



2024 年招收攻读硕士学位研究生入学考试试题 B 卷

招生专业与代码：网络空间安全 083900

考试科目名称及代码：数据结构 830

考生注意：所有答案必须写在答题纸（卷）上，写在本试题上一律不给分。

一、单项选择题（每题 2 分，共 20 分）

1. 下列叙述中正确的是（ ）

- A. 一个逻辑数据结构只能有一种存储结构
- B. 数据的逻辑结构属于线性结构，存储结构属于非线性结构
- C. 一个逻辑数据结构可以有多种存储结构，且各种存储结构影响数据处理的效率
- D. 一个逻辑数据结构可以有多种存储结构，且各种存储结构不影响数据处理的效率

2. 下列程序段的时间复杂度是（ ）

```
int func (int n){
    if (n == 1) return 1;
    else {
        int sum = 0;
        for (int i = 0; i < n; ++i) sum += i;
        return func (n / 2) + func (n / 2) + sum;
    }
}
```

- A. $O(n^{1/2})$
- B. $O(\log n)$
- C. $O(n)$
- D. $O(n \log n)$

3. 已知 p 结点是非空双向链表的中间结点（既有前驱又有后继），删除 p 结点的直接前驱结点的语句序列是（ ）

- A. `p->prior->prior->next=p; p->prior=p->prior->prior; free(p->prior);`
- B. `p->prior=p->prior->prior; p->prior->prior->next=p; free(p->prior);`
- C. `p->prior=p->prior->prior; p->prior->prior->next=p; q=p->prior; free(q);`
- D. 以上都不对

4. 对长度为 n 的线性表 L 执行插入操作，假设以均等概率插入表中任一位置，则插入一个元素时，平均需要移动表中元素的个数为（ ）

- A. n
- B. $n/2$
- C. $(n+1)/2$
- D. $(n-1)/2$

5. 设栈 S 和队列 Q 的初始状态为空，元素 a,b,c,d,e,f 依次通过栈 S，一个元素出栈后即进入队列 Q，若这 6 个元素出队列的顺序是 b,d,e,f,c,a，则栈 S 的容量至少应该是（ ）

- A. 3
- B. 4
- C. 5
- D. 无法确定

6. 设 n, m 为一棵树的两个结点，在中序遍历时，n 在 m 前的条件是（ ）

- A. n 在 m 右方
- B. n 是 m 祖先
- C. n 在 m 左方
- D. n 是 m 子孙

7. 在有向图的逆邻接表存储结构中，顶点 v 在表结点中出现的次数是（ ）

- A. 顶点 v 的度
- B. 顶点 v 的出度
- C. 顶点 v 的入度
- D. 以上都不对

8. 采用堆排序的方法对 n 个记录进行排序，其空间复杂度为（ ）

- A. $O(1)$
- B. $O(\log n)$
- C. $O(n)$
- D. $O(n \log n)$

9. 设 Hash 表长 $m=11$ ，Hash 函数 $H(\text{key})=\text{key} \bmod 11$ 。表中已有 4 个结点，地址为 $\text{addr}(15)=4$ ， $\text{addr}(38)=5$ ， $\text{addr}(61)=6$ ， $\text{addr}(84)=7$ ，其余地址为空，如用二次探测再散列处理冲突，关键字 49

的结点地址是 ()

- A. 3 B. 8 C. 9 D. 10

10. 若数据元素序列{9,10,11,5,6,7,21,2,3}是某排序方法第二趟排序后的结果,则该排序算法可能是 ()

- A. 冒泡排序 B. 直接插入排序 C. 二路归并排序 D. 简单选择排序

二、填空题 (每空 2 分, 共 20 分)

1. 用一维数组 B 存储一个 n 阶三角矩阵, 则一般来说 B 的长度至少为_____。
2. 广义表 A=(a,b,(c,d),(e, (f,g))), 则 Tail(Head(Tail(Tail(A))))=_____。
3. 一棵深度为 h (h>1) 的完全二叉树, 按照层次遍历 (从上往下, 从左到右) 次序给结点编号 (从 1 开始), 则叶子结点的编号最小值是 _____, 最大值为_____。
4. 若将 20 阶上三角矩阵 A 按列优先顺序压缩存放在一维数组 B 中, 假设 A 中第一个元素 a[0][0] 存在 B 数组的 b[0], 则应存在 b[100]的元素是 A 中的_____。
5. 设 G 为有向图, 包含 20 个顶点, 其中顶点 v_i 的出度为 i ($1 \leq i \leq 20$), 入度为 20-i, 则图 G 具有的弧的条数为_____。
6. 采用顺序查找法 (有检测哨) 查找长度为 n 的线性表时, 假设查找成功与不成功的可能性相同, 对每个记录的查找概率也相等, 则顺序查找的平均查找长度为_____。
7. 对于含有 n 个顶点 e 条边的无向连通图 G, 利用 Prim 算法生成 G 的最小生成树的时间复杂度为_____。
8. 对 n 个结点进行快速排序, 最大比较次数是_____。
9. 采用堆排序算法对关键字序列{6, 8, 7, 9, 0, 1, 3, 2, 4, 5}进行排序, 假设采用小顶堆, 则在构建初始堆时初始小顶堆序列是_____。

三. 判断题 (每题 2 分, 共 20 分, 正确的选 T, 错误的选 F)

1. 用带头结点的单链表存储队列, 其队头指针指向头结点, 队尾指针指向队尾结点, 在进行出队操作时可能会修改队尾指针。()
2. 一个稀疏矩阵采用三元组顺序表进行压缩存储后, 失去了随机存取的特性。()
3. 二叉树中叶子结点个数比非叶子节点个数多 1。()
4. 具有 n 个结点的完全二叉树的深度为 $\text{ceil}(\log_2 n)$, 其中 $\text{ceil}(x)$ 表示对 x 向上取整。()
5. 无向图的邻接矩阵中非零元个数等于图的边数。()
6. 用哈希表构造查找表, 并用线性探测再散列的方法处理冲突, 则记录个数 n 越大, 平均查找长度也越大。()
7. 利用克鲁斯卡尔(Kruskal)算法可以判定一个有向图是否有回路。()
8. 对于有向图 G, 广度优先搜索遍历 G 和深度优先搜索遍历 G 的时间复杂度相同。()
9. 基数排序中, 每一趟基于某个关键字的排序都必须使用稳定排序算法。()
10. 快速排序、简单选择排序和希尔排序都是不稳定的排序方法 ()

四、简答题（共 40 分）

1. 阅读下列算法，并回答问题。（本题共两小题，每小题各 3 分，共计 6 分）

```
void f (int R[], n)
{ int i=0, j=0, flag=1, x=0;
while (flag && i<n-1){
    flag=0;
    for (j=i; j<n; j++)
        if (R[j+1].key<R[j].key) {flag=1; x=R[j]; R[j]=R[j+1]; R[j+1]=x;}
    }
}
```

- (1) 描述该算法的功能。
 - (2) 解释 flag 的作用。
2. 已知一颗二叉树的前序遍历序列是 abcdefghij，中序遍历序列是 cbedahgijf，求解下列问题。（本题共两小题，每小题各 4 分，共计 8 分）
- (1) 画出此二叉树。
 - (2) 画出该二叉树的后序线索二叉树。
3. 依次输入关键字序列(16, 3, 7, 11, 9, 26, 18, 14, 15)，构造出对应的平衡二叉树，要求给出每一步的步骤。（本题共 8 分）

4. 如图 1 所示为一个 AOE 网，A 表示工程的开始，I 表示工程的结束，假设工程从时间 0 开始，求出所有事件和活动允许发生的最早及最晚事件，并据此求出所有的关键活动和关键路径。（本题共 10 分）

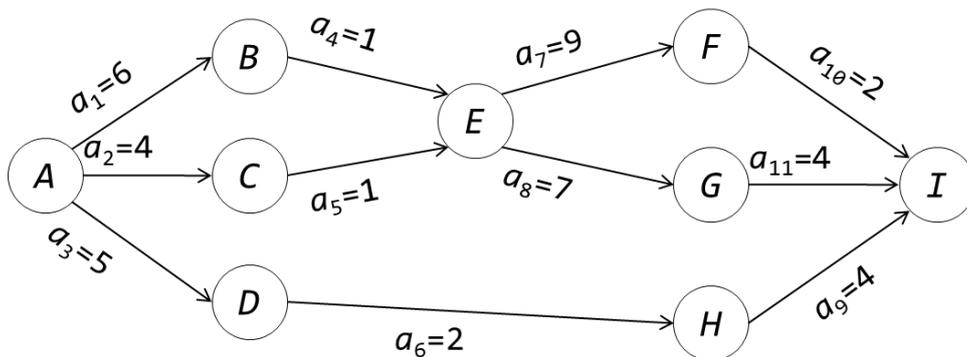


图 1

5. 给定图 2 所示带权有向图，利用 Floyd 算法，画出该图的邻接矩阵并求每一对顶点之间的最短路径长度，并给出顶点 $V_0 \rightarrow V_2$, $V_1 \rightarrow V_0$, $V_3 \rightarrow V_0$, $V_3 \rightarrow V_1$ 的最短路径，要求写出求解过程。（本题共 8 分）

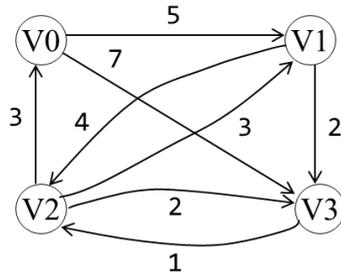


图 2

五、算法填空（共 2 小题，每空 2 分，共 20 分）

1. 下面算法是用 Prim 算法从第 u 个顶点出发构造带权连通图 G 的最小生成树 T ，输出 T 的各条边。其中

```

typedef struct{
    VertexType adjvex;
    VRType lowcost;
}closedge[MAX_VERTEX_NUM];
typedef struct ArcCell{
    VRType adj;
    InfoType *info; //该弧相关信息的指针
}ArcCell, AdjMatrix[MAX_VERTEX_NUM][MAX_VERTEX_NUM];
typedef struct{
    VertexType vexs[MAX_VERTEX_NUM]; //顶点向量
    AdjMatrix arcs; //邻接矩阵
    int vexnum, arcnum; //图的当前顶点数和弧数
    GraphKind kind; //图的种类标志
}MGraph;

void MiniSpanTree_PRIM(MGraph G, VertexType u){
    k=LocateVex(G,u);
    for(j=0;j<G.vexnum;++j)
    { if(j!=k)
        { _____ ① _____;
          _____ ② _____; }
    }
    closedge[k].lowcost=0;
    for(i=1;i<G.vexnum;++i){
    { k=minimum(closedge);
      printf(closedge[k].adjvex,G.vexs[k]);
      _____ ③ _____;
    }
  
```

```

for(j=0;j<G.vexnum;++j){
    if(G.arcs[k][j].adj<closedge[j].lowcost)
    {
        _____④_____
        _____⑤_____
    }
}
}
}

```

2. 以下是借助快速排序的思想，在 n 个元素的序列中，求出第 k 个最小的元素的算法，完成算法的空格部分。

```

typedef struct {KeyType key;} ElemType;
typedef struct {
    ElemType *r; // 数组指针，索引 0 的元素为哨兵
    int length; // 数组长度（排除哨兵元素）
} SqList;
int findMinK(SqList *L, int k) { // findMinK 函数调用 find 函数
return find (L, 1, L->length, k);
}
int find(SqList *L, int low, int high, int k) { //返回第 k 个最小元素索引
    int i = low, j = high;
    KeyType pivotkey;
    L->r[0] = L->r[low];
    pivotkey = L->r[low].key;
    while (i < j) {
        while (_____⑥_____)
            j--;
        L->r[i] = L->r[j];
        while (_____⑦_____)
            i++;
        L->r[j] = L->r[i];
    }
    L->r[i] = L->r[0];
    int left = _____⑧_____;
    if (left == k)
        return i;
    else if (left > k)
        return find(_____⑨_____);
    else
        return find(_____⑩_____);
}

```

六、编写算法（共 3 小题，每小题 10 分，共 30 分）

1. 给定一个整数数组 `asteroids`，表示在同一行的小行星。对于数组中的每一个元素（`int` 类型），其绝对值表示小行星的大小，正负表示小行星的移动方向（正表示向右移动，负表示向左移动）。每一颗小行星以相同的速度移动。设计算法，找出碰撞后剩下的所有小行星。碰撞规则：两个小行星相互碰撞，较小的小行星会爆炸。如果两颗小行星大小相同，则两颗小行星都会爆炸。两颗移动方向相同的小行星，永远不会发生碰撞。例如 `asteroids = [5,10,-5]` 时，输出为 `[5,10]`，因为 10 和 -5 碰撞后只剩下 10。5 和 10 永远不会发生碰撞。`asteroids = [8,-8]` 时，输出为 `[]`。`asteroids = [10,2,-5]` 时，输出为 `[10]`，因为 2 和 -5 发生碰撞后剩下 -5，10 和 -5 发生碰撞后剩下 10。要求算法的时间复杂度为 $O(n)$ ，其中 n 为数组 `asteroids` 的大小。

2. 已知二叉树 `T` 的节点定义如下，给出找到二叉树中数据值为 `x` 的节点的所有祖先节点的算法，要求使用非递归算法进行编写。

```
typedef struct BiTNode { // 结点结构
    TElemType    data; // 结点数据
    struct BiTNode *lchild, *rchild; // 左右孩子指针
} BiTNode, *BiTree;
```

3. 已知一个单向链表 `L` 的首元节点为 `head`，链表节点的结构包含一个数据域 `data` 和一个指向其后继节点的指针 `next`。假设 `L` 可能存在环，即从某个节点开始通过连续跟踪 `next` 指针可以再次到达该节点。编写一个算法，返回 `L` 开始入环的第一个节点，如果 `L` 无环，则返回 `NULL`，要求算法的时间复杂度为 $O(n)$ ，空间复杂度为 $O(1)$ 。例如图示链表，算法运行结束时应该返回数据域为 2 的节点。

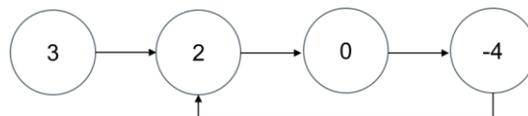


图 3