数据结构&算法

2018 年上半年

● 队列的特点是先进先出,若用循环单链表表示队列,则(58)。

A. 入队列和出队列操作都不需要遍历链表

B. 入队列和出队列操作都需要遍历链表

C. 入队列操作需要遍历链表而出队列操作不需要

D. 入队列操作不需要遍历链表而出队列操作需要

● 设有 n 阶三对角矩阵 A,即非零元素都位于主对角线以及与主对角线平行且紧邻的两条对角线上,现对该矩阵进行按行压缩存储,若其压储空间用数组 B 表示,A 的元素下标从 0 开始,B 的元素下标从 1 开始。已知 A[0,0]存储在 B[1],A[n-1,n-1]存储在 B[3n-2],那么非零元素 A[i,j](0≤i<n,0≤j<n,|i-j|≤1)存储在 B[(59)]。

A. 2i+j-1

B. 2i+j

C. 2i+j+1

D. 3i-j+1

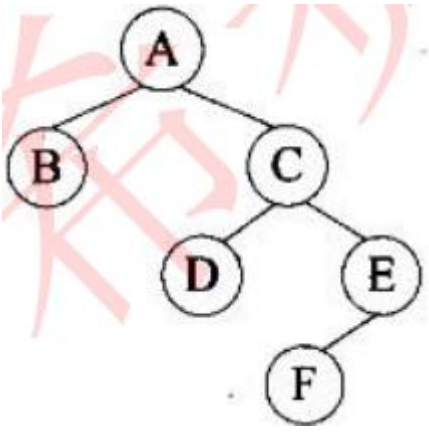
● 对下面的二叉树进行顺序存储(用数组 MEM 表示),已知结点 A、B、C 在 MEM 中对应元素的下标分别为 1、2、3,那么结点 D、E、F 对应的数组元素下标为(60)。

A. 4、5、6

B. 4、7、10

C. 6、7、8

D. 6、7、14



● 用哈希表存储元素时,需要进行冲突(碰撞)处理,冲突是指(61)。

A. 关键字被依次映射到地址编号连续的存储位置

B. 关键字不同的元素被映射到相同的存储位置

C. 关键字相同的元素被映射到不同的存储位置

D. 关键字被映射到哈希表之外的位置

● 现需要申请一些场地举办一批活动,每个活动有开始时间和结束时间。在同一个场地,如果一个活动结束之前,另一个活动开始,即两个活动冲突。若活动 A 从 1 时间开始,5 时间结束,活动 B 从 5 时间开始,8 时间结束,则活动 A 和 B 不冲突。现要计算 n 个活动需要的最少场地数。求解该问题的基本思路如下(假设需要场地数为 m,活动数为 n,场地集合为 P1，P2,…,Pm),初始条件 Pi 均无活动安排:

(1)采用快速排序算法对 n 个活动的开始时间从小到大排序,得到活动 a1，a2,…，an。对每个活动 ai，i 从 1 到 n,重复步骤(2)、(3)和(4)；

(2)从 p1 开始,判断 ai 与 P1 的最后一个活动是否冲突,若冲突,考虑下一个场地 P2,…；

(3)一旦发现 ai 与某个 Pj 的最后一个活动不冲突,则将 ai 安排到 Pj,考虑下一个活动；

(4)若 ai 与所有已安排活动的 Pj 的最后一个活动均冲突,则将 ai 安排到一个新的场地,考虑下一个活动；

(5)将 n 减去没有安排活动的场地数即可得到所用的最少场地数

算法首先采用了快速排序算法进行排序,其算法设计策略是(62)；后面步骤采用的算法设计策略是(63)。整个算法的时间复杂度是(64)。下表给出了 n=11 的活动集合,根据上述算法,得到最少的场地数为(65)。

i	1	2	3	4	5	6	7	8	9	10	11
开始时间 s_i	0	1	2	3	3	5	5	6	8	8	12
结束时间 f_i	6	4	13	5	8	7	9	10	11	12	14

A. 分治
B. 动态规划
C. 贪心
D. 回溯

A. 分治
B. 动态规划
C. 贪心
D. 回溯

A. $O(\lg^n)$　　B. $O(n)$

C. $O(n\lg^n)$　　D. $O(n^2)$

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

2017 年下半年

58. 假设某消息中只包含 7 个字符怡{a,b,c,d,e,f,g}，折 7 个字符在消息中出现的次数为{5,24,8,17,34,f4,13}，利用哈夫曼树（最优二叉树）为该消息中的字符构造符合前缀编码要求的不等长编码。各字符的编码长度分别为（ ）。

A．a:4,b:2,c:3,d:3,e:2,f:4,g:3
B．a:6,b:2,c:5,d:3,e:1,f:6,g:4
C．a:3,b:3,c:3,d:3,e:3,f:2,g:3
D．a:2,b:6,c:3,d:5,e:6,f:1,g:4

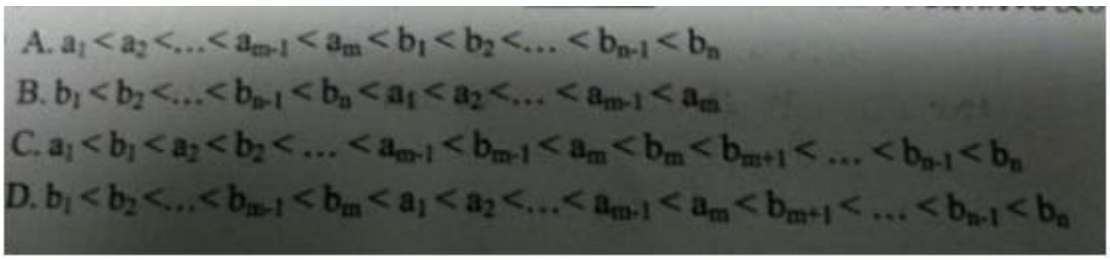
59. 设某二叉树采用二叉链表表示（即结点的两个指针分别指示左、右孩子）。当该二叉树包含 k 个节点时，其二叉链表节点中必有（ ）个空的汉子指针。

A．k-1
B．k
C．k+1
D．2k

60. 以下关于无向连通图 G 的叙述中，不正确的是（ ）。

A．G 中任意两个顶点之间均有边存在
B．G 中任意两个顶点之间存在路径
C．从 G 中任意顶点出发可遍历图中所有顶点
D．G 的临接矩阵是对称矩阵

61 两个递增序列 A 和 B 的长度分别为 m 和 n(m<n 且 m 与 n 接近),将二者归并为一个长度为 m+n 的递增序列。当元素关系为（ ），归并过程中元素的比较次数最少。



64. 现需要对一个基本有序的数组进行排序。此时最适宜采用的算法为（ ）排算法，时间复
杂度为（ ）。

A．插入　　B．快速
C．归并　　D．堆

A．O(n)　　B．O(nlgn)
C．O(n^2)　　D．O(n^2lgn)

2017 年上半年

2016 年下半年

● 以下关于字符串的叙述中，正确的是（ 57 ）

A．包含任意个空格字符的字符串称为空串

B. 字符串不是线性数据结构

C. 字符串的长度是指串中所含字符的个数

D. 字符串的长度是指串中所含非空格字符的个数

● 已知栈 S 初始为空，用 I 表示入栈、O 表示出栈，若入栈序列为 a1a2a3a4a5，则通过栈 S 得到出栈序列 a2a4a5a3a1 的合法操作序列（ 58 ）

A．IIOIIOIOOO

B.IOIOIOIOIO

C.IOOIIOIOIO

D.IIOOIOIOOO

● 某二叉树的先序遍历序列为 ABCDEF ，中序遍历序列为 BADCFE ，则该二叉树的高度(即层数)为（ 59 ）

A.3

B.4

C.5

D.6

● 对于 n 个元素的关键宇序列{k1,k2, ...kn}，当且仅当满足关系 ki≤k2i 且 ki≤k2i+1{i=1.2...[n/2]}时称其为小根堆(小顶堆)。以下序列中，()不是小根堆。

A．16,25,40,55,30,50,45

B.16,40,25,50,45,30,55

C.16,25,39.,41,45,43,50

D.16,40,25,53,39,55,45

● 在 12 个互异元素构成的有序数组 a[1..12] 中进行二分查找(即折半查找，向下取 整),若待查找的元素正好等于 a[9],则在此过程中,依次与数组中的( )比较后，查找成功结束。

A． a[6]、 a[7]、 a[8]、 a[9]

B. a[6]、a[9]

C.a[6]、 a[7]、 a[9]

D. a[6]、 a[8]、 a[9]

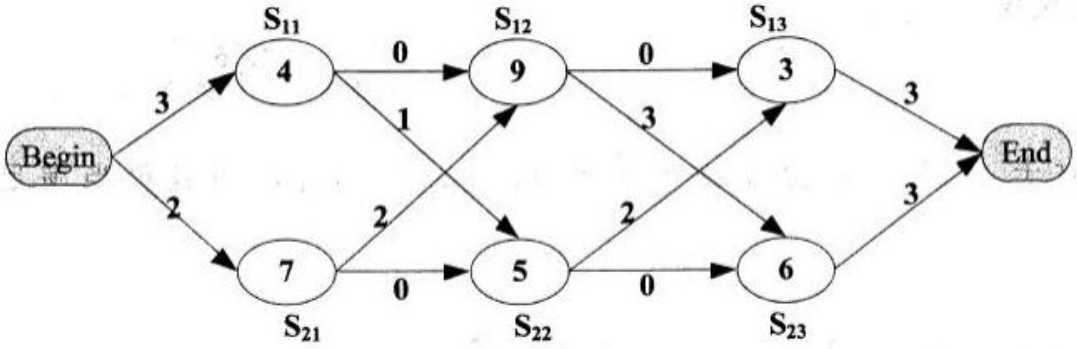
● 某汽车加工工厂有两条装配线 L1 和 L2，每条装配线的工位数均为 n（Sij，i=1 或 2，j=1，2，...，n），两条装配线对应的工位完成同样的加工工作，但是所需要的时间可能不同（aij，i=1 或 2，j=1，2，...，n）。汽车底盘开始到进入两条装配线的时间（e1，e2）以及装配后到结束的时间(X1X2)也可能不相同。从一个工位加工后流到下一个工位需要迁移时间(tij，i=1 或 2，j=2，...n)。现在要以最快的时间完成一辆汽车的装配，求最优的装配路线。分析该问题，发现问题具有最优子结构。以 L1 为例，除了第一个工位之外，经过第 j 个工位的最短时间包含了经过 L1 的第 j-1 个工位的最短时间或者经过 L2 的第 j-1 个工位的最短时间，如式(1)。装配后到结束的最短时间包含离开 L1 的最短时间或者离开 L2 的最短时间如式（2）。

$$f_{1,j}=\begin{cases} e_1+a_{1,j} & \text{若 } j=1 \\ \min\left(f_{1,j-1}+a_{1,j}+t_{1,j-1},\ f_{2,j-1}+a_{1,j}+t_{2,j-1}\right) & \text{其他} \end{cases} \quad (1)$$

$$f_{min}=\min\left(f_{1,n}+x_1,\ f_{2,n}+x_2\right) \quad (2)$$

由于在求解经过 L1 和 L2 的第 j 个工位的最短时间均包含了经过 L1 的第 j-1 个工位的最短时间或者经过 L2 的第 j-1 个工位的最短时间，该问题具有重复子问题的性质，故采用迭代方法求解。

该问题采用的算法设计策略是__(62)__，算法的时间复杂度为__(63)__ 以下是一个装配调度实例，其最短的装配时间为__(64)__，装配路线为__(65)__



A．分治
B.动态规划
C.贪心
D.回溯

A． O(lgn)
B． O(n)
C． O(n2)
D． O(nlgn)

A．21
B.23
C.20
D.26

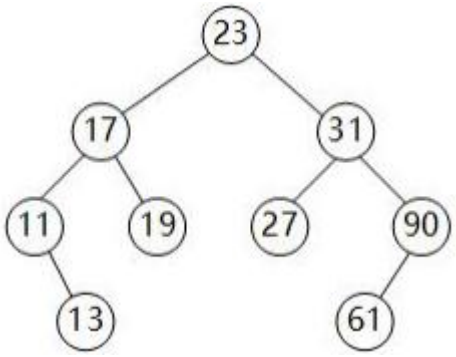
A．S11→S12→S13
B.S11→S22→S13
C.S21→S12→S23
D.S21→S22→S23

2016 年上半年

● 若元素以 a，b，c，d，e 的顺序进入一个初始为空的栈中，每个元素进栈、出栈各 1 次，要求出栈的第一个元素为 d，则合法的出栈序列共有（57）种。

A．4
B．5
C．6
D．24

● 设有二叉排序树（或二叉查找树）如下图所示，建立该二叉树的关键码序列不可能是
（58）。



A．23 31 17 19 11 27 13 90 61
B．23 17 19 31 27 90 61 11 13
C．23 17 27 19 31 13 11 90 61
D．23 31 90 61 27 17 19 11 13

● 若一棵二叉树的高度（即层数）为 h，则该二叉树（59）。

A．有 2h 个结点
B．有 2h-1 个结点
C．最少有 2h-1 个结点
D．最多有 2h-1 个结点

● 在 13 个元素构成的有序表 A[1..13]中进行折半查找（或称为二分查找，向下取整）。那么以下叙述中，错误的是（60）。

A．无论要查找哪个元素，都是先与 A[7]进行比较
B．若要查找的元素等于 A[9]，则分别需与 A[7]、A[11]、A[9]进行比较
C．无论要查找的元素是否在 A[]中，最多与表中的 4 个元素比较即可
D．若待查找的元素不在 A[]中，最少需要与表中的 3 个元素进行比较

● 以下关于图的遍历的叙述中，正确的是（61）。

A．图的遍历是从给定的源点出发对每一个顶点仅访问一次的过程
B．图的深度优先遍历方法不适用于无向图
C．使用队列对图进行广度优先遍历
D．图中有回路时则无法进行遍历

● 考虑一个背包问题，共有 n=5 个物品，背包容量为 W=10，物品的重量和价值分别为：w={2，2，6，5，4}，v={6,3，5，4，6}，求背包问题的最大装包价值。若此为 0-1 背包问题，分析该问题具有最优子结构，定义递归式为

$$c(i,j)=\begin{cases}0 & \text{若}i=0\text{或}j=0\\ c(i-1,j) & \text{若}w[i]>j\\ max\{c(i-1,j),c(i-1,j-w(i))\} & \text{其它}\end{cases}$$

其中 c（i，j）表示 i 个物品、容量为 j 的 0-1 背包问题的最大装包价值，最终要求解 c（n,W)。 采用自底向上的动态规划方法求解，得到最大装包价值为（62），算法的时间复杂度为（63）。 若此为部分背包问题，首先采用归并排序算法，根据物品的单位重量价值从大到小排序，然后依次将物品放入背包直至所有物品放入背包中或者背包再无容量，则得到的最大装包价值为（64），算法的时间复杂度为（65）。

A．11　　B．14
C．15　　D．16.67

A．Θ(nW)　　B．Θ(nlgn)
C．Θ(n2)　　D．Θ(nlgnW)

A．11　　B．14
C．15　　D．16.67

A．Θ(nW)　　B．Θ(nlgn)
C．Θ(n2)　　D．Θ(nlgnW)