

北京工业大学 2022—2023 学年第 1 学期

《数据结构》 考试试卷 A 卷

考试说明： 考试时间：95 分钟；考试形式：闭卷

承诺：

本人已学习了《北京工业大学考场规则》和《北京工业大学学生违纪处分条例》，承诺在考试过程中自觉遵守有关规定，服从监考教师管理，诚信考试，做到不违纪、不作弊、不替考。若有违反，愿接受相应的处分。

承诺人：_____ 学号：_____ 班号：_____

注：本试卷共 五 大题，共 14 页，满分 100 分，考试时必须使用卷后附加的统一答题纸或草稿纸。请将答案统一写在答题纸上，答案写在其他位置造成的成绩缺失由考生自己负责。

卷面成绩汇总表（阅卷教师填写）

题号	一	二	三	四	五	总成绩
满分	10	20	20	20	30	
得分						

得分

一、判断题（每题 1 分，共 10 分）

请在答题纸对应题号下方空格处为正确说法打√，错误说法打×。

- 线性表每个元素都有唯一的前驱和唯一的后继。（ ）
- 线性表的特点是逻辑上相邻的数据元素，其物理位置也是相邻的。（ ）
- 链表的存储空间是动态分配的，其存储密度高于顺序表。（ ）
- 栈和队列都是操作受限的线性表，而串是收据元素类型受限的线性表。（ ）
- 树的每个结点有且仅有一个直接前驱，但所有结点可有多个直接后继。（ ）
- 一棵非空的二叉树的先序遍历序列与后序遍历序列正好相反，则该二叉树的叶子结点个数为 1。（ ）
- 一个连通图的生成树是其极大连通子图， n 个顶点 $n-1$ 条边的连通子图。（ ）
- 中序遍历二叉排序树将得到递增有序的序列。（ ）

9. 有 $n-1$ 条边的图不一定是生成树。()

10. 冒泡排序是不稳定的排序。()

得分

二、单选题（每题 2 分，共 20 分）

请在答题纸对应题号下方空格处填入 A、B、C、D 中的一个选项。

1. 第一本系统阐述数据的逻辑结构、存储结构及其操作的著作《The Art of Computer Programming》的作者是 ()。
A. Prim B. Dijkstra C. Knuth D. Krusgal
2. 循环队列采用顺序存储方式，当利用队列中最后一个存储单元标志满时，判断队列空的条件是 ()。
A. $\text{front} == \text{rear}$ B. $\text{front} == \text{rear} + 1$ C. $\text{rear} == \text{front} + 1$ D. $\text{front} == (\text{rear} + 1) \% \text{maxsize}$
3. 在双向循环链表中，在 p 指针所指的结点后插入 q 所指向的新结点，其修改指针的操作是 ()。
A. $p \rightarrow \text{next} = q; q \rightarrow \text{prior} = p; p \rightarrow \text{next} \rightarrow \text{prior} = q; q \rightarrow \text{next} = q;$
B. $q \rightarrow \text{next} = p \rightarrow \text{next}; q \rightarrow \text{prior} = p; p \rightarrow \text{next} \rightarrow \text{prior} = q; p \rightarrow \text{next} = q;$
C. $p \rightarrow \text{next} = q; p \rightarrow \text{next} \rightarrow \text{prior} = q; q \rightarrow \text{prior} = p; q \rightarrow \text{next} = p \rightarrow \text{next};$
D. $q \rightarrow \text{next} = p \rightarrow \text{next}; q \rightarrow \text{prior} = p; p \rightarrow \text{next} = q; p \rightarrow \text{next} \rightarrow \text{prior} = q;$
4. 若一个栈以向量 $V[0..n-1]$ 存储，初始栈顶指针 top 设为 n ，则元素 x 进栈的正确操作是 ()。
A. $V[--\text{top}] = x;$ B. $V[\text{top}--] = x;$
C. $V[++\text{top}] = x;$ D. $V[\text{top}++] = x;$
5. 若子串“Dreams”在主串“Dreams Do Come True.”中的位置为 1，则子串“Come”的位置是 ()。
A. 2 B. 3 C. 10 D. 11
6. 设有数组 $A[i, j]$ ，数组的每个元素长度为 3 字节， i 的值为 1 到 8， j 的值为 1 到 10，数组从内存首地址 BA 开始顺序存放，当用以行为主存放时，元素 $A[5, 8]$ 的存储首地址为 ()。
A. $BA + 141$ B. $BA + 180$ C. $BA + 222$ D. $BA + 225$

7. G 是一个非连通无向图, 共有 21 条边, 则该图至少有 () 个顶点。
A. 7 B. 8 C. 9 D. 10
8. 对 26 个记录的有序表作折半查找, 当查找失败时, 至少需要比较 () 次关键字。
A. 3 B. 4 C. 5 D. 6
9. 对于分块查找时, 下列说法错误的是 ()。
A. 由于块内是无序, 则块内只能采用顺序查找
B. 索引表是按关键字有序排列的, 则索引表的查找既可采用顺序查找, 也可采用二分查找
C. 索引表只可采用二分查找
D. 设总记录数为 n , 分块大小为 s , 则当 $s^2=n$ 时, 平均查找长度 ASL 最小
10. 元素 16, 10, 12, 14 按顺序进栈, 则该栈不可能的出栈序列为 () (入栈出栈可交替进行)。
A. 16, 12, 10, 14
B. 14, 12, 10, 16
C. 12, 14, 10, 16
D. 10, 14, 10, 12

得分

三、填空题 (每题 2 分, 共 20 分)

请在答题纸对应题号后的空格处顺序填写答案。

- 抽象数据类型形式化为一个三元组 (D, S, P) , 其中 D 为_____, S 为 D 上的_____, P 为 D 上的操作集。
- 长度为 n 的线性表在等概率查找情况下的平均查找长度 ASL 为_____; 查找不成功情况下的平均查找长度 ASL 为_____。
- 对于双向链表任一结点指针 p 满足 $p \rightarrow \text{next} = \text{p} \rightarrow \text{prior}$ 。
- 两个栈共享空间时栈满的条件是_____。
- 设无向图 G 中有 n 个顶点 e 条边, 则其对应的邻接表中的表头结点和表结点的个数分别为_____和_____。
- 广义表操作 $\text{tail}[\text{head}[\text{tail}[(a, b), (c, d)]]$ 的结果是_____。

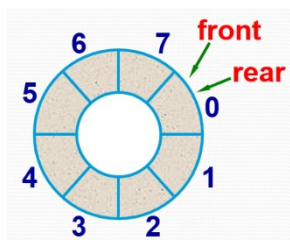
7. 数据的 4 种基本逻辑结构为_____、线性结构、树结构、_____。
8. 算法的 5 个重要特性是：_____、_____、可行、输入和输出。
9. 直接选择排序的平均时间复杂度是_____。
10. 对 n 个不同的排序码进行冒泡排序，在元素无序的情况下比较的次数最多为_____。

得分

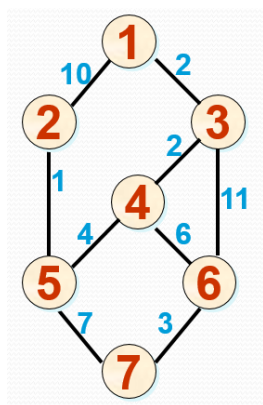
四、操作题（每题 5 分，共 20 分）

请在答题纸对应题号下方标明具体题号、顺序作答。

1. 已知循环队列的初始状态如图，请分别画出 A 进队、BC 进队、A 退队、B 退队的状态图。



2. 已知字符 A、B、C、D、E、F、G 的权值分别为 3、12、7、4、2、8、11，请画出据此构造的一棵哈夫曼树，并给出 FACE 的哈夫曼编码。
3. 假设一棵二叉树的先根序列为 EBADCFHGIKJ 和中序序列 ABCDEFGHIJK，请画出该树。
4. 用 Dijkstra 算法给出下图中从 1 出发到各点的最短路径。



得分

五、算法题（每题 6 分，共 30 分）

请在答题纸对应题号下方标明具体题号、顺序作答。

1. 补全链式队列退队算法。

```

typedef struct node
{
    ElemType data;
    struct node *next;
} QNode;
typedef struct
{
    QNode *rear, *front;
} LiQueue;
LiQueue *q;

bool DeQueue (LiQueue *q, elemtype &e)
{
    if (q->rear==NULL) return false;
    QNode *t = q->front;
    e = t->data;
    _____①_____;
    if ( _____②_____ )
        _____③_____;
    free (t);
    return 1;
}

```

2. 补全算法，删除递增有序链表中值大于
- `mink`
- 且小于
- `maxk`
- 的所有元素。

```

void delete(LinkList &L, int mink, int maxk)
{
    p=L->next;
    while( _____①_____ )

```

```

{pre=p; p=p->next; }
if (p)
{
    while (p && p->data<maxk)
        p=p->next;
    _____②_____;
    _____③_____;
    while(q!=p)
        {s=q->next; delete q; q=s;}
}
}

```

3. 补全中序遍历二叉树的非递归算法。

```

typedef struct node
{
    ElemType data;
    struct node * lchild, * rchild;
} BTreeNode;

inorder(BTreeNode *t)
{
    BTreeNode *p;
    init(S);    //栈初始化
    p=t;
    while(_____①_____)
    {
        while(p!=NULL)
        {
            _____②_____;
            _____③_____;
        }
        if(empty(S)) return;
        _____④_____;
    }
}

```

```
        visit(p);
        p=p->rchild;
    }
}
```

4. 补全二叉排序树的递归查找算法。

```
typedef struct node
```

```
{
    KeyType key;
    InfoType data;
    struct node *lchild, *rchild;
} BSTNode;
```

```
BSTNode* BinSearch(BSTNode *t, keytype k)
```

```
{
    p=t;
    while( p!=NULL )
    {
        if (p->key==k)
            return(p);
        if (k<p->key)
        {
            _____①_____;
            _____②_____;
        }
        else
        {
            _____③_____;
            _____④_____;
        }
    }
    return(NULL);
}
```

5. 补全快速排序划分算法。

```
int partition(rectype R[], int low, int high)
{
    i=low; j=high; temp=R[low];
    while(i<j)
    {
        while(①)
            j--;
        if(i<j )
            {R[i]=R[j]; i++;}
        while(i<j && R[i].key<temp.key)
            i++;
        if(i<j )
            {②}
    }
    ③;
    return i;
}
```


答 题 纸

姓名: _____ 学号: _____

得 分

一、判断题（每题 1 分，共 10 分）

请在题号下方空格处为正确说法打√，错误说法打×。

1	2	3	4	5	6	7	8	9	10

得 分

二、单选题（每题 2 分，共 20 分）

请在题号下方空格处填入 A、B、C、D 中的一个选项。

1	2	3	4	5	6	7	8	9	10

得 分

三、填空题（每题 2 分，共 20 分）

请在题号后的空格处顺序填写答案（用顿号分隔）。

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____

姓名：_____ 学号：_____

得 分

四、操作题（每题 5 分，共 20 分）

请标明具体题号、按要求顺序作答。

姓名： _____ 学号： _____

姓名：_____ 学号：_____

得 分

五、算法题（每题 6 分，共 30 分）

请标明具体题号、按要求顺序作答。

姓名： _____ 学号： _____

草 稿 纸

姓名: _____

学号: _____