

例题

一个高度为 h 的二叉树最小元素数目是 (B)。

- A) $2h+1$ B) h C) $2h-1$ D) $2h$ E) $2h-1$

一个向量第一个元素的存储地址是 100, 每个元素的长度是 2, 则第 5 个元素的地址是 (B)。

- A) 110 B) 108 C) 100 D) 109

设有一个含有 13 个元素的 Hash 表 ($0 \sim 12$), Hash 函数是: $H(\text{key}) = \text{key} \% 13$, 其中 $\%$ 是求余数运算。用线性探查法解决冲突, 则对于序列 (2、8、31、20、19、18、53、27), 18 应放在第几号格中 (B)。

- A) 5 B) 9 C) 4 D) 0

按照二叉树的定义, 具有 3 个结点的二叉树有 (C) 种。

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

在一个有向图中, 所有顶点的入度之和等于所有顶点的出度之和的 (B) 倍。

- A) $1/2$ B) 1 C) 2 D) 4

要使 1...8 号格子的访问顺序为: 8、2、6、5、7、3、1、4, 则下图中的空格中应填入 (C)。

1	2	3	4	5	6	7	8
4	6	1	-1	7		3	2

- A) 6 B) 0 C) 5 D) 3

设栈 S 和队列 Q 的初始状态为空, 元素 $e_1, e_2, e_3, e_4, e_5, e_6$ 依次通过栈 S, 一个元素出栈后即进入队列 Q, 若出队的顺序为 $e_2, e_4, e_3, e_6, e_5, e_1$, 则栈 S 的容量至少应该为 (B)。

- A) 2 B) 3 C) 4 D) 5

设有一棵 k 叉树, 其中只有度为 0 和 k 两种结点, 设 n_0, n_k 分别表示度为 0 和度为 k 的结点个数, 试求出 n_0, n_k 之间的关系 (n_0 =数学表达式, 数学表达式仅含 n_k, k 和数字)

$$N_0 = (K-1) N_k + 1$$

若已知一个栈的入栈顺序是 1, 2, 3, ..., n , 其输出序列为 $P_1, P_2, P_3, \dots, P_n$, 若 P_1 是 n , 则 P_i 是 (C)

- A) i B) $n-1$ C) $n-i+1$ D) 不确定

以下哪一个不是栈的基本运算 (B)

- A) 删除栈顶元素 B) 删除栈底的元素
C) 判断栈是否为空 D) 将栈置为空栈

下面关于算法的错误说法是(B)

- A) 算法必须有输出 B) 算法必须在计算机上用某种语言实现
C) 算法不一定有输入 D) 算法必须在有限步执行后能结束

在顺序表(2, 5, 7, 10, 14, 15, 18, 23, 35, 41, 52)中, 用二分法查找12, 所需的关键码比较的次数为(C)

- A) 2 B) 3 C) 4 D) 5

一棵二叉树的高度为 h , 所有结点的度为 0, 或为 2, 则此树最少有(B)个结点

- A) $2h-1$ B) $2h-1$ C) $2h+1$ D) $h+1$

无 向 图 $G=(V, E)$, 其 中 $V=\{a, b, c, d, e, f\}$
 $E=\{(a, b), (a, e), (a, c), (b, e), (c, f), (f, d), (e, d)\}$

对该图进行深度优先遍历, 得到的顶点序列正确的是(D)

- A) a, b, e, c, d, f B) a, c, f, e, b, d C) a, e, b, c, f, d D) a, b, e, d, f, c

已知一棵二叉树的结点名为大写英文字母, 其中序与后序遍历的顺序分别为: CBGEAFHDIJ 与 CGEBHFJIDA 则该二叉树的先序遍历的顺序为:

ABCEGDFHIJ

在有 N 个叶子节点的哈夫曼树中, 其节点总数为(B)

- A. 不确定 B. $2N-1$ C. $2N+1$ D. $2N$

某数列有 1000 个各不相同的单元, 由低至高按序排列; 现要对该数列进行二分法检索(binary-search), 在最坏的情况下, 需检视(B)个单元。

- A. 1000 B. 10 C. 100 D. 500

线性表若采用链表存贮结构, 要求内存中可用存贮单元地址(D)

- A. 必须连续 B. 部分地址必须连续 C. 一定不连续 D. 连续不连续均可

下列叙述中, 正确的是(D)

- A. 线性表的线性存贮结构优于链表存贮结构 B. 队列的操作方式是先进后出
C. 栈的操作方式是先进先出 D. 二维数组是指它的每个数据元素为一个线性表的线性表

已知, 按中序遍历二叉树的结果为: abc

问: 有多少种不同形态的二叉树可以得到这一遍历结果, 并画出这些二叉树。

5 种

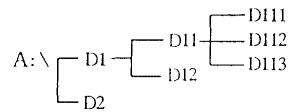
设有一个共有 n 级的楼梯, 某人每步可走 1 级, 也可走 2 级, 也可走 3 级, 用递推公式给出某人从底层开始走完全部楼梯的走法。例如: 当 $n=3$ 时, 共有 4 种走法, 即 1+1+1, 1+2, 2+1, 3。

$F(n)=f(n-1)+f(n-2)+f(n-3), n \geq 4;$

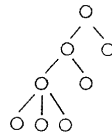
$F(1)=1; f(2)=2; f(3)=4;$

在磁盘的目录结构中，我们将与某个子目录有关联的目录数称为度。

例如下图：



该图表达了 A 盘的目录结构：D1，D11，……D2 均表示子目录的名字. 在这里，根目录的度为 2，D1 子目录的度为 3，D11 子目录的度为 4，D12，D2，D111，D112，D113 的度均为 1。又不考虑子目录的名字，则可简单的图示为如下的树结构：



若知道一个磁盘的目录结构中，度为 2 的子目录有 2 个，度为 3 的子目录有 1 个，度为 4 的子目录有 3 个。

试问：度为 1 的子目录有几个？

$$2*2+3*1+4*3+1*x=(2+1+3+x-1)*2$$

根据 Nocomachns 定理，任何一个正整数 n 的立方一定可以表示成 n 个连续的奇数的和。

例如：

$$1^3 = 1$$

$$2^3 = 3 + 5$$

$$3^3 = 7 + 9 + 11$$

$$4^3 = 13 + 15 + 17 + 19$$

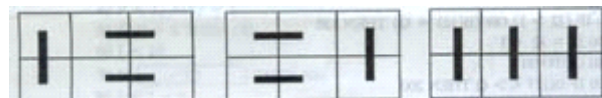
在这里，若将每一个式中的最小奇数称为 X，那么当给出 n 之后，请写出 X 与 n 之间的关系表达式： $n^2 - n + 1$

设循环队列中数组的下标范围是 $1 \sim n$ ，其头尾指针分别为 f 和 r，则其元素个数为 (D)

A. $r - f$ B. $r - f + 1$ C. $(r - f) \text{ MOD } n + 1$ D. $(r - f + n) \text{ MOD } n$

有 $2 \times n$ 的一个长方形方格，用一个 1×2 的骨牌铺满方格。例如 $n=3$ 时，为 2×3 方格。

此时用一个 1×2 的骨牌铺满方格，共有 3 种铺法：



试对给出的任意一个 $n(n \geq 0)$ ，求出铺法总数的递推公式。

$$F(1)=1 \quad F(2)=2 \quad F(n)=F(n-1)+F(n-2), \quad n \geq 3$$

FUNCTION ACK(M, N: INTEGER): INTEGER;

BEGIN

IF M=0 THEN ACK:=N+1

ELSE IF N=0 THEN ACK:=ACK(M-1, 1)

ELSE ACK:=ACK(M-1, ACK(M, N-1))

END;

BEGIN WRITELN(ACK(3, 4)); READLN; END.

输出

125

表达式 $(1+34)*5-56/7$ 的后缀表达式为 (C)。

- A) $1+34*5-56/7$ B) $-*+1\ 34\ 5/56\ 7$ C) $1\ 34\ +5*56\ 7/-$
D) $1\ 34\ 5* +56\ 7/-$ E) $1\ 34+5\ 56\ 7-* /$

已知元素 (8, 25, 14, 87, 51, 90, 6, 19, 20)，问这些元素以怎样的顺序进入栈，才能使出栈的顺序满足：8 在 51 前面；90 在 87 的后面；20 在 14 的后面；25 在 6 的前面；19 在 90 的后面。(D)。(题意是全部进栈，再依次出栈)

- A) 20, 6, 8, 51, 90, 25, 14, 19, 87
B) 51, 6, 19, 20, 14, 8, 87, 90, 25
C) 19, 20, 90, 7, 6, 25, 51, 14, 87
D) 6, 25, 51, 8, 20, 19, 90, 87, 14
E) 25, 6, 8, 51, 87, 90, 19, 14, 20

假设我们用 $d=(a_1, a_2, \dots, a_5)$, 表示无向图 G 的 5 个顶点的度数, 下面给出的哪(些)组 d 值合理 (BE)。

- A) {5, 4, 4, 3, 1} B) {4, 2, 2, 1, 1} C) {3, 3, 3, 2, 2}
D) {5, 4, 3, 2, 1} E) {2, 2, 2, 2, 2}

下列关于程序语言的叙述，不正确的是 (D)。

- A) 编写机器代码不比编写汇编代码容易。
B) 高级语言需要编译成目标代码或通过解释器解释后才能被 CPU 执行。
C) 同样一段高级语言程序通过不同的编译器可能产生不同的可执行程序。
D) 汇编代码可被 CPU 直接运行。
E) 不同的高级语言语法略有不同。

下列哪个程序设计语言不支持面向对象程序设计方法 (C)。

- A. C++ B. Object Pascal C. C D. Smalltalk E. Java

某个车站呈狭长形，宽度只能容下一台车，并且只有一个出入口。已知某时刻该车站状态为空，从这一时刻开始的出入记录为：“进，出，进，进，出，进，进，进，出，出，进，出”。假设车辆入站的顺序为 1, 2, 3, ……，则车辆出站的顺序为 ()。

- A. 1, 2, 3, 4, 5 B. 1, 2, 4, 5, 7 C. 1, 3, 5, 4, 6 D. 1, 3, 5, 6, 7 E. 1, 3, 6, 5, 7

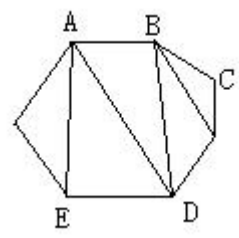
二叉树 T，已知其前序遍历序列为 1 2 4 3 5 7 6，中序遍历序列为 4 2 1 5 7 3 6，则其后序遍历序列为 (B)。

- A. 4 2 5 7 6 3 1 B. 4 2 7 5 6 3 1 C. 4 2 7 5 3 6 1 D. 4 7 2 3 5 6 1 E. 4 5 2 6 3 7 1

满二叉树的叶结点个数为 N，则它的结点总数为 (C)。

- A. N B. $2 * N$ C. $2 * N - 1$ D. $2 * N + 1$ E. $2^N - 1$

在下图中，从顶点（ E ）出发存在一条路径可以遍历图中的每条边一次，而且仅遍历一次。



- A. A 点 B. B 点 C. C 点 D. D 点 E. E 点

某大学计算机专业的必修课及其先修课程如下表所示：

课程代号	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
课程名称	高等数学	程序设计语言	离散数学	数据结构	编译技术	操作系统	普通物理	计算机原理
先修课程			C_0, C_1	C_1, C_2	C_3	C_3, C_7	C_0	C_6

- 请你判断下列课程安排方案哪个是不合理的（ D ）。
- A. $C_0, C_6, C_7, C_1, C_2, C_3, C_4, C_5$ B. $C_0, C_1, C_2, C_3, C_4, C_6, C_7, C_5$
 C. $C_0, C_1, C_6, C_7, C_2, C_3, C_4, C_5$ D. $C_0, C_1, C_6, C_7, C_5, C_2, C_3, C_4$
 E. $C_0, C_1, C_2, C_3, C_6, C_7, C_5, C_4$

- 完全二叉树的结点个数为 $4 * N + 3$ ，则它的叶结点个数为（ E ）。
- A. $2 * N$ B. $2 * N - 1$ C. $2 * N + 1$ D. $2 * N - 2$ E. $2 * N + 2$

- 平面上有五个点 $A(5, 3), B(3, 5), C(2, 1), D(3, 3), E(5, 1)$ 。以这五点作为完全图 G 的顶点，每两点之间的直线距离是图 G 中对应边的权值。以下哪条边不是图 G 的最小生成树中的边（ D ）。
- A. AD B. BD C. CD D. DE E. EA

- 二叉树 T 的宽度优先遍历序列为 $A B C D E F G H I$ ，已知 A 是 C 的父结点， D 是 G 的父结点， F 是 I 的父结点，树中所有结点的最大深度为 3（根结点深度设为 0），可知 F 的父结点是（ C ）。
- A. 无法确定 B. B C. C D. D E. E

- 设栈 S 的初始状态为空，元素 a, b, c, d, e, f, g 依次入栈，以下出栈序列不可能出现的是（ E ）。
- A. a, b, c, e, d, f, g B. b, c, a, f, e, g, d C. a, e, d, c, b, f, g
 D. d, c, f, e, b, a, g E. g, e, f, d, c, b, a

将数组 $\{32, 74, 25, 53, 28, 43, 86, 47\}$ 中的元素按从小到大的顺序排列，每次可以交换任意两个元素，最少需要交换 5 次。

取火柴游戏的规则如下：一堆火柴有 N 根， $A、B$ 两人轮流取出。每人每次可以取 1 根或

2 根，最先没有火柴可取的人为败方，另一方为胜方。如果先取者有必胜策略则记为1，先取者没有必胜策略记为0。当N 分别为100， 200， 300， 400， 500 时，先取者有无必胜策略的标记顺序为 __11011__（回答应为一个由0 和/或1 组成的字符串）

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(BD) 。

A) 希尔排序 B) 起泡排序 C) 插入排序 D) 选择排序