

北京邮电大学
2022 年硕士研究生招生考试试题
考试科目：809 数据结构



北京邮电大学

2022 年硕士研究生招生考试试题

考试科目：809 数据结构

请考生注意：①所有答案(包括选择题和填空题)一律写在答题纸上，否则不计成绩。

②不允许使用计算器。

一. 填空型 (每空 1 分, 共 32 分)

1. 数据结构中评估算法性能的两个重要指标是[]和[]。
2. 程序段：{ $i=1$; while ($i \leq n$) $i=i*2$; }的时间复杂度为[]。
3. 在具有 n 个元素的顺序表中删除一个元素，平均需要移动[]个元素。
4. 设有 5 个元素的进栈序列是 a、b、c、d、e，其输出序列是 c、e、d、b、a，则该栈的容量至少是[]。
5. 设 C++ 中存储三维数组 $A[M][N][P]$ ，则第一个元素为 $a[0][0][0]$ ，若按行优先存储，则 $a[i][j][k]$ 前面共有[]个元素；若按列优先存储，则 $a[i][j][k]$ 前面共有[]个元素。
6. 将数组 $data[0, \dots, n-1]$ 作为循环队列的存储空间，front 为队头指针，rear 为队尾指针，则队列满的条件为[]，队列空的条件为[]，队列长度为[]。
7. 朴素模式匹配算法中，每个串的起始下标均为 1，变量 $i=100$ ， $j=10$ ，分别表示主串和模式串当前比较的字符元素下标，若本次比较两字符不同，则 i 回溯为[]， j 回溯为[]。
8. 一棵度为 4 的树，若度为 1 的结点有 5 个，度为 2 的结点有 8 个，度为 3 的结点有 12 个，度为 4 的节点为 2 个，则该树有[]叶子结点。
9. 含有 200 个结点的完全二叉树，度为 0,1,2 的结点数量分别为[]、[]、[]，树的高度为[]。
10. 已知二叉树的前序遍历序列是 AEFBGCDHIKJ，中序遍历序列是 EFAGBCHKIJD，则该二叉数的后序遍历序列是[]。
11. n 个结点的二叉树若用二叉链表存储，则所有结点中的空指针共有[]个。
12. 对于一个具有 n 个顶点和 e 条边的有向图和无向图，在其对应的邻接表存储结构中，所含的边结点分别有[]个和[]个。

13. 10 个顶点的有向图，最少有[]边，则一定是强连通图。
14. 20 个结点构成的二叉排序树，在等概率查找的条件下，查找成功的平均查找长度最大值是[]；查找不成功的平均查找长度最大是[]。(保留小数点后 2 位)
15. 理想情况下，散列表的平均比较次数是[]。
16. 二叉排序树中，最大值结点的[]为空；最小值结点的[]为空。
17. 长度为 13 的有序序列进行等概率查找，采用顺序查找，则查找成功的平均查找长度是[]；如果采用二分查找，则查找成功的平均查找长度是[]。(保留小数点后 2 位)
18. 按照[]遍历二叉排序树得到的序列是有序序列。
19. 基于关键码比较的排序算法中，[]时间复杂度是 $O(n^2)$ 却是不稳定的排序算法；最好情况下，该算法的移动次数是[]次。

二. 单选题（每空 1 分，共 26 分）

1. 以下说法正确的是[]。
- A. 数据元素是数据的最小单位；
 - B. 数据项是数据的基本单位；
 - C. 数据结构是带有结构的各数据项的集合；
 - D. 数据结构通常包含逻辑结构、存储结构和对数据的操作这三个方面的内容。
2. 在具有 n 个结点的双向链表的某个结点前插入一个结点的处理，时间复杂度是[]。
- A. $O(n)$
 - B. $O(1)$
 - C. $O(\log_2 n)$
 - D. $O(n^2)$
3. 某线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用[]存储方式最节省运算时间。
- A. 单链表
 - B. 仅有头指针的单循环链表
 - C. 顺序表
 - D. 仅有尾指针的单循环链表
4. 如果使用比较高效的算法判断单链表是否有环，则该算法中至少需要几个指针？[]
- A. 1
 - B. 2
 - C. 3
 - D. 4
5. 在双向循环链表中，在 p 指针所指的结点后插入一个指针 q 指向的新结点，修改指针的操作是[]。
- A. $p \rightarrow next = q$; $q \rightarrow prior = p$; $p \rightarrow next \rightarrow prior = q$; $q \rightarrow next = q$;
 - B. $p \rightarrow next = q$; $p \rightarrow next \rightarrow prior = q$; $q \rightarrow prior = p$; $q \rightarrow next = p \rightarrow next$;
 - C. $q \rightarrow prior = p$; $q \rightarrow next = p \rightarrow next$; $p \rightarrow next \rightarrow prior = q$; $p \rightarrow next = q$;
 - D. $q \rightarrow next = p \rightarrow next$; $q \rightarrow prior = p$; $p \rightarrow next = q$; $q \rightarrow next = p$;

6. 在一个含有 n 个元素的单链表中查找某一个值，若查找成功，则需平均比较 [] 个结点。

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

7. 一个栈的输入序列为：a、b、c、d、e，则栈的不可能输出的序列是 []。

- A. a、b、c、d、e B. d、e、c、b、a
C. d、c、e、a、b D. e、d、c、b、a

8. 为了解决计算机主机和键盘输入之间速度不匹配问题，通常设置一个键盘缓冲区，该缓冲区应该是一个 [] 结构。

- A. 线性表 B. 栈 C. 队列 D. 数组

9. 用三元组表表示的稀疏矩阵进行转置时，设一维数组 $\text{pos}[\text{col}]$ 表示转置后各行第一个非零元素在三元组表中的位置，若 $\text{pos}[\text{col}]$ 的各元素值为：

	0	1	2	3
	0	1	3	4

则对应转置前的稀疏矩阵是 []。

- A. $\begin{bmatrix} 0 & -8 & 0 & 6 \\ 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ B. $\begin{bmatrix} 0 & -8 & 0 & 6 \\ 7 & 0 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{bmatrix}$ C. $\begin{bmatrix} 0 & -8 & 0 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ D. $\begin{bmatrix} 0 & -8 & 0 & 6 \\ 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 3 & 0 & 0 \end{bmatrix}$

10. 将两个各有 n 个元素的有序表归并成一个有序表，其最少的比较次数是 []。

- A. n B. $2n-1$ C. $2n$ D. $n-1$

11. 按照二叉树的定义，具有 3 个结点的二叉树有 [] 种不同的形态。

- A. 3 B. 4 C. 5 D. 6

12. 设 a、b 为一棵二叉树上的两个结点，在中序遍历时，a 在 b 前面的条件是 []。

- A. a 在 b 的右方 B. a 在 b 的左方
C. a 是 b 的祖先 D. a 是 b 的子孙

13. 前序遍历和中序遍历结果相同的二叉树一定是 []。

- A. 一般二叉树 B. 只有根结点的二叉树
C. 所有结点只有左子树的二叉树 D. 所有结点只有右子树的二叉树

14. 若由森林转化得到的二叉树是高度为 4 的满二叉树，则该森林是由 [] 棵树构成。

- A. 4 B. 5 C. 3 D. 1

15. 设二叉树中结点包含的两个指针域分别为 lchild 和 rchild，则判断指针变量 p 所指向的结点为叶子结点的条件是 []。

- A. $!p$ B. $p \rightarrow \text{lchild} == p \rightarrow \text{rchild}$

C. !p->lchild&&!p->rchild D. !p->lchild||!p->rchild

16. 如果某图的邻接矩阵是对角线元素均为零的上三角矩阵, 则此图是[]。

- A. 有向完全图 B. 连通图 C. 强连通图 D. 有向无环图

17. 设无向图 $G=(V, E)$ 和 $G'=(V', E')$, 如果 G' 是 G 的生成树, 则下列说法中错误的是[]。

- A. G' 是 G 的连通分量
B. G' 是 G 的一个无环子图
C. G' 是 G 的子图
D. G' 是 G 的极小连通子图且 $V=V'$

18. 如果从无向图的任一顶点出发进行一次深度优先搜索即可访问所有顶点, 则该图一定是[]。

- A. 完全图 B. 连通图 C. 有回路 D. 一棵树

19. 在查找算法中, 查找成功的平均查找长度与结点个数无关的查找方式是[]。

- A. 顺序查找 B. 折半查找 C. 树表查找 D. 散列查找

20. 使用散列函数将元素的关键码值映射为散列地址时, 经常会产生冲突。此时, 冲突是指[]。

- A. 两个元素具有相同的序号
B. 两个元素的具有不同的关键码, 但散列地址相同
C. 装填因子过大, 数据元素过多
D. 两个元素关键码不同, 但非关键码相同

21. 除留余数法的基本思想是: 将元素散列到 $0..m-1$ 的空间中, 元素的关键码为 key , 散列函数记为 $h(key) = key \% p$, 为了减少发生冲突的可能性, 一般 p 取[]。

- A. m B. 小于或等于 m 的最大合数
C. 大于 m 的最小素数 D. 小于或等于 m 的最大素数

22. 使用链地址法处理冲突引起“堆积”的原因是[]。

- A. 同义词之间发生冲突
B. 非同义词之间发生冲突
C. 同义词之间或非同义词之间发生冲突
D. 散列表长度不足

23. 平均时间复杂度为 $O(n \log n)$ 的稳定的排序方法[]。

- A. 希尔排序 B. 归并排序 C. 快速排序 D. 堆排序

24. 下列排序算法中, []算法可能会出现下面的情况, 在最后一趟开始之前, 所有的元素都不在其最终的位置上。

- A. 堆排序 B. 起泡排序 C. 直接插入排序 D. 快速排序

25. 对一组数据 {2, 12, 57, 16, 88, 5} 进行排序, [] 排序的前三趟排序结果如下:

第一趟: 2, 12, 57, 16, 88, 5

第二趟: 2, 5, 57, 16, 88, 12

第三趟: 2, 5, 12, 16, 88, 57

A. 直接插入排序 B. 希尔排序 C. 归并排序 D. 简单选择排序

26. 下列算法中, [] 算法在初始数据基本有序时, 花费的时间反而更多。

A. 堆排序 B. 快速排序 C. 直接插入排序 D. 归并排序

三. 简答题(共 50 分)

1. (4 分) 阅读下面算法, 请回答下列问题:

(1) 说明语句 S1 的功能; (1 分)

(2) 说明语句组 S2 的功能; (1 分)

(3) 设链表表示的线性表为 (a₁, a₂, ..., a_n), 写出算法执行后的返回值所表示的线性表。(2 分)

```
Node* mynote(Node* first)
{
    // first 是不带头结点的单链表的头指针
    if(first && first->next)
    {
        Node* q= first;
        first = first->next;
        Node* p= first;
        S1: while(p->next) p=p->next;
        S2: p->next=q; q->next=NULL;
    }
    return first;
}
```

2. (5 分) 一棵二叉树的先序、中序和后序序列分别如下, 其中有一部分未显示出来。

(1) 试求出空格处的内容, 每空只能填写一个字母; (3 分)

先序序列: B A F E ;

中序序列: A D C E F;

后序序列: D G C。

(2) 画出该二叉树; (1 分)

(3) 将该二叉树转化成森林。(1 分)

3. (4 分) 假设二叉树采用二叉链表作为存储结构, 根据注释, 完成下面的算法, 求中序遍历中的第 k 个元素的值 ($1 \leq k \leq$ 二叉树结点总数)。

```
struct BiNode {
    int data; //结点存储的数据
    BiNode * lch; //左孩子地址
    BiNode * rch; //右孩子地址
}
```

```

};
int i = 0;
void InOrder(BiNode* R, int k)
{
    if (_____ && i < k)
    {
        _____; //递归遍历左子树
        i++;
        if(_____)
            cout<<R->data<<endl;
        _____; //递归遍历右子树
    }
}

```

4. (10 分) 已知图的邻接矩阵如下图所示,

(1) 按照给定顶点位置, 画出其对应的无向图 G; (2 分),

(2) 写出从顶点 1 开始深度优先搜索和广度优先搜索的序列; (2 分)

(3) 画出该图的最小生成树, 并给出最小生成树的权值; (3 分)

(4) 根据 Dijkstra 算法, 按顺序写出图从顶点 1 出发的所有最短路径顶点序列和最短路径长度。(3 分)

$\begin{pmatrix} 0 & 16 & 1 & 15 & \infty & \infty \\ 16 & 0 & 15 & \infty & 3 & \infty \\ 1 & 15 & 0 & 15 & 16 & 4 \\ 15 & \infty & 15 & 0 & \infty & 2 \\ \infty & 3 & 16 & \infty & 0 & 7 \\ \infty & \infty & 4 & 2 & 7 & 0 \end{pmatrix}$	①	②	③
	④	⑤	⑥

图 G

5. (6 分) 已知某电文中出现了 9 种不同的字母, 每个字母出现的频度分别为 A:8, B:4, C:3, D:6, E:7, F:23, G:11, H:2, I:36, 现在对这段电文用二进制进行编码 (即码字由 0,1 组成), 问

(1) 用什么方法可以得到最短总长度的电文编码; (1 分)?

(2) 电文最短总长度是多少位, 并给出具体计算过程 (包括画图)。(5 分)

6. (6 分) 已知关键码序列是 {1, 2, 4, 6, 5, 3}, 请按顺序构造一棵二叉平衡树, 画出每一步的最终二叉平衡树的结果。

7. (7 分) 对给定表 (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec), 设计一个装填因子为 2/3 的散列表 (地址从 0 开始)。这里, 取散列函数为 $H(x)=i/17$, 其中 i 为键值中第一个字母在英语字母表中的序号。计算散列表的长度, 并画出以线性探测法处理的散列表, 计算查找给定表中键值的 ASL。(注 A 的序号为 1)

(1) 散列表长度为: _____; (1 分)

(2) 散列表存储映像 (4 分):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

(3) 计算查找成功的平均查找长度 ASL (2 分)

8. (8 分) 已知序列 { 12, 13, 11, 70, 25, 15, 7, 18, 70, 60 }

(1) 使用最少的调整次数将其调整成为大根堆, 请写出调整后的序列; (2 分)

(2) 写出按大根堆进行堆排序的第一趟排序序列; (1 分)

(3) 写出增量为 3 的第一趟希尔排序序列; (1 分)

(4) 写出第一趟二路归并排序序列; (1 分)

(5) 写出第一趟简单选择排序序列; (1 分)

(6) 写出第一趟快速排序序列(以第一个元素为基准); (1 分)

(7) 写出第一趟起泡排序序列。(1 分)

四. 程序题(共 42 分)

1. (5 分) 线性表 $L = (a_1, a_2, \dots, a_i, \dots, a_n)$, 以单链表存储, 头指针为 first, 指针 p 指向链表第 i 个结点, $i=1, 2, \dots, n-1$; 函数 delete 实现以 $O(1)$ 时间复杂度删除元素 a_i , 并返回 a_i 的值; 请在下面的空格内填入适当语句, 完成算法。

```
struct node{
    int data;
    node* next;
};
int delete(node* first, node*p)
{
    int ai = _____ ;
    node* q = _____;
    _____ = q->data;
    p->next = _____;
    delete q;
    return _____ ;
}
```

2. (7 分) 已知数组 $\text{int } A[] = \{2, 5, 4, 1, 2, 7, 4, 1, 4\}$, 长度为 n, 按照计数排序的思想进行排序, 编程回答下面的问题。前提: 数组 A、B、C 已定义并赋值, 其中 B 和 C 初始化为 0, 各个数组的空间充足, K 为数组元素的最大值。

```
void Sort(int A[], int B[], int C[], int n, int K)
{
    程序片段①
    程序片段②
    程序片段③
}
```

(1) 完成程序片段①, 统计 A 中各个数字的出现次数, 结果保存在数组 $B[] = \{0, 2, 2, 0, 3, 1, 0, 1\}$; (2 分)

(2) 完成程序片段②, 更新 B 数组, 使得 $B = \{0, 2, 4, 4, 7, 8, 8, 9\}$; (2 分)

(3) 根据数组 B, 完成程序片段③, 计算 A 排序后的结果, 保存在数组

C[]={1, 1, 2, 2, 4, 4, 4, 5, 7 }。(3 分)

3. (15 分) 在系统中, 我们经常会遇到这样的需求: 将大量 (比如几十万、甚至上百万) 的对象中取出最大的前 K 个作为排行榜的数据, 这即是一个 TopK 算法。假设数据类型是整数, 数据长度是 n, 设计一个基于直接插入排序的算法, 能够在有限空间内, 充分考虑时间复杂度的情况下, 计算 TopK (K 远小于 n) 个数据。注意: 数组 a 存放原始数据, b 存放结果数据, 空间充足。

(1) 请用自然语言描述, 写出算法思想并计算时间复杂度; (5 分)

(2) 参考下面的 TopK 函数, 编程实现该算法 (可以使用多个函数组合实现, 不能使用 STL 等第三方提供的排序函数)。(10 分)

```
void TopK(int a[], int b[], int n, int K)
{
    //程序片段
}
```

4. (15 分) 已知三叉链表存储的二叉树, 根结点指针为 root, 树中有两个结点的指针分别为 PA 和 PB, 试编写程序打印从 PA 到 PB 的最短路径中所有结点的值。

(1) 请用自然语言描述, 写出算法思想并计算时间复杂度; (设二叉树共有 n 个结点) (5 分)

(2) 参考下面的 Path 函数, 编程实现该算法。(10 分)

```
struct BiNode{
    int data;
    BiNode* lch; //左孩子指针
    BiNode* rch; //右孩子指针
    BiNode* parent; //双亲指针
};
void Path(BiNode *root, BiNode *PA, BiNode* PB)
{
    //程序片段
}
```