

2003 年试题参考答案

一、数据结构

1、参考答案：

广义表：广义表是 $n(n \geq 0)$ 个元素 $a_1, a_2, \dots, a_i, \dots, a_n$ 的有限序列。其中， a_i 或者是原子或者是一个广义表。

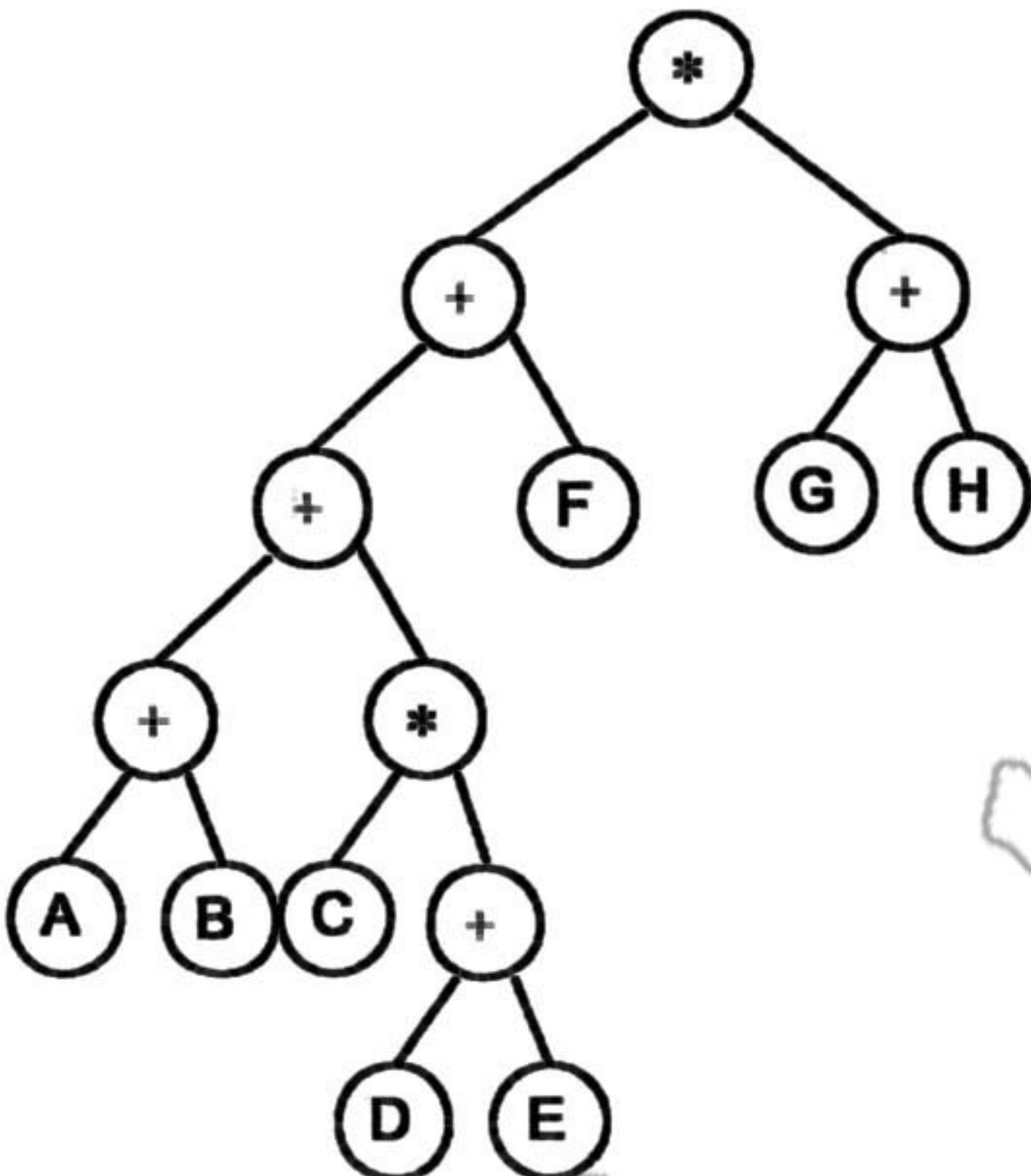
线性表：线性表是具有相同类型的 $n(n \geq 0)$ 个数据元素 a_0, a_1, \dots, a_{n-1} 组成的有限序列。其中 n 称为线性表的长度，当 $n=0$ 时称为空线性表， $n>0$ 时称为非空表。

2、参考答案：

步骤	操作数栈	运算符栈	动作
1	A		A 进操作数栈
2	A	-	- 进运算符栈
3	AB	-	B 进操作数栈
4	AB	-*	* 进运算符栈
5	ABC	-*	C 进操作数栈 归约，令 $T_1 = B * C$
6	AT_1	-/	/ 进运算符栈
7	AT_1D	-/	D 进操作数栈 归约，令 $T_2 = T_1 / D$

8	AT_2		归约，令 $T_3=A-T_2$
9	T_3	+	+进运算符栈
10	T_3E	+	E进操作数栈 归约，令 $T_4=T_3+E$
11	T_4	#	输出 T_4

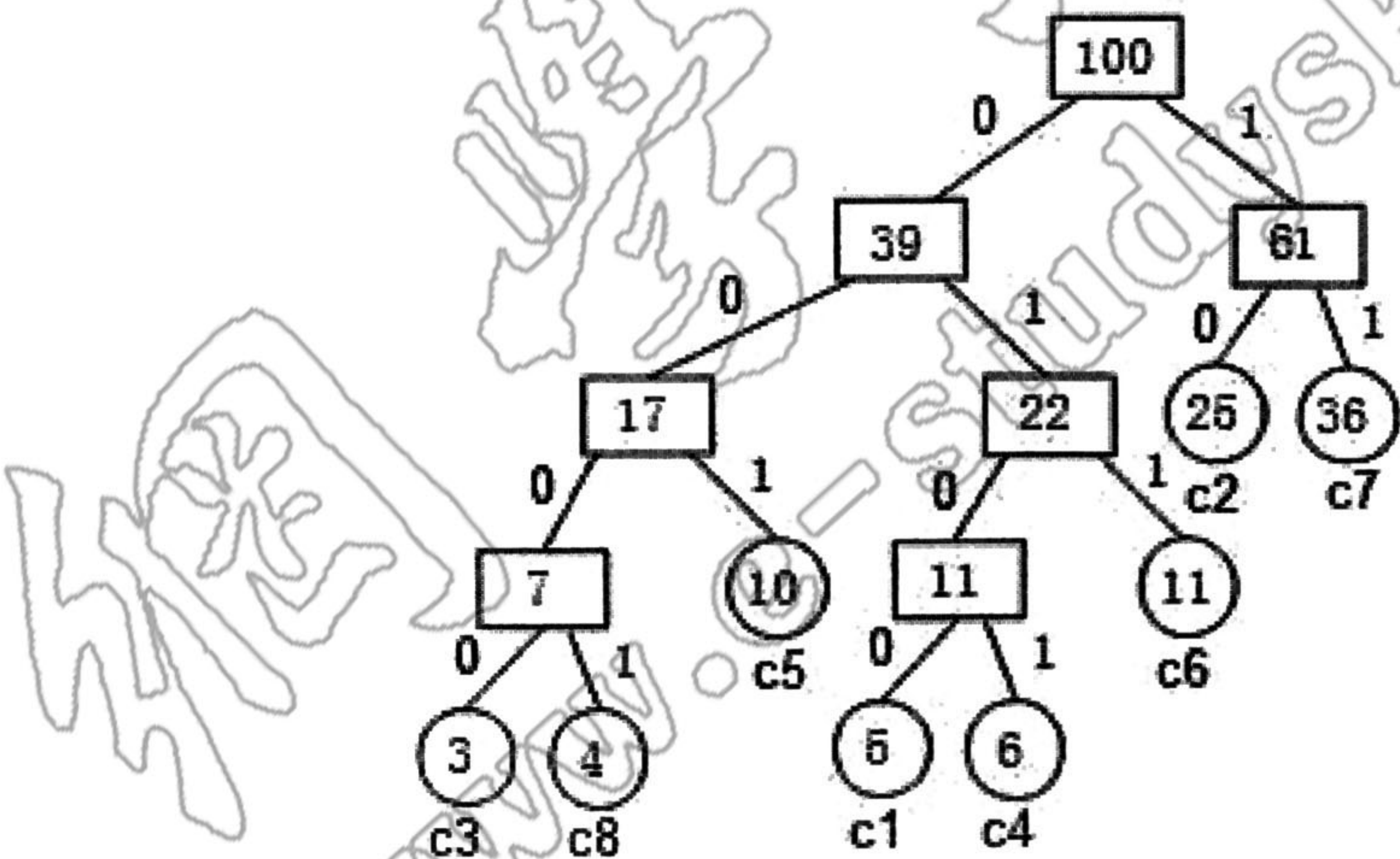
3、参考答案：



后序遍历：A B + C D E + * + F + G H + *

4、参考答案：

哈弗曼树：



哈弗曼编码：

字母	频率	码字
c1	5	0100
c2	25	10
c3	3	0000
c4	6	0101
c5	10	001
c6	11	011
c7	36	11
c8	4	0001

总码数 WPL=2*(25+36)+3*(10+11)+4*(5+3+6+4)=257

5、参考答案：

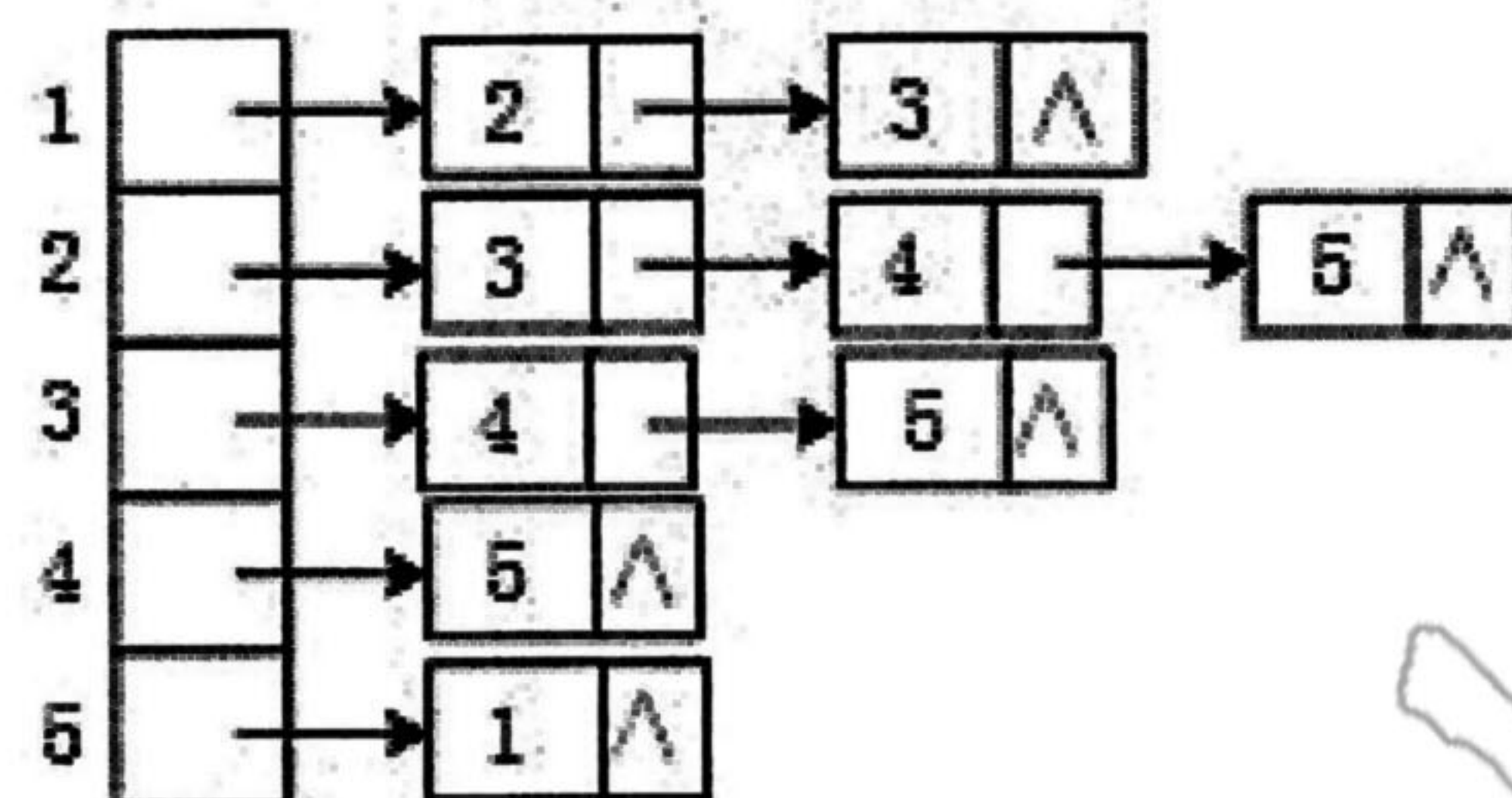
查找 $K=84$ ： $H(84)=84 \text{ MOD } 13=6$ ，对比 $\text{elem}[6]=19 \neq 84$ ，继续对比 $\text{elem}[7]=20 \neq 84$ ，继续对比 $\text{elem}[8]=84$ ，查询成功；

查找 $K=38$ ： $H(38)=38 \text{ MOD } 13=12$ ，对比 $\text{elem}[12]=10 \neq 38$ ，继续对比 $\text{elem}[13]$ 为空，查找失败。

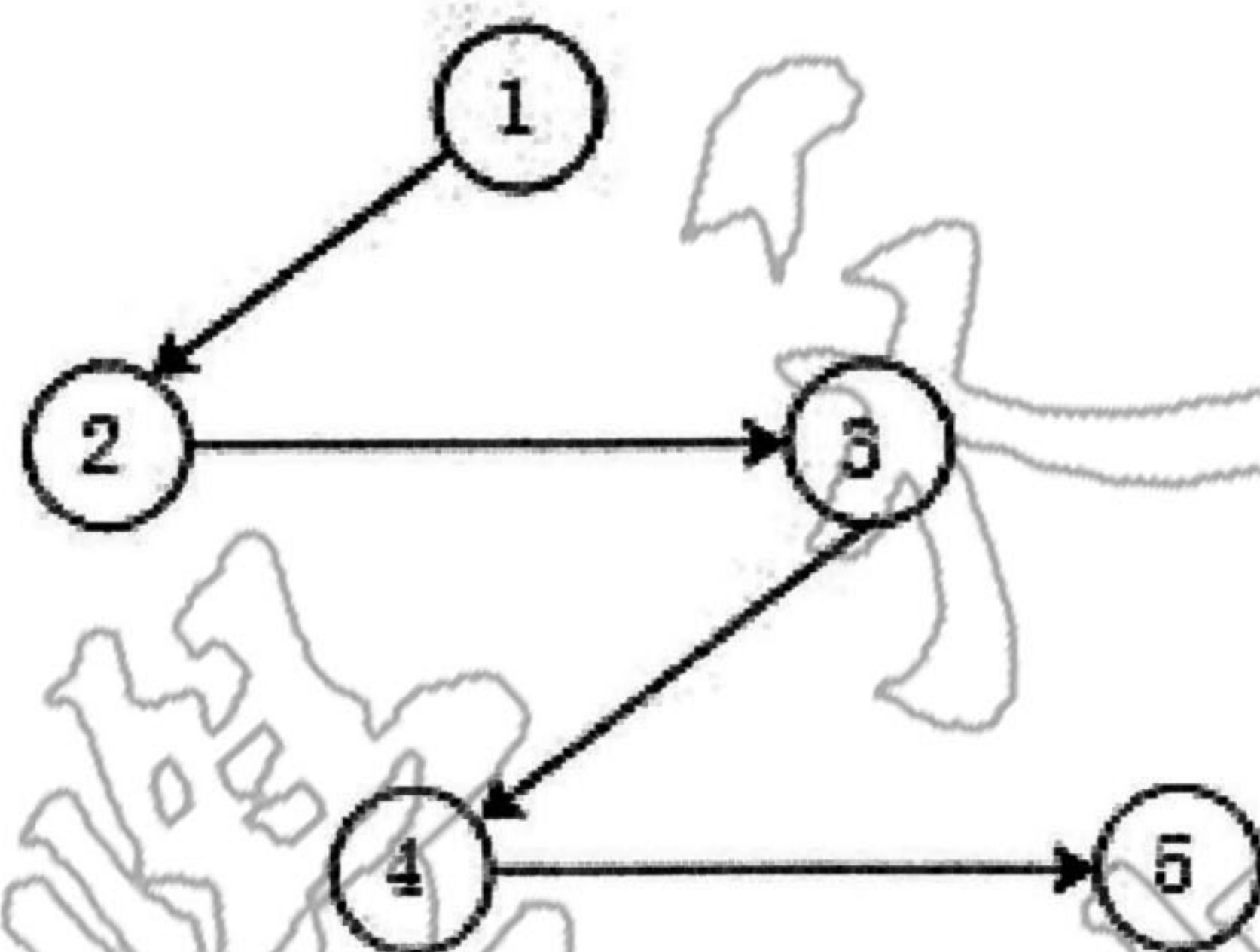
6、参考答案：参照 98 真题程序设计题第 10 题改进后的“冒泡”排序算法。

7、参考答案：

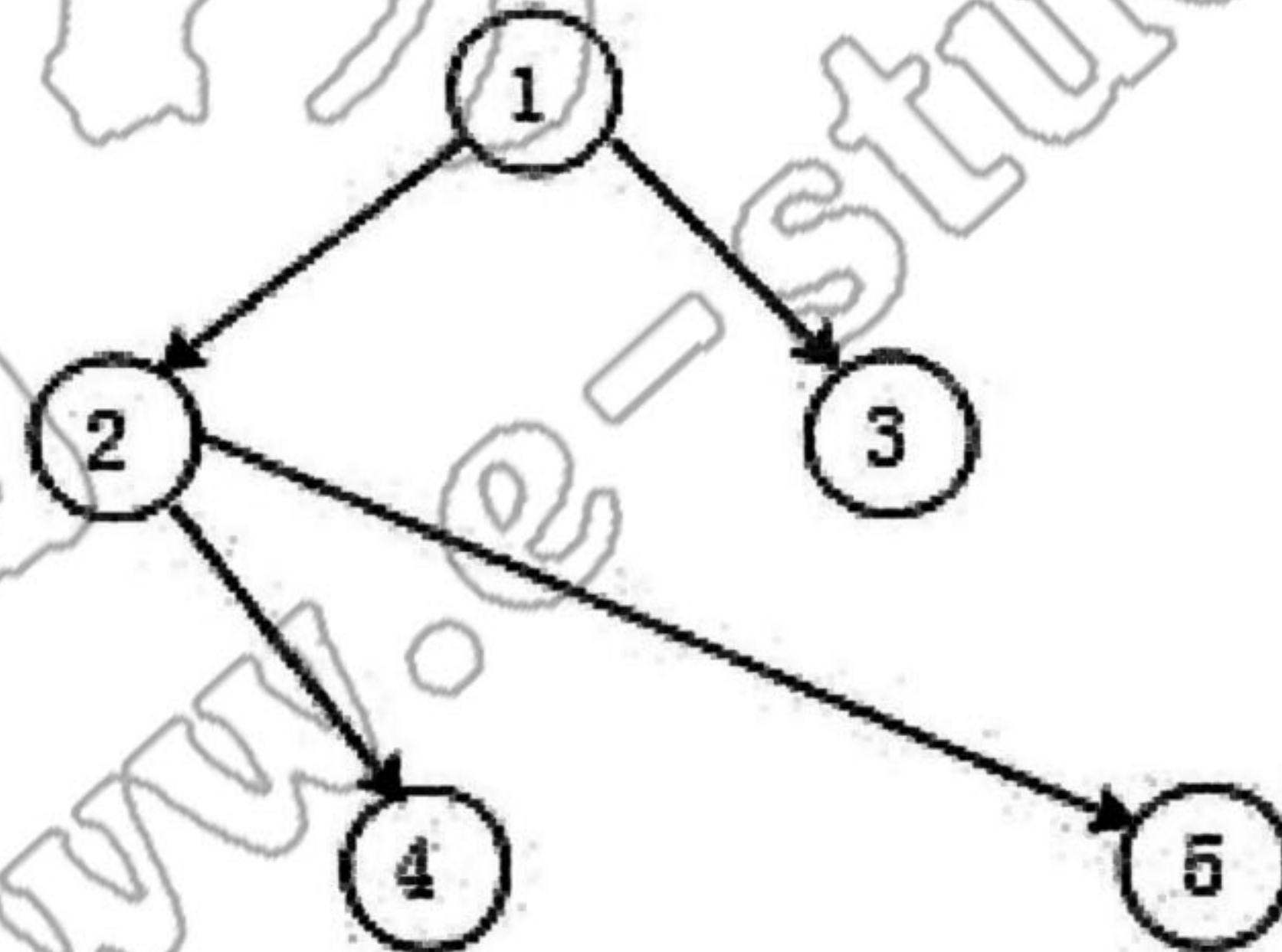
(1) 邻接表：



(2) 深度优先生成树：



(3) 广度优先生成树：



(4) 对于有向图 G 中任意顶点 u, v

①假设结点 u 是结点 v 的祖先，则在调用 DFS 访问 u 的过程中，必然会在这个过程结束之前递归的对 v 调用 DFS 访问，即 v 的 DFS 函数结束时间先于 u 的 DFS 结束时间。也就是说，祖先的结束时间必然大于子孙的结束时间。

②如果 u 是 v 的子孙，由①知 v 的结束时间大于 u 的结束时间；

③如果 u, v 没有关系，则 u 和 v 在拓扑序列的关系任意。

```
bool visited[MAX_VEXNUM];
void DFSTraverse(Graph G)
{
    for(v=0;v<G.vexnum;v++)
```


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；咨询QQ：3505993547

```
visited[v]=false;//初始化访问标记数组
time=0;
for(v=0;v<G.vexnum;v++)
    if(!visited[v])
        DFS(G,v);
}
void DFS(Graph G,int v)
{
    visited[v]=true;
    visit(v);
    for(w=FirstNeighbor(G,v);w>=0;w=NextNeighbor(G,v,w))
        if(!visited[w])
            DFS(G,w);
    time++;
    FinishTime[v]=time;
}
```

8、参考答案：参照 00 真题设计题第 2 题。

二、程序设计

(二) 程序阅读题

1、

运行结果：

12,20
14,30
16,40
18,50
20,60
22,70
24,80
26,90

2、

运行结果：

constructor B2.2
constructor B1.1
constructor B3.3
constructor A.4
1
2
4,3

结果分析：

构造函数：首先按照派生类继承顺序 B1、B2 执行相应构造函数，其参数根据 A(int i,int j,int k,int l):B1(i),B2(j),bb(k)对应前两个参数；继续执行派生类 A 的子对象 B3 bb 的构造函数，对应参数为第三个参数；最终执行本类的构造函数。

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；咨询QQ：3505993547

输出函数：注意输出顺序与 `endl` 换行符即可。

3、

运行结果：

B::~~B(a)called.

A::~~A(C)called.

4、

运行结果：

Constructing Base 5

Constructing Derived 7

42

Destructing Derived 6

Destructing Base 6

结果分析：

首先，注意在派生类 `Derived` 中的访问声明：`Base::set_i`，这样派生类的对象与成员函数就能访问了。构造函数执行顺序先基类后派生类，输出的时候数一语句 `(j--)*(++i)`，`j` 是先运算后自减，`i` 是先自增再运算。最终执行析构函数时注意与构造函数反序操作，并且 `ij` 的值均已经发生变化。