

## 习题解答 (部分)

张光河

[guanghezhang@163.com](mailto:guanghezhang@163.com)

2018 年 08 月

目录

习题一 .....3

习题二 .....5

习题三 .....7

习题四 .....9

习题五 .....11

习题六 .....14

习题七 .....18

习题八 .....20

习题九 .....22

## 习题一

### 一、选择题

1. 下列有关说法不正确的是 ( **D** )。
  - A. 数据元素是数据的基本单位
  - B. 数据项是数据中不可分割的最小可标识单位
  - C. 数据可由若干个数据元素构成
  - D. 数据项可由若干个数据元素组成
2. 计算机所处理的数据一般具备某种内在联系, 这是指 ( **B** )。
  - A. 数据和数据之间存在某种关系
  - B. 元素和元素之间存在某种关系
  - C. 元素内部存在某种关系
  - D. 数据项和数据项之间存在某种关系
3. 从逻辑上可以把数据结构分为 ( **C** ) 两大类。
  - A. 动态结构和静态结构
  - B. 顺序结构和链式结构
  - C. 线性结构和非线性结构
  - D. 初等结构和构造型结构
4. 下面关于算法的说法正确的是 ( **D** )。
  - A. 算法最终必须由计算机程序执行
  - B. 算法就是为解决某一问题而编写的程序
  - C. 算法的可行性是指不能有二义性指令
  - D. 以上几个都是错误的
5. 算法的时间复杂度取决于 ( **C** )。
  - A. 问题的规模
  - B. 待处理数据的初态
  - C. A 和 B
  - D. 以上都不是

## 二、填空题

1. 数据项是数据元素中 不可分割 的最小标识单位, 通常不具备完整、确定的实际意义, 只是反映数据元素某一方面的属性。

2. 数据的逻辑结构通常分为 集合、线性结构、树形结构 和 图状(或网状)结构。

3. 数据的存储结构通常分为 顺序存储结构、链式存储结构、索引存储结构 和 哈希(或散列)存储结构。

4. 一个算法有 5 个特性, 即 有穷性、确定性、可行性、输入 和 输出。

5. 在对算法的空间复杂度进行分析时, 只需考虑 临时变量 所占用的存储空间而不用考虑 形参 占用的存储空间。

## 三、编程题(略)

## 习题二

### 一、选择题

1. 顺序表比链表的存储密度更大, 是因为 ( B )。
  - A. 顺序表的存储空间是预先分配的
  - B. 顺序表不需要增加指针来表示元素之间的逻辑关系
  - C. 链表的所有结点是连续的
  - D. 顺序表的存储空间是不连续的
2. 假定顺序表中第一个数据元素的存储地址为第 1000 个存储单元, 若每个数据元素占用 3 个存储单元, 则第五个元素的地址是第 ( C ) 个存储单元。
  - A. 1015
  - B. 1005
  - C. 1012
  - D. 1010
3. 若将某一数组 A 中的元素, 通过头插法插入至单链表 B 中 (单链表初始为空), 则插入完毕后, B 中结点的顺序 ( A )。
  - A. 与数组中元素的顺序相反
  - B. 与数组中元素的顺序相同
  - C. 与数组中元素的顺序无关
  - D. 与数组中元素的顺序部分相同、部分相反
4. 与单链表相比, 双链表 ( B )。
  - A. 可随机访问表中结点
  - B. 访问前后结点更为便捷
  - C. 执行插入、删除操作更为简单
  - D. 存储密度等于 1
5. 在一个含有  $n$  个结点的有序循环双链表中插入一个结点后, 仍保持循环双链表的有序, 其算法的时间复杂度为 ( A )。
  - A.  $O(n)$

- B .  $O(1)$
- C .  $O(\log_2 n)$
- D .  $O(n^2)$

## 二、填空题

- 1 . 我们将以顺序存储结构实现的线性表称为 顺序表 。
- 2 . 我们将以链式存储结构实现的线性表称为 链表 。
- 3 . 在单链表中 , 我们若想在头结点之前插入一个新结点 nNode 可通过执行 nNode.next=self.head.next 和 self.head.next=nNode 两条语句实现。

注意: 此处 “在头结点之前插入一个新结点 nNode” 是指将新结点 nNode 插入至头结点后的第一个位置。

- 4 . 在某一双链表中 , 假定 cNode 已经指向了当前待删除的结点 , 若想成功将该结点删除 , 需要执行的操作对应的代码为 pNode=cNode.prev、qNode=cNode.next、pNode.next=qNode、qNode.prev=pNode、del cNode。

- 5 . 循环单链表是在单链表的基础上 将其自身的第一个结点的地址存入表中最后一个结点的指针域中 , 循环双链表是在双链表的基础上 将双链表中最后一个结点的后继指针指向双链表的头结点 , 并将其头结点的先驱指针指向表中最后一个结点。

## 三、编程题 ( 略 )

## 习题三

### 一、选择题

1. 对于一个顺序栈, 栈中能存储的元素个数最多不超过正整数  $\text{MaxStackSize}$  ( 栈顶指针  $\text{top}$  的初值为  $-1$  ), 对于栈满条件的判断应该为 ( C )。

A .  $\text{top} \neq \text{MaxStackSize} - 1$

B .  $\text{top} \neq \text{MaxStackSize}$

C .  $\text{top} < \text{MaxStackSize} - 1$

D .  $\text{top} < \text{MaxStackSize}$

2. 让元素  $a$ 、 $b$ 、 $c$ 、 $d$ 、 $e$  依次进入一个链式栈中, 则出栈的顺序不可能是 ( C )。

A .  $e$ 、 $d$ 、 $c$ 、 $b$ 、 $a$

B .  $b$ 、 $a$ 、 $e$ 、 $d$ 、 $c$

C .  $d$ 、 $c$ 、 $a$ 、 $b$ 、 $e$

D .  $b$ 、 $c$ 、 $e$ 、 $d$ 、 $a$

3. 设栈  $S$  和队列  $Q$  的初始状态均为空, 元素  $a$ 、 $b$ 、 $c$ 、 $d$ 、 $e$ 、 $f$ 、 $g$  依次进入栈  $S$ 。若每个元素出栈后立即进入队列  $Q$ , 且 7 个元素出队的顺序是  $b$ 、 $d$ 、 $c$ 、 $f$ 、 $e$ 、 $a$ 、 $g$ , 则栈  $S$  的容量至少是 ( C )。

A . 1

B . 2

C . 3

D . 4

4. 带头结点的链式队列, 其队头指针指向实际队头元素所在结点的前一个结点, 其队尾指针指向队尾结点, 则在进行出队操作时 ( D )。

A . 修改队头指针

B . 修改队尾指针

C . 队头和队尾指针都要修改

D . 队头和队尾指针可能都要修改

5 . 设有一个递归算法如下。

```
def fact(self,n):  
    if n<=0:  
        return 1  
    else:  
        return self.fact(n-1)*n
```

计算 fact(n)需要调用该函数的次数为 ( A )。

A .  $n+1$

B .  $n-1$

C .  $n$

D .  $n+2$

## 二、填空题

1 . 一个栈的进栈序列为  $1, 2, 3, \dots, n$  , 对应的出栈序列为  $S_1, S_2, S_3, \dots, S_n$  。若  $S_2=3$  , 则  $S_3$  可能取值的个数为  $n-1$  。

2 . 引入循环队列的目的是 提高存储空间的利用率 。

3 . 利用长度为  $n$  的列表存储循环队列的元素, 队头指针 front 指向实际队头元素所在位置的前一个位置, 队尾指针 rear 指向实际队尾元素, 则入队时的操作为  $rear=(rear+1)\%n$  , 出队时的操作为  $front=(front+1)\%n$  。

4 . 栈和队列的共同点是 都是操作受限的线性表 。

5 . 一个递归算法必须包括 终止条件 和 递归部分 。

## 三、编程题 ( 略 )



## 习题四

### 一、选择题

1. 现有两个串分别为  $S1 = \text{"abdcefg"}$ ,  $S2 = \text{"MLHWP"}$ , 对其执行以下操作( $S1.SubString(0, S2.GetStringLentgh())$ ).StringConcat( $S1.SubString(S2.GetStringLentgh()-1, 2)$ )后的结果为 ( **B** )。

- A . bcdef
- B . bdcefg
- C . bcMLHWP
- D . bcdefef

2. 若串  $S = \text{"software"}$ , 则其子串和真子串数目分别为 ( **B** )。

- A . 8 , 7
- B . 37 , 36
- C . 36 , 35
- D . 9 , 8

3. 模式串  $T = \text{"ABABAABAB"}$  的 ListNextValue 值为 ( **A** )。

- A . (0,1,0,1,0,4,1,0,1)
- B . (0,1,0,1,0,2,1,0,1)
- C . (0,1,0,1,0,0,0,1,1)
- D . (0,1,0,1,0,1,0,1,1)

注意: 教材中介绍时是以 “-1” 代表没有位置可以移动, 而这里分别是以 “0” 表示没有位置移动, 下标从 “1” 开始。

4. 设矩阵  $A$  是一个对称矩阵, 为了节省存储空间, 将其下三角部分按照行优先存放在一个一维数组  $B[0, \dots, n(n+1)/2-1]$  中, 对于下三角部分中任意一元素  $a[i][j]$  ( $i \geq j$ ), 在一维数组  $B$  中的下标  $k$  的值为 ( **B** )。

- A .  $i(i-1)/2+j-1$
- B .  $i(i+1)/2+j$

C .  $i(i+1)/2+j-1$

D .  $i(i-1)/2+j$

5 . 广义表((a,b,c,d))的表头和表尾分别为 ( **D** )。

A . a , (b,c,d)

B . a , ((b,c,d))

C . (a,b,c,d) , 表尾为空

D . (a,b,c,d) , ()

## 二、填空题

1 . 两个串相等的充分必要条件为 两个串的长度相等且对应位置的字符依次相同。

2 . 模式串 T="ababaab" 的 ListNext 和 ListNextValue 函数值分别为 0,1,1,2,3,4,2; 0,1,0,1,0,4,1。

注意：教材中介绍时是以“-1”代表没有位置可以移动，而这里分别是以“0”表示没有位置移动，下标从“1”开始。

3 . 设有二维数组 A[30][50]，其元素长度为 4 字节，按行优先顺序存储，基址为 100，则元素 A[23][42] 的存储地址为 4868。

4 . 稀疏矩阵常用的压缩存储方式为 三元组顺序表。

5 . 广义表(a,(a,b),d,e,(i,j),k)的长度为 6，其表头和表尾分别为 a; ((a,b),d,e,(i,j),k)。

## 三、编程题 (略)

## 习题五

### 一、选择题

1. 深度为  $h$  的满  $m$  叉树的第  $k$  层有 ( **A** ) 个结点。 (  $1 \leq k \leq h$  )

A.  $m^{k-1}$

B.  $m^k - 1$

C.  $m^{h-1}$

D.  $m^h - 1$

2. 一棵具有 1028 个结点的二叉树的深度  $h$  为 ( **C** )。

A. 11

B. 10

C.  $11 \sim 1028$

D.  $10 \sim 1027$

3. 关于二叉树的说法正确的是 ( **B** )。

A. 所有二叉树的度均为 2

B. 一棵二叉树的度可以小于 2

C. 一棵二叉树中至少有一个结点的度为 2

D. 一棵二叉树的根结点的度必为 2

4. 一棵满二叉树的层次遍历的结果为 ABCDEFG, 则先序遍历该满二叉树得到的先序序列为 ( **B** )。

A. ABCEFDG

B. ABDECFG

C. ACGFBED

D. ABEDCGF

5. 假定在一棵二叉树中, 度为 2 的结点的数目为 6, 则该二叉树中叶子结点的数目是 ( **C** )。

A. 6

B. 5

C . 7

D . 8

## 二、填空题

1 . 图 5-52 所示的树的深度和度分别为 4 和 8 , 若将该树转换为森林, 则转换后得到的森林中共包含 3 棵树。

2 . 遍历图 5-53 所示的二叉树, 得到的先序序列为 ABCDEF、中序序列为 CBDAEF、后序序列为 CDBFEA。

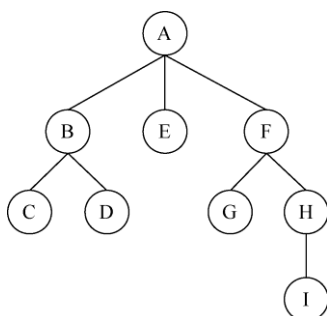


图5-52 一棵树

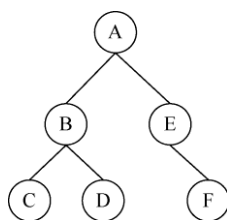


图 5-53 一棵二叉树

3 . 已知有一棵深度为 5 的完全二叉树, 共有 23 个结点, 则该树一共有 12 个叶子结点。

4 . 将图 5-52 所示的树转换为二叉树后, 值为 F 的结点的左孩子的值为 G。

5 . 假设在某次通信时的一份报文中只包含 A、B、C、D、E 这 5 种字符, 它们在该报文中出现的频率分别为 0.1、0.2、0.43、0.09、0.18。若为上述 5 种字符设计哈夫曼编码, 则字符 D 的编码为 1110 ( 低频率在左, 高频率在右, 向左走为 “0”, 向右走为 “1” 对应的编码结果 )。

### 三、算法设计题 (若无特别说明, 本题中均采用二叉链表存储二叉树, [解答过程略](#)。)

## 习题六

### 一、选择题

1. 图 6-58 所示的有向图中, 顶点 A 的入度为 ( **B** )。  
A. 4  
B. 1  
C. 3  
D. 0
2. 若某图有 4 个顶点, 它们的度分别为 3、1、2、2, 则该图共有 ( **B** ) 条边或弧。  
A. 6  
B. 8  
C. 9  
D. 10
3. 一棵有  $n$  个顶点的生成树有且仅有 ( **D** ) 条边。  
A.  $n+2$   
B.  $n+1$   
C.  $n$   
D.  $n-1$
4. 对于图 6-59 所示的无向图, 以顶点 A 为起点, 对其进行深度优先遍历所得的序列不可能是 ( **A** )。  
A. AEBCD  
B. ACDEB  
C. ABCDE  
D. ADECB

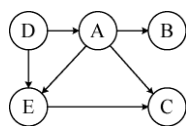


图 6-58 有向图

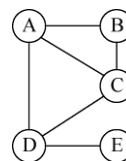


图 6-59 无向图

5. 对于图 6-60 所示的连通网, 以顶点 A 为起点的最小生成树为 ( **B** )。

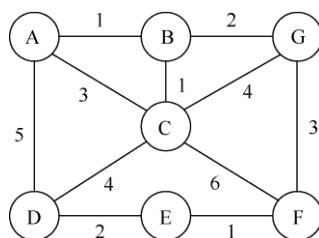
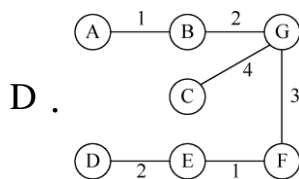
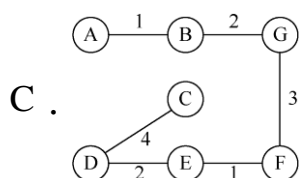
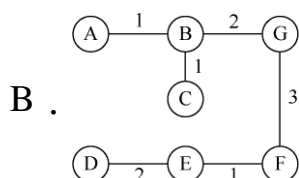
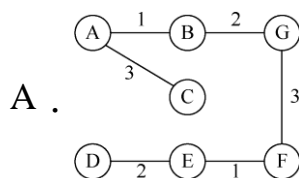


图 6-60 连通网

## 二、填空题

1. 对图 6-61( a )所示的无向图, 若其对应的邻接表如图 6-61( b )所示, 则对其进行广度优先遍历的结果为 **AEBDC**。

注意：图 6-61 ( b ) 所示的邻接表中的数字与字母对应关系为 ( A:0;B:1;C:2;D:3;E:4 ), 题目没有给出遍历起点, 给的答案是以“A”为起点的。

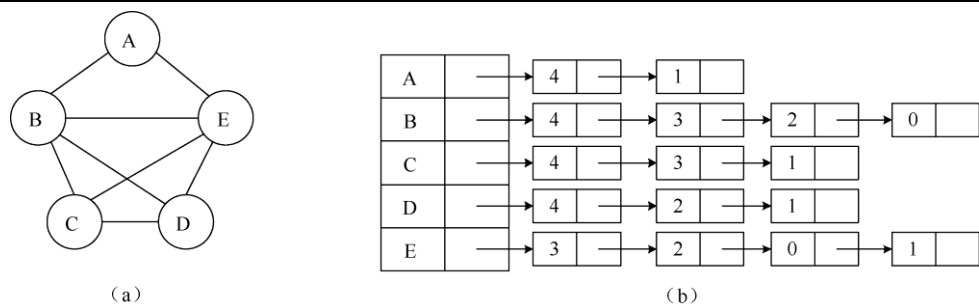


图 6-61 无向图及其对应的邻接表

- 2 对图 6-62 所示的连通网 其最小生成树为\_\_\_\_\_。
- 3 . 对图 6-63 所示的有向网 , 以顶点 A 为起点 , 并以顶点 G 为终点求最短路径的结果为 ABDFG。
- 4 .对图 6-64 所示的 AOV 网 ,其可能的拓扑序列为 ACEBDFG、ACEBFDG、ACEDBFG、ACEDFBG、ACEFBDG、ACEFDBG、ACBEDFG、ACBEFDG、ABCEDFG、ABCEFDG。
- 5 .对图 6-65 所示的 AOE 网 ,对其求关键路径的结果为 ADEF。

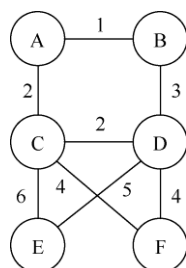


图6-62 连通网

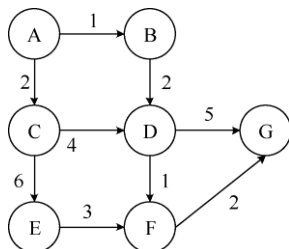


图 6-63 有向网



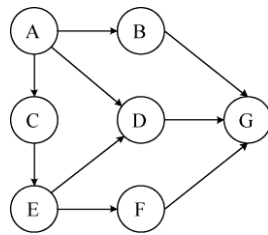


图6-64 AOV网

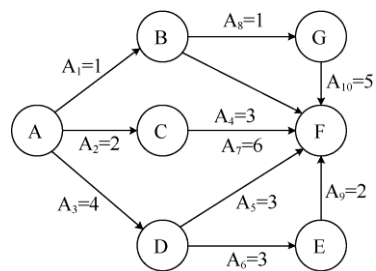


图 6-65 AOE 网

### 三、算法设计题(略)

## 习题七

### 一、选择题

1. 在下列查找方法中, 适用于静态查找的方法有 ( **B** )。
  - A. 折半查找、二叉排序树查找
  - B. 折半查找、索引查找
  - C. 二叉排序树查找、顺序查找
  - D. 哈希表查找、索引查找
2. 对含有 10 个数据元素的有序查找表执行折半查找, 当查找失败时, 至少需要比较 ( **C** ) 次。
  - A. 2
  - B. 3
  - C. 4
  - D. 5
3. 下列选项中 ( **C** ) 可能是在二叉排序树中查找 35 时所比较的关键字序列。
  - A. 2,25,40,39,53,34,35
  - B. 25,39,2,40,53,34,35
  - C. 53,40,2,25,34,39,35
  - D. 39,25,40,53,34,2,35
4. 在平衡二叉树中, 每个结点的平衡因子的取值范围为 ( **A** )。
  - A.  $-1 \sim 1$
  - B.  $0 \sim 1$
  - C.  $-2 \sim 2$
  - D.  $-2 \sim 1$
5. 下列关于 B-树和 B+树的叙述中, 不正确的一项是 ( **C** )。
  - A. 都是平衡多叉树
  - B. 都可用于文件的索引结构
  - C. 都能有效地支持顺序检索

## D . 都能有效地支持随机检索

### 二、填空题

1 . 对含有  $n$  个元素的查找表执行顺序查找时 , 假定每个元素的查找概率相同 , 其平均查找长度为  $3(n+1)/4$ 。

2 插入结点后引起 AVL 树失去平衡的调整方式分别为 LL 型、RR 型、LR 型、RL 型。

3 .  $m$  阶 B-树的非叶子结点至多有  $m-1$  个关键字。

4 . 构造哈希函数最常用的方法是直接定址法、除留余数法、数字分析法、平方取中法、折叠法。

5 . 解决哈希冲突的两类方法是开放地址法、拉链法。

### 三、编程题 ( 略 )

## 习题八

### 一、选择题

1. 在待排序序列大致有序的情况下, 直接插入排序算法所需的时间较少。对下列 ( **D** ) 序列进行直接插入排序时, 所需移动记录的次数最少。

A . 56,23,87,90,17,33

B . 90,17,56,23,87,33

C . 23,87,33,56,90,17

D . 56,23,17,87,33,90

2. 在下述排序算法中, ( **A** ) 是稳定的排序算法。

A . 归并排序

B . 快速排序

C . 希尔排序

D . 堆排序

3. 若对序列{90,17,56,23,87,33}建小根堆, 其结果为 ( **B** )。

A . 17,56,23,87,90,33

B . 17,23,33,90,87,56

C . 17,23,33,56,87,90

D . 17,56,23,87,33,90

4. 对含有  $n$  个记录的序列, 进行冒泡排序的平均时间复杂度为 ( **D** ), 进行二路归并排序的平均时间复杂度为 ( **D** )。

A .  $O(n\log n)$ ,  $O(n)$

B .  $O(n\log n)$ ,  $O(n\log n)$

C .  $O(n\log n)$ ,  $O(n^2)$

D .  $O(n^2)$ ,  $O(n\log n)$

5. ( **C** ) 在完成第一趟排序后, 至少能保证一个记录在最终位置上。

A . 快速排序

B . 二路归并排序

C . 简单选择排序

D . 折半插入排序

## 二、填空题

1 . 若对序列{90,17,56,23,87,33}进行初始增量为 2 的希尔排序 , 则完成一趟排序后的序列为 56,17, 87, 23, 90,33。

2 . 若对含有 50 个记录的序列进行堆排序 , 建立初始堆的高度为 6 , 最后一个非终端结点的下标为 24 ( 假定起始下标为 0 )。

3 . 若对序列{89,17,56,23,28,31}进行基数排序 , 则对此序列的个位数关键字进行分配和收集后的结果为 31,23,56,17,28,89。

4 . 在对含有 10 个记录的序列进行直接插入排序时 , 最少需要进行 9 次记录的比较。

5 . 若将序列{10,37,56,66,98}和序列{14,16,40,49,77}进行归并 , 得到的序列为 10,16,40,49,77,14,37,56,66,98。

## 三、编程题(略)

## 习题九

### 一、选择题

1. 外排序最主要的特点是 ( C )。
  - A. 排序速度较快
  - B. 所需内存较小
  - C. 需涉及内、外存数据交换
  - D. 进行外排序的数据需全部存储在内存中
2. 进行多路平衡归并是为了 ( C )。
  - A. 创建败者树
  - B. 减少归并段的个数
  - C. 减少归并总次数
  - D. 创建最佳归并树
3. 若初始归并段为  $n$  个, 此时采用  $k$  路归并, 需归并的总次数  $s$  应为 ( B )。
  - A.  $n_k$
  - B.  $\log_k n$
  - C.  $\log_n k$
  - D.  $\sqrt[k]{n}$
4.  $m$  个归并段采用  $k$  路平衡归并时, 对应的败者树共有 ( B ) 个结点。
  - A.  $2k$
  - B.  $2k-1$
  - C.  $2m-1$
  - D.  $2m$
5. 现有一个记录序列(43,48,80,61,42,58,21,65,96,50), 若内存工作区可容纳的记录个数为 5, 则对该序列采用置换-选择算法可产生 ( A ) 个递增有序段。
  - A. 2

B . 3

C . 4

D . 1

## 二、填空题

1 . 采用归并算法进行外排序时 , 需经过文件读入、子文件排序和子文件归并 3 个阶段。

2 . 外排序可采用归并排序的方法实现对数据的排序处理 , 但在进行归并处理前 , 首先需生成\_\_\_\_\_ 初始归并段 \_\_\_\_\_。

3 .  $n$  个归并段进行 3 路排序 , 其所需的归并次数  $s$  为  $\lceil \log_3 n \rceil$ 。

4 . 败者树中的胜者是关键字较小的结点。

5 . 现有一组序列(62,96,74,66,92,87,40,72,75) , 若此时内存工作区最多可容纳两个记录 , 则采用置换-选择排序算法时 , 产生的归并段为 {62, 74, 96} , {66, 87, 92} , {40,72,75}。

## 三、简答题(略)