

北京工业大学 2017 年硕士研究生招生考试试题

科目代码： 896

科目名称： 数据结构

★所有答案必须做在答题纸上，做在试题纸上无效

一、单项选择题（20 分，每题 2 分）

下列每个题目中有四个选项，其中只有一个是正确的。试根据题目的陈述，选择正确的答案。

1. 数据的存储结构是指（ ）。
 - A. 从问题空间中抽象出来的数学模型
 - B. 性质相同的数据元素的集合
 - C. 数据结构在计算机内存中的表示
 - D. 相互之间存在一种或多种特定关系的数据元素的集合
2. 假设某个循环队列采用数组 $Q[0...9]$ 表示，当前的队头指针 $front$ 和队尾指针 $rear$ 分别为 4 和 8。首先执行两次入队操作，然后再执行两次出队操作，队头指针 $front$ 和队尾指针 $rear$ 应该分别变为（ ）。
 - A. 5, 9
 - B. 6, 0
 - C. 0, 6
 - D. 6, 4
3. 指针 p 指向双向链表中某结点，在其后插入由指针 q 指向的新结点，下列方法正确的是（ ）。
 - A. $p \rightarrow next = q; q \rightarrow pre = p$
 - B. $q \rightarrow pre = p; q \rightarrow next = q \rightarrow pre \rightarrow next; q \rightarrow next \rightarrow pre = q; p \rightarrow next = q$
 - C. $q \rightarrow next = p \rightarrow next; q \rightarrow pre = p \rightarrow pre; p \rightarrow next = q; p \rightarrow next \rightarrow pre = q$
 - D. $p \rightarrow next \rightarrow pre = q; q \rightarrow pre = p; q \rightarrow next = p \rightarrow next; p \rightarrow next = q \rightarrow next$
4. 由 n 个具有权值的叶子结点构造的赫夫曼树包含的结点总数是（ ）。
 - A. $n-1$
 - B. $n+1$
 - C. $2n-1$
 - D. $2n+1$
5. 深度为 h (设根的层数为 1) 的完全二叉树至少包含的结点数是（ ）。
 - A. 2^h-1
 - B. 2^h
 - C. 2^{h-1}
 - D. $2^{h-1}-1$

北京工业大学 2017 年硕士研究生招生考试试题

科目代码: 896 科目名称: 数据结构

五、算法设计题(45 分, 每题 15 分)

1. (算法设计) 利用二叉树遍历的思想, 设计算法, 判断一棵二叉树是否为平衡二叉树, 如果是平衡二叉树, 则返回 true, 否则, 返回 false。二叉树采用二叉链表存储表示。

二叉链表表示法中结点的定义为:

```
typedef struct BiTNode{
    int data;
    struct BiTNode *lchild,*rchild; //左孩子, 右孩子
} BiTNode, *BTree;
```

算法的函数原型定义为: bool isBalance(BiTNode *t, int& h); // h 记录树的高度

2. (算法设计) 背包问题求解: 设有一个背包可以放入的物品的总体积为 T, 现有 n 件物品, 其各自的体积值存储于数组 W 中, 分别为 W[0], W[1], ..., W[n-1]。设计非递归算法, 求解是否存在若干件物品可以放入背包中, 使得放入的体积之和刚好为背包的总容量 T。如果有解, 请输出, 否则输出无解信息。提示: 可以利用栈作为辅助数据结构。

算法的函数原型定义为: void KNAP(int T, int W[], int n);

3. (数据结构设计) 设计一个计算机专业的课程选课系统, 要求符合下列要求: 有些课程属于基础课程, 独立于其他课程, 例如“高等数学”; 有些课程必须完成其它的先修课程才可以开始, 例如: “数据结构”课程的先修课程为“离散数学”和“程序设计基础”。如何得到符合要求的选课序列, 回答下列问题。

- (1) 通过分析, 确定所需要的数据结构。(文字描述即可)
- (2) 写出数据结构的抽象数据类型定义。

北京工业大学 2017 年硕士研究生招生考试试题

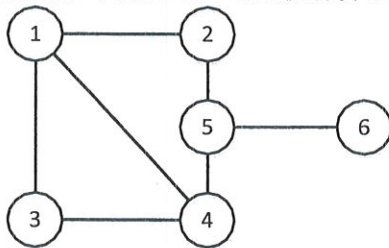
科目代码: 896 科目名称: 数据结构

```

        break;
    }
    w = NextAdjVex(G, v, w);
} //end while
foo = -1;
if(w == -1){
    Pop(stack, foo);
    G.mark[foo] = UNVISITED;
}
}
if (GetTop(stack, w) != ERROR) {
    if(w == j){
        a = -1;
        while(StackEmpty(stack) != TRUE){
            Pop(stack, path[++a]);
        }
        while(a >= 0){
            Push(stack, path[a]);
            printf("%d ", path[a]);
            a--;
        }
        printf("\n");
        foo = j;
        Pop(stack, w);
        G.mark[w] = UNVISITED;
    }
}
} //end while
}

```

(1) 给定如下的图 G, 写出执行算法 Algorithm(G, 1, 4)之后的输出结果。



(2) 说明该算法的功能。

(3) 该算法中, 栈 stack 的作用是什么?

北京工业大学 2017 年硕士研究生招生考试试题

科目代码: 896 科目名称: 数据结构

2. 根据字母集{ e, m, o, r, y }中每个字母在电文中出现的频度所构建的赫夫曼编码为:

e: 00
m: 01
o: 11
r: 101
y: 100

回答下列问题:

(1) 画出对应的赫夫曼树。(约定构造过程中, 左分支编码为‘0’, 右分支编码为‘1’)

(2) 当接收到的电文为 01000111101100 时, 根据赫夫曼编码给出对应的译文。

3. 已知一个有向图 G 的顶点集 V, 弧集 E 如下:

 $V = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8\};$ $E = \{ \langle V_1, V_2 \rangle, \langle V_1, V_3 \rangle, \langle V_2, V_6 \rangle, \langle V_3, V_7 \rangle, \langle V_4, V_3 \rangle, \langle V_4, V_5 \rangle, \langle V_5, V_8 \rangle, \langle V_6, V_7 \rangle, \langle V_8, V_7 \rangle \}$ 分别采用邻接矩阵表示和邻接表表示, 注意: 假定每个顶点邻接表中的结点是按顶点序号从大到小的次序链接的。回答下列问题:(1) 采用邻接矩阵存储表示, 写出从顶点 V_1 出发按深度优先搜索遍历得到的顶点序列和按广度优先搜索遍历得到的顶点序列。(2) 采用邻接表存储表示, 写出从顶点 V_1 出发按深度优先搜索遍历得到的顶点序列和按广度优先搜索遍历得到的顶点序列。

(3) 画出采用邻接矩阵存储表示时, 深度优先搜索 (DFS) 生成树 (或森林) 和广度优先搜索 (BFS) 生成树 (或森林)。

4. 对如下关键字构建哈希表, 地址空间为 0-16。

{Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec}

哈希函数为 $H(k) = \lfloor (\text{Ord}(\text{关键字 } k \text{ 的第一个字母}) - \text{Ord}('A') + 1) / 2 \rfloor$ 。Ord(x)表示 x 的 ASCII 码值, 如: Ord('A') 为字母'A'的 ASCII 码值。

回答下列问题。

(1) 用线性探测开放定址法处理冲突, 画出构造的哈希表。

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| | | | | | | | | | | | | | | | | |

北京工业大学 2017 年硕士研究生招生考试试题

科目代码： 896 科目名称： 数据结构

- (2) 写出查找关键字“Nov”时，在查找过程中依次进行比较的关键字序列。
 (3) 假定查找每个关键字的概率相等，计算该哈希表查找成功的平均查找长度 ASL。

5. 对于关键字序列 (28, 45, 32, 37, 29, 64, 39, 79, 62, 55)，按照下列格式，写出利用堆排序方法将其重新排列为非递减顺序的前 3 个堆的状态。

初始状态

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 28 | 45 | 32 | 37 | 29 | 64 | 39 | 79 | 62 | 55 |

第 1 个堆 (初始堆)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | | | | | | | | | |

第 2 个堆

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | | | | | | | | | |

第 3 个堆

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | | | | | | | | | |

注意：请将所有答案做在答题纸上，在答题纸上绘制表格填写，做在试题纸上无效！

四、算法阅读题 (15 分，每题 5 分)

阅读如下有关图的算法，回答问题。

```
#define VISITED 1    //标记顶点被访问过
#define UNVISITED 0 //标记顶点未被访问过
#define MAX 100    //顶点数量上限
```

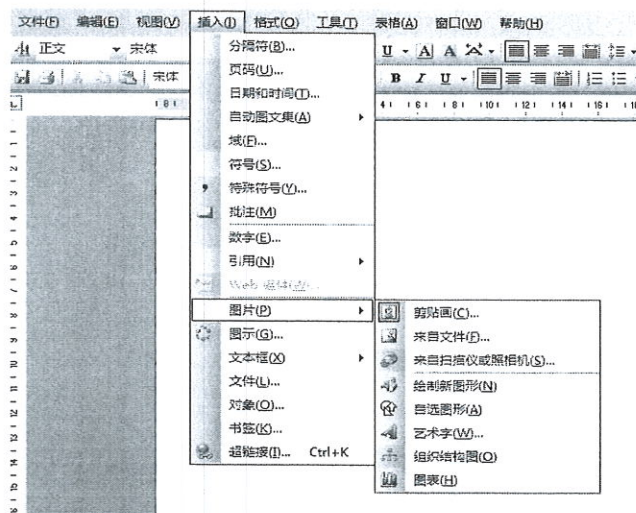
北京工业大学 2017 年硕士研究生招生考试试题

科目代码: 896 科目名称: 数据结构

6. 按层次次序将一棵包含 n 个结点的完全二叉树中所有结点从1到 n 编号, 则某个结点 i 没有左孩子的条件是_____⑥_____。
7. 由权值序列 (3, 8, 2, 6, 7) 构造一棵赫夫曼树, 其带权路径长度WPL值是_____⑦_____。
8. 图的简单路径是指_____⑧_____。
9. 如果在构造哈希表时采用链地址法解决冲突, 且哈希函数为 $H(\text{key}) = \text{key} \text{ MOD } 11$, 则需要建造的链表数目 (包括空链表) 是_____⑨_____。
10. 在10阶B树中, 根结点所包含的关键字个数的最大值和最小值分别是_____⑩_____。

三、解答题 (50 分, 每题 10 分)

1. 菜单栏可以采用广义表描述。下图中菜单栏



可以使用广义表描述为:

(文件 (...), 编辑 (...), 视图 (...), 插入 (分隔符, 页码, ..., 图片 (剪贴画, ...)), 格式 (...), ...)

回答下列问题:

- (1) 广义表的长度、深度分别描述了菜单栏中的什么信息?
- (2) 说明广义表中的子表和原子分别对应菜单栏中的哪类信息?
- (3) 为广义表设计存储结构, 写出相应的类型定义。

北京工业大学 2017 年硕士研究生招生考试试题

科目代码: 896 科目名称: 数据结构

```

typedef enum{ FALSE, TRUE, ERROR, OK }Status; //状态
typedef struct{
    int* base; //在栈构造之前和销毁之后, base 的值为 NULL
    int* top; //栈顶指针
}SqStack; //顺序栈定义
typedef struct{
    int vexs[MAX]; //顶点向量
    int arcs[MAX][MAX]; // 邻接矩阵, 其中值为 1 表示存在关联边, -1 表示没有关联边
    int vexnum, arcnum; // 顶点数量和弧的数量
    int mark[MAX]; //标记图中顶点是否被访问过, 初始值为 UNVISITED
}MGraph; //邻接矩阵定义

Status InitStack(SqStack &s); //构造一个空栈
Status Push(SqStack& s, int e); //栈中压入元素 e 为新的栈顶元素
Status GetTop(SqStack& s, int& e); //若栈不空, 则取栈顶元素并返回 OK; 否则返回 ERROR
Status Pop(SqStack& s, int& e); //若栈不空, 则栈顶元素出栈并返回 OK; 否则返回 ERROR
Status StackEmpty(SqStack s); //若栈为空栈, 则返回 TRUE; 否则返回 FALSE
int FirstAdjVex(MGraph& G, int v); //返回图 G 中顶点 v 的第一个邻接点; 若不存在则返回-1
int NextAdjVex(MGraph& G, int v, int w); //返回图 G 中顶点 v 的(相对于 w 的)下一个邻接点; 若 w
                                     是 v 的最后一个邻接点, 则返回-1

void Algorithm (MGraph& G, int i, int j)
{
    int v, w, foo, a;
    SqStack stack;
    int *path;
    path = (int*)malloc(sizeof(int)*G.vexnum);
    InitStack(stack);
    Push(stack,i);
    G.mark[i] = VISITED;
    foo = -1;
    while(StackEmpty(stack) != TRUE){
        w = -1;
        GetTop(stack, v);
        if((w=FirstAdjVex(G, v)) != -1){
            while(w != -1){
                if(G.mark[w] == UNVISITED && w > foo){
                    Push(stack, w);
                    G.mark[w] = VISITED;
                }
            }
        }
    }
}

```