

计算机科学与技术专业数据结构试题

2002 年 1 月

Table with 7 columns: 题号, 一, 二, 三, 四, 五, 六, 总分. Row 1: 分数

Table with 2 columns: 得分, 评卷人

一、单选题(每小题 2 分,共 20 分)

- 1. 与数据元素本身的形式、内容、相对位置、个数无关的是数据的()。
A. 存储结构 B. 逻辑结构
C. 算法 D. 操作
2. 链式栈与顺序栈相比,一个比较明显的优点是()。
A. 插入操作更加方便 B. 通常不会出现栈满的情况
C. 不会出现栈空的情况 D. 删除操作更加方便
3. 对待排序的元素序列进行划分,将其分为左、右两个子序列,再对两个子序列施加同样的排序操作,直到子序列为空或只剩一个元素为止。这样的排序方法是()。
A. 直接选择排序 B. 直接插入排序
C. 快速排序 D. 起泡排序
4. 若采用邻接矩阵法存储一个 n 个顶点的无向图,则该邻接矩阵是一个()。
A. 上三角矩阵 B. 稀疏矩阵
C. 对称矩阵 D. 对角矩阵
5. 在一个顺序存储的循环队列中,队头指针指向队头元素的()。
A. 前一个位置 B. 后一个位置
C. 队头元素位置 D. 队尾元素的前一位置

- 6. 用链表表示线性表的优点是()。
A. 便于随机存取
B. 花费的存储空间比顺序表少
C. 便于插入与删除
D. 数据元素的物理顺序与逻辑顺序相同
7. 对 5 个不同的数据元素进行直接插入排序,最多需要进行()次比较。
A. 8 B. 10
C. 15 D. 25
8. 下列存储形式中,()不是树的存储形式。
A. 双亲表示法 B. 左子女右兄弟表示法
C. 广义表表示法 D. 顺序表示法
9. 在一棵具有 5 层的满二叉树中结点总数为()。
A. 31 B. 32
C. 33 D. 16
10. 设有 100 个数据元素,采用折半搜索时,最大比较次数为()。
A. 6 B. 7
C. 8 D. 10

Table with 2 columns: 得分, 评卷人

二、判断题(判断下列各个叙述的正误。对,在题号前的括号内填入“√”;错,在题号前的括号内填入“×”。每小题 1 分,共 10 分)

- () 1. 算法的运行时间涉及加、减、乘、除、转移、存、取等基本运算。要想准确地计算总运行时间是不可行的。
() 2. 二维数组是数组元素为一维数组的线性表,因此它是线性结构。
() 3. 顺序表用一维数组作为存储结构,因此顺序表是一维数组。
() 4. 通常使用两个类来协同表示单链表,即链表的结点类和链表类。
() 5. 栈和队列都是顺序存储的线性表,但它们对存取位置的限制不同。
() 6. 在使用后缀表示实现计算器类时用到一个栈的实例,其作用是暂存运算对象。
() 7. 具有 n 个结点的完全二叉树的高度为 floor(log2 n) + 1。(n >= 0,根在第 0 层)
() 8. 为度量一个搜索算法的性能,需要在时间和空间方面进行权衡。
() 9. 闭散列法通常比开散列法时间效率更高。
() 10. 一棵 m 阶 B 树中每个结点最多有 m 个关键字,最少有 2 个关键字。

Table with 2 columns: 得分, 评卷人

三、阅读理解题(说明下列递归过程的功能。10 分)

```
void unknown(BinTreeNode *T,int a[],int i){
//指针 T 是完全二叉树的根指针。
if(T! = NULL){
a[i] = T->data;
unknown(T->leftChild,a,2*i+1);
unknown(T->rightChild,a,2*i+2);
}
}
主程序调用方式 unknown(BT.root,a,0);
//将完全二叉树所有结点从根开始,自顶向下,同一层自左向右连续编号,
//根结点的编号为 0。
```

Table with 2 columns: 得分, 评卷人

四、简答题(共 35 分)

1. 对下面的带权无向图采用 prim 算法从顶点 1 开始构造最小生成树。(写出加入生成树顶点集合 S 和选择边 Edge 的顺序)(10 分)

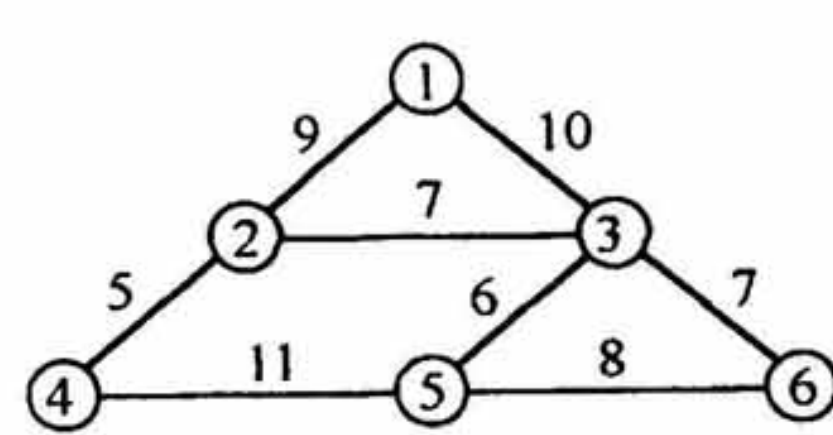


Table with 2 columns: S: 顶点号, Edge: (顶点, 顶点, 权值). Rows for S: 1, 2, 3, 4, 5, 6. Rows for Edge: (1,2,9), (1,3,10), (2,4,5), (2,3,7), (3,5,6), (3,6,7), (4,5,11), (5,6,8).

2. 某二叉树的结点数据采用顺序存储表示如下:

Table with 17 columns: 0-16. Row 1: E, A, F, D, H, C, G, I, B

- (1) 试画出此二叉树的图形表示。(3 分)
(2) 写出结点 D 的双亲结点及左、右子女。(3 分)
(3) 将此二叉树看作森林的二叉树表示,试将它还原为森林。(3 分)
3. 设待排序序列为 {10, 18, 4, 3, 6, 12, 1, 9, 15, 8}, 请给出用希尔排序每一趟的结果。增量序列取为 5, 3, 2, 1。(每一趟 2 分,共 8 分)
4. 设散列表的长度为 13,散列函数为 H(k) = k%13, 给定的关键字序列为 19, 14, 23, 01, 68, 20, 84, 27。试画出用线性探查法解决冲突时所构成的散列表。(8 分)

Table with 13 columns: 0-12. Row 1: (empty)

Table with 2 columns: 得分, 评卷人

五、综合算法题(每小题 5 分,共 15 分)

对于二维整数数组 A[m][n],对下列三种情况,分别编写相应的函数。

- (1) 求数组所有边缘元素的和。(5 分)
int sum1(int A[M][N], int m, int n)//M 和 N 分别大于等于 m 和 n
(2) 求从 A[0][0] 开始的互不相邻的所有元素的和。(5 分)
注:一个元素的八个方向上的第一个元素均为相邻元素。
int sum2(int A[M][N], int m, int n)
(3) 假定 m = n,请分别计算正、反两条对角线上的元素的和。(5 分)
int sum3(int A[M][N], int n)

Table with 2 columns: 得分, 评卷人

六、填空题(每空 2 分,共 10 分)

已知一棵完全二叉树存放于一个一维数组 T[n]中,T[n]中存放的是各结点的值。下面的算法的功能是:从 T[0] 开始顺序读出各结点的值,建立该二叉树的二叉链表表示。

```
#include<iostream.h>
typedef struct node {
int data;
struct node * leftChild, * rightChild;
} BinTreeNode;
typedef BinTreeNode * BinaryTree;
void ConstructTree(int T[], int n, int i, BinTreeNode * &ptr);
int main(void){
BinaryTree t; int n;
cout << "Please enter the number of node : \n"; cin >> n;
int * A = new int[n];
for(int i=0; i<n; i++) ①; //从键盘输入结点值
for(i=0; i<n; i++) cout << A[i];
cout << endl;
ConstructTree(A, n, 0, t); //以数组建立一棵二叉树
②; //删除数组 A
return 1;
}
void ConstructTree(int T[], int n, int i, BinTreeNode * &ptr)
{
if(i >= n) ③; //置根指针为空
```

```
else
ptr = new BinTreeNode;
ptr->data = T[i];
ConstructTree(T, n, 2*i+1, ④);
ConstructTree(T, n, ⑤, ptr->rightChild);
}
}
缺失的语句为:
①
②
③
④
⑤
```

计算机科学与技术专业数据结构试题答案及评分标准

(供参考)

2002 年 1 月

一、单选题(每小题 2 分,共 20 分)

- 1. B 2. B 3. C 4. D 5. A
6. C 7. B 8. D 9. A 10. B

二、判断题(每个 1 分,共 10 分)

- 1. √ 2. × 3. × 4. √ 5. √
6. √ 7. × 8. √ 9. × 10. ×

三、阅读理解题(说明下列递归过程的功能。10 分)

将用二叉链表表示的完全二叉树转换为二叉树的顺序(数组)表示。

四、简答题(共 35 分)

1. (10 分) 每加对一个顶点和一条边得 2 分,全对 10 分。

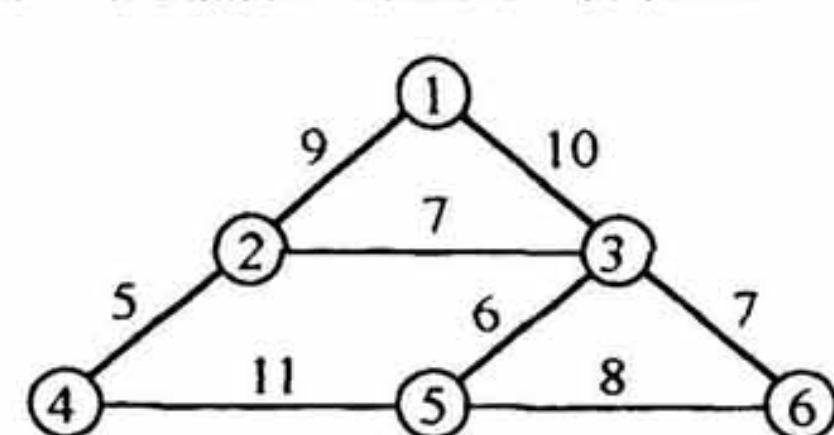


Table with 2 columns: S: 顶点号, Edge: (顶点, 顶点, 权值). Rows for S: 1, 2, 3, 4, 5, 6. Rows for Edge: (1,2,9), (1,3,10), (2,4,5), (2,3,7), (3,5,6), (3,6,7), (4,5,11), (5,6,8).

Table with 2 columns: 得分, 评卷人

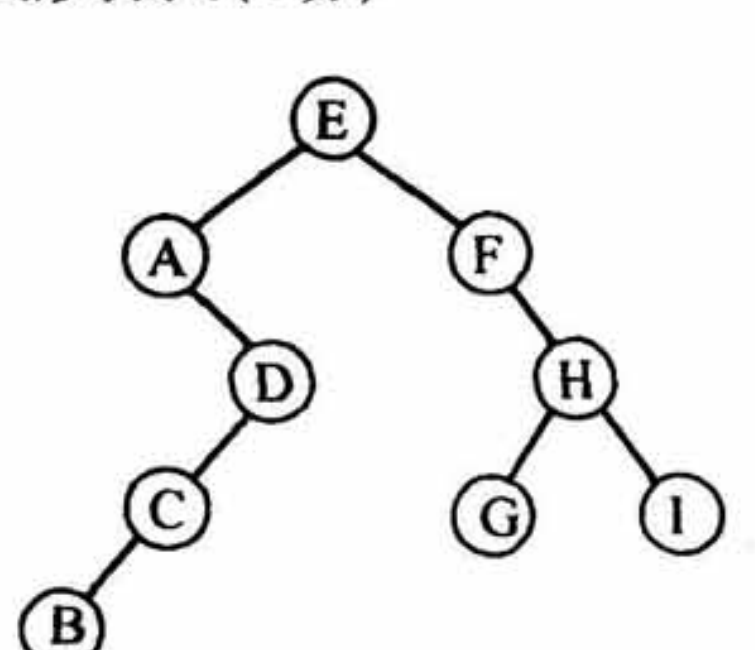
三、阅读理解题(说明下列递归过程的功能。10 分)

```
void unknown(BinTreeNode *T,int a[],int i){
//指针 T 是完全二叉树的根指针。
if(T! = NULL){
a[i] = T->data;
unknown(T->leftChild,a,2*i+1);
unknown(T->rightChild,a,2*i+2);
}
}
主程序调用方式 unknown(BT.root,a,0);
//将完全二叉树所有结点从根开始,自顶向下,同一层自左向右连续编号,
//根结点的编号为 0。
```

Table with 2 columns: 得分, 评卷人

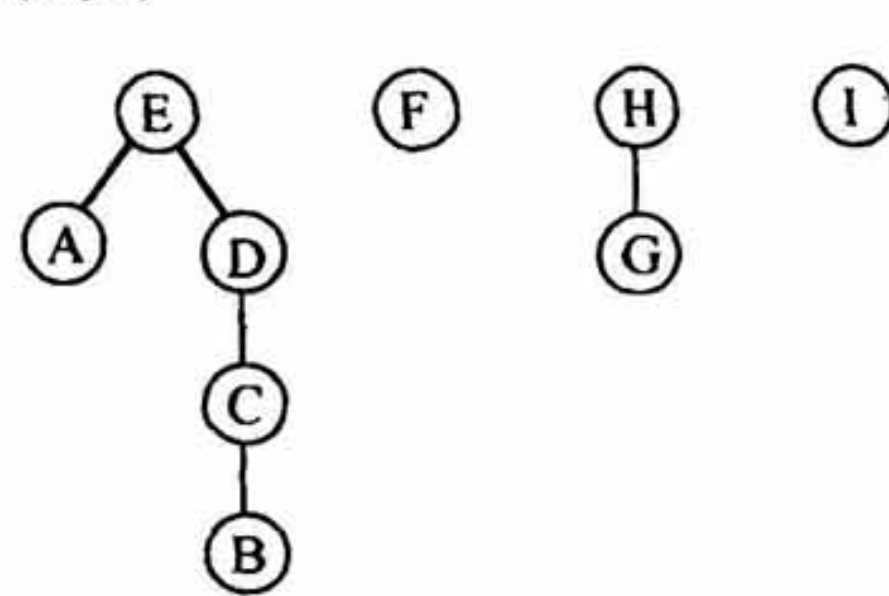
四、简答题(共 35 分)

2. (1) 二叉树的图形表示:(3 分)

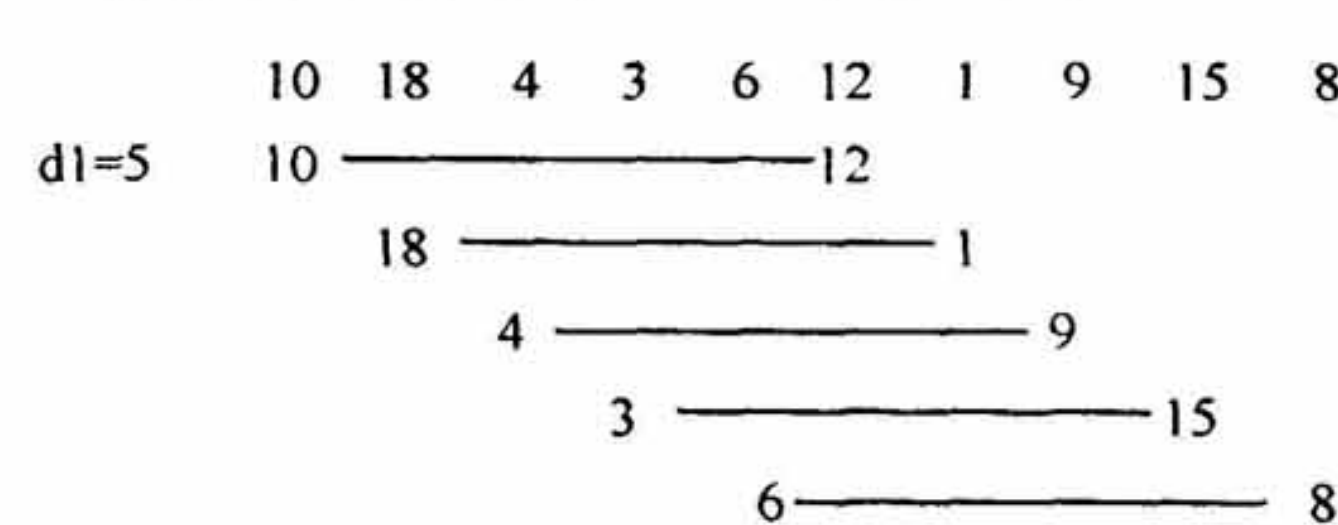


(2) 结点 D 的双亲结点为 A,左子女结点为 C,右子女为空。(3 分)

(3) 对应森林为:(3 分)



3. 各趟排序结果如下:(每一趟 2 分,共 8 分)



2. 某二叉树的结点数据采用顺序存储表示如下:

Table with 17 columns: 0-16. Row 1: E, A, F, D, H, C, G, I, B

- (1) 试画出此二叉树的图形表示。(3 分)
(2) 写出结点 D 的双亲结点及左、右子女。(3 分)
(3) 将此二叉树看作森林的二叉树表示,试将它还原为森林。(3 分)
3. 设待排序序列为 {10, 18, 4, 3, 6, 12, 1, 9, 15, 8}, 请给出用希尔排序每一趟的结果。增量序列取为 5, 3, 2, 1。(每一趟 2 分,共 8 分)
4. 设散列表的长度为 13,散列函数为 H(k) = k%13, 给定的关键字序列为 19, 14, 23, 01, 68, 20, 84, 27。试画出用线性探查法解决冲突时所构成的散列表。(8 分)

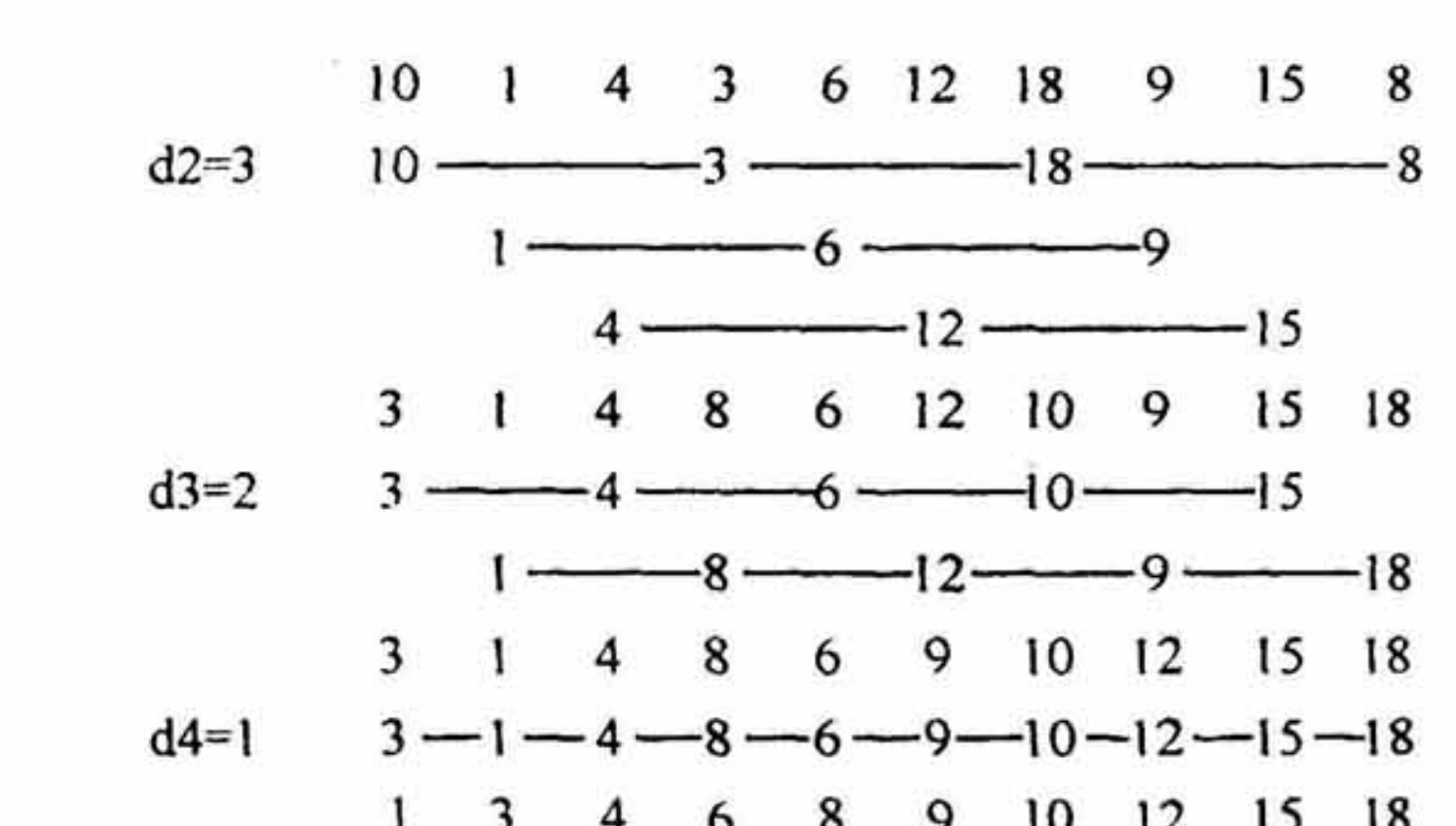
Table with 13 columns: 0-12. Row 1: (empty)

Table with 2 columns: 得分, 评卷人

五、综合算法题(每小题 5 分,共 15 分)

对于二维整数数组 A[m][n],对下列三种情况,分别编写相应的函数。

- (1) 求数组所有边缘元素的和。(5 分)
int sum1(int A[M][N], int m, int n)//M 和 N 分别大于等于 m 和 n
(2) 求从 A[0][0] 开始的互不相邻的所有元素的和。(5 分)
注:一个元素的八个方向上的第一个元素均为相邻元素。
int sum2(int A[M][N], int m, int n)
(3) 假定 m = n,请分别计算正、反两条对角线上的元素的和。(5 分)
int sum3(int A[M][N], int n)



4. 计算各关键字得到的散列地址(8 分)

Table with 2 columns: 关键字, 散列地址. Rows: (19, 6), (14, 1), (23, 10), (01, 1), (68, 3), (20, 7), (84, 6), (27, 1)

在散列表中散列结果

Table with 13 columns: 0-12. Row 1: 14, 01, 68, 27, 19, 20, 84, 23

评分标准:每对一个元素得 1 分,全对 8 分。

五、综合算法题(每小题 5 分,共 15 分)

(1) 本小题是计算数组 A 的最外圈的 4 条边的所有元素之和。可以先累加各个靠边的元素的值,再减去位于 4 个角上重复相加的元素的值。

```
int sum1(int A[M][N], int m, int n){
int s=0, i, j;
for(i=0; i<m; i++) s+=A[i][0]; s+=A[i][n-1];
for(j=0; j<n; j++) s+=A[0][j]; s+=A[m-1][j];
s=s-A[0][0]-A[0][n-1]-A[m-1][0]-A[m-1][n-1];
return s;
}
```

评分标准:5 分,根据情况酌情给分。

Table with 2 columns: 得分, 评卷人

六、填空题(每空 2 分,共 10 分)

已知一棵完全二叉树存放于一个一维数组 T[n]中,T[n]中存放的是各结点的值。下面的算法的功能是:从 T[0] 开始顺序读出各结点的值,建立该二叉树的二叉链表表示。

```
#include<iostream.h>
typedef struct node {
int data;
struct node * leftChild, * rightChild;
} BinTreeNode;
typedef BinTreeNode * BinaryTree;
void ConstructTree(int T[], int n, int i, BinTreeNode * &ptr);
int main(void){
BinaryTree t; int n;
cout << "Please enter the number of node : \n"; cin >> n;
int * A = new int[n];
for(int i=0; i<n; i++) ①; //从键盘输入结点值
for(i=0; i<n; i++) cout << A[i];
cout << endl;
ConstructTree(A, n, 0, t); //以数组建立一棵二叉树
②; //删除数组 A
return 1;
}
void ConstructTree(int T[], int n, int i, BinTreeNode * &ptr)
{
if(i >= n) ③; //置根指针为空
```

(2) 本小题的互不相邻是指上、下、左、右、对角线均互不相邻,即求第 0, 2, 4, ... 的各行中第 0, 2, 4, ... 列的所有元素的值之和。

```
int sum2(int A[M][N], int m, int n){
int s=0, i, j;
for(i=0; i<m; i+=2)
for(j=0; j<n; j+=2)
s+=A[i][j];
return s;
}
```

评分标准:5 分,根据情况酌情给分。

(3) 本小题中一条对角线是 A[i][i], i=0, 1, ..., n-1; 另一条对角线是 A[i][n-i-1], i=0, 1, ..., n-1。可以用循环实现。

```
int sum3(int A[M][N], int n){
int s=0, i;
for(i=0; i<n; i++)
{s+=A[i][i]; s+=A[i][n-i-1];}
return s;
}
```

评分标准:5 分,根据情况酌情给分。

六、填空题(每空 2 分,共 10 分)

- 缺失的语句为:①in > A[i] ②delete[] A ③ptr = NULL
④ptr->leftChild ⑤2 * i + 2