

北京邮电大学 20——20 学年第 学期

《 数据结构与算法 》 期末考试试题（A 卷）

| | | | | | | | | | | | |
|----------------|--|----|----|----|------|---|---|-------|---|---|----|
| 考试 注意 事项 | 一、学生参加考试须带学生证或学院证明，未带者不准进入考场。学生必须按照监考教师指定座位就坐。 二、书本、参考资料、书包等物品一律放到考场指定位置。 三、学生不得另行携带、使用稿纸，要遵守《北京邮电大学考场规则》，有考场违纪或作弊行为者，按相应规定严肃处理。 四、学生必须将答题内容做在试题答卷上，做在试题及草稿纸上一律无效 | | | | | | | | | | |
| 考试 课程 | 数据结构与算法 | | | | 考试时间 | | | 年 月 日 | | | |
| 题号 | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 八 | 九 | 十 | 总分 |
| 满分 | 23 | 20 | 12 | 30 | 15 | | | | | | |
| 得分 | | | | | | | | | | | |
| 阅卷 教师 | | | | | | | | | | | |

一 填空题（每空 1 分，共 23 分）

- 数据的逻辑结构中，_____适合表示一对一的关系，_____适合表示一对多的层次关系，_____适合对标多对多的关系。
- 代码 `for (int i=0;i<n;i*=3);` 的时间复杂度为：_____。
- 设 n 阶对称矩阵按行优先方式存储下三角元素，元素 a_{00} 存储在 `sa[0]` 元素中，元素 a_{ij} 存储在 `sa[100]` 元素中，则下标 i 的值为____， j 的值为_____。
- 设一棵完全二叉树的顺序存储结构中存储数据元素为 `abcdefg`，则该二叉树的前序遍历序列为_____，中序遍历序列为_____，后序遍历序列为_____。
- 含有 100 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_____。
- 含有 100 个结点的二叉树，若采用二叉链表存储，则各个结点中所

- 有的空指针域共有_____个。采用三叉链表存储，则各个结点中所有的空指针域共有_____个。
7. 由权值分别为 1,26,5,9,12,1 的叶子结点生成一棵哈夫曼树，它的带权路径长度为：_____。
 8. 关键字序列为：{30, 25, 80, 63, 52, 48}，哈希函数为 $H(k)=k\%7$ ，哈希表的长度为 7，若用线性探测处理冲突，等概率情况下查找成功的平均查找长度为（保留两位小数）_____，若用链地址法处理冲突，等概率情况下查找成功的平均查找长度为_____。
 9. 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_____。
 10. 若一个栈用数组 `int data[n]` 存储，若初始栈顶指针 `top` 为 `n`，则经过若干次出入栈操作后，元素 `x` 进入栈的操作使用一条 C/C++ 语句是_____。
 11. 循环队列使用数组 `int data[n]` 存储，设 `front` 队首指针指向队首元素的前一位置，`rear` 队尾指针指向队尾元素，则队满条件是_____（不允许负数求余），队空的条件是_____。
 12. 若待排序序列为 `r[1...n]`，采用堆排序，则根结点编号为 1，最后一个结点编号为 `n`；请问建堆过程中应该从第_____个结点开始筛选。

二 单选题（每空 1 分，共 20 分）

| | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|
| 题号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | 8 | 9 |
| 答案 | | | | | | | | | | |
| 题号 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 答案 | | | | | | | | | | |

1. 算法分析的两个主要方面是[]。
 - A. 空间复杂度和时间复杂度
 - B. 正确性和简单性
 - C. 可读性和文档性
 - D. 数据复杂性和程序复杂性
2. 已知带头节点的单链表，`p` 节点既不是首节点（第一个节点），也不是尾节点，`p` 节点直接后继节点从链表中去掉的语句序列是[]
 - A. `P = P -> next;`

- B、P -> next = P;
- C、P -> next = P -> next -> next;
- D、P=P -> next -> next;

3. 如果使用比较高效的算法判断单链表有没有环的算法中，至少需要几个指针？ []

- A 1 B 2 C 3 D 4

4. 在双向链表存储结构中，将 p 所指的结点从链表中去掉，需要修改指针 []。

- A p->next->prior=p->prior; p->prior->next=p->next
- B p->next=p->next->next; p->next->prior=p
- C p->prior->next=p; p->prior=p->prior->prior
- D p->prior=p->next->next; p->next=p->prior->prior

5. 不属于链表特点的是 []

- A 适用频繁插入 B 适用于频繁删除
- C 存取速度快 D 方便扩充

6. 若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算,则利用 [] 存储方式最节省时间。

- A 顺序表 B 双链表
- C 带头结点的双循环链表 D 单循环链表

7. 前序遍历和中序遍历结果相同的二叉树为 []; 前序遍历和后序遍历结果相同的二叉树为 []。

- A. 一般二叉树
- B. 只有根结点的二叉树
- C. 根结点无左孩子的二叉树
- D. 根结点无右孩子的二叉树
- E. 所有结点只有左子树的二叉树
- F. 所有结点只有右子树的二叉树

8. 分别用以下序列构造二叉排序树，与其他三个序列构造结果不同的是[]

A 100 80 90 60 120 110 130

B 100 120 110 130 80 60 90

C 100 60 80 90 120 110 130

D 100 80 60 90 120 130 110

9. 对于关键字序列(14,26,38,54,91)，按序列次序创建一颗平衡二叉排序树，在等概率情况下查找成功时，其平均查找长度是[]。

A. 7/5

B. 9/5

C. 11/5

D. 13/5

10. 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 $V=V'$

11. 设有 5 个结点的无向图，该图至少应有[]条边才能确保是一个连通图。

A. 5

B. 6

C. 7

D. 8

12. 折半查找有序表 (3,4,5,13,16,32,81,727,2021)，为了找到数字 727，被比较的数字顺序为 ()。

A. 16,81,727

B. 16,727

C. 16,32,81,727

D. 16,2021,727

13 对于关键字序列(49,38,65,97,76,13,27,49)，完成创建的大根堆是 []。

A.(97,76,65,49,49,13,27,38)

B.(97,76,65,49,49,38,27,13)

C.(97,65,76,49,49,13,27,38)

D.(13,27,38,49,49,65,76,97)

14 对关键字序列 (30, 26, 18, 16, 5, 66)，进行 2 遍[]排序后得

到序列 (5, 16, 18, 26, 30, 66)。

- A.插入 B.归并 C.冒泡 D.选择

15 在下列排序算法中, []排序算法可能出现如下情况: 在最后一趟排序之前, 所有元素均不在其最终的位置上。

- A.快速 B.插入 C.堆 D.冒泡

16 在下列排序方法中, []排序方法的平均时间复杂度不是 $O(n^2)$ 。

- A.直接插入 B.冒泡 C.快速 D.直接选择

17. 若从二叉树的任一结点出发到根的路径的结点序列都按关键字有序, 则该二叉树是[]

- A.二叉排序树 B. 哈夫曼树 C. 堆 D.完全二叉树

18、将递归算法转换成对应的非递归算法时, 通常需要使用[]来保存中间结果。

- A. 队列 B. 栈 C. 链表 D. 树

19、一个待散列的线性表为 $k=\{18,25,63,50,42,32,9\}$, 散列函数为 $H(k)=k \text{ MOD } 9$, 与 18 发生冲突的元素有[]个。

- A. 1 B. 2 C. 3 D. 4

三 简答题 (12 分)

1. (4 分) 设先序遍历某棵树的结点序列为 SACEFBDGHIJK, 后序遍历该树的结点序列为 CFEABHGKJDS, 要求画出这棵树。

2、(4 分) 现有一个稀疏矩阵，按行优先存储，起始行列号均为 0，请给出它的三元组表。

$$\begin{bmatrix} 0 & 3 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & -2 & 0 \end{bmatrix}$$

3. (4 分) 已知一棵度为 m 的树中有 N_1 个度为 1 的结点， N_2 个度为 2 的结点， \dots ， N_m 个度为 m 的结点。试问该树中有多少个叶子结点？

四. 综合题 (30 分)

1、(8 分) 设哈希表表长 m 为 13，哈希函数为 $H(k)=k \text{ MOD } m$ ，给定的关键字序列为 {19, 14, 23, 10, 68, 20, 84, 27, 55, 11}。试写出用线性探测法解决冲突时所构造出的哈希表，并求出在等概率的情况下查找成功的平均查找长度 ASL。

(1) Hash 表 (5 分)

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|
| | | | | | | | | | | | | |

(2) 查找成功的平均查找长度 ASL (3 分)

2. (10 分) 已知某电文中出现了 10 种不同的字母, 每个字母出现的频率分别为 A:8, B:5, C:3, D:7, E:7, F:23, G:9, H:11, I:2, J:35, 现在对这段电文用二进制进行编码 (即码字由 0, 1 组成), 问电文编码总长度至少有多少位 (5 分)? 请画出相应的图 (5 分)。

3、(12 分) 已知图 G 的邻接矩阵如下所示：

(1) 求从顶点 1 (顶点序号从 1 开始) 出发的深度优先遍历和广度优先遍历序列；(4 分)

(2) 根据 prim 算法，求图 G 从顶点 1 出发的最小生成树，要求表示出其每一步生成过程。(用图或者表的方式均可)。(4 分)

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

$$\begin{pmatrix} 0 & 8 & 1 & 6 & \infty & \infty \\ 8 & 0 & 6 & \infty & 3 & \infty \\ 1 & 6 & 0 & 7 & 6 & 4 \\ 6 & \infty & 7 & 0 & \infty & 2 \\ \infty & 3 & 6 & \infty & 0 & 6 \\ \infty & \infty & 4 & 2 & 6 & 0 \end{pmatrix}$$

五. 编程题 (15 分)

1. 具有 n 个结点的完全二叉树, 已经顺序存储在一维数组 $a[1\cdots n]$ 中, 算法 `create` 将 a 中顺序存储变为以二叉链表存储的完全二叉树。请在下面的空格内填入适当语句, 完成算法。(5 分)

```
struct Node
{
    char data;
    Node *lch, *rch;
}
void create(Node *&r, int i, char a[], int n)
{
    if (i > n) _____;
    else
    {
        r = new Node;
        r->data = a[i];
        Create(_____, _____, a, n);
        Create(_____, _____, a, n);
    }
}
```

2. 大根堆筛选算法用 C++ 语言描述如下, 建堆元素存储在数组 $r[s\cdots m]$ 内, 请在下面的空格内填入适当语句, 完成算法。(5 分)

```
void Sift ( int r[], int s, int m )
{
    r[0] = r[s];    //暂存 r[s]
    for (int j = 2*s; _____; j *= 2 )
    {
        if ( j < m && r[j] < r[j+1])
            _____;    //令 j 指示关键字较大元素的位置
        if ( r[0] < r[j])
        {
            _____;
            s = j;    //元素上移, 继续向下调整
        }
        else _____;    //找到 r[0] 的插入位置 s , 停止向下调整
    }
    _____;    //将调整前的堆顶元素插入到 s 位置
}
```

3. 线性表 $L = (a_1, a_2, \dots, a_n)$ ，以单链表存储，头指针 `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 ，并将 a_i 返回调用函数；请在下面的空格内填入适当语句，完成算法。

(5 分)

```
struct node{
    int data;
    node* next;
}
int delete(node* first, node*p)
{
    int ai = _____;    //暂存被删结点
    node* q = _____;
    _____ = q->data;    //拷贝 p 的直接后继结点的 data
    p->next = _____;    // 摘链
    delete q;
    return ____;
}
```