

## 华北计算技术研究所 2006 年专业课试题及参考答案

### 一、填空题（15 分）

1. 数据结构是相互之间存在一种或多种特定关系的 数据元素 的集合。通常有下列四种基本结构：集合、线性结构、树状结构 和 图状结构（或网状结构）。
2. 在顺序表中插入或删除一个元素，需要平均移动 表中一半（或  $n/2$  个） 元素，具体移动的元素个数与 表长和该元素在表中的位置 有关。
3. 0 个字符的串称为空串，它的长度为 0。
4. 矩阵压缩存储的基本思想是：值相同 的多个元素只分配一个存储空间，零元素 不分配空间。
5. 深度为  $k$  的二叉树至多有  $2^k - 1$  个结点，至少有  $k$  个结点。
6. 图的深度优先搜索遍历类似于树的 先根 遍历；图的广度优先搜索遍历类似于树的 按层次 遍历。

### 二、选择题（20 分）

1. 时间复杂性最好，即执行时间最短的是：B  
(A)  $O(n)$  (B)  $O(\log_2 n)$  (C)  $O(n \log_2 n)$  (D)  $O(n^2)$
2. 具有 6 个顶点的无向图至少有 D 条边才能确保是一个连通图。  
(A) 15 (B) 7 (C) 6 (D) 5
3. 在所有排序方法中，关键字比较的次数与记录的初始排列次序无关的是：D  
(A) 希尔排序 (B) 起泡排序 (C) 插入排序 (D) 选择排序
4. 若用一个大小为 6 的数组来实现循环队列，且当前 rear 和 front 的值分别为 0 和 3，当从队列中删除一个元素，再加入两个元素后，rear 和 front 的值分别为：C。  
(A) 1 和 5 (B) 2 和 4 (C) 4 和 2 (D) 5 和 1
5. 设栈的长度为 3，入栈序列为 A、B、C、D、E、F，不可能产生的出栈序列是：D。  
(A) A, B, C, D, E, F (B) B, A, D, C, F, E  
(C) C, B, A, F, E, D (D) D, C, B, A, F, E

### 三、判断题。（10 分）

请判断下列说法的对错。

1. 数据元素是数据的最小单位。 错
2. 串的三种存储表示方法为定长顺序存储表示、堆分配存储表示和块链存储表示。 对
3. 一个稀疏矩阵的转置矩阵仍然是稀疏矩阵。 对
4. 树状结构中，度为 0 的结点称为树根。 错
5. 完全二叉树不一定是满二叉树，但满二叉树一定是完全二叉树。 对

四、根据下列要求分别编写算法。(20 分)

1. 设计算法，判断一个算术表达式的圆括号是否正确配对。

参考答案：

```
#include <string.h>
#include "stack.h"
Int PairBracket(char *S)
{
    //检查表达式中括号是否配对
    int i;
    SeqStack T; //定义一个栈
    InitStack (&T);
    for(i=0;i<strlen(S);i++)
    {
        if ( S[i]=='(' ) Push(&T,S[i]); //遇'('时进栈
        if ( S[i]==')' ) Pop(&T); //遇')'时出栈
    }
    Return !EmptyStack(&T); //由栈空否返回正确配对与否
}
```

2. 已知一棵完全二叉树存于顺序表 sa 中，sa.elem[sa.last]中存放各结点的数据元素。编写算法由此顺序存储结构建立该二叉树的二叉链表。

参考答案：

```
Status CreateBitree_SqList(Bitree &T, SpList sa)    //根据顺序存储结构建立
二叉链表
```

```

{
    Bitree ptr[sa.last+1]; //该数组存储与 sa 中各结点对应的树指针
    if(!sa.last)
    {
        T=NULL;
        return;
    }
    ptr[1]=(BTNode*)malloc(sizeof(BTNode));
    ptr[1]->data=sa.elem[1];
    T=ptr[1];
    for(i=2;i<=sa.last;i++)
    {
        if(!sa.elem[i] return ERROR;
        ptr[i]=(BTNode*)malloc(sizeof(BTNode));
        ptr[i]->data=sa.elem[i];
        j=i/2;
        if(i-j*2) ptr[j]->rchild=ptr[i]; //i 是 j 的右孩子
        else ptr[j]->lchild=ptr[i]; //i 是 j 的左孩子
    }
    return OK;
}

```

## 五、回答问题。(20 分)

1. 设有上三角矩阵 $(a_{ij})_{n \times n}$ ，将其上三角元素逐行存于数组  $B[m]$  中 ( $m$  充分大)，使得  $B[k]=a_{ij}$  且  $k=f_1(i)+f_2(j)+c$ 。试推导出函数  $f_1$ ， $f_2$  和常数  $c$  (要求  $f_1$  和  $f_2$  中不含常数项)。

参考答案：

$$k = ni - (n - j) - \frac{i(i-1)}{2} - 1, (i \leq j)$$

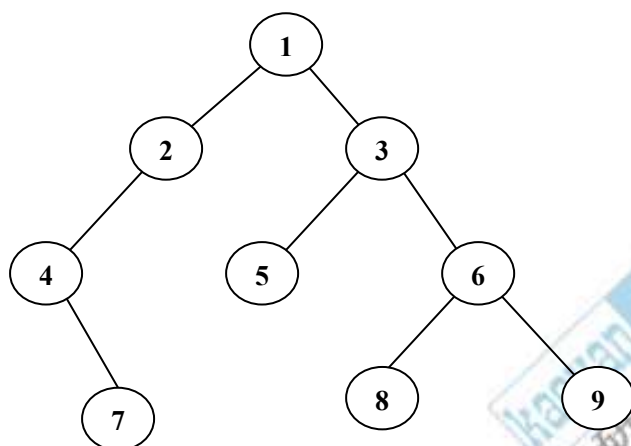
则得：

$$f_1(i) = (n + \frac{1}{2})i - \frac{1}{2}i^2$$

$$f_2(j) = j$$

$$c = -(n + 1)$$

2. 写出下图中所示的二叉树的先序序列、中序序列和后序序列。



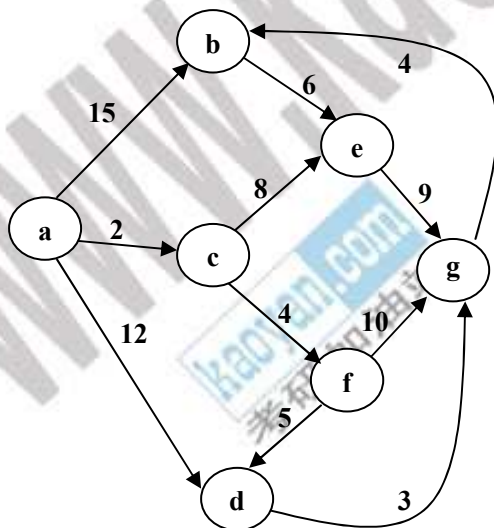
参考答案:

前序: 124735689

中序: 472153869

后序: 742589631

六、 下图是一个有向图,其中每条弧段上的数字表示该弧段的权值。利用 Dijkstra 算法求图中从顶点 a 到其他各顶点间的最短路径, 写出执行算法过程中各步的状态。(15 分)



参考答案:

终点 Dist	b	c	d	e	f	g	S
K=1	15 (a,b)	<b>2</b> <b>(a,c)</b>	12 (a,d)				{a,c}
K=2	15 (a,b)		12 (a,d)	10 (a,c,e)	<b>6</b> <b>(a,c,f)</b>		{a,c,f}
K=3	15 (a,b)		11 (a,c,f,d)	<b>10</b> <b>(a,c,e)</b>		16 (a,c,f,g)	{a,c,f,e}
K=4	15 (a,b)		<b>11</b> <b>(a,c,f,d)</b>			16 (a,c,f,g)	{a,c,f,e,d}
K=5	15 (a,b)					<b>14</b> <b>(a,c,f,d,g)</b>	{a,c,f,e,d,g}
K=6	<b>15</b> <b>(a,b)</b>						{a,c,f,e,d,g,b}

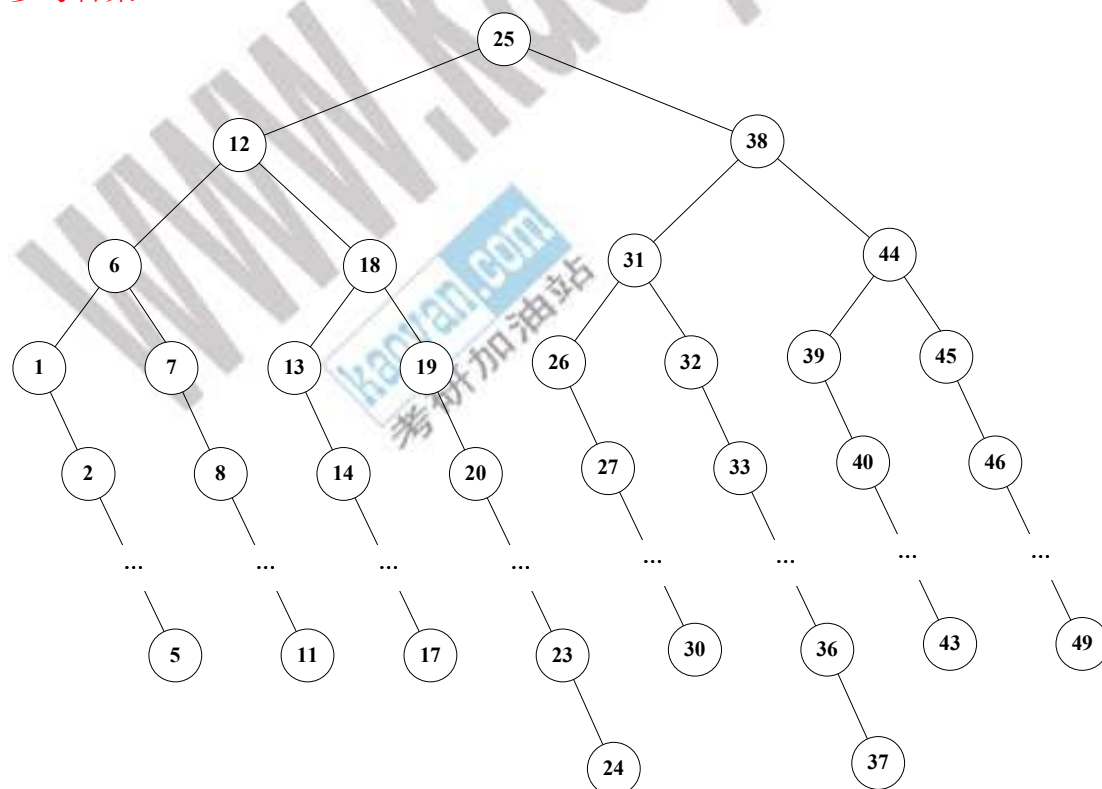
故：

a 到各点最短路径分别为：

b	c	d	e	f	g
15	2	11	10	6	14

七、假设按下述递归方法进行顺序表的查找：若表长 $\leq 10$ ，则进行顺序查找，否则进行折半查找。试画出对表长 $n=50$ 的顺序表进行上述查找时，描述该查找的判定树，并求出在等概率情况下查找成功的平均查找长度。（20 分）

参考答案：





其等概率查找时查找成功的平均查找长度为：

$$ASL_{succ} = \frac{1}{50} (1 \times 1 + 2 \times 2 + 3 \times 4 + (4 + 5 + 6 + 7 + 8) \times 8 + 9 \times 3) = 5.68$$

八、将下列 C++ 程序的类定义和主函数分离，改成多文件结构。主函数使用类的方式采取包含类定义头文件的方法。并写出运行结果。（15 分）

```
#include <iostream.h>
class Cat
{
    public:
        int GetAge( );
        void SetAge(int age);
        void Meow( ); //喵喵叫
    protected:
        int itsAge;
};

int Cat::GetAge( )
{
    return itsAge;
}

void Cat::SetAge(int age)
{
    itsAge=age;
}

void Cat::Meow( )
{
    cout<<"Meow.\n";
}

void main( )
{
    Cat frisky;
    Frisky.SetAge(5);
    Frisky.Meow( );
    Cout<<"frisky is a cat who is "
        <<frisky.GetAge( )
        <<"years old.\n";
    frisky.Meow( );
}
```

参考答案：

//头文件 cat.h

#include <iostream.h>

class Cat

{

public:

int GetAge( );

void SetAge(int age);

void Meow( ); //喵喵叫

protected:

int itsAge;

};

int Cat::GetAge( )

{

return itsAge;

}

void Cat::SetAge(int age)

{

itsAge=age;

}

void Cat::Meow( )

{

cout<<"Meow.\n";

}

//主程序

#include <iostream.h>

#include "cat.h"

void main( )

```
{
    Cat frisky;
    Frisky.SetAge(5);
    Frisky.Meow( );
    Cout<<"frisky is a cat who is "
        <<frisky.GetAge( )
        <<"years old.\n";
    frisky.Meow( );
}
```

运行结果：

Meow.

Frisky is a cat who is 5 years old.

Meow.

九、用 C++语言编写应用程序。定义一个 Document 类，包含成员变量 name。  
从 Document 类派生出 Book 类，增加 PageCount 变量。程序运行输出为：  
Name of Book: Book1。（15 分）

参考答案：

```
#include <iostream.h>
```

```
#include <string.h>
```

```
class Document
```

```
{
```

```
public:
```

```
    Document(){ };
```

```
    Document(char *name);
```

```
    char *Name; //Document name.
```

```
    void PrintNameOf(); //Print name.
```

```
};
```

```
Document::Document(char *name)
```

```
{
```

```
    Name=new char[strlen(name)+1];
```

```
    Strcpy(Name,name);
```



```
};

void Document::PrintNameOf()
{
    cout<<Name<<endl;
}

class Book:public Document
{
public:
    Book(char *name,long pagecount);
    void PrintNameOf();
private:
    long PageCount;
};

Book::Book(char *name,long pagecount):Document(name)
{
    PageCount=pagecount;
}

void Book::PrintNameOf()
{
    cout<<"Name of book:";
    Document::PrintNameOf();
}

void main()
{
    Document a("Document");
    Book b("Book1",100);
    b.printNameof();
}
```