

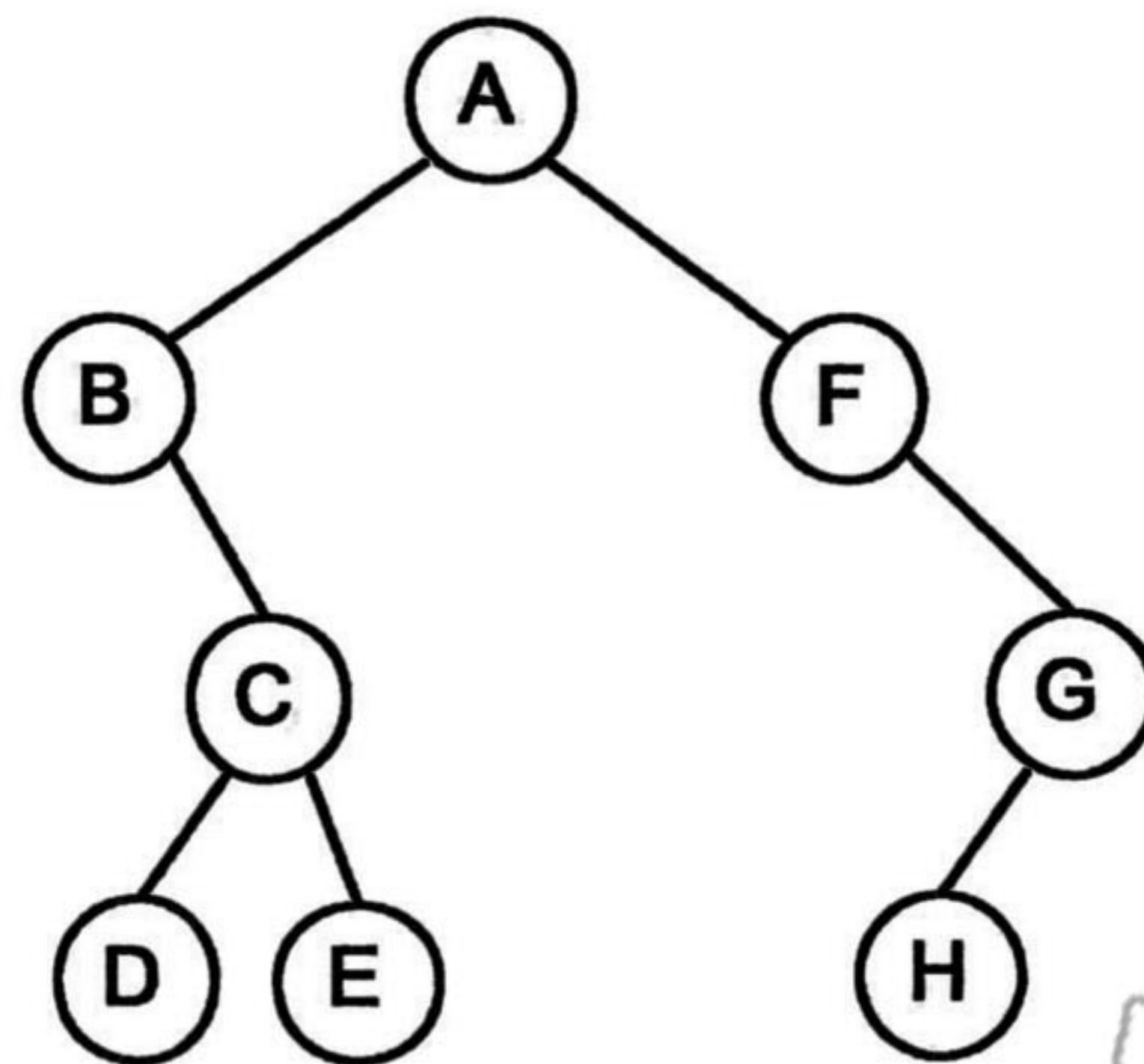
## 2000 年试题参考答案

### 题一、简答题

#### 1.1

- (1) 边的条数=邻接矩阵中“1”的个数的一半；
- (2) 若邻接矩阵  $\text{edges}[i][j]$  或者  $\text{edges}[j][i]$  等于 1，则顶点  $i$ 、 $j$  之间有边相连，反之则无；
- (3) 顶点对应邻接矩阵对应行或对应列“1”的个数。

#### 1.2



#### 1.3

```
int binarySearch(ElemType []srcArray,int des)
{
    //第一个位置.
    int low=0;
    //最高位置.数组长度-1,因为下标是从 0 开始的.
    int high=srcArray.length-1;
    //当 low"指针"和 high 不重复的时候.
    while(low<=high)
    {
        //中间位置计算,low+ 最高位置减去最低位置,右移一位,相当于除 2.也可以用(high+low)/2
        int middle=low+((high-low)>>1);
        //与最中间的数字进行判断,是否相等,相等的话就返回对应的数组下标.
        if(des==srcArray[middle])
        {
            return middle;
            //如果小于的话则移动最高层的"指针"
        }
        else if(des<srcArray[middle])
            high=middle-1;//移动最低的"指针"
        else
            low=middle+1;
    }
    return -1;
}
```

#### 1.4、



算法思想：

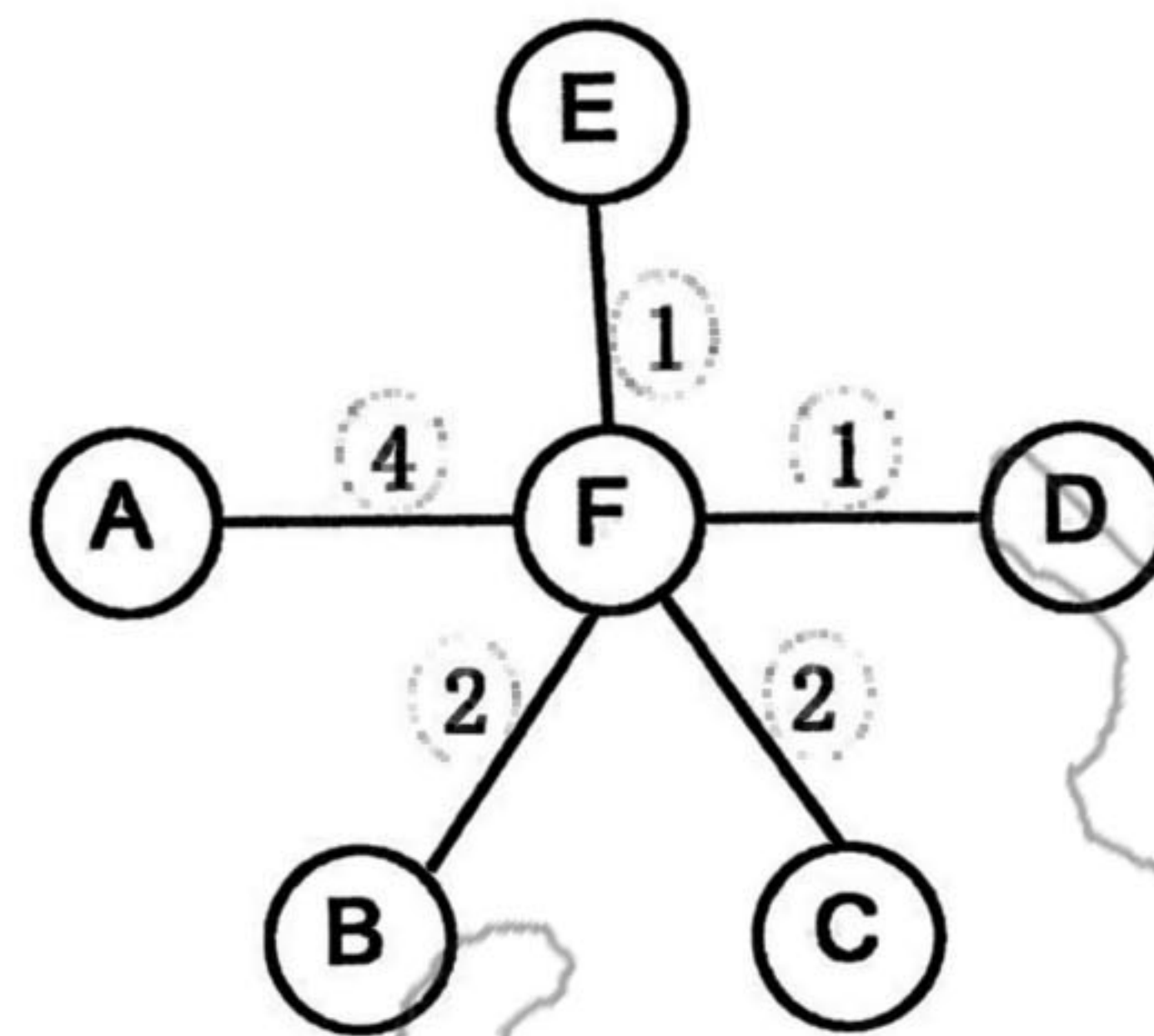
①普利姆算法（Prim）：

算法思想：在所给图中任选一个结点作为一棵树，然后在与结点相邻接的边中选取一条权值最小的边，同时把边上的另一个结点纳入这棵树。那么，重点来了，这时候继续选取边，边的选择方式：选择树中的所有结点所邻接的边中权值最小的边纳入新的结点，以此类推直至图中所有点均在树中。这样，便得到了最小生成树。

②克鲁斯卡尔算法（Kruskal）：

算法思想：在所给图中选取权值最小的边，纳入生成树中，在不构成环路的情况下，继续选取权值最小的边及其结点纳入生成树中，以此类推直至图中所有点均在树中。这样，便得到了最小生成树。

最小生成树：



## 题二、设计题

### 2.1

```
#include<iostream.h>
typedef struct DLNode
{
    int data;
    struct DLNode *ff;
    struct DLNode *rr;
}DLNode;

DLNode* search(DLNode *&C,int x)
{
    DLNode *p=C->rr;
    while(p)
    {
        if(p->data==x)
            break;
        p=p->rr;
    }
    return p;
}

void del(DLNode *p)
{
    if(p==NULL)
        return;
    p->ff->rr=p->rr;
```



本答案由学长友情提供，我们未核实其全部正确性。免费提供，仅供参考！  
更多专业课视频和资料，请见：[www.e-studysky.com](http://www.e-studysky.com)；咨询QQ：3505993547

```
p->r->ff=p->ff;
free(p);
}

void main()
{
    int x;
    cin>>x;
    DLNode *p=search(C,x);
    del(p);
}
```

2.2

假设采用二叉链表存储结构：

```
typedef struct BTNode
{
    ElemType data;
    struct BTNode *lchild;
    struct BTNode *rchild;
} BTNode;
```

算法如下：

```
int similar(BTNode *T1,BTNode *T2)
{
    int lefts,rights;
    if(T1==NULL&&T2==NULL)
        return 1;
    else if(T1==NULL&&T2!=NULL||T1!=NULL&&T2==NULL)
        return 0;
    else
    {
        lefts=similar(T1->lchild,T2->lchild);
        rights=similar(T1->rchild,T2->rchild);
        return lefts&&rights;
    }
}
```

#### 题四、程序阅读题

1、

运行结果：

4 10 13

0 -6 13

6 10 35

结果分析：



本答案由学长友情提供，我们未核实其全部正确性。免费提供，仅供参考！  
更多专业课视频和资料，请见：[www.e-studysky.com](http://www.e-studysky.com)；咨询QQ：3505993547

本题考查全局变量、静态变量的使用。

从主函数走起，首先遇到了静态变量  $a$  的定义及其初始化，这里并未见到初始化的影子，但是函数并没有错，而且读者不能参照 `func()` 函数中的静态变量初始化，注意这两个静态变量根本就没有任何关系，在主函数中的静态变量  $a$  因为没有初始化语句，因此默认其值为 0。继而定义了一个局部变量  $b$ ，执行操作  $b+=4$  即  $b=b+4=-6$ ；执行 `func` 函数，进入函数主体定义了新的静态变量  $a$  赋值为 2，紧接着定义了局部变量  $b$  赋值为 5，执行函数内静态变量  $a$  的  $a+=2$  即  $a=a+2=4$ ，执行全局变量  $n$  的  $n+=12$  即  $n=n+12=13$ ，执行局部变量  $b$  的  $b+=5$  即  $b=b+5=10$ ，输出即可。

回到主函数时，输出的  $a$  为主函数定义的静态变量为 0， $b$  为 -6， $n$  为全局变量，变为 13。

执行  $n+=10$  即  $n=n+10=23$ ，进入 `func` 函数，执行函数内静态变量  $a$  的  $a+=2$  即  $a=a+2=4+2=6$ ，执行全局变量  $n$  的  $n+=12$  即  $n=n+12=35$ ，执行局部变量  $b$  的  $b+=5$  即  $b=b+5=10$ ，输出即可。

2、

运行结果：

```
xxxx
abcdqrstijklbcd
```

3、

运行结果：

```
5
5
5
```

4、

运行结果：

```
666
```

结果分析：

本题考查传指针函数调用。

执行 `fun(5,6,&x)`，执行  $b+=a$  即  $b=b+a=6+5=11$ ，执行  $*c=b-a=11-5=6$ ，得  $x=6$ ；

执行 `fun(7,x,&y)`，执行  $b+=a$  即  $b=b+a=6+7=13$ ，执行  $*c=b-a=13-7=6$ ，得  $y=6$ ；

执行 `fun(x,y,&z)`，执行  $b+=a$  即  $b=b+a=6+6=12$ ，执行  $*c=b-a=12-2=6$ ，得  $z=6$ 。