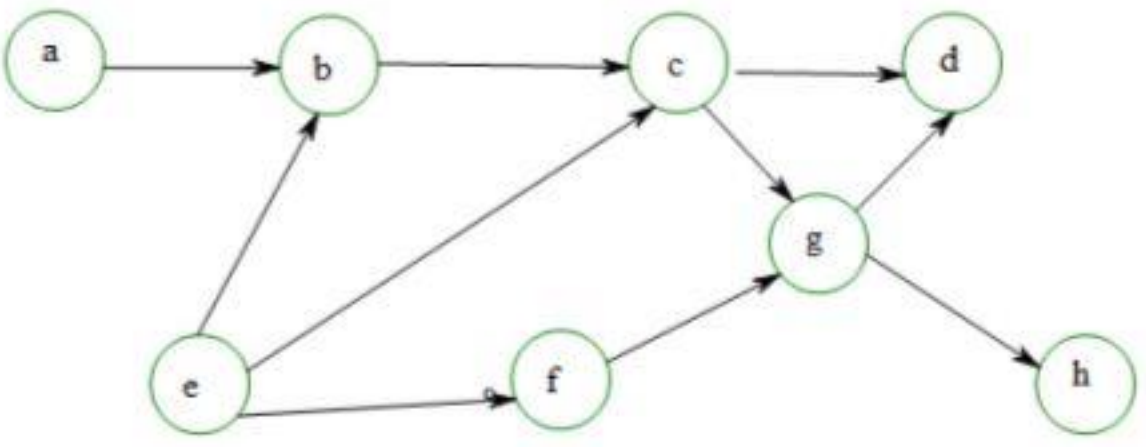
next[6] = -1，即"cacab"，前缀是“caca”，后缀是“acab”，最长重复的子串“”；

4.已知整型变量占 4 个存储单元(字节)。C 语言定义数组 int a[6][5],若数组按行优先存储，起始地址为 1000 则数组元素 a[2][3]的存储地址为__1052__.

5.设有 n 个结点的完全二叉树，如果从上到下、从左到右为每个结点编号，第一个结点

编号为 1，其他结点的编号依次递增。那么编号为 i(i>1)的结点的双亲结点编号为__n/2取下值__.

6 某有向图如下所示。该图采用邻接表实现存储。设其邻接表存储结构中各单链表中的表结点按顶点字母升序连接，写出以顶点 e 出发进行宽度优先搜素时的顶点访问次序__ebcfdgha__.



7.在分块检索中，若索引表和各分块均用顺序查找，则有 975 个元素的线性表均匀分成 25 块，各元素等概率查找，则查找成功时的平均查找长度为__33__.

8.已知一个带头结点的单链表，下面的函数对单链表实现就地逆置(即不开辟新的结点将原有的结点链接成一个新的单链表，使得所有结点的排列顺序与原来的顺序相反)。请填空使算法完整。

已知结点结构定义为 pedef struct node{ int data； stroct node*next； }Inode；

void reverse(lnode*L)/* L 传入单链表的头指针*

```
{    P= L->next;
     L->next=___NULL___;
     While(___p___){
          q=p->next;
          p->next= _L->next__;
          L->next=__p__;
          P=q;
     }
```

}

9.下面的程序段的功能是递归实现二分直找算法：在记录中查找关键字为 k 的记录，如果找到则返回该记录的位置，否则返回-1.请填空使算法完整。

```
struct record {int key} ;
int biSearcb(record* r, int low, int high int k) {
if(__low>high__) return -1;          /*记录未找到，递归终止*/
int mid =__(low+high )/2__;
if(k==- r[mid].key)   return mid;    /* 记录找到*/
if(k<r[mid].key)return __
biSearch(r,low,mid-1,key)__;         /*递归维续进行二分查找*/
else return __biSearch(r,mid+1,high,key)__;
}
```

上述函数的调用方法为

biSearch(r,0,n-1,k)；　　/*r 为记录数组的首地址，n 为记录的长度*/

10.请写出下面数学表达式 x1=-b+a/2a（说明：a、b、c 都是实数）的 c 语言表达式 x1=(-b+a)/(2*a) .

11.结构化程序设计思想，自顶向下、逐步细化 模块化 .

12 C 语言语句序列如下： double s；

scanf(" %lf ",&s)；

13.执行 C 语言语句序列： int x= 5,y=2；

y*=x + 1；执行后 y 的值 12 .

14.执行下面程序段： int x= 1，y；y= x+ 3/2；

执行后 y 的值是 2 .

15.有如下定义：

struct num

{int a； float b； } data,*p；

若有 p=&data；则对 data 中的 a 域的正确引用是__p->a__.

16.定义语句： int a[3][6]；按在内存中的存放顺序，a 数组第 10 个元素是__a[1][3]__.

17.定义带参的宏 S（r）(求半径为 r 的圈的面积，π 的值用 3.14 常量表示)__S（r）=π*r*r__.

18、我行下列程序段,请写山程序运行结果__a<=b__.

```
int x=3,y=20,a=1,b= 10;
if(x<y)
    if(a>b)
        printf("a>b");
    else
        printf("a<=b");
```

19 将一个长度为 5 的数组逆序输出，请填空使之完整。

```
#include"stdio.b"

#define N 5

void main()

    {    int a[N]={9,6,5,4,1},i,temp;

         pintf("\n original array:  \n");

         for（i=0;  i<N;  i+t)

              printf("%4d",a[i]);

         for(i=0;  _i<N_;  i++)

         {    temp=a[i];

              a[i]= a[N-i-1]  ;

         a[N-i-1]=temp;

         }

printf("\n sorted array:  \n");

for(i=0;  i<N;  i++)

         printf("%4d",a[i]);

}
```

20.下面 length 函数的功能是求一字符串长度，在 main 函数中输入字符串并输出其长度。请填空使之完整，

```
#include"stdio.h"
int length(char *p)
    {
        int n;
        __n=0__;
        while(*p!='\0')
        {
            n++;
            __p++__;
        }
        return n;
    }
void main()
{
```

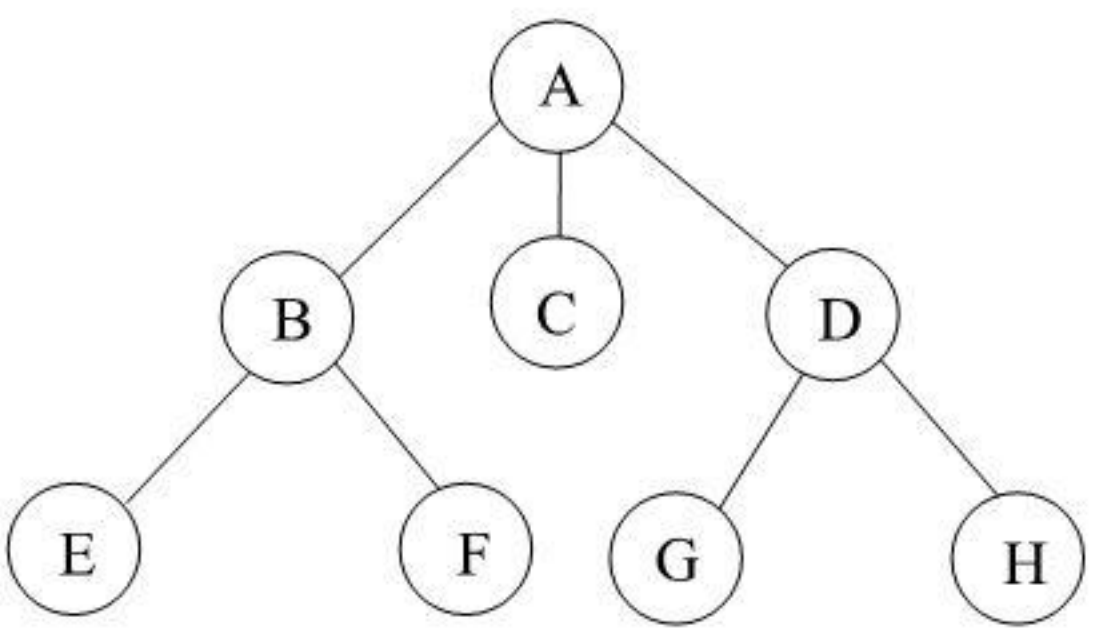
```
    int len;
    char str[20];
    printf("please input a string:  \n");
    gets(str);
     len=length(str)  ;
    prntf("the string has %d characters.",len);
}
```

三、简答题(40 分共 9 小题) (答在试卷上的内容无效)

1、下图所示的树(6 分)

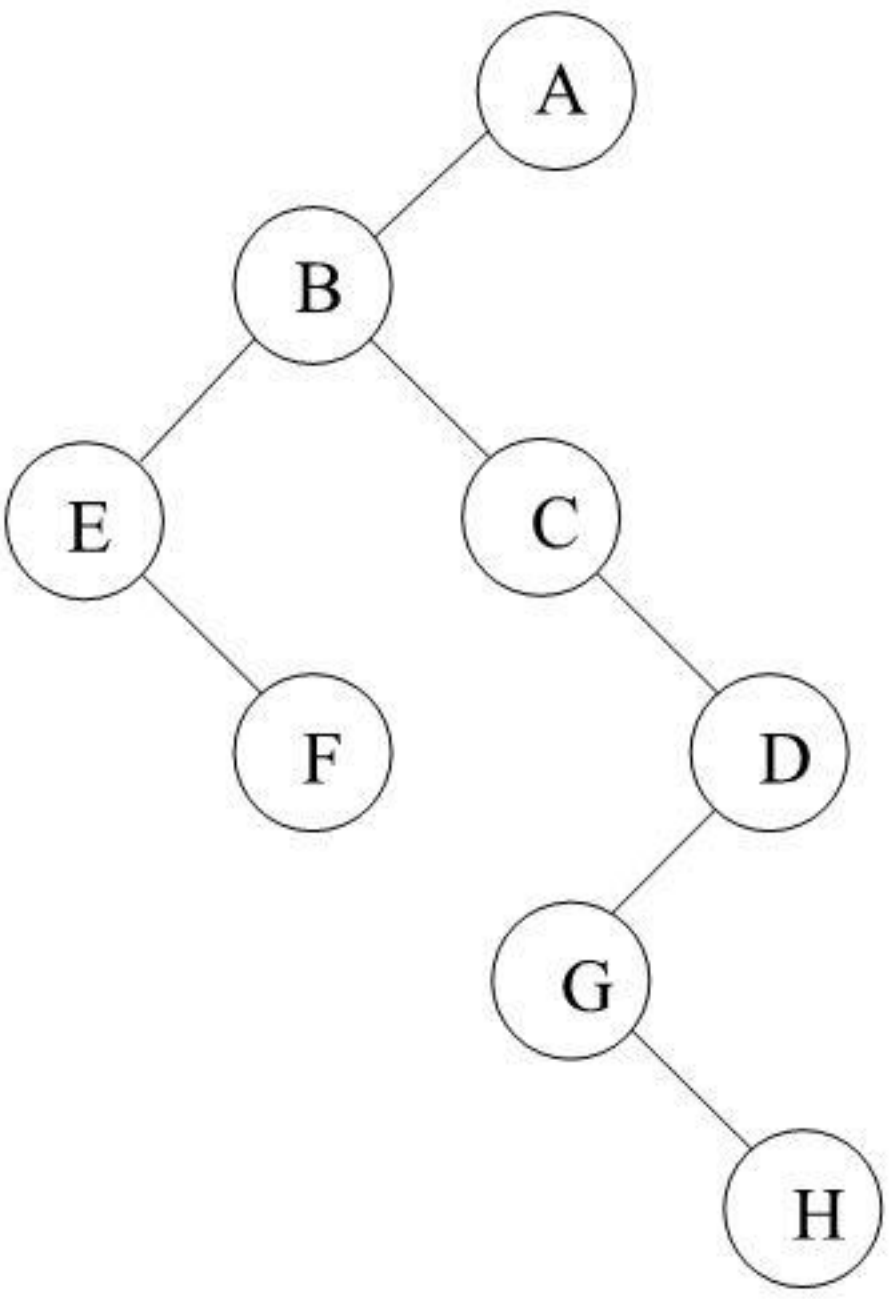
(1)求其先根遍历序列和后根遍历序列

(2)将该树转换成二叉树



答案：先根遍历：ABEFCDGH　　后跟遍历：EFBCGHDA

（2）

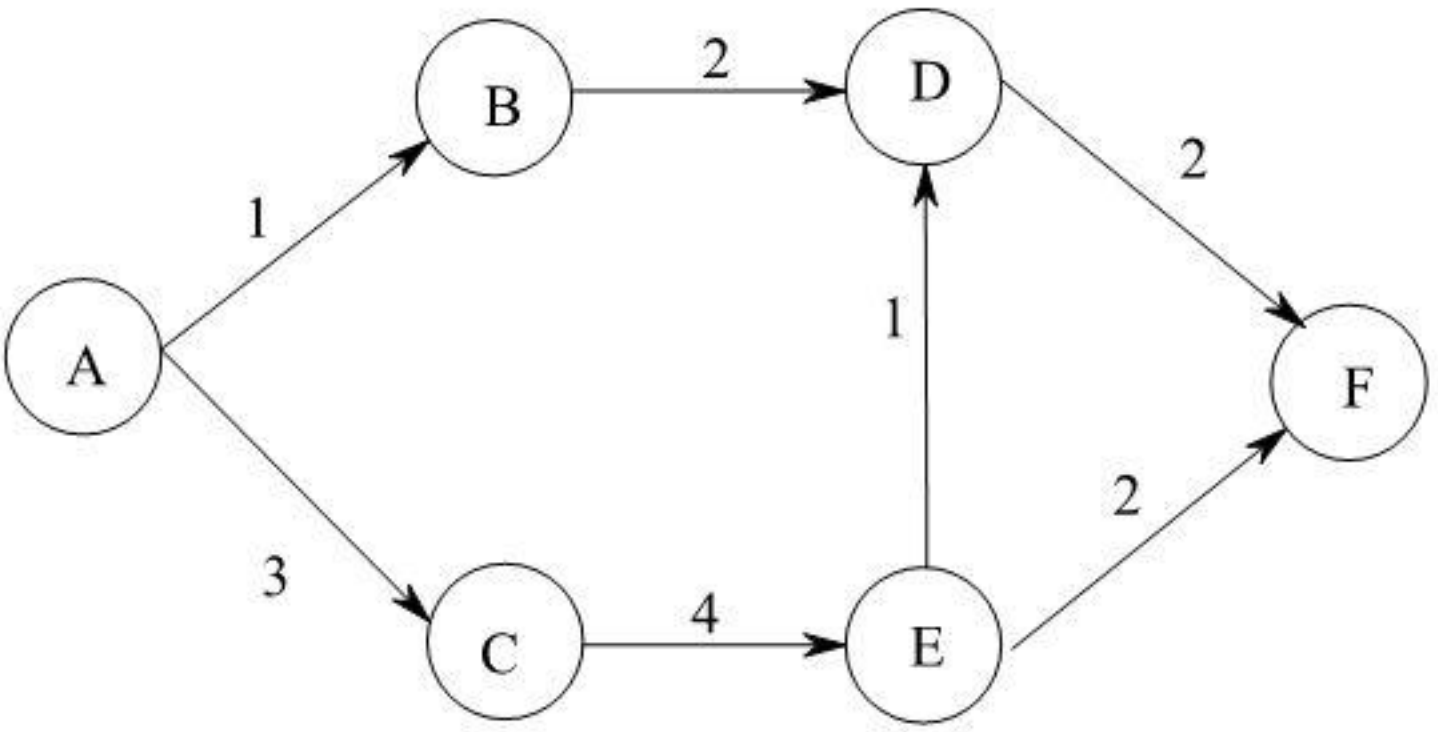


2 已知 AOE 网如下图所示(6 分)

(1)写出图中所有项点的一种拓扑排序序列。

(2)求出所有项点的最早开始时间 ve(i)和最

晚开始时间 vl（i），列出关键路径。



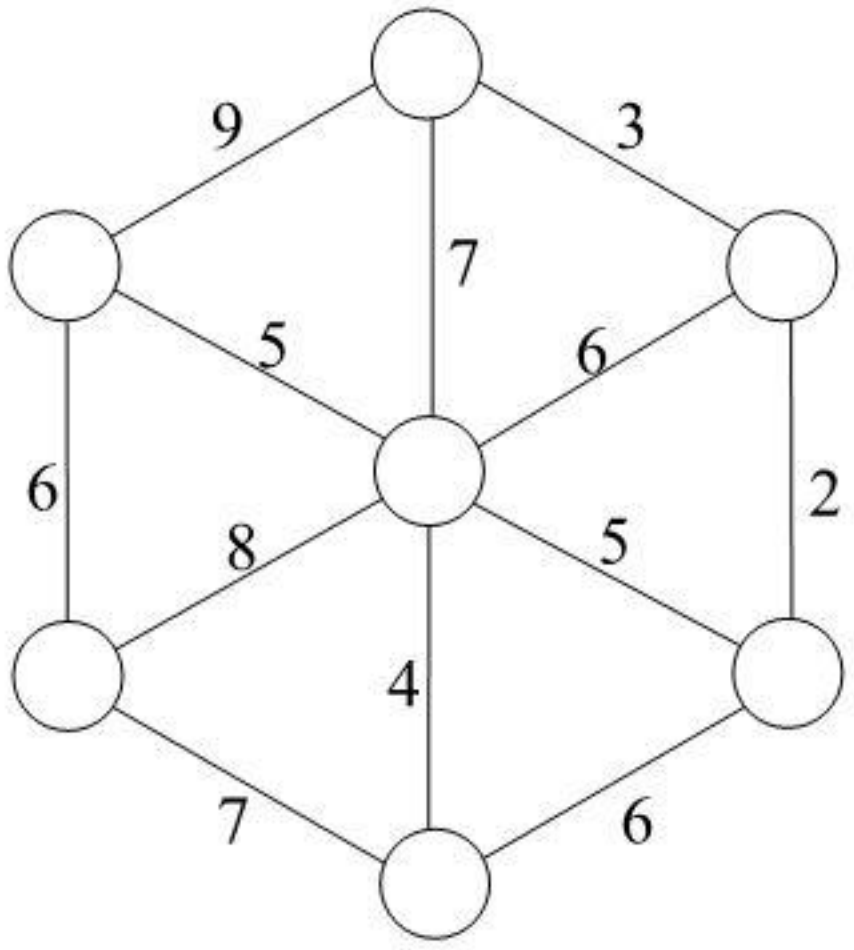
答案：（1）ABCEDF

(2)

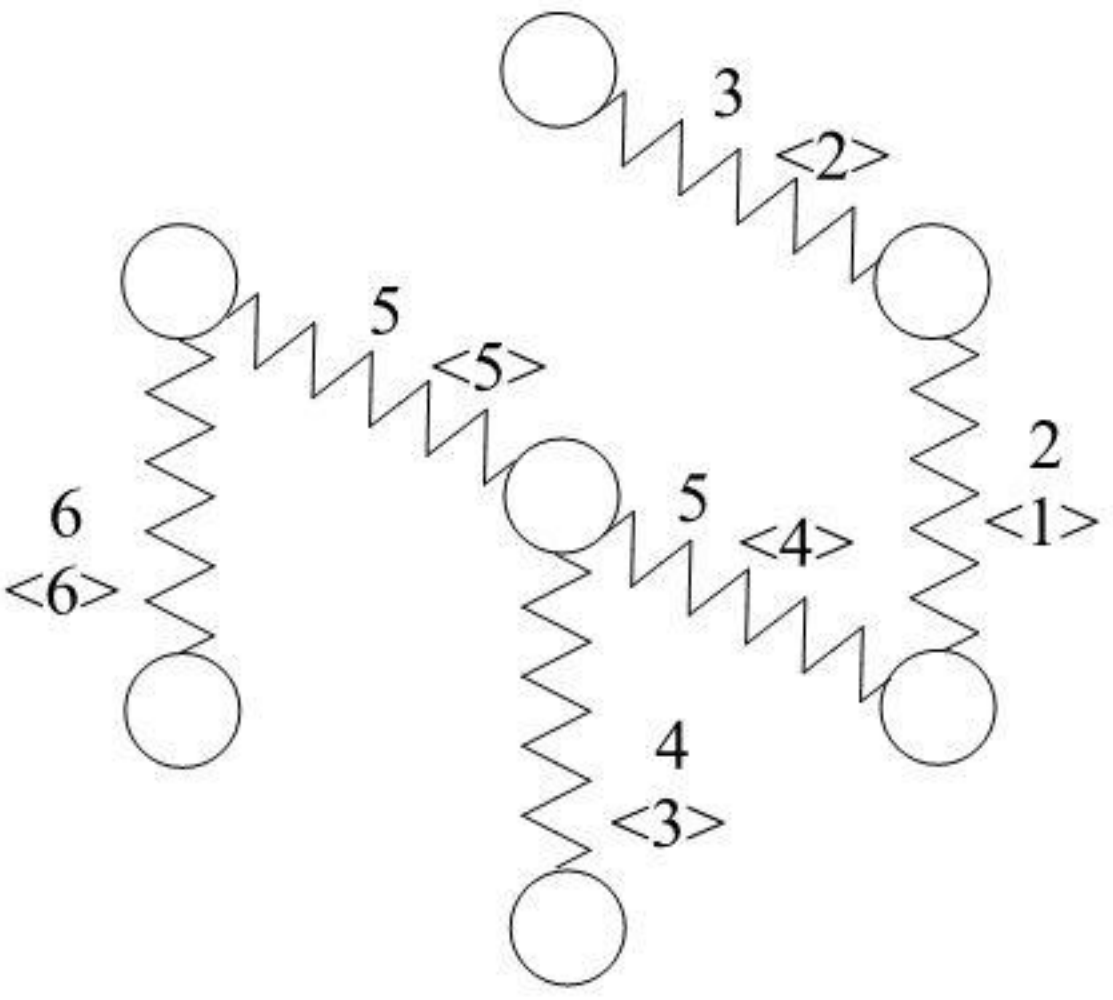
	A	B	C	E	D	F
ve	0	1	3	7	8	10
vl	0	6	3	7	8	10

3.已知带权图如下图所示，用普里姆(Prim)算法(从最上方的结点开始)和克鲁斯卡尔(Kruskl)算法分别求带权图的最小生成树。要求答题时画出原图，将最小生成树的边用波浪线表示，并<>括起来的数字标号

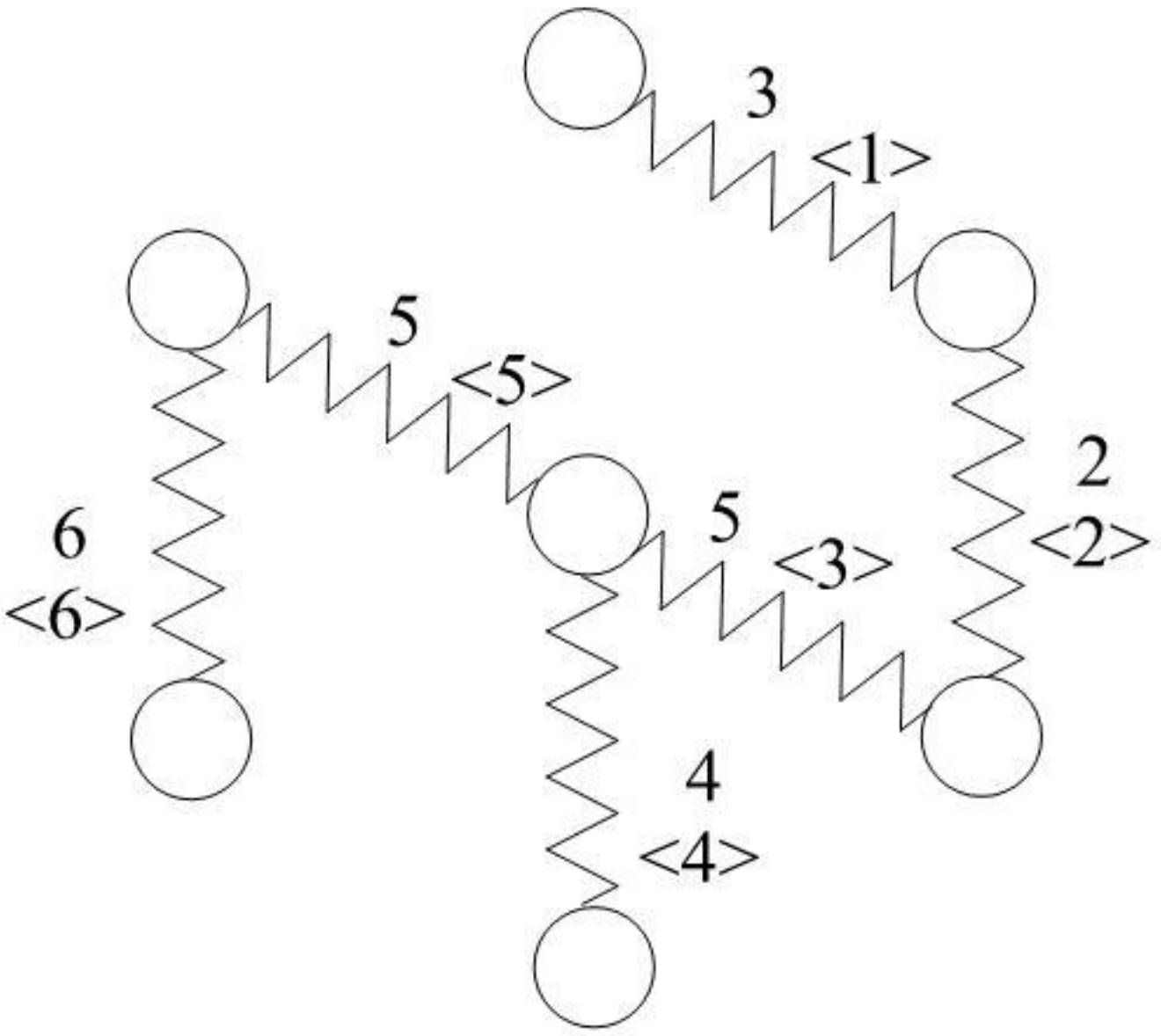
反映最小生成树中各条边的求取次序（7 分)



答案：Krusk 算法



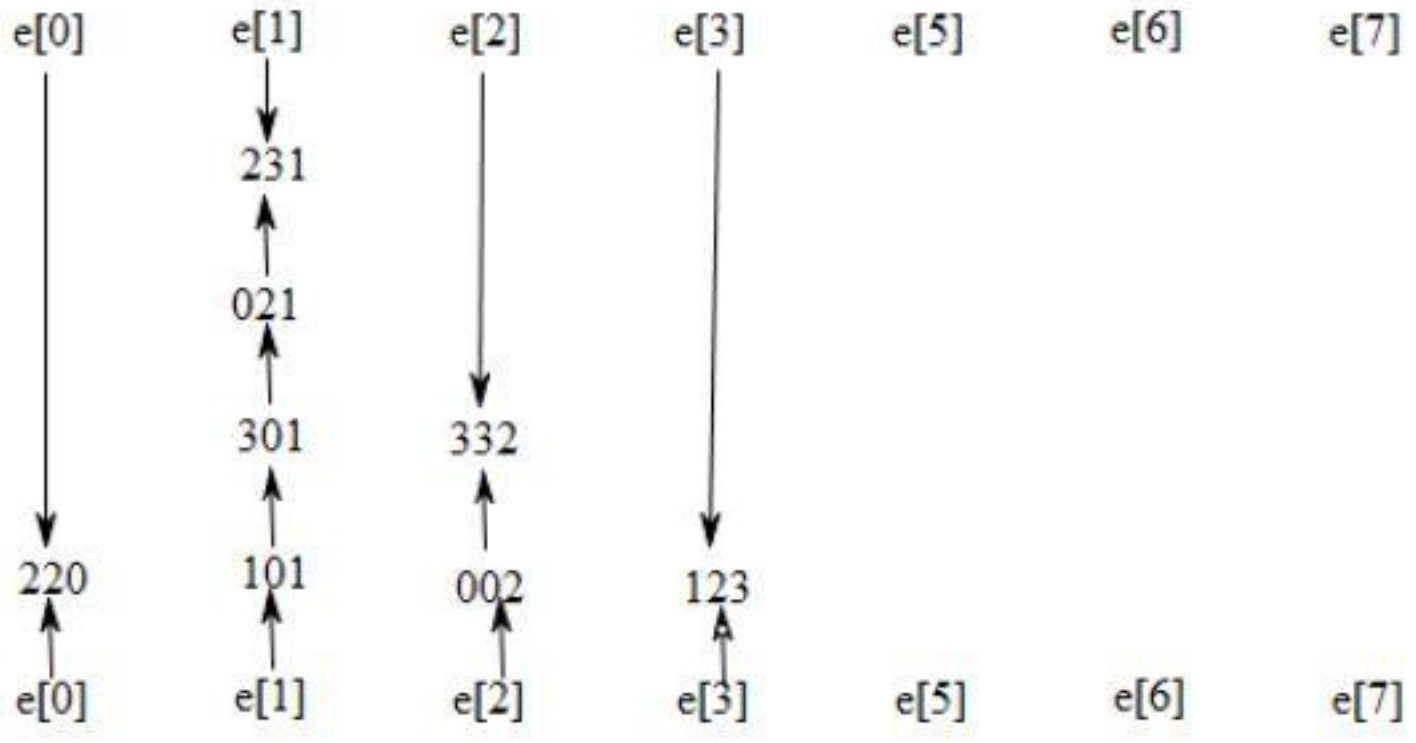
Prim 算法：



4.对以下 8 个 3 位 4 进制数采用基数排序进行有小到大排序，写出对最低位、中间位以及最高位分别进行分配和收集的结果(7 分)

101,002,332,220,123,301,021,231

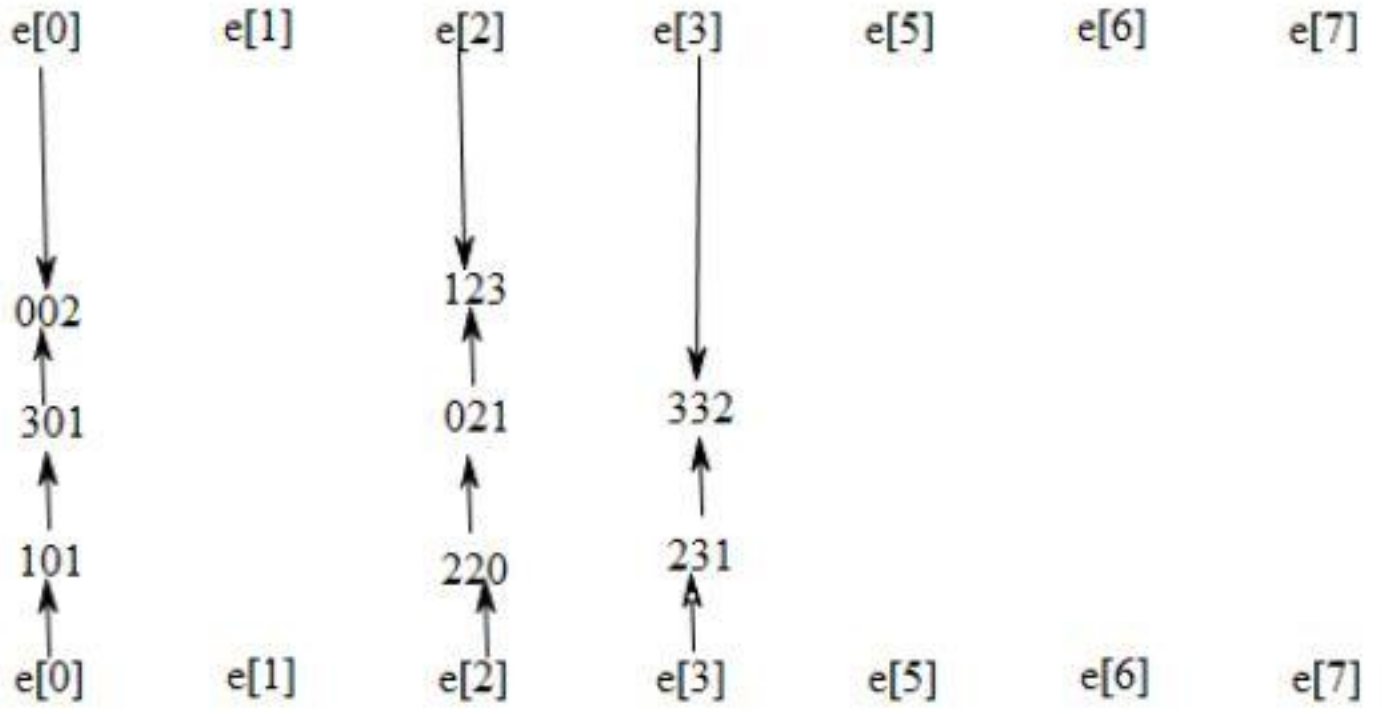
答案：最低位分配结果：



最低位收集结果：

220->110->301->021->231->002->332->123

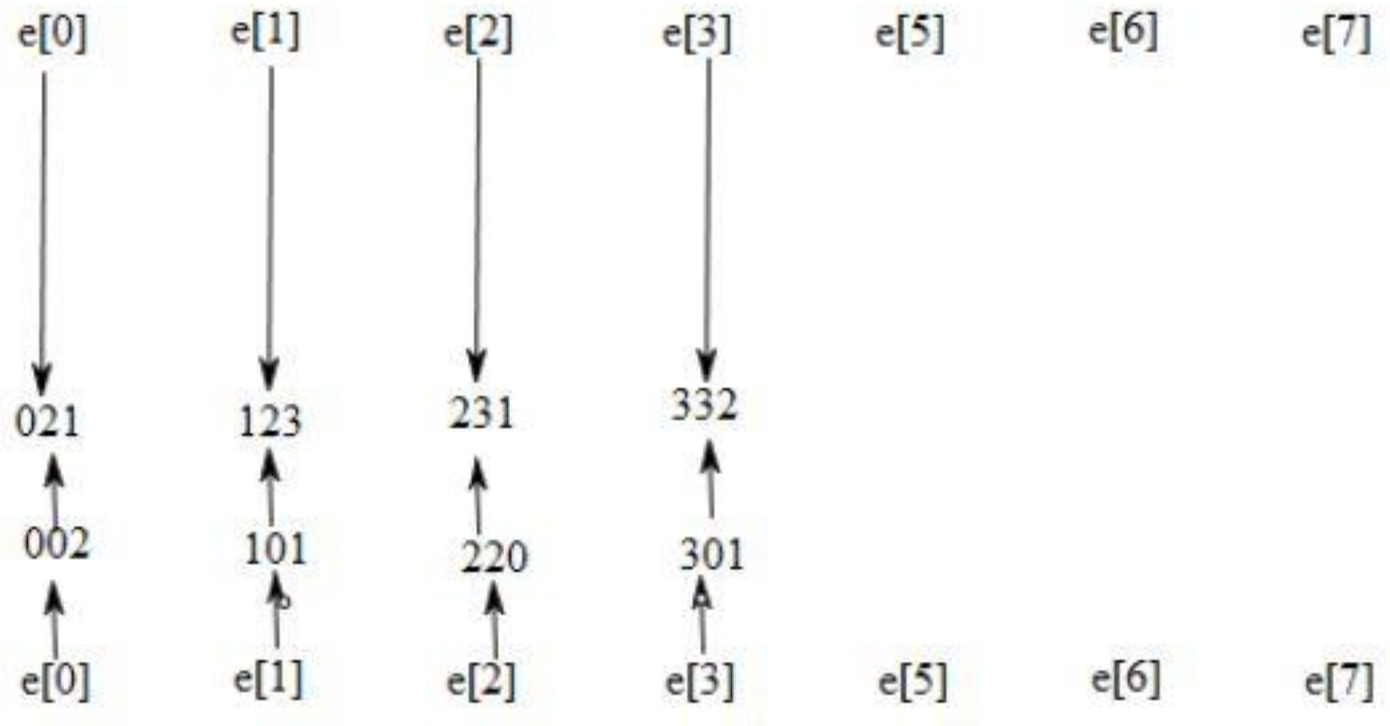
中间位分配结果：



中间位收集结果：

101->301->002->220->021->123->231->332

最高位分配结果：



最 高 位 收 集 结 果 ：

002->021->101->123->220->231->301->332

5.给出下面程序的运行结果(字符‘0'的 ASCII 码为十六进制的 30) (4 分)

```
union pw
{
    int i;
    char ch;
    }a;
    void main（）
    {
```

```
        a.i=100;
        printf("%d\t",a.i);
          a.    ch='0';
         printf("%d\t",a.i);
        printf("%c\t",a.ch);
    }
```

答案：100　　48　　0

6.给出下面程序的运行结果(4 分)

```
#include "stdio.h"
void main()
{
char a[]="good";
char *p;
for(p=a;  p<=a+1;  p++)
printf("%s",p);
}
```

答案：goodood

解析：由于 p 是指针类型，printf("%s",p)；输出遇到'\0'才停止。

7.给出下面程序的运行结果（4 分)

```
#include "stdio.h"
int func(int a,int b)
{ return (a*b)；  }
void main( )
{
    int x=2,y=5,z=8,r;
    r=func(func(x,y),z);
    printf("%\d\n",r);
}
```

答案：80

8.给出下面程序的运行结果(4 分)

```
#include <stdio.h>
void main()
{
```

```
FILE *fp;
int i=20,j=30,k,n;
fp=fopen("dl.dat","w");
fprintf(fp,"%d\n",i+j);
fprintf(fp,"%d\n",i-j);
fclose(fp);
fp=fopen("dl.dat","r");
fscanf(fp,"%d%d",&k,&n);
printf("%d %d\n",k,n);
fclose(fp);
}
```

答案：50　-10

解析：加入头文件#include "stdlib.h"便可运行，先将 i+j 和 i-j 的值写入 fp，再以读模式打开文件，将 i+j 的值写入 k，i-j 的值写入 n，在输出 k，n

9，给出下面程序的运行结果(4 分)

```
#include"stdio.h"
void main()
{    int s;
     scanf("%d",&s);
     while(s>0)
     {    switch (s)
          {    case 1:  printf("%d ",s+5);
               case  2 :  printf("%d  ",s+4) ;
break;
                    case 3:  printf("%d ",s+3);
                    default:  printf("%d  ",s+1);
break;
          }
          scanf("%d",&s);
     }
}
```

运行时，若输入 6 个整数：2 1 3 4 0<回车>

答案：6 6 5 6 4 5

四、程序与算法设计题(30 分，每题 10 分，共 3 小题) (答在试卷上内容无效)

1.编写算法，利用队列实现二叉树按层次遍历。(10 分)

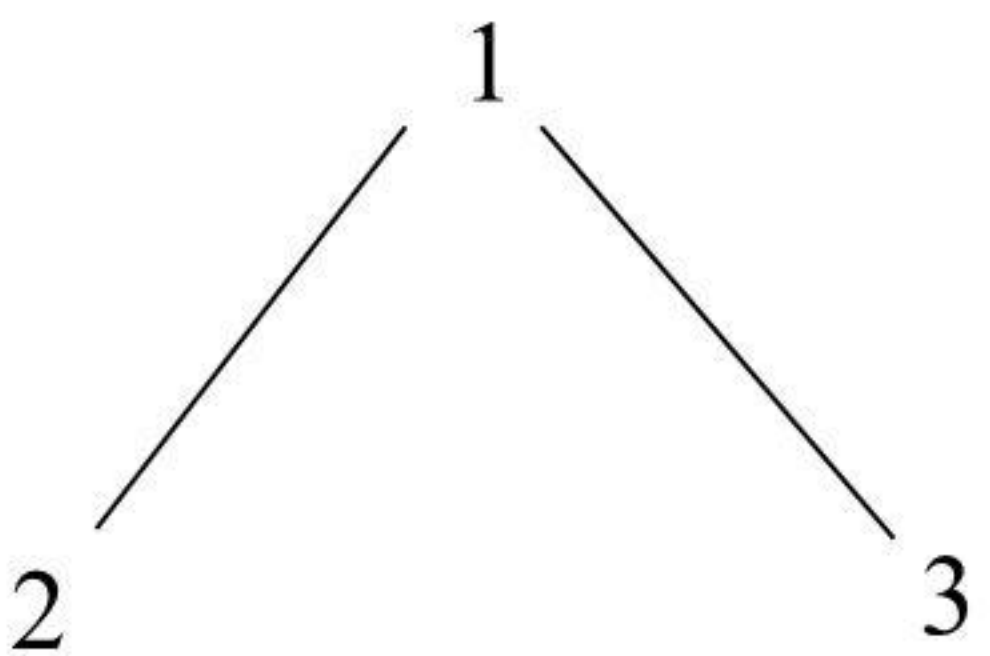
二叉树的结点数据类型定义如下

typedef struct node {int val；　struct node *lchild；　struct node *rchild；　}bitree；

队列可直接使用，无需给出其实现细节(即假设队列已有正确定义，所用操作请加适当注释即可)。

(1) 算法依次输出所有遍历的结点

例：　　输出：1　2　3

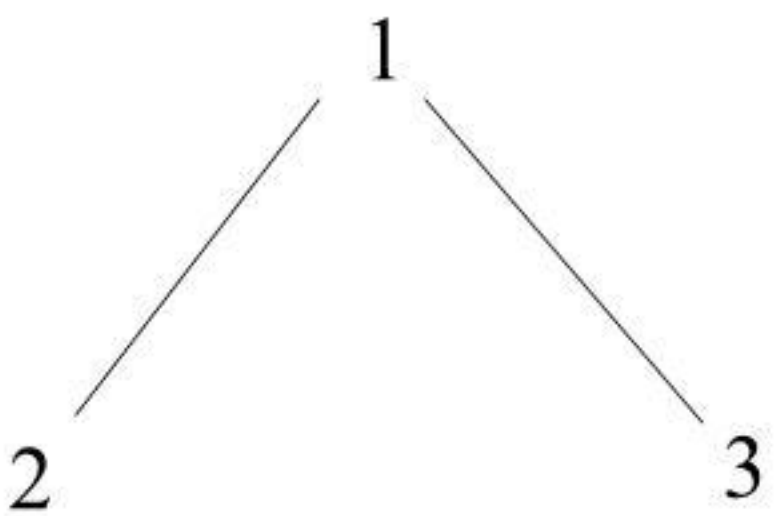


(2)算

法按层次输出每层遍历的结点

例：　　输出：1

2　3



答案：(1)

```
void Traversal(TreeNode *root)
{
    TreeNode *cur;
    EnQueen(root);                    //根节点
入队，如果为空入队失败
    while(!QueenNull())               //如果
队列不为空
    {
        cur = OutQueen();             //出队
```

```
            print(cur);
//访问当前出队节点
            EnQueen(cur->left);          //入队
左节点
            EnQueen(cur->right);         //入队右
节点
        }
}
```

（2）

```
#include <math.h>
void Traversal(TreeNode *root)
{
        TreeNode *cur;
        EnQueen(root);                   //根节点
入队，如果为空入队失败
        while(!QueenNull())              //如果
队列不为空
```

```
{
        int k=0,i=1;
        cur = OutQueen();                    //出队
        k++;
       if(pow(2,i-1)==k)                     //判断 2
```

的 i-1 次方是否等于//k，2 的 i-1 次方为第 i 层的叶子结点数，k 为统计每一层的叶子节//点数。

```
        {
              Printf("\n"); i++; k=0;
        }
         print(cur);
```

//访问当前出队节点

```
        EnQueen(cur->left);          //入队
```

左节点

```
        EnQueen(cur->right);         //入队右
```

节点

```
        }
}
```

2.输入三个 1 位非负整数，输出由这三个 1 位整数组成的一个三位整数（能构成十进制 3 位整数）且该三位整数的百位数、十分位数、个位数分别按升序排列。如，

输入： 3 9 6　　输出：369

输入： 9 0 3　　输出： error

答案：

```
#include"stdio.h"
void main()
{
    int a,b,c,temp;
    scanf("%d%d%d",&a,&b,&c);
    if(a>b)   //以下三个 if 是将 a，b，c 按从小到大排序
    {
```

```
temp=a；  a=b；  b=temp；
}
if(b>c)
{
temp=b；  b=c；  c=temp；
}
if(a>b)
{
    temp=a；  a=b；  b=temp；
    }
    if(a==0)
        printf("error")；
    else
        printf("%d",100*a+10*b+c)；
}
```

3.给定程序中，函数 fun 的功能是：将 N×N 矩阵对角线元素中的值与反向对角线对应

位置上元素中的值进行交换。例如，若 N=3，有以下矩阵：

1 2 3

4 5 6

7 8 9

交换后为：

3 2 1

4 5 6

9 8 7

要求：(1)N 要在程序中通过常量定义给出：

(2) 按照上述题目要求，定义函数 fun（）完成相应的功能；

(3) N×N 矩阵在 min 函数中给出，在 min 函数中调用函数 fun（）完成交换，并在 main 函数中输出交换后的矩阵。

答案：

```
#include"stdio.h"
```

```
#define N 3
void fun(int t[][N], int    n)                  //二维数
组与一维数组的调//用不一样，请注意区别
{    int    i,s;
     for(i=0；i<N；    i++)                          //每一
行的相应位置进行互换
     {    s=t[i][i]；
          t[i][i]=t[i][n-i-1]；
          t[i][n-1-i]=s；
     }
}
main( )
{    int    t[][N]={1,2,3,4,5,6,7,8,9}, i, j；
     fun(t,N)；
     for(i=0；    i<N；    i++)     //输出
     {    for(j=0；    j<N；    j++)    printf("%d
",t[i][j])；
```

```
            printf("\n");
        }
    }
```