江苏大学
硕士研究生入学考试样题

A 卷

科目代码：851
科目名称：数据结构　　满分：150 分

注意：①认真阅读答题纸上的注意事项；②所有答案必须写在答题纸上，写在本试题纸或草稿纸上均无效；③本试题纸须随答题纸一起装入试题袋中交回！

一、单项选择题(每小题 1 分，共 10 分，下列每小题给出的四个选项中，只有一项符合题目要求)

1. 计算机算法必具备输入、输出和(　　)等五个特性。
(A)有效性、可移植性和可扩充性　　(B)有效性、确定性和有穷性
(C)确定性、稳定性和有穷性　　(D)易读性、稳定性和安全性

2. 链表不具有的特点是(　　)。
(A)插入、删除不需要移动元素　　(B)可随机访问任一元素
(C)不必事先估计存储空间　　(D)所需空间与线性长度成正比

3. 设 head 是带头结点的单循环链表的头指针，结点的指针域为 next，数据域为 data，则指针 p 是指向链表尾结点的条件是(　　)。
(A) p->next == head　　(B) p->next == head->next
(C) p == head　　(D) p == head->next

4. 针对下述结论，正确答案是(　　)。
①两个栈共享一片连续内存空间时，为提高内存利用率，减少溢出机会，应把两个栈的栈底分别设在这片内存空间的两端
②队列逻辑上是一个下端和上端既能增加又能减少的线性表
(A) 只有①正确　(B) 只有②正确　(C) ①②都正确　(D) ①②都不正确

5. 已知广义表 LS=((a，b)，c，d，(e，f))，运用 head 和 tail 函数取出 LS 中原子 e 的运算是(　　)。
(A) head(tail(LS))　　(B) tail(tail(head(LS)))
(C) head(tail(head(tail(LS)))　　(D) head(head(tail(tail(tail (LS)))))

6. 在下述结论中，正确的是(　　)。
①只有一个结点的二叉树的度为 0
②二叉树的度为 2
③二叉树的左右孩子可任意调换
④具有 n 个结点的二叉链表具有 n+1 个空指针域
(A) ④　(B) ②③　(C) ②④　(D) ①④

7. 在二叉树结点的先序序列、中序序列和后序序列中，所有叶子结点的先后顺序(　　)。
(A)都不相同　　(B)完全相同
(C)先序和中序相同，而与后序不同　　(D)中序和后序相同，而与先序不同

8. 对具有 n 个顶点、e 条边的无向网用邻接矩阵为存储结构时，求最小生成树的 Prim 算法的时间复杂度为(　　)。

(A) O(n)　　(B) O(n+e)　　(C) $O(n^2)$　　(D) O(eloge)

9. 当采用折半查找法查找一个数据时，要求数据存储结构(　　)

(A) 一定采用顺序存储结构　　(B) 一定采用链式存储结构

(C) 一定采用三元组存储结构　　(D) 既可采用链式又可采用顺序存储结构

10. 下列排序算法中，在关键字基本无序的情况下，经第一趟排序完毕后，其最大或最小关键字的元素一定在其最终位置上的算法是(　　)。

(A) 直接插入排序　　(B) 归并排序　　(C) 直接选择排序　　(D) 快速排序

二、填空题(每小题 2 分，共 10 分)

1. 一个数据元素可由若干个_____组成。
2. 对于一个具有 n 个结点的单链表，删除已知地址为 p 的结点后的一个结点（假设存在）的时间复杂度为_____，删除给定值为 x 的结点（假设存在）的时间复杂度为_____。
3. 分别采用堆排序、快速排序、直接插入排序和归并排序，对初态为有序的表，则最省时间的是_____算法。
4. 字符串的子串是指该字符串中_____。
5. 二维数组 W 中，每个元素 W[i][j]的长度为 4 个字节，行下标 i 从 0 到 7，列下标 j 从 0 到 3，若按行顺序存放二维数组 W，其起始地址为 100，则二维数组元素 W[6][3]的起始地址为_____。

三、应用题(共 80 分)

1. (8 分)已知森林的先根遍历的序列为 EADCBFHGI，中根遍历的序列为 ABCDEFGHI，要求画出该森林以及对应的二叉树。
2. （7 分）某二叉树的数组表示法如图 1 所示，其中空白表示此处不存在结点，根结点为 A。要求：

下标	0	1	2	3	4	5	6	7	8	9	10	11	12	13
	A	C	B	D			E							F

图 1

(1) 画出该二叉树的图形表示。

(2) 画出该二叉树的后序后继线索二叉链表。

3. (6 分)以数据集{5，10，7，6，15}为叶结点的权值，构造一棵哈夫曼树，要求结点的左孩子的权值小于右孩子的权值，请写出构造过程。并计算该哈夫曼树的带权路径长度 WPL。
4. （12 分）设 G=(V, N)为带权有向图，V 是顶点集合，N 是弧的集合。顶点集合 V={A，B，C，D，E}，各顶点编号依次为 1，2，3，4，5。其对应的邻接矩阵如图 2 所示（见下页），要求：

(1) 画出该带权有向图。

(2) 利用 dijkstra 算法，求从源点 A 出发到其它各终点的最短路径以及长度，请给出逐步求解最短路径以及长度的过程。

	1	2	3	4	5
1	0	90	60	10	30
2	∞	0	10	∞	∞
3	∞	15	0	∞	80
4	∞	∞	45	0	15
5	∞	40	10	∞	0

图 2

5. （8 分）已知无向图邻接表如图 3 所示，现要求基于该无向图的邻接表从顶点 A 出发对无向图进行广度优先搜索（BFS），请写出 BFS 顶点序列，并画出 BFS 生成树。

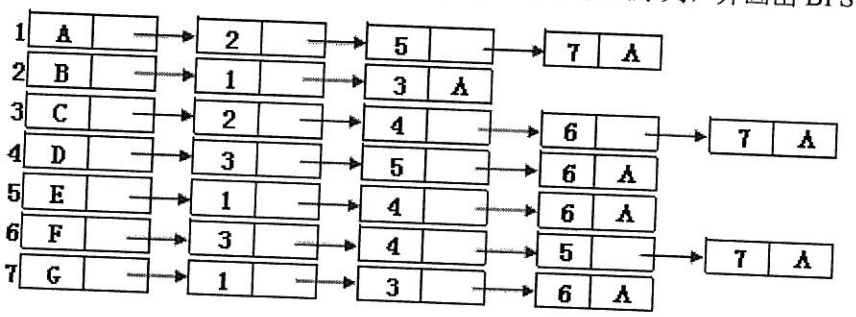

图 3

6. (7 分)已知中缀表达式为#A-B/C#，其中#表示表达式开始符和结束符。现利用栈将该中缀表达式转换成后缀表达式。初始时将开始符#预先放到栈中，然后对 A-B/C#进行处理，按表 1 所示的格式写出各步骤的结果。

表 1　中缀表达式转换成后缀表达式

步骤	符号序列	栈	输出结果
初始	无	#	无
1			
2			
……			

7. （12 分）已知关键字序列 F={31，15，21，26，18，48，37，20}，散列函数为除留余数法 H(Key)= Key % 11， 即关键字 Key 对 11 取模，散列地址空间为 0～10。要求：

(1) 如果采用二次探测再散列法解决冲突，探测的增量序列为 $d_i=+1^2, -1^2, +2^2, -2^2, +3^2, -3^2$，……，请画出相应的散列表。

(2) 计算出装填因子。

(3) 求出等概率下查找成功时的平均查找长度 ASL。

8. (12 分)已知关键字序列 F={28，07，39，10，65，14，61，17，50，21}，要求最后排序结果是按关键字从小到大的次序排列，请写出对关键字序列 F 进行快速排序的第一趟排序过程（以关键字 28 为基准）。

9. (8 分)已知关键字序列 F={44，20，26，36，22，34，28，18，32，46，24}，要求最后排序结果是按关键字从小到大的次序排列，若采用链式基数排序方法排序，请写出第一次“分配”和“回收”的结果。

四、简答题(共 30 分)

1. （12 分）在线性表的单链表存储结构中，涉及到头指针、头结点、首元结点这样几个概念，试论述这些概念之间的区别与联系。单链表存储结构中加上头结点有什么作用？
2. （8 分）若对大小均为 n 的有序的顺序表和无序的顺序表分别进行顺序查找，试在下面两种情况下，分别讨论在等概率时，有序的顺序表和无序的顺序表的平均查找长度是否相同，各为多少？请给出分析。
   (1) 查找成功，且表中只有一个关键字等于给定值 K 的记录。
   (2) 查找不成功，即表中没有关键字等于给定值 K 的记录。
3. （5 分）请问对具有 10 个不同键值的关键字序列构造大顶堆时，最多会发生多少次关键字的比较，请给出分析过程。
4. （5 分）一棵深度为 H 的满 k 叉树有如下性质：H 层上的结点都是叶子结点，其余各层上的每个结点都有 k 棵非空子树。现设一棵满 k 叉树上的叶子结点数为 n_0 ，非叶子结点数为 n_1，则 n_0 和 n_1 之间应满足什么关系？请给出推导过程。

五、算法设计题(共 20 分)

1. (8 分)假设有一个非空的线性表，表中各个元素值都不相同，现以顺序表为存储结构存储，且从顺序表的数组下标为 0 的单元开始存储线性表中的元素。试编写算法，从顺序表中删除具有最小值的元素，并返回被删除元素在线性表中的序号。
2. (7 分)限定二叉排序树的各个结点的值都不相同，二叉排序树采用二叉链表形式存储。试编写算法将指针 s 所指的结点插入到以指针 t 所指的结点为根的二叉排序树中，使得以指针 t 所指的结点为根的二叉树仍然是二叉排序树。
3. (5 分)限定有向图中不能有相同的弧。试编写算法在以邻接表为存储结构的带权有向图中插入一条顶点序号为 v1 到顶点序号为 v2 的弧，弧的权值是 w。如果插入成功，则返回 1，否则返回 0（设图的当前顶点数为 VerNum，顶点序号 v 的范围是 0≤v≤VerNum-1）。

注：

(1) 可采用类 C 语言或 C 语言或 C++语言描述你的算法，关键之处请给出简要注释。
(2) 算法中可使用下面给出的存储结构。
(3) 若算法中使用了其他的存储结构和运算，请给出其定义和实现。

//顺序表存储结构描述如下：

```
const   int   maxlen = 100 ;          //maxlen 是顺序表中最大项数
template <class Type>   class   seqlist { //顺序表类
private :
        Type data[maxlen] ;        //抽象类型 Type 定义的数组
        int   len;                          //当前数据元素个数
public:
    seqlist(void);                          //构造函数，初始化空表
    ~seqlist(void);                         //析构函数
};
```

```
//二叉排序树的二叉链表存储结构描述如下：
template <class Type> class BSTree;   //二叉排序树前视声明，以便使用友元
template <class Type> class BSTreeNode { //结点类
friend class BSTree<Type>;
private:
   BSTreeNode<Type> *leftChild,   *rightChild;      //结点的左、右孩子指针域
   Type data;              //结点的数据域
public:
   BSTreeNode ( ) : leftChild (NULL), rightChild (NULL) { } //构造函数，构造一个空结点
   BSTreeNode (Type d, BSTreeNode<Type> *lp = NULL, BSTreeNode<Type> *rp
         =NULL ) : data (d),   leftChild (lp), rightChild (rp) { }    //构造函数，构造一个数
         据域的值为 d 的结点
};
template <class Type> class BSTree { //二叉排序树类
private:
   BSTreeNode <Type> *root;    //二叉排序树根结点指针
   void   destroy(BSTreeNode<Type> *p);    //删除以 p 为根的二叉排序树
public:
   BSTree ( ) : root (NULL) { }    //构造函数
   ~BSTree ( ) { destroy ( root ); }      //析构函数
};

//带权有向图的邻接表存储结构描述如下：
const int MaxVertexes = 20;   //最大的顶点数
template <class vertexType, class arcType> class Graph;
template < class arcType> struct ArcNode {// 定义弧结点
     friend class Graph <class vertexType, class arcType>;
     int adjvex;   //和弧相关联的另一个顶点序号
     arcType weight; //弧上的信息(权)
     ArcNode<arcType> *nextarc ; //指向下一条弧结点的指针
     ArcNode( ) { }   //构造函数
     ArcNode( int v , arcType w ) : adjvex( v ) , weight( w ) , next( NULL ){ }   //构造函数
};
template < class arcType , class vertexType > struct VertexNode {
// 定义顶点结点
     friend class Graph <class vertexType, class arcType>;
     vertexType data; //顶点的信息
     ArcNode<arcType> *firstarc ; //指向依附该顶点的弧链表
};
```

```
template <class vertexType, class arcType> Graph{//定义图
private:
    VertexNode < arcType , vertexType > * VertexesTable;//顶点表
    int VerNum;  当前的顶点数
    int ArcNum; //当前的弧数
public:
    Graph:VerNum (0),ArcNum(0){};//构造函数
    ~Graph ( );//析构函数
};
```