

安徽大学 20_23—20_24 学年第 2 学期

《数据结构》考试试卷 (A 卷试题) (闭卷 时间 120 分钟)

考场登记表序号 _____

一、算法阅读题 (每小题 5 分, 共 20 分)

1. 设 n 是描述问题规模的非负整数, 请仔细阅读下面的算法, 给出其算法时间复杂度。

```
int sum=0, i=1;
while (i<n) {
    sum+=i;
    i=i*2;
}
```

2. 设 n 是描述问题规模的非负整数, 请仔细阅读下面的算法, 给出其算法时间复杂度。

```
x=300;
y=3;
while(x≥(y+2)*(y+2))
    y++;
```

3. 阅读算法, 将下面的算法补充完整。其中, 链栈的结点表示为(*data, next*), *top* 指向栈顶, 插入一个新结点 (用指针 *x* 指向) 的操作为:

```
void push_S(LinkStack & S, ElemtType e) {
    x=(LinkList)malloc(sizeof(SNode));
    x->data=e;
    _____;
    _____;
}
```

4. 以下程序是递归交换二叉树每个结点的左孩子和右孩子。要求在下面程序中下划线标记处填上合适的语句以使其完整:

```
void ChangeLR( BiTree & T ) {
    if (T==NULL) return;
    if (T->lchild==NULL && T->rchild==NULL) return;
    else {
        Temp=T->lchild;
        T->lchild=T->rchild;
        T->rchild=Temp;
    }
    _____;
    _____;
}
```

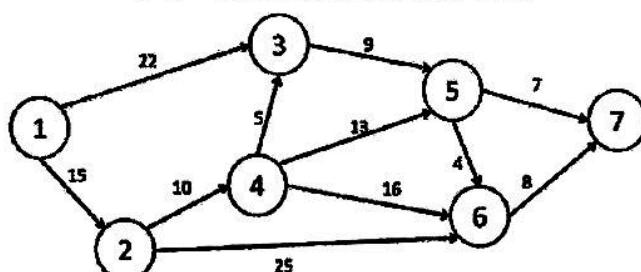
二、简答题（每小题 5 分，共 20 分）

5. 在对称数组 A 中，元素 $A[i, j]$ 的长度为 32 个二进位，其中行和列下标均从 0 到 5 编号。现从首地址 S 开始按行优先连续存放下三角矩阵到主存储器中（储器字长为 16 位）。求：
 (1) 存放数组 A 共需要多少个字节？
 (2) $A[3][5]$ 的起始存储地址为多少？
6. 对 n 个互不相同的符号进行哈夫曼编码。若生成的哈夫曼树共有 117 个结点，则 n 的值是多少？
7. 请将西瓜 watermelon 用工具 H()—Head(), T()—Tail() 从 L 中取出。其中，L 为：
 $L=(apple, banana, (orange, (strawberry, (watermelon)), peach), pear)$
8. 在堆排序算法中，请写出将下列初始序列调整成大根堆的结果。
 {56, 32, 69, 41, 88, 47, 84, 30, 66, 22, 42, 54, 60, 27, 86, 19}

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
56	32	69	41	88	47	84	30	66	22	42	54	60	27	86	19

三、应用题（每题 10 分，共 30 分）

9. 设一棵二叉树中序序列：DBGIEAFHC，后序序列：DIGEBHFCA
 (1) 画出这棵二叉树；
 (2) 画出这棵二叉树的先序线索二叉树；
 (3) 将这棵二叉树转换成对应的树（或森林）。
10. 关键字输入顺序为：(25, 36, 18, 39, 55, 60, 22, 66, 33, 72)，请按步骤画出建立3阶B-树的过程。
11. 设有如下所示的 AOE 图，活动上的权值表示其持续时间。



- (1) 在下表中列出各事件的最早发生时间。

事件	1	2	3	4	5	6	7
最早发生时间	0						

- (2) 求出该 AOE 图的关键路径。

四、算法设计题（每小题 10 分，共 30 分）

12. 回文是指正读反读均相同的字符序列，如“abba”和“bacab”均为回文，但“good”不是回文。试写一个算法判定给定的字符序列是否为回文。要求：
 (1) 给出算法的基本设计思想；

(2) 以下函数可直接使用:

```
strlen(t);
InitStack(S);
EmptyStack(s)
Push(S, x); //将x压入栈顶
Pop(S); //摘取栈顶元素，并返回
根据设计思想，采用C或者C++语言描述算法，关键之处给出注释。
int IsPalindrome(char *t) //判断t是否回文，若是则返回1，否则返回0.
```

13. 2024年4月25日20时59分，我国成功发射了神舟18号载人飞船，把叶光富、李聪、李广苏3名80后航天员送到“中国”空间站内生活、工作6个月。载人航天工程是包含众多子工程的复杂系统工程，为了保证工程的有序开展，需要明确各子工程的前导工程，以协调各子工程的实施。该问题可以简化、抽象为有向图的拓扑序列问题。已知有向图G采用邻接矩阵存储，存储类型定义如下：

```
typedef struct {
    int numVertices, numEdges;           //图的类型定义
    char VerticesLists[MAXV];           //图的顶点数和有向边数
    int Edge[MAXV][MAXV];                //顶点表，MAXV为已定义常量
} MGraph;
//邻接矩阵
```

请设计算法，即int uniquely(MGraph G)，以判定G是否存在唯一的拓扑序列，若是则返回1，否则返回0。具体要求如下：

(1) 给出算法的基本设计思想；

(2) 根据设计思想，采用C或者C++语言描述算法，关键之处给出注释。

14. 已知关键字为整数类型，现要编写算法，以便于对n个记录进行整理，进而使所有关键字为负数的记录排在关键字为非负数的记录之前。对编写的算法具体要求为：算法采用顺序存储结构，其空间、时间复杂度分别为O(1)、O(n)。

(1) 给出算法的基本设计思想；

(2) 根据设计思想，采用C或者C++语言描述算法，关键之处给出注释。

```
void SortProcess(int a[], int n) //数组a中有n个元素
```

2023—2024第2学期
《数据结构》A卷参考答案及评分标准

一、算法阅读题(每小题5分, 共20分)

1. $O(\log_2 n)$ (5分)

2. $O(1)$ (5分)

3. $x->next=top;$ (3分) $top=x;$ (2分)

或 : $x->next=S;$ $S=x;$ 或 : $x->next=S->top;$ $S->top=x$

4. $\text{ChangeLR}(T->lchild);$ (3分)

$\text{ChangeLR}(T->rchild);$ (2分)

二、简答题(每小题5分, 共20分)

5. $(1+2+3+4+5+6)*2=42$ (3分)

存放 $A[5][3], S+(1+2+3+4+5+3)*2=S+36$ (2分)

6.59 (5分)

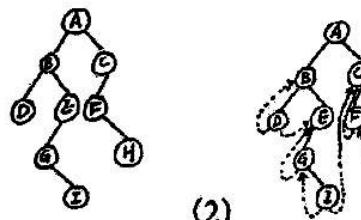
7. $H(H(T(H(T(H(T(L)))))))$ (5分)

8. (5分)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
56	32	69	41	88	47	84	30	66	22	42	54	60	27	86	19
88	66	86	56	42	60	84	30	41	22	32	54	47	27	69	19

三、应用题(每题10分, 共30分)

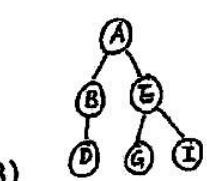
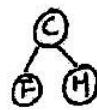
9.



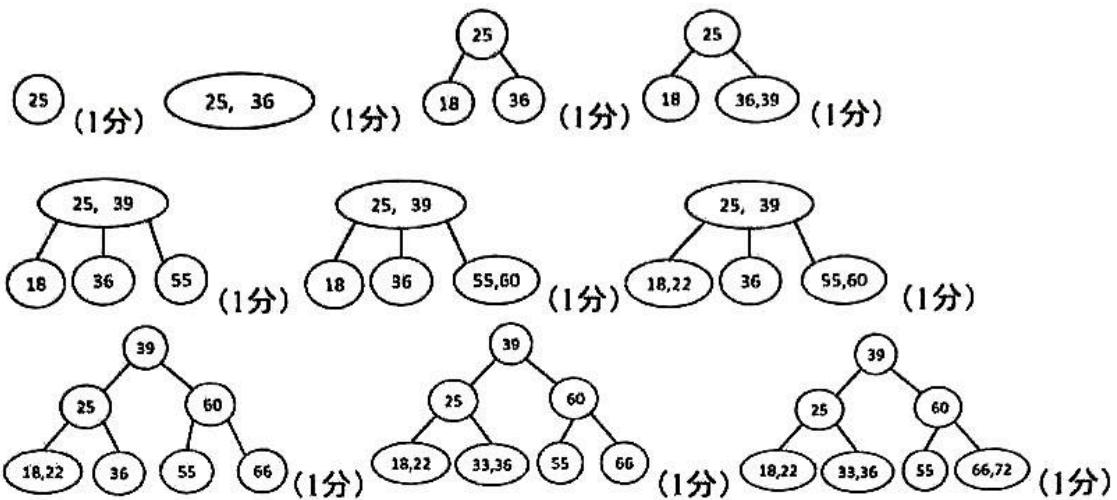
答案:(1)

(2)

(3)



10. 每正确添加一个关键字得1分, 共10分。



11.

(1)列出事件的最早发生时间。(6分);

事件	1	2	3	4	5	6	7
最早发生时间	0	15	30	25	39	43	51

(2)指出关键路径。(4分);

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

四、算法设计题(每小题10分, 共30分)

12.

(1)将字符串前一半入栈, 然后, 将栈中元素和后一半元素比较(思想正确均可得分)

(2) int IsPalindrome(char *t)

{

```

    InitStack(s);
    len=strlen(t);
    for (i=0;i<len/2;i++)           //前半部分入栈
        Push(S,t[i]);
    if (len%2!=0)
        i++;
    while (! EmptyStack(s)) //前半部分元素和后半部分元素比较
    {
        temp=Pop(S);
        if (temp!=t[i])
            return 0;          //不是回文
        else
            i++;
    }
    return 1;                  //是回文
}

```

13.

(1)

初始化:首先,我们需要一个数组来存储每个顶点的入度。我们遍历邻接矩阵,计算每个顶点的入度。

初始一个计数器作为变量用于记录有多少个入度为0的节点

遍历每个节点:首先计算每个顶点的入度,然后在每次迭代中找到并移除了入度为0的顶点。

如果在任何时刻存在多个入度为0的顶点,那么就返回0,表示存在多个有效的拓扑序列。如果成功完成了拓扑排序,那么就返回1,表示存在唯一的拓扑序列。

```
(2) int uniquely(MGraph G){  
    int inDegree[MAXV]={0};      //初始化入度数组  
    //计算每个顶点的入度  
    for(int i=0;i<G.numVertices;i++){  
        for(int j=0;j<G.numVertices;j++){  
            if(G.Edge[i][j]!=0)  
                inDegree[j]++;  
  
        //对每个顶点进行处理  
        for(int i=0;i<G.numVertices;i++){  
            int zeroInDegreeCount=0;      //入度为0的顶点数量  
            int zeroInDegreeIndex=-1;     //入度为0的顶点索引  
            //找到入度为0的顶点  
            for(int j=0;j<G.numVertices;j++){  
                if(inDegree[j]==0){  
                    zeroInDegreeCount++;  
                    zeroInDegreeIndex=j;  
  
                }  
            //如果存在多个入度为0的顶点或者没有找到入度为0的点,则返回0  
            if(zeroInDegreeCount!=1) return 0;  
            //将找到的入度为0的顶点从图中移除  
            inDegree[zeroInDegreeIndex]=-1;  
            for(int j=0;j<G.numVertices;j++){  
                if(G.Edge[zeroInDegreeIndex][j]==1){  
                    inDegree[j]--;  
  
            //如果成功完成了拓扑排序,则返回1  
            return 1;  
        }
```

14.

(1) 借助快速排序子表划分的算法思想对表中数据进行划分

```
(2) { low=0;high=n-1;  
      while(low<high){  
          while (low<high && a[low]<0)           //从左边找>0的
```

```
low++;
while (low<high && a[low]>0)      //从右边找<0的
    high--;
if (low<high)          //交换
{
    temp=a[low];
    a[low]=a[high];
    a[high]=temp;
    low++;
    high--;
}
}
```