

其中,  $T$  是以顶点  $v_j$  为头的所有弧的集合。

$$v_i(n-1) = v_e(n-1)$$

$$v_i(i) = \text{Min} \{ v_i(j) - \text{dut}(\langle i, j \rangle) \mid \langle i, j \rangle \in S, i = n-2, \dots, 0$$

其中,  $S$  是以顶点  $v_i$  为尾的所有弧的集合。

这两个递推公式的计算必须分别在拓扑有序和逆拓扑有序的前提下进行。也就是说,  $v_e(j-1)$  必须在  $v_j$  的所有前驱的最早发生时间求得之后才能确定, 而  $v_i(j-1)$  则必须在  $v_j$  的所有后继的最迟发生时间求得之后才能确定。因此, 应该在拓扑排序的基础上计算  $v_e(j)$  和  $v_i(j)$ 。

由此得到如下所述的求关键路径的算法:

- ① 输入  $e$  条弧  $\langle j, k \rangle$ , 建立 AOV 网的存储结构。
- ② 从源点  $v_1$  出发, 令  $v_e(0) = 0$ , 按拓扑有序求其余各顶点的最早发生时间  $v_e(i)$  ( $1 \leq i \leq n-1$ )。如果得到的拓扑有序序列中顶点个数小于网中顶点个数  $n$ , 则说明网中有环, 不能求出关键路径, 算法结束。
- ③ 从汇点  $v_n$  出发, 令  $v_i(n-1) = v_e(n-1)$ , 按逆拓扑有序序列求其余各顶点的最迟发生时间  $v_i(i)$  ( $0 \geq i \geq n-2$ )。
- ④ 根据各顶点的  $v_e$  和  $v_i$  值, 求每条弧  $S$  的最早开始时间  $e(S)$  和最迟开始时间  $l(S)$ 。若某条弧满足, 则它是关键活动。

如上所述, 计算各顶点的  $v_e$  值是在拓扑排序的过程中进行的, 而各顶点的  $v_i$  值需要根据逆拓扑有序序列求得。因此, 必须对上一节的拓扑排序算法进行如下修改:

- ① 在拓扑排序之前设初值, 令  $v_e(i) = 0$  ( $0 \leq i \leq n-1$ )。
- ② 在算法中增加一个计算  $v_j$  的直接后继  $v_k$  的最早发生时间的操作: 若  $v_e(j) + \text{dut}(\langle j, k \rangle) > v_e(k)$ , 则  $v_e(k) = v_e(j) + \text{dut}(\langle j, k \rangle)$ 。
- ③ 为了能按逆拓扑有序序列的顺序计算各顶点的  $v_i$  值, 需保存在拓扑排序过程中求得的拓扑有序序列, 因此用顺序队列替代栈来保存入度为 0 的顶点, 原算法中有关栈的操作将改为相应的队列的操作。

## 2.3 试题分析

### 1. 给定 C 语言的数据结构

```
struct T {
    int w;
    union T { char c; int i; double d; } U;
};
```

假设 char 类型变量的存储区大小是 1 字节, int 类型变量的存储区大小是 4 字节, double 类型变量的存储区大小是 8 字节, 则在不考虑字对齐方式的情况下, 为存储一个 struct T 类型变量所需要的存储区域至少应为 \_\_\_\_\_ 字节。(2007 年下半年)

- A. 4                      B. 8                      C. 12                      D. 17

【分析】: 本题考查数据结构的相关知识。

struct 和 union 都是由多个不同的数据类型成员组成,但在任何同一时刻,union 中只存放了一个被选中的成员,而 struct 的所有成员都存在。在 struct 中,各成员都占有自己的内存空间,它们是同时存在的。一个 struct 变量的总长度等于所有成员长度之和。在 union 中,所有成员不能同时占用它的内存空间,它们不能同时存在。union 变量的长度等于最长的成员的长度。根据此本题即可得解,故选项 C 是正确的。

【参考答案】:C

2. 设栈 s 和队列 q 的初始状态为空,元素 a、b、c、d、e 依次进入栈 s,当一个元素从栈中出来后立即进入队列 q。若从队列的输出端依次得到元素 c、d、b、a、e,则元素的出栈顺序是 (1) ,栈 s 的容量至少为 (2) 。(2007 年下半年)

- (1) A. a、b、c、d、e    B. e、d、c、b、a    C. c、d、b、a、e    D. e、a、b、d、c  
 (2) A. 2    B. 3    C. 4    D. 5

【分析】:本题考查队列和栈的相关知识。

队列是一种特殊的线性表,它只允许在表的前端进行删除操作,而在表的后端进行插入操作。进行插入操作的一端称为队尾,进行删除操作的一端称为队头。队列中没有元素时,称为空队列。队列具有先进先出(FIFO)的特点。

栈是一种数据结构,它按照先进后出的原则存储数据,先进入的数据被压入栈底,最后的数据在栈顶,需要读数据的时候从栈顶开始弹出数据,最后一个数据被第一个读出来。栈也称为后进先出(LIFO)表。

根据题目“元素 a、b、c、d、e 依次进入栈 s,当一个元素从栈中出来后立即进入队列 q。若从队列的输出端依次得到元素 c、d、b、a、e”,则元素的出栈顺序也应该是 c、d、b、a、e,故(1)中选项 C 是正确的。栈的特点是先进后出,分析出队的序列,第一个出栈的是 c,因为元素 a、b、c、d、e 依次进入栈 s,那么 a、b 肯定已在栈中,然后出栈的是 d,这样,a、b、d 必须同时在栈中,才能保证 d 出栈,然后 b 出栈,这时栈中只有 a,最后出栈的是 e,因此栈 s 容量至少为 3。故(2)中选项 B 是正确的。

【参考答案】:(1) C (2) B

3. 若将某有序树 T 转换为二叉树  $T_1$ ,则 T 中结点的后(根)序序列就是  $T_1$  中结点的 \_\_\_\_\_ 遍历序列。例如图 2.14(a)所示的有序树转换为二叉树后如图 2.14(b)所示。(2008 年上半年)

- A. 先序    B. 中序    C. 后序    D. 层序

【分析】:本题考查树和二叉树的相关知识。

树的后序遍历定义:若树 T 非空,则

- ① 依次后序遍历根 T 的各子树  $T_1, T_2, \dots, T_k$ ;
- ② 访问根结点。

所以有序树 T 的后序遍历序列是:2563741。

先序遍历二叉树的操作定义为:若二叉树为空,则空操作;否则先访问根结点,再先序遍历左子树,最后先序遍历右子树。

中序遍历二叉树的操作定义为:若二叉树为空,则空操作;否则先中序遍历左子树,再访问根结点,最后中序遍历右子树。

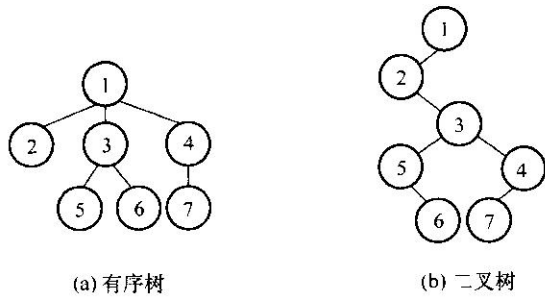


图 2.14 有序树转换二叉树

后序遍历二叉树的操作定义为:若二叉树为空,则空操作;否则先后序遍历左子树,再后序遍历右子树,最后访问根结点。

所以二叉树的中序遍历序列为 2563741,先序遍历序列为 1235647,后序遍历序列为 6574321。故选项 B 是正确的。

【参考答案】:B

4. 某项目主要由 A ~ I 任务构成,其计划图(如图 2.15 所示)展示了各任务之间的前后关系以及每个任务的工期(单位:天),该项目的关键路径是 (1)。在不延误项目总工期的情况下,A 最多可以推迟开始的时间是 (2) 天。(2009 年上半年)

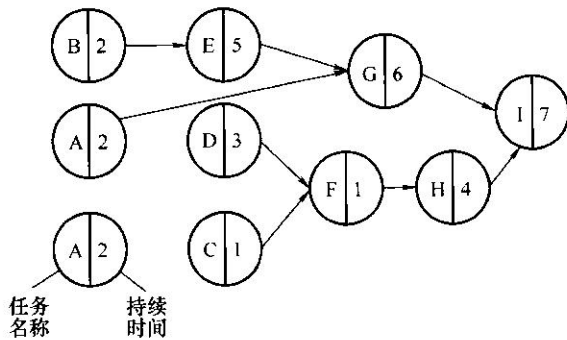


图 2.15 项目计划图

- (1) A. A→G→I                      B. A→D→F→H→I  
 C. B→E→G→I                      D. C→F→H→I  
 (2) A. 0                      B. 2                      C. 5                      D. 7

【分析】:本题考查图的关键路径计算。

在项目管理中,关键路径是指关键活动的序列,该序列具有最长的总工期并决定了整个项目的最短完成时间。任何关键路径上的关键活动的延迟将直接影响项目的预期完成时间。根据这一原则,把 ABCD 四个选项代入原图分析,就可得到, B→E→G→I 这一序列具有最长的总工期 20。故(1)中选项 C 是正确的。

根据上图和关键路径的算法:只有任务 B 和任务 E 都完成了,任务 G 才能开始,而要开始执

行到任务 G,则必须经过执行任务 B 和任务 E 总共的 7 天,任务 A 的执行时间只有 2 天,故而 A 最多可以推迟  $7-2=5$  天开始。故(2)中选项 C 是正确的。

【参考答案】:(1) C (2) C

5. 关于查找运算及查找表的说法,错误的是\_\_\_\_\_。(2009 年上半年)

- A. 散列表可以动态创建
- B. 二叉排序树属于动态查找表
- C. 二分查找要求查找表采用顺序存储结构或循环链表结构
- D. 顺序查找方法既适用于顺序存储结构,也适用于链表结构

【分析】:本题考查查找运算及查找表的相关知识。

散列就是把任意长度的输入通过散列算法,变换成固定长度的输出,该输出就是散列值,按这个思想建立的表为散列表,散列表是可以动态创建的。

二叉排序树或者是一棵空树,或者是具有下列性质的二叉树:若它的左子树不空,则左子树上所有结点的值均小于它的根结点的值;若它的右子树不空,则右子树上所有结点的值均大于它的根结点的值;它的左、右子树也分别为二叉排序树。二叉排序树属于动态查找表。

二分查找又称折半查找,要求关键字必须采用顺序存储结构并且必须按关键字大小有序排列,所以二分查找不能采用循环链表结构。

顺序查找算法的基本过程为:从表中的最后一条记录开始,逐条将记录的关键字与给定值进行比较,若某条记录的关键字与给定值相等,则查找成功,取出相应的记录值,直到第一条记录都被比较结束;若关键字和给定值比较都不相等,则表明表中没有所查的记录,查找失败。所以顺序查找方式既适用于顺序存储结构,也适用于链表结构。综上所述,选项 C 是正确的。

【参考答案】:C

6. 关于二叉排序树的说法,错误的是\_\_\_\_\_。(2009 年上半年)

- A. 对二叉排序树进行中序遍历,必定得到结点关键字的有序列
- B. 依据关键字无序的序列建立二叉排序树,也可能构造出单支树
- C. 若构造二叉排序树时进行平衡化处理,则根结点的左子树结点数与右子树结点数的差值一定不超过 1
- D. 若构造二叉排序树时进行平衡化处理,则根结点的左子树高度与右子树高度的差值一定不超过 1

【分析】:本题考查二叉排序树的相关知识。

根据二叉排序树的定义,通常采取二叉链表作为二叉排序树的存储结构,中序遍历二叉排序树可得到一个关键字的有序序列,一个无序序列可以通过构造一棵二叉排序树变成一个有序序列,构造树的过程即为对无序序列进行排序的过程,二叉排序树在最差情况下成为单支树,检索效率等同顺序表,所以 A、B 的说法都是正确的。

最佳二叉排序树难于构造,因此提出一种称为平衡二叉树的二叉排序树,它的检索效率介于最佳二叉排序树和一般二叉排序树之间,要求树中任一结点的左右子树的高度差都不大于 1,所以 D 的说法也是正确的。故此题选择 C。

【参考答案】:C

## 2.4 模拟训练

1. 数据结构是指\_\_\_\_\_。
  - A. 数据元素的组织形式
  - B. 数据类型
  - C. 数据存储结构
  - D. 数据定义
2. 数据在计算机存储器内表示时,物理地址与逻辑地址不相同的,称为\_\_\_\_\_。
  - A. 存储结构
  - B. 逻辑结构
  - C. 链式存储结构
  - D. 顺序存储结构
3. 树状结构是数据元素之间存在一种\_\_\_\_\_。
  - A. 一对一关系
  - B. 多对多关系
  - C. 多对一关系
  - D. 一对多关系
4. 线性表是\_\_\_\_\_。
  - A. 一个有限序列,可以为空
  - B. 一个有限序列,不可以为空
  - C. 一个无限序列,可以为空
  - D. 一个无限序列,不可以为空
5. 在一个长度为  $n$  的顺序表中删除第  $i$  个元素 ( $0 \leq i \leq n$ ) 时,需向前移动\_\_\_\_\_个元素。
  - A.  $n-i$
  - B.  $n-i+1$
  - C.  $n-i-1$
  - D.  $i$
6. 线性表采用链式存储时,其地址\_\_\_\_\_。
  - A. 必须是连续的
  - B. 一定是不连续的
  - C. 部分地址必须是连续的
  - D. 连续与否均可以
7. 从一个具有  $n$  个结点的单链表中查找其值等于  $x$  的结点时,在查找成功的情况下,需平均比较\_\_\_\_\_个元素结点。
  - A.  $n/2$
  - B.  $n$
  - C.  $(n+1)/2$
  - D.  $(n-1)/2$
8. 空串与空格字符组成的串的区别在于\_\_\_\_\_。
  - A. 没有区别
  - B. 两串的长度不相等
  - C. 两串的长度相等
  - D. 两串包含的字符不相同
9. 一个子串在包含它的主串中的位置是指\_\_\_\_\_。
  - A. 子串的最后那个字符在主串中的位置
  - B. 子串的最后那个字符在主串中首次出现的位置
  - C. 子串的第一个字符在主串中的位置
  - D. 子串的第一个字符在主串中首次出现的位置
10. 下面的说法中,只有\_\_\_\_\_是正确的。
  - A. 字符串的长度是指串中包含的字母的个数
  - B. 字符串的长度是指串中包含的不同字符的个数
  - C. 若  $T$  包含在  $S$  中,则  $T$  一定是  $S$  的一个子串
  - D. 一个字符串不能说是其自身的一个子串
11. 设二维数组  $A[0..m-1][0..n-1]$  按行优先顺序存储在内存中,第一个元素的地址为

$p$ , 每个元素占  $k$  个字节, 则元素  $a_{ij}$  的地址为\_\_\_\_\_。

- A.  $p + [i * n + j - 1] * k$                       B.  $p + [(i-1) * n + j - 1] * k$   
 C.  $p + [(j-1) * n + i - 1] * k$                       D.  $p + [j * n + i - 1] * k$

12. 已知二维数组  $A[10, 10]$  中, 元素  $a_{20}$  的地址为 560, 每个元素占 4 个字节, 则元素  $a_{10}$  的地址为\_\_\_\_\_。

- A. 520              B. 522              C. 524              D. 518

13. 若数组  $A[0..m][0..n]$  按列优先顺序存储, 则  $a_{ij}$  地址为\_\_\_\_\_。

- A.  $LOC(a_{00}) + [j * m + i]$                       B.  $LOC(a_{00}) + [j * n + i]$   
 C.  $LOC(a_{00}) + [(j-1) * n + i - 1]$                       D.  $LOC(a_{00}) + [(j-1) * m + i - 1]$

14. 若下三角矩阵  $A[n \times n]$ , 按列顺序压缩存储在数组  $Sa[0..(n+1)n/2]$  中, 则非零元素  $a_{ij}$  的地址为\_\_\_\_\_。(设每个元素占  $d$  个字节)

- A.  $\left[ (j-1) * n - \frac{(j-2)(j-1)}{2} + i - 1 \right] * d$   
 B.  $\left[ (j-1) * n - \frac{(j-2)(j-1)}{2} + i \right] * d$   
 C.  $\left[ (j-1) * n - \frac{(j-2)(j-1)}{2} + i + 1 \right] * d$   
 D.  $\left[ (j-1) * n - \frac{(j-2)(j-1)}{2} + i - 2 \right] * d$

15. 在一棵度为 3 的树中, 度为 3 的结点数为 2, 度为 2 的结点数为 1, 度为 1 的结点数为 2, 则度为 0 的结点数为\_\_\_\_\_。

- A. 4              B. 5              C. 6              D. 7

16. 假设在一棵二叉树中, 双分支结点数为 15, 单分支结点数为 30, 则叶子结点数为\_\_\_\_\_。

- A. 15              B. 16              C. 17              D. 47

17. 假定一棵三叉树的结点数为 50, 则它的最小高度为\_\_\_\_\_。

- A. 3              B. 4              C. 5              D. 6

18. 在一棵二叉树上第 4 层的结点数最多为\_\_\_\_\_。

- A. 2              B. 4              C. 6              D. 8

19. 在一个具有  $n$  个顶点的有向图中, 若所有顶点的出度数之和为  $s$ , 则所有顶点的入度数之和为\_\_\_\_\_。

- A.  $s$               B.  $s-1$               C.  $s+1$               D.  $n$

20. 在一个具有  $n$  个顶点的有向图中, 若所有顶点的出度数之和为  $s$ , 则所有顶点的度数之和为\_\_\_\_\_。

- A.  $s$               B.  $s-1$               C.  $s+1$               D.  $2s$

21. 在一个具有  $n$  个顶点的无向图中, 若具有  $e$  条边, 则所有顶点的度数之和为\_\_\_\_\_。

- A.  $n$               B.  $e$               C.  $n+e$               D.  $2e$

22. 在一个具有  $n$  个顶点的无向完全图中, 所含的边数为\_\_\_\_\_。

- A.  $n$               B.  $n(n-1)$               C.  $n(n-1)/2$               D.  $n(n+1)/2$

23. 在一个具有  $n$  个顶点的有向完全图中,所含的边数为\_\_\_\_\_。

- A.  $n$       B.  $n(n-1)$       C.  $n(n-1)/2$       D.  $n(n+1)/2$

参考答案:

1. A    2. C    3. D    4. A    5. A    6. D    7. C    8. B    9. D    10. C  
11. A    12. A    13. A    14. B    15. C    16. B    17. C    18. D    19. A  
20. D    21. D    22. C    23. B