

中南大学考试试卷

参考答案与评分标准

2019--2020 学年 2 学期 时间 100 分钟 2020 年 9 月 6 日

数据结构 课程 56 学时 3.5 学分 考试形式: 闭卷

专业年级: 计算机信息类 19 级 总分 100 分, 占总评成绩 60 %

注: 此页不作答题纸, 请将答案写在答题纸上

一、填空题: (每题 1 分, 共 10 分)

1. 某算法时间复杂度 $T(n)$ 表达式是 $100n^4+3n^3+1000n+2\log_2n+999$, 可用大 O 表示成_____。
2. 循环队列用数组 $A[0, \max-1]$ 存放其元素值, 已知其头尾指针分别是 $front$ 和 $rear$, 则该队列满的条件是_____。
3. 设数组 $a[0..10, 2..20]$ 的基地址为 10, 每个元素占 4 个存储单元, 若以行序为主序顺序存储, 则元素 $a[5, 5]$ 的存储地址为_____。
4. 已知一个有向图的邻接矩阵表示, 计算结点 v 的入度的方法是_____。
5. 给定 n 个不同关键字, 则建立的哈夫曼树总结点数为_____。
6. 一个深度为 h 的完全二叉树结点数目最多为_____。
7. 对广义表 $A=((a), (b, (c, d)), ((e), f))$, 运算 $\text{tail}(\text{head}(\text{tail}(\text{head}(\text{tail}(A))))))$ 的结果是_____。
8. 写出带头结点的双向循环链表 L 为空表的条件_____。
9. 表达式 $x*(y/z+a*b)-c$ 的后缀形式是_____。
10. 在长度为 n 的顺序表中的第 i 个元素 ($1 \leq i \leq n$) 之前插入一个元素时, 需向后移动_____个元素。

Solution:

1. $O(n^4)$
2. $(rear+1)\%max==front$
3. 402
4. 顶点 v 所在列中 1 的数目
5. $2n-1$
6. 2^h-1
7. (d)
8. $L->prior==L->next==L$
9. $xyz/ab*+*c-$
10. $n-i+1$

二、选择题: (每题 2 分, 共 20 分)

1. 一个栈的输入序列是 a, b, c, d, e, f , 则栈的不可能输出序列是 ()
A. f, e, d, c, b, a B. a, b, d, c, f, e C. a, c, d, b, f, e D. c, d, a, b, e, f
2. 下列说法正确的是 ()

- A.带权的连通无向图的最小（代价）生成树唯一。
 B.无向图的邻接矩阵一定是对称矩阵，有向图的邻接矩阵一定是非对称矩阵。
 C.拓扑排序可以用来判断有向图有没有环。
 D.深度优先遍历生成树描述了从起点到各顶点的最短路径。
- 3.已知有序表（6, 10, 17, 19, 26, 33, 49, 51, 68），当折半查找值为 19 的元素时，查找成功的比较次数为（ ）。
 A. 1 B. 2 C. 3 D. 4
4. 已知串S='aaab'，则next数组值为（ A ）。
 A. 0123 B. 1123 C. 1231 D. 1211
- 5.给定输入序列{30, 27, 46, 15, 29, 62, 88, 57, 9, 90}构造二叉排序树，则所建二叉排序树的平均查找长度为（ ）。
 A. 3 B. 3.1 C. 3.2 D. 3.3
6. 若广义表 K 满足 $\text{head}(K)=\text{tail}(K)$,则 K 为（ ）。
 A. () B. (()) C. ((), ()) D. ((), (), ())
- 7.在下面的排序方法中，关键字比较的次数与记录的初始排列次序无关的是（ ）。
 A. 快速排序 B. 冒泡排序 C. 插入排序 D. 选择排序
8. 在一棵度为 3 的树中，度为 3 的结点个数为 3，度为 2 的结点个数为 2，则度为 0 的结点个数为（ ）。
 A. 6 B. 7 C. 8 D. 9
- 9.设二叉树的先序遍历序列和后序遍历序列正好相同，则该二叉树满足的条件是（ ）。
 A. 空或只有一个结点 B. 高度等于其结点数
 C. 任一结点无左孩子 D. 任一结点无右孩子
- 10.通过构建最小堆对关键字序列 (49, 56, 69, 32, 28, 77, 89, 18, 46)进行排序，则排序过程中初始堆的序列状态为()
 A. (18, 28, 69, 32, 46, 77, 89, 49, 56) B. (18, 32, 46, 69, 77, 89, 28, 56, 49)
 C. (18, 32, 69, 46, 28, 77, 89, 56, 49) D. (18, 28, 69, 32, 49, 77, 89, 56, 46)

1-5 DCDAB 6-10 BDDAD

三. 计算题（共 5 小题，每小题 8 分，共 40 分）

1. 假设以数组 $\text{seqn}[m]$ 存放循环队列的元素，设变量 rear 和 quelen 分别指示循环队列中队尾元素的位置和元素的个数。

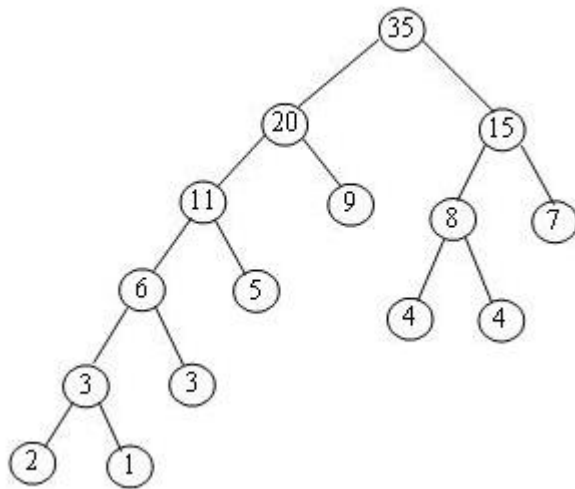
- (1)写出队满的条件表达式；
 (2)写出队空的条件表达式；
 (3)设 $m=40, \text{rear}=13, \text{quelen}=19$,求队头元素的位置；
 (4)写出一般情况下队头元素位置的表达式。

解答：

- (1) $m=\text{quelen}$ (2 分)
 (2) $\text{quelen}=0$ (2 分)
 (3) 队头元素的位置 $\text{front}=34$ (2 分)
 (4) 队头元素的位置 $\text{front}=(\text{rear}+m-\text{quelen})\%m$ (2 分)

2. 已知某字符串 S 中共有 8 种字符，各种字符分别出现 2 次、1 次、4 次、5 次、7 次、3

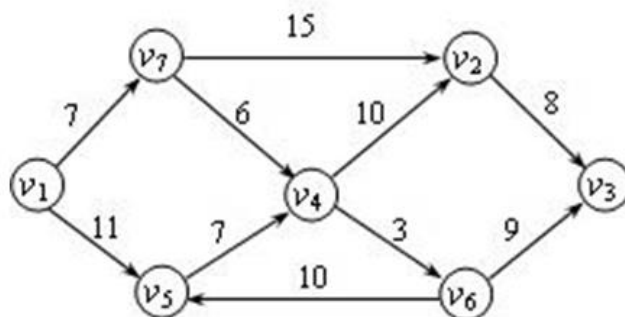
次、4 次和 9 次，对该字符串用[0, 1]进行前缀编码，问该字符串的编码至少有多少位。以
 解答：如图所示，由哈夫曼编码树可得各字母的编码如下：2+6



各字符出现的次数作为叶子结点的权值构造的哈夫曼编码树如图所示。其带权路径长度
 $= 2 \times 5 + 1 \times 5 + 3 \times 4 + 5 \times 3 + 9 \times 2 + 4 \times 3 + 4 \times 3 + 7 \times 2 = 98$ ，所以，该字符串的编码长度至少为 98 位。

3. 试对图 3 所示的带权有向图，

- (1) 按 Dijkstra 算法执行过程求从源点 v_1 到其他各顶点的最短路径及最短路径长度。
- (2) Dijkstra 算法在应用中有什么限制？



第 3 题图

解答：(1) 列表求解 5 分，给出路径 2 分，参考答案均见下表。

(2) 图中含负权边时，算法不一定获得正确结果。1 分

		V1	V2	V3	V4	V5	V6	V7
V1	1	0	∞	∞	∞	11 (V1,V5)	∞	7 (V1,V7)
V7	2	0	22 (V1,V7,V2)	∞	13 (V1,V7,V4)	11 (V1,V5)	∞	7 (V1,V7)
V5	3	0	22 (V1,V7,V2)	∞	13 (V1,V7,V4)	11 (V1,V5)	∞	7 (V1,V7)
V4	4	0	22 (V1,V7,V2)	∞	13 (V1,V7,V4)	11 (V1,V5)	16 (V1,V7,V4,V6)	7 (V1,V7)
V6	5	0	22 (V1,V7,V2)	25 (V1,V7,V4,V6,V3)	13 (V1,V7,V4)	11 (V1,V5)	16 (V1,V7,V4,V6)	7 (V1,V7)
V2	6	0	22 (V1,V7,V2)	25 (V1,V7,V4,V6,V3)	13 (V1,V7,V4)	11 (V1,V5)	16 (V1,V7,V4,V6)	7 (V1,V7)
V3	7	0	22 (V1,V7,V2)	25 (V1,V7,V4,V6,V3)	13 (V1,V7,V4)	11 (V1,V5)	16 (V1,V7,V4,V6)	7 (V1,V7)

4. 设有一组关键字 {9, 7, 26, 16, 37, 22, 89, 56}，采用哈希函数： $H(\text{key}) = (3 \times \text{key}) \bmod 11$ ，表长为 12，用二次探测再散列方法 $H_i = (H(\text{key}) + d_i) \bmod 11$ ($d_i = 1^2, -1^2, 2^2, -2^2, 3^2, \dots$) 解决冲突。

(1) 构造哈希表；(2) 计算查找成功的平均查找长度。

解答：6+2 分

0	1	2	3	4	5	6	7	8	9	10
22	26	37	89	16	9		56			7

平均查找长度：12/8=3/2

5. 算法填空题 8 分

如下为二分查找的非递归算法，试将其填写完整。

Int Binsch(ElemType A[], int n, KeyType K)

```

{
    int low=0;
    int high=n-1;
    while (_____)
    {
        int mid=_____
        if (K==A[mid].key) return mid; //查找成功，返回元素的下标
        else if (K<A[mid].key)
            _____; //在左子表上继续查找
        else _____; //在右子表上继续查找
    }
}

```

```

return -1;        //查找失败，返回-1
}

```

low<=high ; **(low+high)/2 ;** **high=mid-1 ; low=mid+1** 各 2 分

四. 算法设计题（共 2 小题，每题 15 分，共 30 分）酌情给分。

1. 输入一个字符串，内有数字和非数字字符，如：ak123x456 17960?302gef4563，将其中连续的数字作为一个整体，依次存放到一个数组 a 中，如 123 放入 a [0]，456 放入 a [1]，… …。编程统计其共有多少个整数，并输出这些数。

参考算法

```

int CountInt ( )
// 从键盘输入字符串，连续的数字字符算作一个整数，统计其中整数的个数。
{int i=0, a[];          // 整数存储到数组 a, i 记整数个数
scanf ("%c", &ch); // 从左到右读入字符串
while (ch!='#') // '#'是字符串结束标记
{if (isdigit (ch)) // 是数字字符
{num=0;          // 数初始化
while (isdigit (ch) && ch!='#') // 拼数
{num=num*10+'ch'-'0';
scanf ("%c", &ch);
}
a[i]=num; i++;
if (ch!='#') scanf ("%c", &ch); // 若拼数中输入了'#'，则不再输入
} // 结束 while (ch!= '#')
printf ("共有%d 个整数，它们是： "i);
for (j= 0 ; j<i; j++)
{printf ("%6d", a[j]);
if ((j+1) %10==0) printf ("\n"); }// 每 10 个数输出在一行上
} // 算法结束

```

2. 编写根据无向图 G 的邻接表，判断图 G 是否连通的算法。

将图的深度优先遍历或广度优先遍历算法稍加改造，另设 m 保存访问的顶点个数，若已访问顶点个数 m 小于图中顶点个数 n，则图 G 不连通，否则为连通。现将深度优先遍历的非递归算法改造如下：

采用邻接表表示，类型定义如下：

```

#define MaxV 20
typedef struct node
{int adjvex;
struct node *next;
} Enode;
typedef struct
{int data;
Enode *firste;

```

```

    }Vnode;
typedef struct
    {Vnode adjlist[MaxV];
    int n;
    int e;
    }ALGraph;
int connect(ALGraph *G)
{int i,j,Visited[MaxV],m=0,top=0;
  Enode *p,*S[MaxV];
  For(i=0;i<n;i++)
    Visited[i]=0;
  Visited[0]=0;
  Printf("%c",G->adjlist[0].data);
  m++;
  S[top++]=p->next;
  j=p->adjvex;
  if(visited[j]==0)
    {visited[j]=1;
    printf("%c",G->adjlist[j].data);
    m++;
    s[top++]=G->adjlist[j]->firste;
    }
  }
}
if(m<G->a)
  return(0);
else
  return(1);
}

```