

一、请回答下列关于图(Graph)的一些问题:

(1) (4 分) 有 n 个顶点的有向强连通图最多有多少条边? 最少有多少条边?

(2) (4 分) 表示一个有 1000 个顶点、1000 条边的有向图的邻接矩阵有多少个矩阵元素? 是否稀疏矩阵?

(3) (4 分) 对于一个有向图, 不用拓扑排序, 如何判断图中是否存在环?

二、斐波那契数列 F_n 定义如下:

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}, \quad n = 2, 3, \dots$$

请就此斐波那契数列, 回答下列问题。

(1) (7 分) 在递归计算 F_n 的时候, 需要对较小的 F_{n-1} , F_{n-2} , ..., F_1, F_0 精确计算多少次?

(2) (5 分) 如果用大 O 表示法, 试给出递归计算 F_n 时递归函数的时间复杂度是多少?

三、有一种简单的排序算法, 叫做计数排序 (count Sorting)。这种排序算法对一个待排序的表 (用数组表示) 进行排序, 并将排序结果存放另一个新的表中。必须注意的是, 表中所有待排序的关键码互不相同。计数排序算法针对表中的每个记录, 扫描待排序的表一趟, 统计表中

有多少个记录的关键码比该记录的关键码小。假设针对某一个记录,统计出的计数值为 c ,那么,这个记录在新的有序表中的合适的存放位置即为 c 。

(1) (3 分) 给出适用于计数排序的数据表定义;

(2) (7 分) 使用 Pascal 或 C 语言编写实现计数排序的算法;

(3) (4 分) 对于有 n 个记录的表,关键码比较次数是多少?

(4) (3 分) 与简单选择排序相比较,这种方法是否更好?为什么?

四、(10 分) 在一棵表示有序集 S 的二叉搜索树 (binary search tree) 中,任意一条从根到叶结点的路径将 S 分为 3 部分:在该路径左边结点中的元素组成的集合 S_1 ;在该路径上的结点中的元素组成的集合 S_2 ;在该路径右边结点中的元素组成的集合 S_3 。 $S = S_1 \cup S_2 \cup S_3$ 。若对于任意的 $a \in S_1, b \in S_2, c \in S_3$,是否总有 $a \leq b \leq c$? 为什么?

五、请回答下列关于堆 (Heap) 的一些问题:

(1) (4 分) 堆的存储表示是顺序的,还是链接的?

(2) (4 分) 设有一个最小堆,即堆中任意结点的关键码均大于它的左子女和右子女的关键码。其具有最大值的元素可能在什么地方?

(3) (4 分) 对 n 个元素进行初始建堆的过程中,最多做多少次数据比较(不用大 O 表示法)?

六、(12 分) 已知 Q 是一个非空队列, S 是一个空栈。仅用队列和栈的 ADT 函数和少量工作变量,使用 Pascal 或 C 语言编写一个算法,将队列 Q 中的所有元素逆置。

栈的 ADT 函数有

`makeEmpty(s:stack);`

置空栈

$push(s:stack; value:datatype);$ 新元素 $value$ 进栈
 $pop(s:stack):datatype;$ 出栈, 返回栈顶值
 $isEmpty(s:stack):boolean;$ 判栈空否

队列的 ADT 函数有

$enqueue(q:queue; value:datatype);$ 元素 $value$ 进队
 $deQueue(q:queue):datatype;$ 出队列, 返回队头值
 $isEmpty(q:queue):boolean;$ 判队列空否

八、设散列表为 $HI[0..12]$, 即表的大小为 $m=13$ 。现采用双散列法解决冲突。散列函数和再散列函数分别为:

$$H_0(key) = key \% 13; \quad \text{注: \% 是求余数运算 (=mod)}$$

$$H_i = (H_{i-1} + REV(key + 1) \% 11 + 1) \% 13;$$

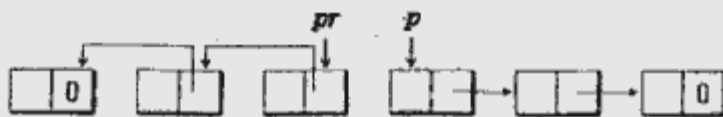
$$i = 1, 2, 3, \dots, m-1$$

其中, 函数 $REV(x)$ 表示颠倒 10 进制数 x 的各位, 如 $REV(37) = 73$, $REV(7) = 7$ 等。若插入的关键码序列为 $\{2, 8, 31, 20, 19, 18, 53, 27\}$ 。

(1) (8 分) 试画出插入这 8 个关键码后的散列表。

(2) (5 分) 计算搜索成功的平均搜索长度 ASL 。

九、从左到右及从右到左遍历一个单链表是可能的, 其方法是在从左向右遍历的过程中将连接方向逆转, 如下图所示。在图中的指针 p 指向当前正在访问的结点, 指针 pr 指向指针 p 所指结点的左侧的结点。此时, 指针 p 所指结点左侧的所有结点的链接方向都已逆转。



(1) (6 分) 使用 Pascal 或 C 语言编写一个算法, 从任一给定位置 (pr, p) 开始, 将指针 p 右移 1 个结点。如果 p 移出链表, 则将 p 置为 $NULL$, 并让 pr 停留在链表最右边的结点上。

(2) (6 分) 使用 Pascal 或 C 语言编写一个算法, 从任一给定位置 (pr, p) 开始, 将指针 p 左移 1 个结点。如果 p 移出链表, 则将 p 置为 $NULL$, 并让 pr 停留在链表最左边的结点上。