

西南交通大学 2006 年年硕士研究生入学考试试题解析

试题名称：程序设计与数据结构

一、填空题

1. (0)

解析：从右向左依次赋值计算

2. (17)

解析：先执行++p 指向 a[1]的地址，*q=3,++*q=4

3. (0) (48)

解析：讲“E”转化为 ASCII 的值来计算，这是一个逗号运算符，从左到右计算，最后表达式的值为最右边的一个

4. (7)

解析： $x > y ? x : y$ ，如果 $x > y$ 为真，则值为 x ，否则为 y

5. $\text{sqrt}((y_1 - y_2) * (y_1 - y_2) + (x_1 - x_2) * (x_1 - x_2))$

解析：A、B 的距离公式 $\sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2}$ ，再用 C 语言表达式写出

6. (数组类型、结构体类型、构造体类型)

解析：见课本 P37

7. (\0)

解析：C 语言用 \0 表示字符串的结束

8. (void)

9. (数据首元素地址)

解析：数组名可以作为指针常量

10. (100)

解析：每次循环 sum 被重新赋值 0

11. (d)

解析：数据元素是处理数据的最小单位

12. (n) ($\lfloor \log_2 n \rfloor + 1$)

解析：单支树时深度对最大，完全二叉树时，深度最小

13. $O(\log_2 n)$ n

解析：见 P232

14. (YES)

解析：见 P233

15. (1)

解析：二叉平衡树上的所有结点的平衡因子只可能是 -1, 0, 1

16. (一样)

解析：平衡二叉树查找的过程和排序树相同，二排序树又类似于折半查找

17. (集合) (线性结构) (树形结构) (网状结构)

解析：见课本 P5

18. (插入) (删除) (查找) (修改) (排序)

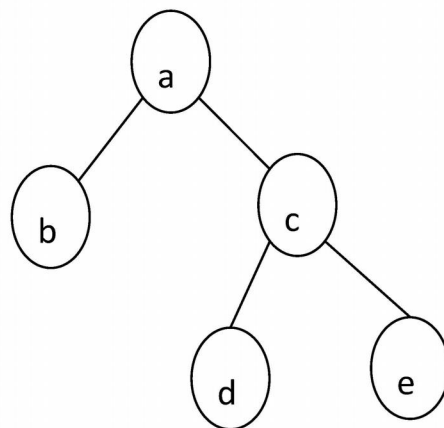
19. (YES)
20. (只在表尾一端进行插入、删除操作)
解析：根据栈的定义

二、单项选择题

1. (C)
解析：见课本 P335
2. (D)
解析：c 语言表达式中只有 && 表示并列关系
3. (D)
解析：共用变量中有三种数据类型，故内存存储空间应为三种之和
4. (C)
解析：*s[1] 是用来存放 hello 这个字符的首地址，而 *s[2] 用来寻访 world 这个字符的首地址
5. (D)
解析：*p=a 表示将数组首地址给了指针 p，p+3 表示元素 a[3] 的地址，所以它和 &a[3] 是等价
6. (B)
解析：p 表示二维数组的首地址，而 M*6+N 表示元素所在的位置
7. (D)
解析：当执行 k=g 时，由于 k 为 int 型，所以 k 为 3，又赋值给 float 后值为 3.0
8. (D)
解析：循环条件永为真，故为死循环
9. (D)
解析：++x 表示先加 1 后执行 return x，故输出为 1,2,3，主函数循环了三次，故输出三个数
10. (C)
解析：这是 Func(x) 的递归调用， $7*4*1*(-2)=-56$

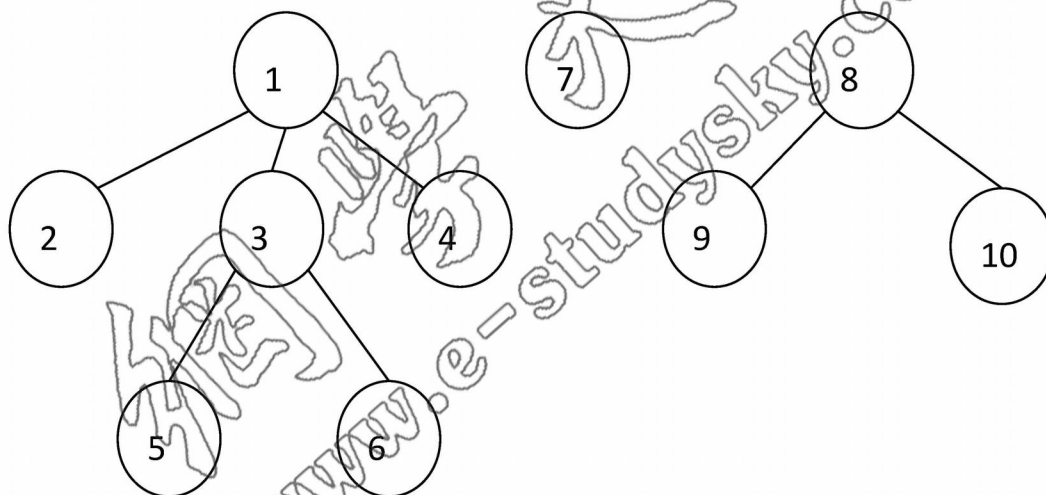
三、简要回答题

1.
解析：Next[j]=0 1 1 1 2 2 1 2 3 1
见课本 P82-P84
2.
解析：二叉树如下



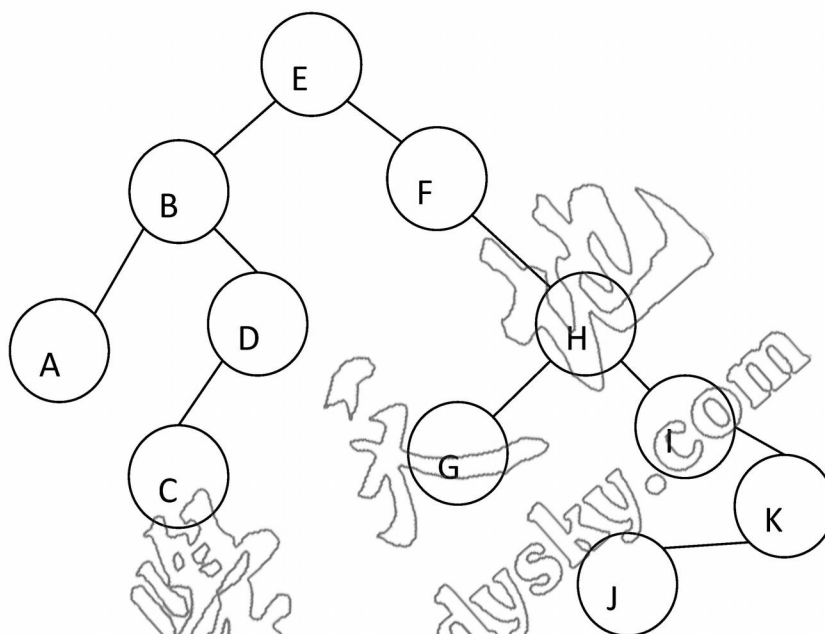
D

3.解析：根据左孩子、右兄弟的思想，将二叉树→树，从而可得到相应的森林

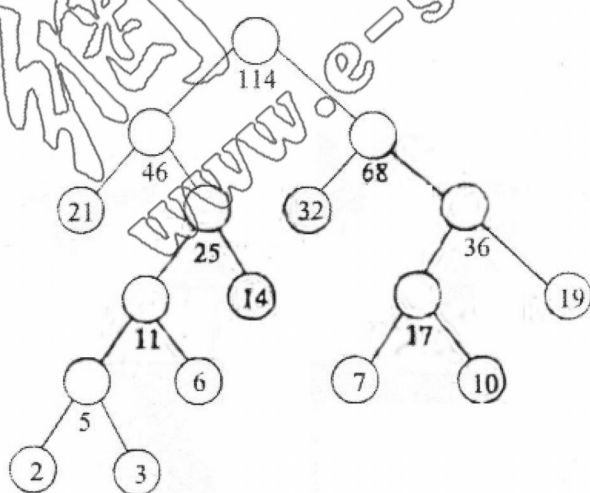


4. 解析：一趟快速排序算法的时间复杂度为 $O(n \log_2 n)$

5. 解析：二叉树如下

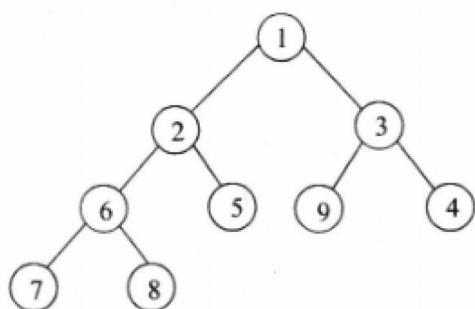


6.解析：最优二叉树如下



7.

解析：堆为



建堆得过程见课本 P28

8.解析：2i+j-3

参考 2003 年二、3)

四、阅读程序

1. 10

2. bb bc Dc

HOW

3. *ARE*

YUE

五、程序填空

1. i<100 i%7!=0

2. " c:\F.txt","r" (fp=fopen(" c:\F.txt","r"))==NULL (c=='\n'||c=='\t')

3. st.length

low<=high

(low+high)

mid-1

low=mid+1

4. 65,30,18,10,43,90,70,50,80,20

5

解析：一趟快速排序算法，见课本 P275

函数值是返回的 low 所在的位置，即为 5

六、程序设计

1.解析：

```
#include<stdio.h>
```

```
int mail()
```

```
{
```

```
double x,y;
```

```
Printf( "请输入 x 的值: " );
```

```
Scanf( "%1f" ,&x);
```

```
if(x<1.5)
```

```
{printf( "y=%f" ,x);}
```

```
else if(x<1.8 & &x>=1.5)
```

```
{
```

```
double m=3*x-5.5;
```

```
Printf( "y=%5.1f" ,m)
```

```
}
```

```
else if(x>=1.8 & &x<80)
```

```
{
```

```
double r=6*x+11.0;
```

```
Printf( "y=%5.1f" ,r);
```

```
}
```

```
}
```

2.解析:

```
int digit (int n,int k)
{
    char ch[10];
    int value,index=0;
    while((value=n%10)!=0)
    {
        ch[index]=(char)value;
        n=n/10;index++;
    }
    if(k>index||k==0)
        return -1;
    return ch[index-k];
}
```

3.解析:

```
struct MyNode *ChangeHead(struct MyNode *pHead)
{
    struct MyNode*p,*q;
    p=pHead->pNext;
    while(p)
    {
        q=p->pNext;
        p->pNext=pHead;
        pHead=p;
        p=q;
    }
    return pHead;
}
```

4.解析:

```
#include<stdio.h>
#include<ctype.h>
void main()
{
    char ch;
    int numberofword=0;
    wordstart=0;
    FILE*fp1=fopen( "test.tet" ,"r");
    FILE*fp2=fopen( "result.txt" ,"w");
    if(fp1==NULL||fp2==NULL)
    {
        puts( "cannot open file!" );
        return;
    }
}
```

```
while(!feof(fp1))
{
    ch=fgetc(fp1);
    if(isalpha(ch)&&wordstart==0)
        Wordstart=1;
    else if(!isalpha(ch)&&wordstart==0)
        {numberofword++;wordstart=0;}
    fprintf(fp2,"%d",numberofword);
    fclose(fp1);
    fclose(fp2);
}
}
```

5.解析：

```
leaf_Num(BiTree T)
{
    if(NULL)return 0;
    else if(T->lchild==NULL && T->rchild==NULL)
        return 1;
    else
        return leaf_Num(T->lchild)+leaf_Num(T->rchild);
}

int Depth(BiTree)
{
    int m,n;
    if(T=NULL) return 0;
    else
        m=Depth(T->lchild);
        n=Depth(T->rchild);
        return (m>n?m:n)+1;
}
```

6.解析：A)

```
typedef struct
{
    ElemType*p;
    int Maxlength;
}
```

B)

```
struct student
{
    char Name[10];
    float Score;
    int addr;
}
```

C)

typedef struct

{ElemType*p;

int Maxlength;

struct struct s T;

}HashTable;

D)线性探测再散列

E)

status InsertHash(ElemType e)

{

int index=0,count=0;

if(!SearchHash(e,index,count)&&count<Maxlength)

{

p[index]=e;

count++;

return success;

}

return duplicate;

}

F)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Zhang Yang											Li Pin	Luo Hao										Wang Lin	Wu Xin	Weng Kai	Zhao Gang
2											1	1										1	2	3	1