

15.00元

南京邮电大学 2016 / 2017 学年第一学期

《数据结构 B》期末试卷 () 答案

本试卷共 4 页; 考试时间 110 分钟;

专业 _____ 班级 _____ 学号 _____ 姓名 _____

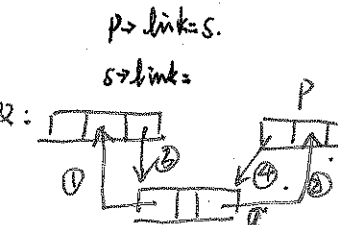
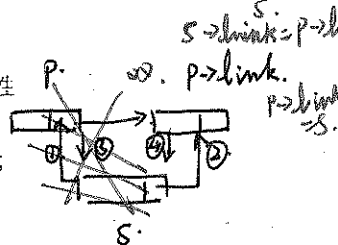


填空题 (20 分, 共 10 题)

1. 数据结构主要研究数据的 逻辑 结构, 数据的存储结构以及在数据上执行的运算。
2. 设顺序表 A 长度为 100, 若下标从 0 开始计, 则删除元素 A[10] 需要移动 89 个元素。
 $A[100-11]=89$
3. 一棵二叉树中, 若叶结点的个数为 2011, 则度为 2 的结点个数为 2010。
 $2010/10 = 201$
4. 无向图的连通分量是其 有向图 连通子图。
5. 线性表采用二分搜索必须满足两个条件: 线性表关键字必须是 有序表, 存储结构必须采用顺序存储结构。
6. 二叉搜索树的 中序 遍历序列是一个按关键字递增排列的有序序列。
7. 设有一组记录的关键字为 {19, 14, 1, 69, 20, 27, 55, 79}, 散列函数为 $h(key) = key \% 11$, 散列函数值为 3 的有 2 个。
 $19 \% 11 = 8, 14 \% 11 = 3, 69 \% 11 = 3$
8. 快速排序算法平均情况下的渐近时间复杂度为 $O(n \log n)$ 。
 $27 \% 11 = 5, 55 \% 11 = 0$
9. 采用二次探查法解决冲突可能产生 二次 聚集。
 $79 \% 11 = 2$
10. 图常见的两种存储结构有邻接矩阵和 邻接表。

二、选择题 (20 分, 共 10 题)

1. 一个算法必须在执行有限步之后结束。这是算法的 A。
A. 有穷性 B. 正确性 C. 确定性 D. 可行性
2. 在指针 p 所指示的结点之后插入新结点 s 的操作是 B。
A. $s \rightarrow link = p; p \rightarrow link = s;$
B. $s \rightarrow link = p \rightarrow link; p \rightarrow link = s;$
C. $s \rightarrow link = p; p \rightarrow link = s;$
D. $p \rightarrow link = s; s \rightarrow link = p;$
3. 栈和队列的共同点是 C。
A. 都是先进后出 B. 都是先进先出
C. 只允许在端点处插入和删除元素 D. 没有共同点
4. 后缀表达式: $532 * 3 + 3 / +$ 的值为 D。
A. 18 B. 7 C. 9 D. 8



532.
53. +3
5913
7-5=8.

$q \rightarrow llink = p \rightarrow llink.$
 $q \rightarrow rlink = p. 1$
 $p \rightarrow llink \Rightarrow rlink = q.$
 $p \rightarrow llink = q$

5. 高度为 5 的二叉树至多有 C 个结点。
A. 5 B. 10 C. 31 D. 32
6. 采用对半查找方法查找长度为 n 的线性表时, 时间复杂度为 B。
A. $O(n^2)$ B. $O(n \log_2 n)$ C. $O(n)$ D. $O(\log_2 n)$
7. n 个顶点的无向图采用邻接矩阵表示, 则该矩阵的大小是 B。
A. n B. $(n-1)^2$ C. n^2 D. $n-1$
8. 一个无向连通图的生成树是一个 B 连通子图。
A. 极大 B. 极小 C. 有时极大 D. 有时候极小
9. 下列排序方法中, 排序过程中的比较次数与排序方法无关的是 C。
A. 简单选择排序法 B. 直接插入排序法
C. 快速排序法 D. 冒泡排序法
10. 散列表的长度为 11, 下标范围是 $[0, 10]$, 散列函数为 $h(key) = key \% 11$ 。采用线性探查法解决冲突, 依次将关键字 7, 38, 5, 16 插入空的散列表中。则关键字 16 在散列表中存放的下标是 D。
A. 5 B. 6 C. 7 D. 8

$$7 \% 11 = 7$$

$$38 \% 11 = 5$$

$$5 \% 11 = 5$$

$$(5+1) \% 11 = 6$$

$$16 \% 11 = 5$$

$$8 \% 11 = 8$$

$$9 \% 11 = 8$$

三、简答题 (30 分, 共 5 题)

1. 有二叉树如图 1 所示, 写出该二叉树的先序遍历序列和中序遍历序列。

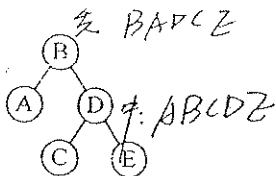


图 1

	0	1	2	3	4
0	21	0	0	0	0
1	0	0	11	0	0
2	0	1	0	3	8
3	0	0	0	2	0

图 2

2. 写出图 2 中稀疏矩阵的行三元组表示及快速转置算法中 $K[col]$ 数组中各元素值。
3. 设有向图的邻接表表示如图 3 所示, 请给出每个顶点的入度。

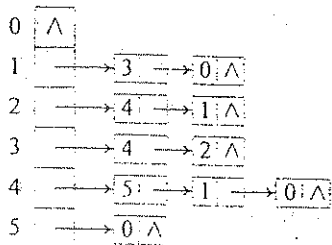
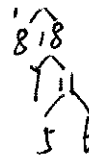
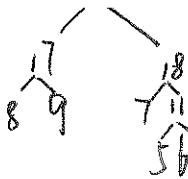


图 3

4. 空二叉搜索树中依次插入 33, 44, 99, 22, 11, 55, 画出最终所构建二叉搜索树。



$$9 + 16 + 21 + 11 \times 4 = 46 + 44 = 90$$

5. 设 $W = \{5, 6, 7, 8, 9\}$, 要求左子树根节点的权值小于等于右子树根节点权值。

- (1) 画出由权值集合 W 构造的哈夫曼树。
- (2) 计算加权路径长度。

四、判断题 (10 分, 共 5 题, 对的记 “√”, 错的记 “×”)

1. 线性结构只能用顺序结构来存放, 非线性结构只能用非顺序结构来存放. ~~×~~
2. 简单选择排序是稳定的排序算法. ~~×~~
3. 散列函数越复杂越好, 因为这样随机性好, 冲突概率小. ~~×~~
4. 完全二叉树一定存在度为 1 的结点. ~~×~~
5. 在一非空二叉树的中序遍历中, 根结点的右边是其右子树上的所有结点. ~~√~~

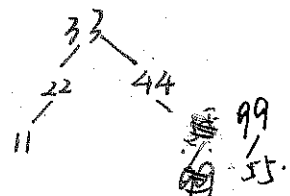
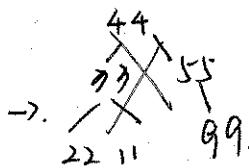
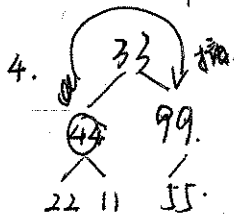
五、程序填空题 (10 分, 共 1 题)

1. 以下程序是对半搜索的迭代实现, 请填写完整。

```

BOOL BSearch2(List lst, KeyType k, T *x)
{
    int mid, low=(1) 0, high = lst.Size-1;
    while ((2) low < high)
    {
        mid=(low+high)/2;
        if (k < lst.Elements[mid].Key) high = (3) mid-1;
        else if ((4) k > lst.Elements[mid].Key) low = mid+1;
        else {
            *x = lst.Elements[mid]; return TRUE;
        }
    }
    (5) return false;
}

```



六、编程题 (10 分, 共 1 题)

1. 用二叉链表方式存储二叉树。试编写函数Count1, 求一棵二叉树的结点总数。并编写Count接口函数, 让其调用Count1函数。

```
typedef int K;
typedef struct btnode{
    K Element;
    struct btnode* LChild, *RChild;
}BTNode;
typedef struct btree{
    struct btnode* Root;
}BTree;
```

自觉遵守考场规则，诚信考试，绝不作弊

南京邮电大学 2016 / 2017 学年第一学期

《 数据结构 B 》 期末试卷 () 答案

一、填空题 (20 分, 共 10 题)

1	2	3	4	5	6	7	8	9	10
逻辑	89	2010	有向回路	有序	中	2	$n \log_2 n$	二次	邻接表

二、选择题 (20 分, 共 10 题)

1	2	3	4	5	6	7	8	9	10
A	B	C	D	C	D	C	B	A	D

三、简答题 (30 分, 共 5 题)

1.

前序遍历序列: BADCE (3 分)

中序遍历序列: ABCDE (3 分)

2.

row, col

0	0	21
1	2	11
2	1	1
2	3	3
2	4	8
3	3	2

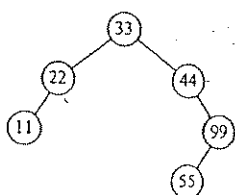
row, col

col	0	1	2	3	4
K[col]	0	1	2	3	5

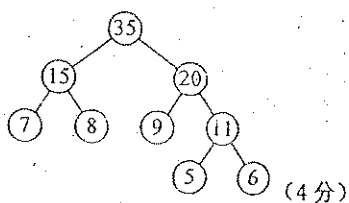
3. 每个 i 分

顶点	入度
0	3
1	2
2	1
3	1
4	2
5	1

4.



5.



$$WPL = (5+6) * 3 + (7+8+9)*2 = 33 + 48 = 81 \quad (2分)$$

四、判断题 (10 分, 每题 2 分)

1	2	3	4	5
×	×	×	×	√

五、程序填空题 (10 分, 每空 2 分)

(1) 0

(2) low <= high

(3) mid-1

(4) k>lst.Elements[mid].Key

(5) return FALSE;

六、编程题 (10 分, 共 1 题)

1.

```

int Count(BTree Bt){
    return Count1(Bt.Root);
}
  
```

```

int Count1(BTNode* p){
    if(!p) return 0;
    else return Count1(p->LChild)+ Count1(p->RChild) + 1;
}
  
```

南京邮电大学 2015 / 2016 学年第一学期

《数据结构 B》期末试卷 ()

本试卷共 4 页; 答题纸共 4 页; 考试时间 110 分钟;

专业 _____ 班级 _____ 学号 _____ 姓名 _____

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

- 注意: 1. 答案一律填写在答题纸上, 直接填写在试卷上成绩无效。
2. 试卷和答题纸上均填写完整的专业、班级、学号、姓名。
3. 考试完成后将试卷、答题纸、草稿纸交监考老师方可离开考场。

得分

一、填空题 (每小空 2 分, 共 20 分)

1. 通常将数据结构中的逻辑结构分为 (1) 集合结构、线性结构、树形结构、图结构, 这 4 种基本的结构关系可以分为两大类, 其中集合结构、树形结构、图结构属于 (2) 非线性结构。
2. 一个有 n 个顶点, e 条边的图 $G = (V, E)$ 的邻接矩阵为一个 $n \times n$ 的矩阵 A 。如果 G 是带权的无向图, 且 $A[2][3] = 5$, 则 $A[3][2]$ 的值为 (3) 5。所有顶点的出度和入度之和为 (4) $2e$ 。
3. 以 (5) 中序 遍历一棵二叉搜索树, 将得到一个以关键字值递增排列的有序序列。
4. 给定权值集合 $W = \{2, 3, 4, 6\}$, 构建的哈夫曼树的加权路径长度是 (6) 29。
5. 快速排序算法平均情况下的时间复杂度为 (7) $n \lg n$ 。
6. 某二叉树结点个数为 100 个, 叶子结点的个数为 50 个, 那么度为 1 的结点个数为 (8) 1 个。
7. 已知二维数组 $A[20][30]$, $\text{Loc}(A[0][0])$ 的地址为 100, 设每个元素占 1 个单元, 则 $\text{Loc}(A[10][12])$ 的地址为 412。
8. 一棵有 n 个结点的二叉树采用二叉链表方式存储, 有 (10) $n-1$ 个非空指针域。

得分

二、选择题 (每小题 2 分, 共 20 分)

1. 用对半搜索算法在表 $[5, 12, 20, 33, 54, 69, 76, 88, 96]$ 中查找 20, 经 B 次比较后搜索成功。
A. 1 B. 2 C. 3 D. 4
2. 设栈的输入序列是 a, b, c, d , (进栈后可立即出栈), 下列出栈列序列中 C 序列不可能得到。
A. a, b, c, d B. a, c, b, d C. a, d, c, b D. a, b, d, c

- A. c、b、a、d
 (C) d、b、c、a

- B. c、b、d、a
 D. b、c、d、a

3. 设有如下遗产继承规则，丈夫和妻子可以相互继承遗产，子女可以继承父亲或母亲遗产，子女间不能互相继承遗产，则表示该遗产继承关系的最合适的的数据结构是 B。

- A. 树 B. 图 C. 数组 D. 二叉树

4. 下面叙述中错误的是 C。

- A. 线性表采用顺序存储，必须占用一片连续的存储单元。
 B. 当线性表的元素总数基本稳定，并且很少进行插入和删除操作，但要求以最快的速度存取线性表中的元素时，应该采用顺序存储结构。
 C. 线性表采用链接存储，所占用的存储单元一定是非连续的。
 D. 线性表采用顺序存储，称为顺序表。

5. 下列说法错误的是 D。

- A. 堆栈和队列从逻辑上来说都是线性结构。
 B. 若进队列的序列是 w, x, y, z，则出队列的序列是 w, x, y, z。
 C. 在循环队列中队尾指针进行操作为 $Rear = (Rear + 1) \% \text{MaxQueue}$ 。
 D. 在循环队列中当 $Rear \% \text{MaxQueue} = \text{Front}$ 时认为队列已满。

6. 有关二维图形变换有以下几种说法：①树的子树之间是无序的，二叉树中结点的子树要区分左、右子树；②二叉树是树的一种特例；③二叉树可以为空二叉树；④森林可以为空森林；⑤高度为 n 的二叉树上至多有 2^{n-1} 个结点；... D 中哪一项所包含的说法均是正确的。

- A. ①②③④； B. ①②④； C. ①③⑤； D. ①③④。

7. 下面程序段的渐进时间复杂度为 B。

```
i=1; x=100;
do{
  x++; i=2*i;
}while(i<n)
```

- A. $O(n)$ B. $O(\log_2 n)$ C. $O(n \log_2 n)$ D. $O(n^2)$

8. 长度为 n 的线性表采用顺序存储结构存储，设插入表中每一个元素的概率是相等，则平均情况下从表中删除一个元素需移动元素的个数为 C。

- A. n B. n/2 C. (n-1)/2 D. 不能确定。

9. 对线性表 (35, 29, 72, 49, 28, 39, 65, 82) 进行一趟一般冒泡排序后得到的

序列为 A。

- A. (29, 35, 49, 28, 39, 65, 72, 82)
 B. (29, 35, 72, 49, 28, 39, 65, 82)
 C. (28, 29, 72, 49, 35, 39, 65, 82)
 D. (28, 29, 35, 39, 45, 65, 72, 82)

10. 具有 n 个顶点的无向图至少应有 A 条边才能确保是一个连通图。

A. $n-1$

B. n

C. $n+1$

D. $2n$

得分

三、简答题 (每小题 6 分, 共 30 分)

1. 已知某 5×6 稀疏矩阵的三元组表如表 1 所示, 请写出执行快速转置时用到的数组 $k[]$ 的值。

表 1

	行号	列号	值
0	0	1	1
1	0	5	2
2	1	0	3
3	2	3	4
4	3	4	5
5	4	0	6

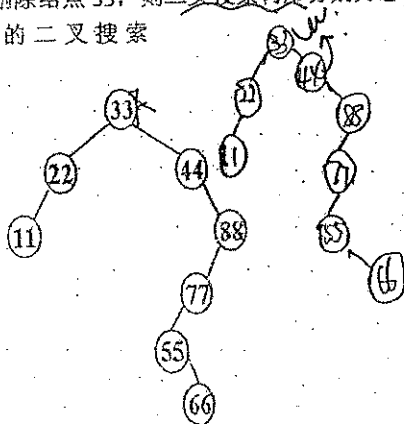
0	1	2	3	4	5
2	1	0	1	1	1
0	2	2	3	4	5

列号	0	1	2	3	4	5
$k[]$	0	2	②	③	④	5

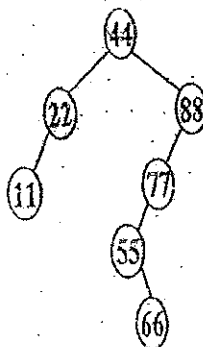
列号 0 1 2 3 4 5
 $k[]$ 0 2 ② ③ ④ 5
 列号 0 1 2 3 4 5
 0 → 0 列 为 0 个。
 0 → 0 列 为 0 个。
 0 → 0 列 为 0 个。
 0 → 0 列 为 0 个。
 0 → 0 列 为 0 个。
 0 → 0 列 为 0 个。

2. 建立 33, 44, 88, 22, 11, 77, 55, 66 为输入的二叉搜索树, 再从该二叉搜索树上删除结点 33, 则二叉搜索树形分别为怎样。

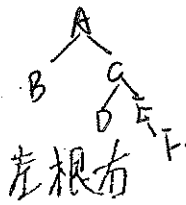
建立的二叉搜索



删除结点 33

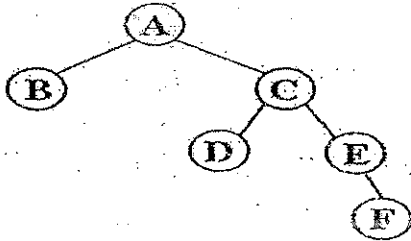


3. 对一棵二叉树进行中序遍历和后序遍历的结果如下, 画出该二叉树。

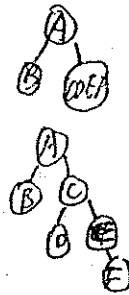
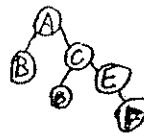


(1) 中序遍历: BADCEF

(2) 后序遍历: BDFECA



左右根



BADCEF
BDFECA

4. (1) 设 F 是一个森林, B 是由 F 转换得到的二叉树, F 中有 m 个结点, B 的根结点为 p, p 的右子树结点的个数为 n, 则森林 F 中第一棵树的结点个数为 m-n。

(2). 将图 1 中的二叉树转换为森林。

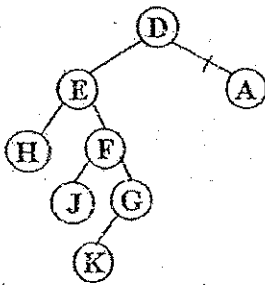
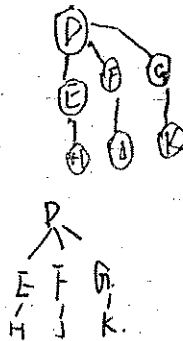
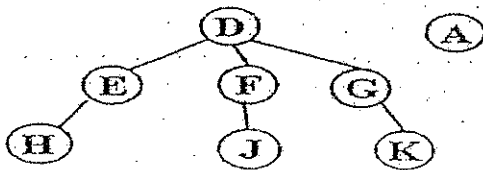


图 1



A



5. 设有散列表 ht[7], 散列函数为 $h(\text{key}) = \text{key} \% 7$, 采用线性探测法处理冲突, 试对关键字序列 70, 74, 53, 35, 42, 50 建立散列表, 请将下表数据补充完整。

0	1	2	3	4	5	6
70	35	42	50	74	53	

70 35 42 50 74 53

70 35 42 50 74 53

得分

四、按照要求解答下面关于图的问题。(共 14 分)

已知某 5 个顶点没有边的图，以边 $\langle 0, 3 \rangle$, $\langle 0, 1 \rangle$, $\langle 0, 2 \rangle$, $\langle 2, 4 \rangle$, $\langle 2, 3 \rangle$, $\langle 4, 3 \rangle$, $\langle 4, 1 \rangle$ 的次序，通过使用书中程序的 Add 函数插入这些边，建立该图的邻接表存储结构。(本小题 6 分)

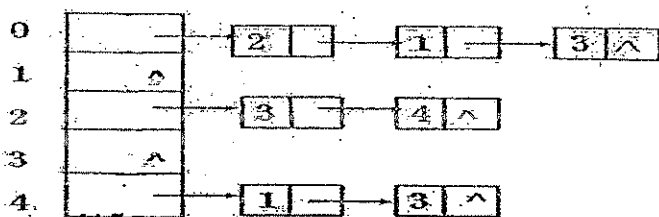


图 4

2. 根据有向图的深度优先搜索遍历算法，从顶点 0 出发，得到的顶点序列。(本题 3 分)

0, 2, 3, 4, 1

3. 使用普里姆 (Prim) 算法以 B 为源点，构造图 2 的最小代价生成树，按生成次序写出各条边构造图 5 的最小代价生成树。(本小题 5 分)

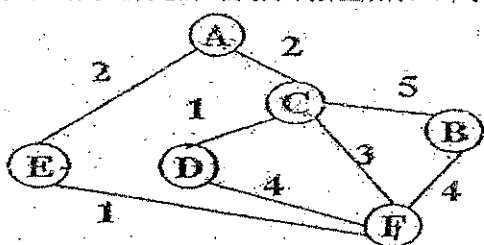
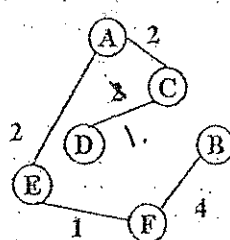
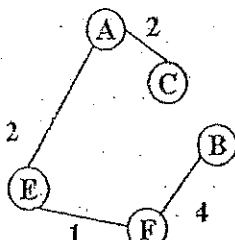
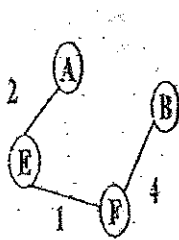
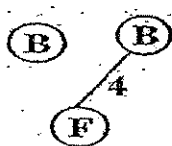
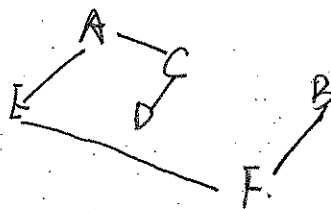


图 2



得分

五、程序填空题（每小空 2 分，共 8 分）

请将下面程序补充完整，利用对半搜索算法在一个有序表中插入一个元素 x ，并保持表的有序性。

```
typedef int KeyType;
typedef struct entry{
    KeyType Key;
    DataType Data;
}Entry;
typedef Entry T;
typedef struct list{
    int Size, MaxList;
    T Elements[MaxSize];
} List;
BseaInsert(List lst, KeyType x)
{
    int mid, low=0, high=lst.Size-1;
    int i, inplace, find=0;
    while (low<=high)
    {
        mid=(low+high)/2;
        if (x<lst.Elements[mid].Key) high=mid-1;
        else if (x>lst.Elements[mid].Key) low=mid+1;
        else {
            find=1;
        }
    }
    if (find) inplace=mid;
    else inplace=low;
    for(i=lst.Size-1; i>=inplace; i--)
    {
        lst.Elements[i+1].Key=lst.Elements[i].Key;
    }
    lst.Elements[inplace].Key=x;
}
```

得分

六、请按照要求编写程序交换二叉树中每个结点的左右子树。（本题 8 分）

```
typedef char K;
typedef int T;
```

要 答
| 觉 遵 守
则, 诚 信 考

```
typedef struct btnode
{
    K Element;
    struct btnode* LChild, *RChild;
}BTNode;

typedef struct btree{
    struct btnode* Root;
}BTree;

void Exch (BTNode *t);    // 实现算法的具体函数
void ExchOfLeaf(BTree Bt);    // 提供给用户使用的函数
void Exch(BTNode* p)
{
    if (p){
        BTNode *q=p->LChild;
        p->LChild=p->RChild;
        p->RChild=q;
        Exch(p->LChild);
        Exch(p->RChild);
    }
}

void ExchOfLeaf (BTree Bt)
{
    Exch(Bt.Root);
}
```

往下走
(每个结点)

南京邮电大学 2014 / 2015 学年第一学期

《数据结构 B》期末试卷 (B)

本试卷共 6 页; 考试时间 110 分钟;

专业 _____ 班级 _____ 学号 _____ 姓名 _____

题号	一	二	三	四	五	总分
得分						

注意: 1. 答案一律填写在试卷上。

2. 试卷上均填写完整的专业、班级、学号、姓名。

3. 考试完成后将试卷、草稿纸交监考老师后方可离开考场。

得分	阅卷人

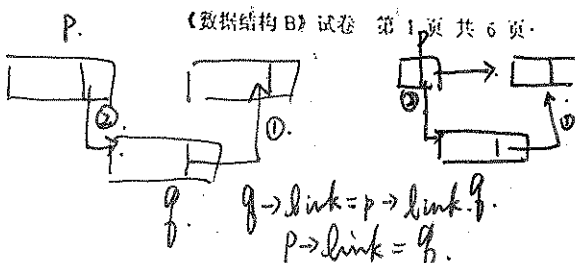
一、判断题 (每小题 1 分, 共 10 分, 正确的记 "T", 错误的记 "F")

- (F) 1. 线性表的逻辑顺序与存储顺序总是是一致的。
顺序存储是一致的, 链式不一定。
- (F) 2. 队列的插入操作在队尾进行, 删除操作在队头进行。
顺序插入。
- (F) 3. 二维数组是其数据元素可以表示为线性表的线性表。
- (F) 4. 二叉搜索树的形态取决于集合中元素关键字值的大小关系, 而与元素输入次序无关。
形态。
- (T) 5. 二叉树可以为空二叉树。
- (F) 6. 数据结构中四种基本逻辑结构是线性结构、集合结构、树形结构、图状结构。
图状结构。
- (F) 7. 对半搜索只适用于有序表, 包括有序的顺序表和有序的链表。
有序顺序。
- (F) 8. 循环队列是队列在循环链表上的实现。
有序顺序。
- (T) 9. 一棵有 n 个结点的二叉树采用二叉链表方式存储, 空指针域为 $n+1$ 。
有序顺序。
- (F) 10. 简单选择排序是稳定的排序算法。
不稳定, 三路元素交换, 相邻交换。

得分	阅卷人

二、选择题 (每小题 2 分, 共 20 分)

1. 线性表若采用链式存储结构时, 要求内存中可用存储单元的地址 _____。
A. 必须是连续的 B. 部分地址必须是连续的
C. 一定是不连续的 D. 连续或不连续都可以
2. 设单链表的每个结点有两个域: element 和 link, 在指针 p 所指示的结点之后插入新结点 q 的操作是 _____。
A. $q \rightarrow \text{link} = p; p \rightarrow \text{link} = q;$ B. $q \rightarrow \text{link} = p \rightarrow \text{link}; p \rightarrow \text{link} = q;$
C. $q \rightarrow \text{link} = p \rightarrow \text{link}; p \rightarrow \text{link} = q;$ D. $p \rightarrow \text{link} = q; q \rightarrow \text{link} = p;$



①. $q \rightarrow \text{link} = p \rightarrow \text{link};$
 $p \rightarrow \text{link} = q;$

3. 设栈的输入序列是 a、b、c、d，(进栈后可立即出栈)，下列出栈序列中 C 序列不可能得到。

A. c、b、a、d ✓

B. c、b、d、a

C. d、b、c、a

D. b、c、d、a

4. 散列函数有一个共同的要求，即函数值应当以 C 取其值域的每个值。

A. 最大概率

B. 最小概率

C. 随机概率

D. 均等概率

5. 设有如下遗产继承规则，丈夫和妻子可以相互继承遗产，子女可以继承父亲或母亲遗产，子女间不能互相继承遗产，则表示该遗产继承关系的最合适的数据结构是 B。

A. 线性结构

B. 图结构

C. 树形结构

D. 集合结构

6. 二叉搜索树中，最小元素位于 A。

A. 根结点的最左下结点处

B. 根结点的最右下结点处

C. 根结点的右结点的最左下结点处

D. 根结点的左结点的最右下结点处

7. 已知元素序列为 (20, 40, 30, 60, 50, 70, 90)，利用快速排序的方法，以第一个元素为基准得到的第一次划分结果为 C。

A. 40, (20, 30, 60, 50, 70, 90)

B. 30, (-10, 20, 60, 50, 70, 90)

C. 20, (-10, 30, 60, 50, 70, 90)

D. (20, 30), 10, (60, 50, 70, 90)

8. 采用邻接表存储的图的宽度优先遍历算法类似于二叉树的 D。

A. 先序遍历

B. 中序遍历

C. 后序遍历

D. 按层次遍历

9. 深度为 5 (根为第一层) 的二叉树至多有 C 个结点。

A. 16

B. 32

C. 31

D. 33

10. 长度为 n 的线性表采用顺序存储结构存储，设删除表中每一个元素的概率是相等，则平均情况下从表中删除一个元素需移动元素的个数为 C。

A. n

B. n/2

C. (n-1)/2

D. 不能确定。

得分	阅卷人

三、解答题 (50 分)

1. (1) 给出以下稀疏矩阵的行三元组表示。(5 分)

$$\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 21 & 0 & 0 & 25 & 0 \\ 0 & 12 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

0	0	21
1	1	12
2	1	-1
3	4	8

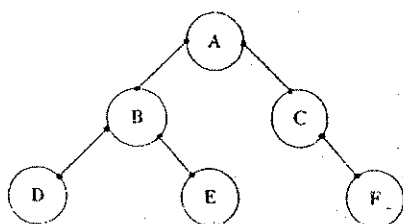
(2) 给出该稀疏矩阵快速转置算法中的 K[col] 数组中各元素的值。(5 分)

col	0	1	2	3	4
K[col]	0	1	3	3	4

2. 设散列表的长度为 11，散列函数 $h(key) = key \% 11$ ，采用线性探测法处理冲突，试用关键字序列 21, 15, 44, 65, 22, 40, 58, 67 建立该散列表。(6 分)
(说明：在下表的合适位置直接填入关键字)

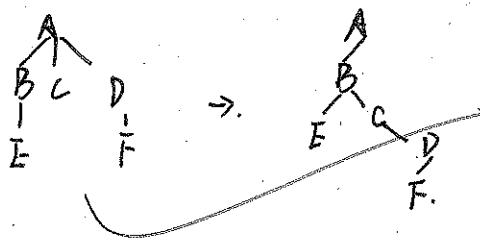
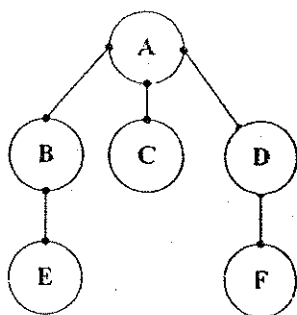
0	1	2	3	4	5	6	7	8	9	10
44	65	22	58	15	67		40			21

3. 写出下图中的二叉树所对应的先序遍历。(6 分)



A B D E. C F.

4. 画出下列树对应的二叉树。(6 分)



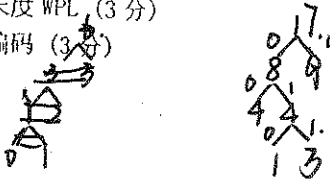
5. 设有字符集 $S=\{W, X, Y, Z\}$, 其出现的频率分别是 $w=\{9, 4, 3, 1\}$ (共 10 分)

(1). 画出哈夫曼树 (构建新二叉树时, 新二叉树根的左子树根的权值小于等于右子树根的权值) (4 分)

(2). 计算带权路径长度 WPL (3 分)

(3). 给出各字符的编码 (3 分)

解: 1. 11.

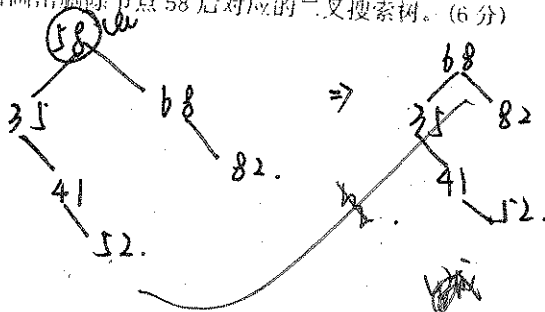


(2). $WPL = 9 \times 1 + 4 \times 2 + 4 \times 3 = 25 + 4 = 29$.

(3).

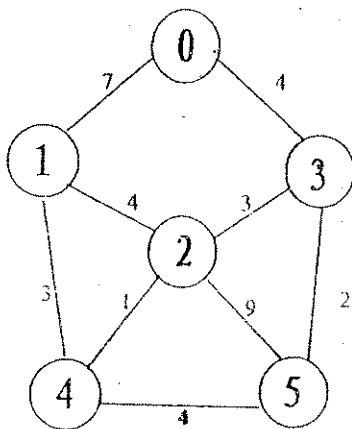
W	X	Y	Z
1	00	010	010

6. 请按照以下序列构造一颗二叉搜索树 (58 35 41 52 68 82), 并画出此二叉搜索树, 然后画出删除节点 58 后对应的二叉搜索树. (6 分)

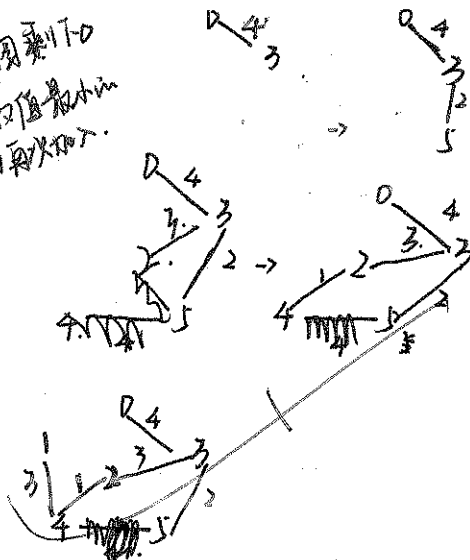


7. 用普里姆算法从下图中的顶点 C 开始逐步构造最小代价生成树, 并计算该生成树的代价. (6 分)

(要求: 画出构造的每一步)



↑ 马路上产生的路口和联系
找法 将已经生成的图乘以下一步
的图中权值最小的边再次加入。



带格式的
框线)

生成树代价:

$$1+2+3+3+4=13$$

得分	阅卷人

四、算法填空 (10 分)

int bSch(List lst, KeyType k, int low, int high)

```

{
    int mid;
    if (low <= high)
    {
        mid = (low + high) / 2;
        if (k < lst.Elements[mid].Key) { return bSch(lst, k, low, mid-1); }
        else {
            if (k > lst.Elements[mid].Key)
                return bSch(lst, k, mid+1, high);
            else return mid;
        }
    }
    return -1;
}

```

带格式的：边框：底端：框线

BOOL BSearch(List lst, KeyType k, T *x)

```
{
    int i;
    i=bSch(lst, k, 0, lst.Size);
    if (i== -1) return FALSE;
    else {
        *x=lst.Elements[i];
        return TRUE;
    }
}
```

得分	阅卷人

五、算法设计 (10 分)

设二叉树以二叉链表存储，在二叉树的根指针为 Bt，试编写求二叉树中度数为 1 的节点个数的递归算法（函数定义的首部已给出，其中 BTreeNode 是二叉树的结点类型，BTree 为二叉树类型）。

int Degree1 (BTreeNode * p)

```
{
    int i=0;
```

```
}
```

int BtDegree1 (BTree Bt)

```
{
```

```
    return Degree1 ( Bt.root );
}
```

南京邮电大学 2014 / 2015 学年第一学期

《数据结构 B》参考答案 (B)

一、判断题 (每小题 1 分, 共 10 分, 正确的记“T”, 错误的记“F”)

- () 1. 线性表的逻辑顺序与存储顺序总是一致的。F
- () 2. 队列的插入操作在队尾进行, 删除操作在队头进行。T
- () 3. 二维数组是其数据元素可以表示为线性表的线性表。T
- () 4. 二叉搜索树的形态取决于集合中元素关键字值的大小关系, 而与元素输入次序无关。F
- () 5. 二叉树可以为空二叉树。T
- () 6. 数据结构中四种基本逻辑结构是线性结构、集合结构、树形结构和顺序结构。F
- () 7. 对半搜索只适用于有序表, 包括有序的顺序表和有序的链表。F
- () 8. 循环队列是队列在循环链表上的实现。F
- () 9. 一棵有 n 个结点的二叉树采用二叉链表方式存储, 空指针域为 $n+1$ 。T
- () 10. 简单选择排序是稳定的排序算法。F

二、选择题 (每小题 2 分, 共 20 分)

1. D 2. C 3. C 4. D 5. B 6. A 7. C 8. D 9. C 10. C

三、解答题 (50 分)

1. (1) [评分] 共 5 分, 每个元素 1 分

0	0	21
0	3	25
1	1	12
2	1	-1
2	4	8

- (2) [评分] 共 5 分, 每个元素 1 分

col	0	1	2	3	4
K[col]	0	1	3	3	4

2. [评分] 共 6 分, 每个元素 1 分

散列函数 $h(key) = key \% 11$, 线性探测法处理冲突, 序列 21, 15, 44, 65, 22, 40, 58, 67

0	1	2	3	4	5	6	7	8	9	10
44	65	22	58	15	67		40			21

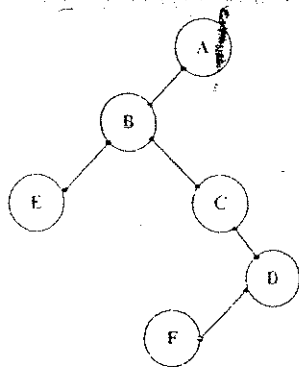
3. [评分] 共 6 分, 每个元素 1 分

二叉树所对应的先序遍历:

A B D E C F

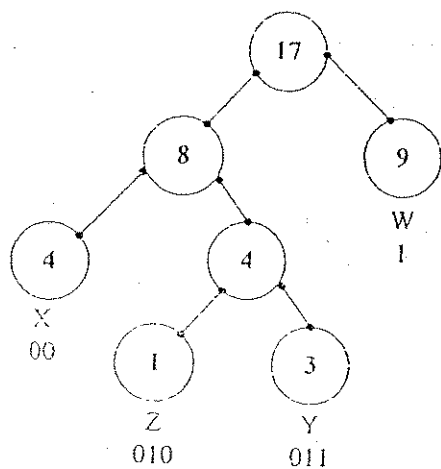
4. [评分] 共 6 分, 每个元素 1 分

所求树对应的二叉树为:



5. [评分] 共 10 分，其中：(1) 画出哈夫曼树 4 分；(2) 计算带权路径长度 3 分；(3) 给出各字符的编码 3 分。

(1) 画出哈夫曼树 (4 分)



(2) 计算带权路径长度 (3 分)

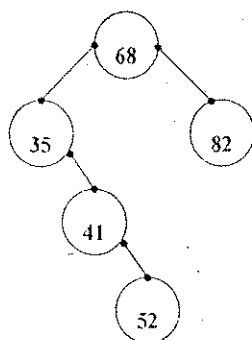
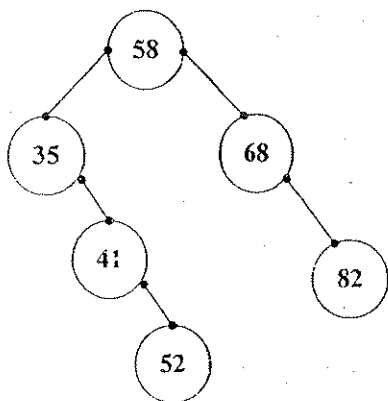
$$WPL = \sum_{i=1}^4 W_i L_i = 9 \times 1 + 4 \times 2 + 3 \times 3 + 1 \times 3 = 29$$

(3) 给出各字符的编码 (3 分)

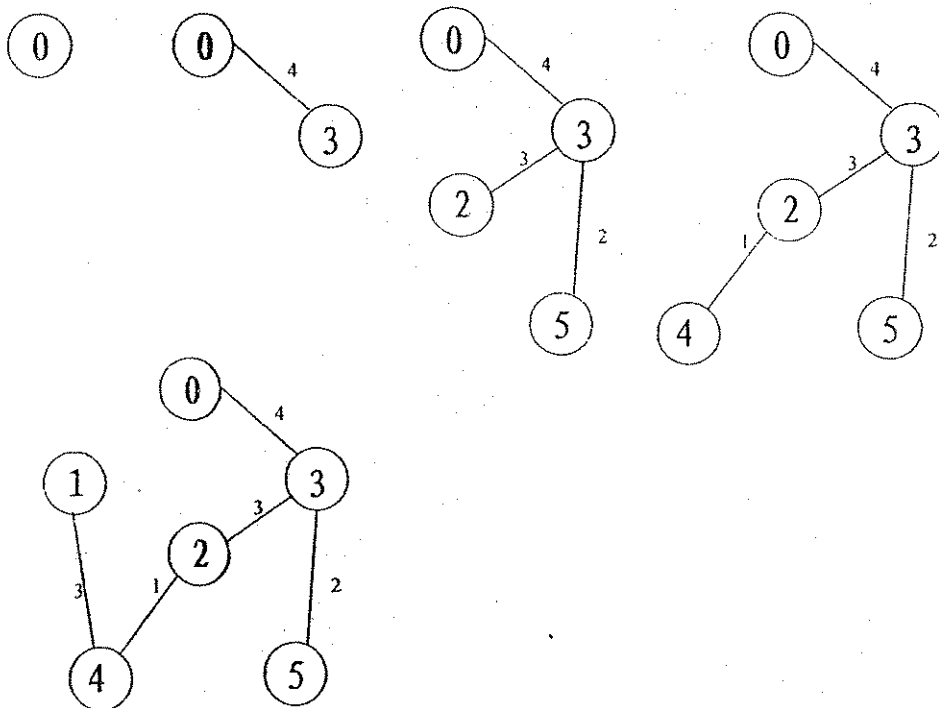
W: 1 X: 00 Y: 011 Z: 010

6. [评分] 共 6 分，每个元素 1 分

(58 35 41 52 68 82)，构造一颗二叉搜索树



7. [评分] 共 6 分, 5 个步骤每步骤 1 分, 生成树的代价 1 分
用普里姆算法, 最小代价生成树



该生成树的代价为: $1+2+3+3+4=13$

四、算法填空 (每空 2 分, 共 10 分)

以下 C 程序实现对半搜索的递归算法, 试将其补充完整。

```

int bSch(List lst, KeyType k, int low, int high)
{
    int mid;

```

```

if ( low <= high )
{
    mid = (low+high)/2;
    if (k < lst.Elements[mid].Key) {return bSch(lst, k, low, mid-1); }
    else {
        if (k > lst.Elements[mid].Key)
            return bSch(lst, k, mid+1, high);
        else return mid;
    }
}

return -1;
}

BOOL BSearch(List lst, KeyType k, T *x)
{
    int i;
    i = bSch(lst, k, 0, lst.Size);
    if (i == -1) return FALSE;
    else {
        *x = lst.Elements[i];
        return TRUE;
    }
}

```

五、算法设计 (共 10 分, BtDegree1 函数 2 分; Degree1 函数 8 分)

```

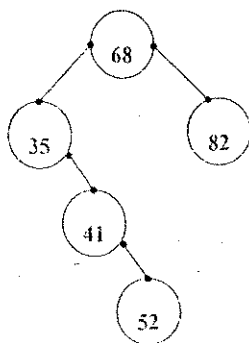
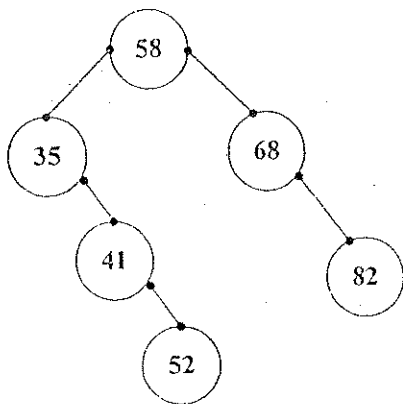
int Degree1 (BTreeNode * p)
{
    int i=0; //1 分
    if (!p) return 0; //1 分
    else if ((!p->LChild && p->RChild) || (p->LChild && !p->RChild)) i=1; //2 分
    return Degree1 (p->LChild) + Degree1 (p->RChild) + i; //4 分
}

int BtDegree1 (BTree Bt)
{
    return Degree1 (Bt.Root); //2 分
}

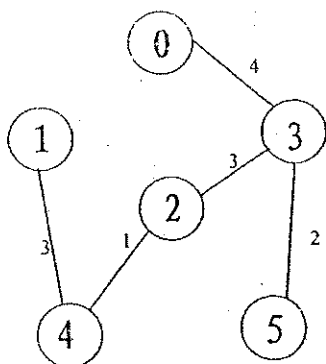
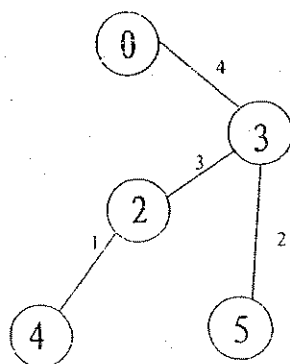
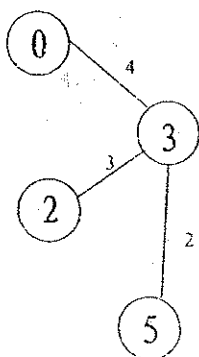
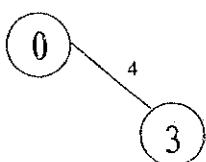
```

8.3 8.4

	0	1	2	3	4	5	6	7	8	9	10
线性探查法	55	45	35	25	70	80	60	50			
二次探查法	45	35	80	25	70	60	50				55
双散列法	55	80	35	25	70	60	45	50			



7. [评分] 共 6 分, 5 个步骤每步骤 1 分, 生成树的代价 1 分
用普里姆算法, 最小代价生成树



该生成树的代价为: $1+2+3+3+4=13$

四、算法填空 (每空 2 分, 共 10 分)

以下 C 程序实现对半搜索的递归算法, 试将其补充完整。

```

int bSch(List lst, KeyType k, int low, int high)
{
    int mid;

```

```

if ( low<=high )
{
    mid= (low+high)/2 ;
    if (k<lst.Elements[mid].Key) {return bSch(lst, k, low, mid-1); }
    else{
        if (k>lst.Elements[mid].Key)
            return bSch(lst, k, mid+1, high);
        else return mid;
    }
}

return -1;
}

BOOL BSearch(List lst, KeyType k, T *x)
{
    int i;
    i=bSch(lst, k, 0, lst.Size);
    if (i== -1 ) return FALSE;
    else {
        *x=lst.Elements[i];
        return TRUE;
    }
}

```

五、算法设计（共 10 分，BtDegree1 函数 2 分；Degree1 函数 8 分）

```

int Degree1 (BTreeNode * p)
{
    int i=0; //1 分
    if(!p) return 0; //1 分
    else if((!p->LChild && p->RChild) || (p->LChild && !p->RChild)) i=1; //2 分
    return Degree1 (p->LChild)+ Degree1 (p->RChild)+i; //4 分
}

int BtDegree1 (BTree Bt)
{
    return Degree1 (Bt.Root); //2 分
}

```

8.3 8.4

	0	1	2	3	4	5	6	7	8	9	10
线性探查法	55	45	35	25	70	80	60	50			
二次探查法	45	35	80	25	70	60	50				55
双散列法	55	80	35	25	70	60	45	50			

期 末 《 数据结构 B 》 期末 试卷 (B)

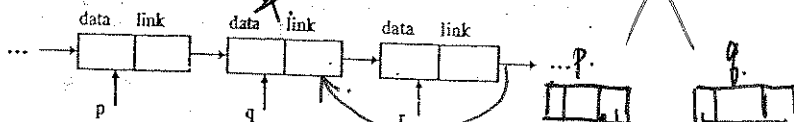
班级	学号	姓名	得分							
题号	一	二	三	四	五	六	七	八	九	十
得分										

一、填空题 (6小题, 每小题2分, 共12分)

1. 顺序表中各元素之间的地址是 连续的。
2. 若有n个表并存, 并且处理时各表的长度和表的总数都会动态变化, 则应采用 链式 存储结构。
3. 栈的TOP运算的功能是 取栈顶元素, 但此运算不删除栈顶元素。
4. 对树进行后序遍历时, 最后出现的结点, 在先序遍历中将 最先 出现。
5. AOV 网络中活动之间的领先关系是一种 拟序, 它具有传递性和反自反性。
6. 若待排序序列为 0 81 15 8 9 15 71, 经过某种排序后, 得到的序列为 0 8 9 15 15 71 81, 则此排序算法的稳定性如何? 答案是 不稳定。

二、单项选择题 (7小题, 每小题2分, 共14分)

1. 将 q 和 r 所指结点的先后位置交换且不出现断链现象, 以下错误的程序段是 D。

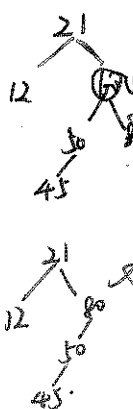


- (A) $q \rightarrow \text{link} = r \rightarrow \text{link}; p \rightarrow \text{link} = r; r \rightarrow \text{link} = q;$
 (B) $r \rightarrow \text{link} = q; q \rightarrow \text{link} = r \rightarrow \text{link}; p \rightarrow \text{link} = r;$
 (C) $p \rightarrow \text{link} = r; q \rightarrow \text{link} = r \rightarrow \text{link}; r \rightarrow \text{link} = q;$
 (D) $q \rightarrow \text{link} = r \rightarrow \text{link}; r \rightarrow \text{link} = q; p \rightarrow \text{link} = r;$
2. WINDOWS 操作系统中页面调度时, 先来先服务算法是 C 的典型应用实例。
 (A) 数组 (B) 堆栈 (C) 队列 (D) 二叉树
3. 中缀表达式 $A + (B - C / D) * E$ 的后缀形式是 B。
 (A) $ABCD / E + -$ (B) $ABCD - E * +$
 (C) $ABCD + - * E /$ (D) $ABCD + E - / *$
4. 在对非空二叉树进行中序遍历的序列中, 根结点右边 D。
 (A) 只有左子树上的部分结点 (B) 只有根的左子树上的全部结点或无结点
 (C) 只有右子树上的部分结点 (D) 只有根的右子树上的全部结点或无结点

5. 有 t 个非零元素的稀疏矩阵 $A_{m \times n}$, 采用三元组表示, 则快速转置的时间复杂度是 A.
- (A) $O(m \times n)$ (B) $O(m+n)$
(C) $O(n+t)$ (D) $O(n \times t)$
6. 二叉搜索树中, 最小元素结点的左子树 C, 它的右子树 有比其大.
- (A) 一定为空, 不一定为空 (B) 不一定为空, 一定不为空
(C) 一定不为空, 不一定为空 (D) 不一定为空, 不一定为空
7. 对非连通图进行广度优先搜索得到 A.
- (A) 树 (B) 森林
(C) 生成树 (D) 生成森林

三、简答题 (8 小题, 每小题 6 分, 共 48 分)

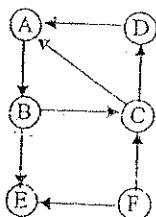
1. 输入序列为 (21 60 12 50 45 80), 请先建立 二叉搜索树, 再从此树上将 60 删除.



2. 设散列表的长度为 11, 请采用 双散列法 解决冲突, 试以散列函数 $h_1(key) = key \% 11$, $h_2(key) = key \% 9 + 1$, 从空表开始依次插入下列关键字值序列: 81 28 80 35 60 45, 建立散列表. 请画出该散列表.

同该结点在列的最后结点. key 81 28 80 35 60 45

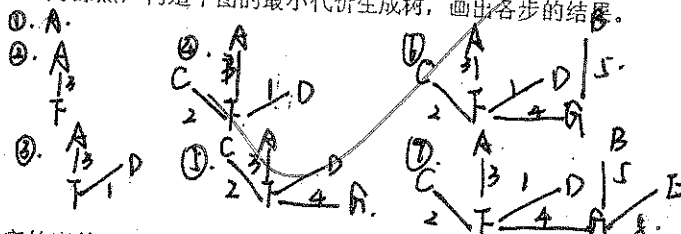
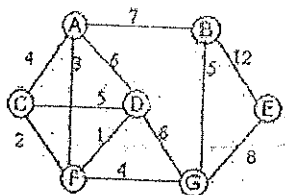
3. 有向图见下图. 给出强连通分量的定义并画出强连通分量.



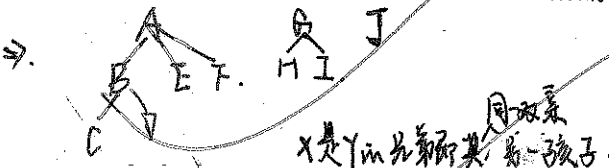
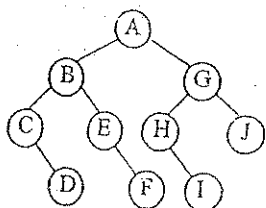
强连通分量: 有向图中极大强连通子图称为其强连通分量.



4. 使用普里姆 (Prim) 算法以 A 为源点, 构造下图的最小代价生成树, 画出各步的结果.



5. 画出下图中的二叉树所对应的森林; 若 X 结点是其双亲 Y 的右孩子, 则在对应的树或森林中 X 是 Y 的什么结点?



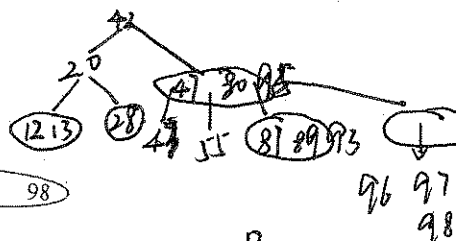
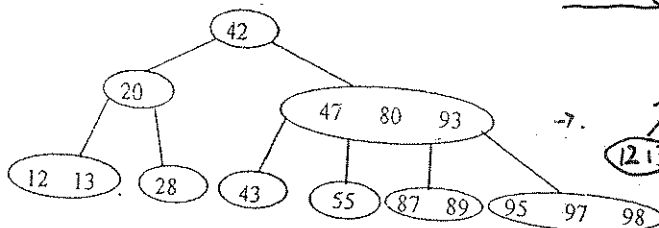
6. 当以边 $\langle 0, 1 \rangle$, $\langle 1, 3 \rangle$, $\langle 1, 2 \rangle$, $\langle 2, 5 \rangle$, $\langle 5, 0 \rangle$, $\langle 4, 2 \rangle$, $\langle 4, 3 \rangle$, $\langle 2, 0 \rangle$ 的次序从只有 6 个顶点没有边的图开始, 通过插入这些边, 建立邻接表.

(1) 画出该邻接表;

(2) 在所建立的邻接表上, 进行以 0 为起始顶点的 深度 优先遍历, 写出遍历结果.

7. 设字符集合S={A, B, C, D, E}, 各字符的使用频率为W={5, 7, 19, 6, 3}
- 画出哈夫曼树; (生成新结点时, 新结点的左子树根的权值小于等于右子树根的权值)
 - 求该哈夫曼树的带权路径长度;

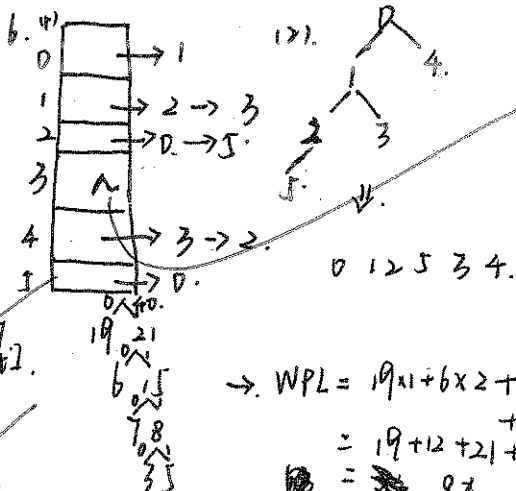
8. 设有四阶B-树, 如下图所示 (未画出失败结点)。画出插入关键字96的B-树;



四、程序填空题 (每空2分, 共8分)

下面是快速排序程序, 请补充完整:

```
template <class T>
void QSort(T A[], int left, int right)
{
    int i, j;
    if (left < right)
    {
        i = left; j = right + 1;
        do
        {
            do i++; while ( (1) );
            do j--; while ( (2) A[j] < A[left] );
            if (i < j) Swap(A[i], A[j]);
        } while (i < j);
        Swap( (3) A[left], A[j] );
        QSort(A, left, j-1); QSort(A, j+1, (4) right);
    }
}
```



$$\begin{aligned} WPL &= 19 \times 1 + 6 \times 2 + 7 \times 3 \\ &\quad + 8 \times 4 \\ &= 19 + 12 + 21 + 32 \\ &= 84 \end{aligned}$$

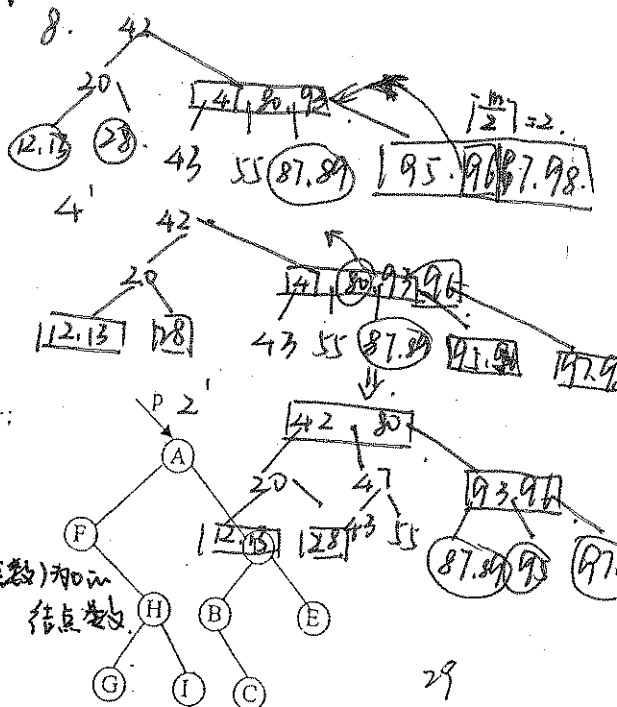
五、程序阅读题 (每个小题3分, 共6分)

算法程序如下:

int k; // 设k是Btree类中的数据成员, 并有初值0

```
template <class T>
int BTree<T>::F(BTNode<T>* u)
{
    if (u)
    {
        if (u->lchild == null && u->rchild == null) k++;
        F(u->lchild);
        F(u->rchild);
    }
}
```

- 说明该程序的作用是什么? 返回叶子结点的个数
- 执行F(p)后, k的值是多少?



六、程序设计题(12分)

已知带表头的单链表(SingleList)的结点(Node)有两个私有的数据成员: data 和 link, 其中 data 是结点关键字, link 是指向 Node 的指针。SingleList 中私有的数据成员有两个: first 和 length, 其中 first 是指向第一个结点的指针, length 是当前单链表中结点的个数。请完成:

- (1) 写出结点 Node 和单链表 SingleList 的 C++ 类模板; (Node 只要求写出私有的数据成员, SingleList 不用从 LinearList 继承, 其成员函数只要列出(2)中的 Insert 函数即可)
- (2) 假设单链表中的结点是有序递增的, 设计成员函数 Insert(const T& x), 在单链表中插入元素 x 且保证链表的有序性。(单链表中任意两个结点的关键字都不相等)

一、填空题 (6小题, 每小题2分, 共12分)

1. 连续的
2. 链式
3. 取得栈顶元素的值
4. 最先
5. 拟序
6. 不稳定

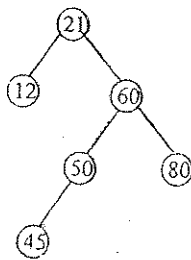
二、单项选择题 (7小题, 每小题2分, 共14分)

BCBDCAD

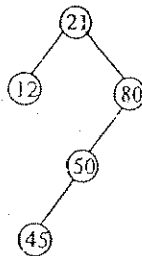
三、简答题 (8小题, 1每题6分, 共48分)

1.

4分



2分



2.

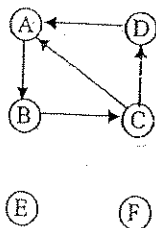
(每个1分)

0	1	2	3	4	5	6	7	8	9	10
	80	35	25	81	60	45				

3. (1) 有向图的一个极大强连通子图称为该图的一个强连通分量。(2分)

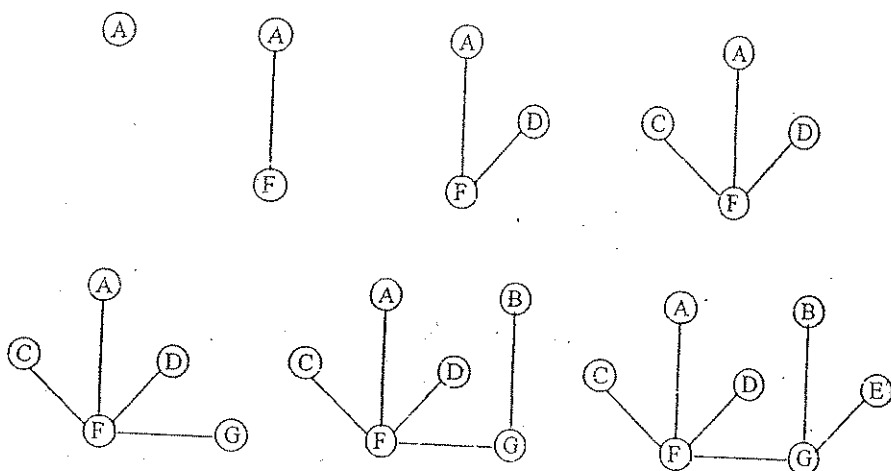
(2)

(4分)



4.

(每步 1 分)

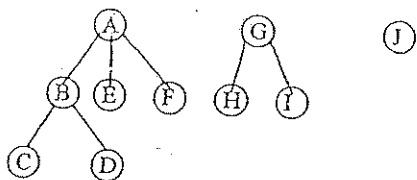


5.

2 分

1 分

1 分



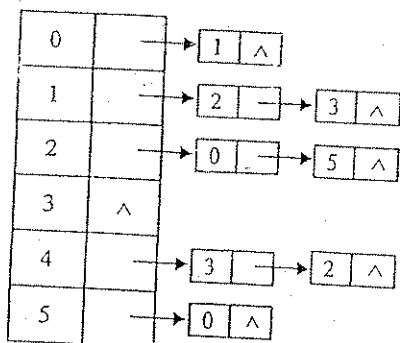
(2) 兄弟

6.

(2 分)

(1)

(4 分)

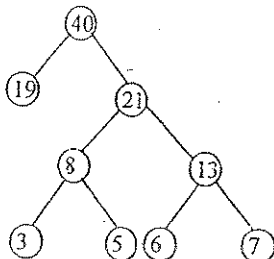


(2) 012534

(2 分)

7. (1)

(4分)

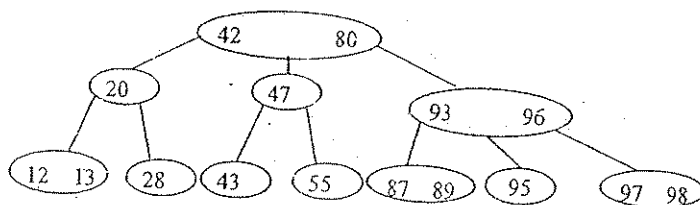


(2) $WPL = (3+5+6+7)*3 + 19*1 = 82$

8. (1). 插入 96

(2分)

(第一次分裂 4分, 第二次分裂 2分)



四、程序填空题 (8分)

(每空 2分)

- (1) $A[i] < A[\text{left}]$
- (2) $A[j] > A[\text{left}]$
- (3) $A[\text{left}], A[j]$
- (4) $j+1, \text{right}$

五、程序阅读题 (6分)

(1) 计算叶子结点数

(3分)

(2) 4

六、程序设计题 (12分)

(3分)

(1)

template <class T> class SingleList;

template <class T>

class Node

(2分)

private:

```

class SingleList
{ public:
    SingleList();
    ~SingleList();
    bool Insert(const T& x);
private:
    Node<T>* first;
    int length;
};

```

(2 分)

```

(2) template<class T>
bool SingleList<T>::Insert(const T&x)
{
    Node<T>* p,q,r;
    p=first->link;
    q=first;
    while(!p&&p->data<x)
    { q=p;
      p=p->link;
    }
    if(p->data==x) return false;
    r=new Node<T>;
    r->data=x;
    r->link=q->link;
    q->link=r;
    length++;
    return true;
}

```

(3 分)

(2 分)

(2 分)

(1 分)

数据结构B 期末试卷

班级 _____ 学号 _____ 姓名 _____ 得分 _____

题号	一	二	三	四	五	六
分数						

一、解答题：（共 82 分）

1、下列程序段或函数的时间复杂度。（10%）

(1) for (int k=0; k<m; k++)
for (int j=0; j<n; j++)
a[k][j]=k*j;

$T(n) = O(mn)$
 ~~$O(m^2)$~~

(2) int fac(unsigned int n)
{ if (n==0||n==1) return 1;
else return n*fac(n-1); }

$T(n) = O(n)$

(3) int Prime(int n)
{ int k=2, x=(int)sqrt(n);
while (k<=x) {
if (n % k==0) break;
k++; }
if (k>x) return 1;
else return 0; }

$T(n) = O(\sqrt{n})$

(4) k=1; x=0;
do {
x++; k*=2;
}
while (k<n);

$T(n) = O(\log_2 n)$

2、有 A、B、C、D 四个元素依次入栈，即入栈序列唯一，问共能得到多少种出栈序列？能否得到以下四种出栈序列：ABCD、BDAC、CBDA、DBAC。对能得到的序列，请写出 Push、Pop 序列；对不能得到的序列，请说明理由。（6%）

~~ABCD, DCBA, BADC~~
14种

ABCD. push pop. push pop. push pop.
push pop.

BDAC. 不能. D不可入栈

CBDA. push push push pop pop
push pop pop.

DBAC. 不可以
A不能入栈

依据
压栈
出栈
规则

At 17.

3、矩阵 $A_{m \times n}$ 以行优先方式从 1000H 处开始存放，元素类型未知，已知：A[2][3] 存放在 1011H 处，A[1][1] 存放在 1005H 处，求元素 A[2][0] 的存放位置。(6%)

$$(2 \times n + 3) \times k = 11$$

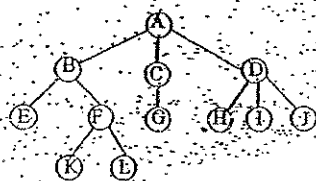
$$(1 \times n + 1) \times k = 5$$

$$\frac{2n+3}{n+1} = \frac{11}{5}$$

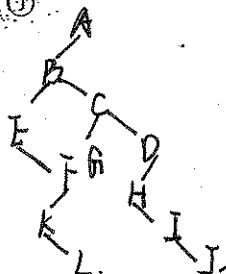
$$\begin{cases} n = 4 \\ k = 1 \end{cases}$$

4、根据下图所示的树回答问题。(共 13%)

(1) 画出该树等效的二叉树。(3%)



等效的二叉树



(2)、写出对该树进行先序、后序遍历的结点序列。(4%)

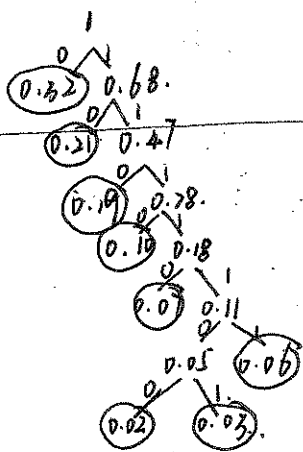
先序：A B E F K L C G D H I J

后序：L K F E G J I H D C B A

(3) 用带右链的先序表示法来存储此树，填写下表。(6%)

下标	0	1	2	3	4	5	6	7	8	9	10	11
sibling												
element												
ltag												

5、假设用于通讯的电文仅由 {ABCDEFGH} 8 个字母组成，字母在电文中出现的频率分别为 0.07, 0.19, 0.02, 0.06, 0.32, 0.03, 0.21, 0.10。请画出哈夫曼树并在树中标明编码情况，给出这 8 个字母的哈夫曼编码，最后求出 WPL。(9%)



0.32 0.21 0.19 0.10
0. 10. 110. 1110.

0.07 0.06 0.02.

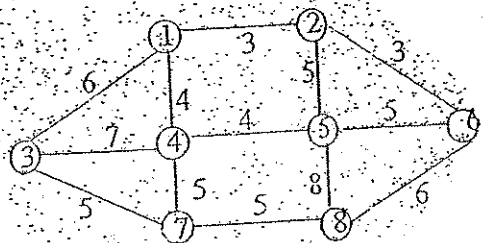
11110. 11111 1111100.

0.03

111111

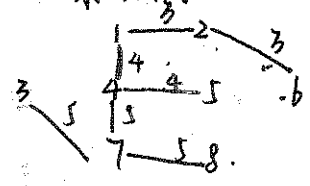
$$WPL = 0.32 \times 1 + 0.21 \times 2 + 0.19 \times 3 + 0.10 \times 4 + 0.07 \times 5 + 0.06 \times 6 + 0.03 \times 7$$

6. 对下图, 要求: (共 13%)



(1) 画出它的邻接表. (3%)

邻接表



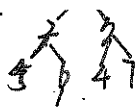
1	→	2 (3)	→	3 (6)	→	4 (4)
2	→	1 (3)	→	5 (5)	→	6 (3)
3	→	1 (6)	→	4 (7)	→	7 (5)
4	→	4 (4)	→	5 (4)	→	7 (4)
5	→	2 (5)	→	4 (4)	→	6 (5)
6	→	2 (3)	→	5 (5)	→	8 (6)
7	→	3 (5)	→	4 (7)	→	8 (5)
8	→	5 (8)	→	6 (6)	→	7 (5)

(2) 写出从顶点 1 到顶点 8 的四条路径长度为 3 的简单路径. (2%)

- 1378
- 1458
- 1258
- 1268

12543100.

~.



12356478 X.

12345678.

(3) 分别写出从顶点 1 出发根据 (1) 所示的邻接表用深度优先搜索和广度优先搜索遍历图得到的顶点序列。(4%)

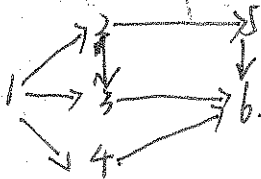
(4) 求出它的一棵最小代价生成树(方法任选), 其代价是多少? 你所求出的最小代价生成树是唯一的吗?(4%)

是唯一

7. 一项工程 P 由 P1, P2, P3, P4, P5, P6 六个子工程组成, 这些工程之间有下列关系: $P1 > P2, P1 > P3, P1 > P4, P2 > P3, P2 > P5, P3 > P6, P4 > P6, P5 > P6$. 其中符号 ">" 表示先于关系例如 $P1 > P2$ 表示只有在工程 P1 完成之后才能进行 P2 的工作。请: (7%)

(1) 画出该工程的 AOV 网

(2) 给出工程 P 的其中四种可能的施工顺序。



(2). 12346.

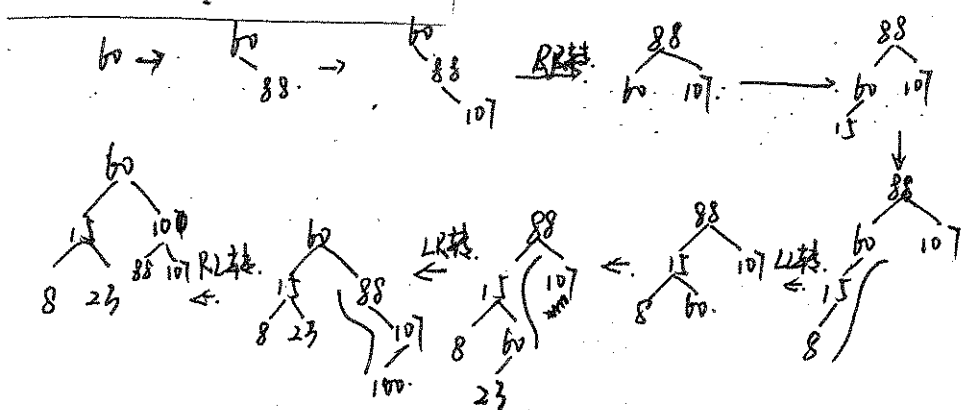
123456.

132456.

142356.

125346.

8. 按如下关键字序列 (60, 88, 107, 15, 8, 23, 100) 从空树开始建立一棵 AVL 搜索树, 画出建树的步骤以及调整平衡的过程 (6%)



9. 设散列表 $ht[13]$, 散列函数 $h(key) = key \% 13$. 采用二次探查法解决冲突, 试用关键字值序列: {56, 78, 14, 27, 41, 70, 51, 86, 24, 30, 36} 建立散列表. (6%)

i	0	1	2	3	4	5	6	7	8	9	10	11	12
ht[i]	78	14	27	41	56	70		51		36	66	24	51

10. 元素序列: {55, 71, 12, 98, 4, 70, 51}; 请写出用冒泡排序法和 2 路合并排序法进行排序的各趟排序结果. (6%)

冒泡排序法

- ① [55 12 71 4 70 51] 98
- ② [12 55 4 70 51] 71 98
- ③ [12 4 55 51] 70 71 98
- ④ [4 12 51] 55 70 71 98
- ⑤ [4 12] 51 55 70 71 98
- ⑥ [4] 12 51 55 70 71 98

2 路合并排序法

- ① (55, 71), (12, 98), (4, 70), (51)
- ② (12, 55, 71, 98) (4, 51, 70)
- ③ (4, 12, 51, 55, 70, 71, 98)

二、算法填空 (8%) ①

以下算法实现二叉搜索树的删除, 根据给定的关键字 k , 找到待删除元素后将元素值通过参数 e 返回, 若成功删除则返回 true; 找不到待删除元素则返回 false.

```
template <class E, class K>
bool BSTree<E, K>::Delete(const K &k, E &e)
```

```
{
    BTreeNode<E> *p=root, *q=0;
    while ( p && p->element!=k )
```

```
{
    q=p;
```

```

    if (k < p->element) p = p->lchild;
    else p = p->rchild;
}
if (!p)
{
    cerr << "No element with key k\n";
    return false;
}
c = p->element;
while (p->lchild && p->rchild)
{
    BTreeNode<E> *s = p->rchild; *r = p;
    while (s->lchild)
    {
        r = s;
        s = s->lchild;
    }
    *s = p; p->element = s->element;
    p = s;
    q = r;
}
BTreeNode<E> *c;
if (p->lchild) c = p->lchild;
else c = p->rchild;
if (p = root) root = c;
else if (p == q->lchild) q->lchild = c;
    else q->rchild = c;
delete p;
return true;
}

```

三、算法设计 (10%)

编程实现将两个按元素递增排序的单向循环链表合并成一个单向循环链表, 合并后元素仍递增有序。注意: 不允许再增加新的结点, 相同元素只保留一份。该算法为 SingleList 类的成员函数 Merge, 该函数的作用是将形参 r 代表的单向循环链表合并到当前单向循环链表中, 合并后的结果存于当前单循环链表。

```

template <class T> class SingleList;
template <class T>
class Node{
private:
    T data;
    Node<T> *link;
    friend class SingleList<T>;
};
template <class T>

```

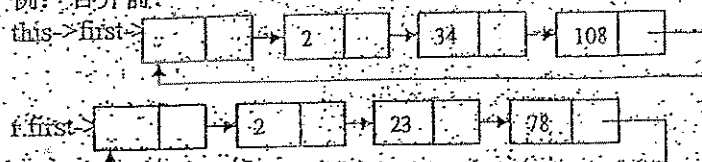


```

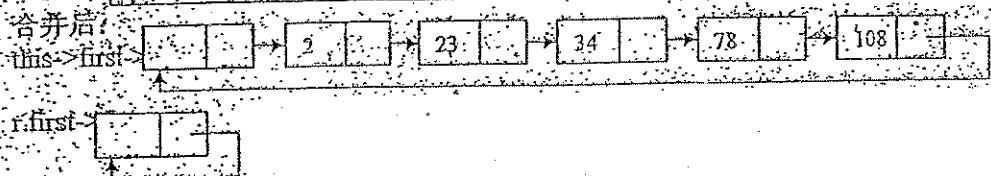
class SingleList:public LinearList<T>
{public:
void Merge(const SingleList<T> &r);
private:
Node<T> *first;
};

```

例：合并前：



合并后：



算法实现：

```

template <class T>
void SingleList<T>::Merge(const SingleList<T> &r)
{

```


数 据 结 构 B 期 末 试 卷

一、解答题: (共 82 分)

1、下列程序段或函数的时间复杂度。(10%)

- (1) $T(n)=O(m \cdot n)$ (2) $T(n)=O(n)$ (3) $T(n)=O(\sqrt{n})$ (4) $T(n)=O(\log_2 n)$

2、(6%) 答: 共 14 种出栈序列,

ABCD 出栈序列可以得到, 操作序列: Push Pop Push Pop Push Pop Push Pop
BDA C 出栈序列不能得到, 因为: D 出栈后, C 处于栈顶, 故 D 后不能得到 A
CBDA 出栈序列可以得到, 操作序列: Push Push Push Pop Pop Push Pop Pop
DBA C 出栈序列不能得到, 因为: D 出栈后, C 处于栈顶, 故 D 后不能得到 B

3、解: 设每个元素占用 k 个字节, 根据已知条件有:

$$\begin{cases} \text{loc}(A[2][3]) = (2 \cdot n + 3) \cdot k + 1000 = 1011 \\ \text{loc}(A[1][1]) = (1 \cdot n + 1) \cdot k + 1000 = 1005 \end{cases} \quad \text{解此方程组得: } \begin{cases} n=4 \\ k=1 \end{cases}$$

所以: $\text{loc}(A[2][0]) = (2 \cdot 4 + 0) \cdot 1 + 1000 = 1008H$

4、(共 13%)

(1) 等效的二叉树

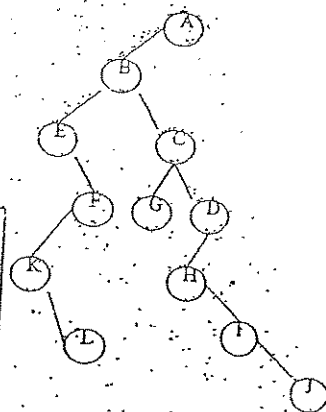
(2) 先序遍历的结点序列: ABEFKLCGDHJ

后序遍历的结点序列: EKLFBCGHIJDA

(3)

下标

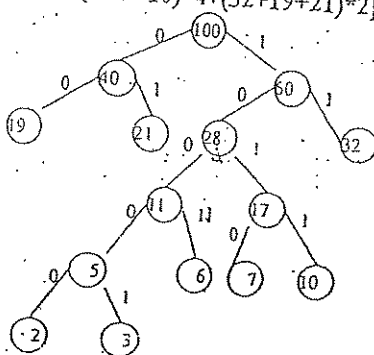
	0	1	2	3	4	5	6	7	8	9	10	11
sibling	-1	6	3	-1	5	-1	8	-1	-1	10	11	-1
element	A	B	E	F	K	L	C	G	D	H	I	J
itag	0	0	1	0	1	1	0	1	0	1	1	1



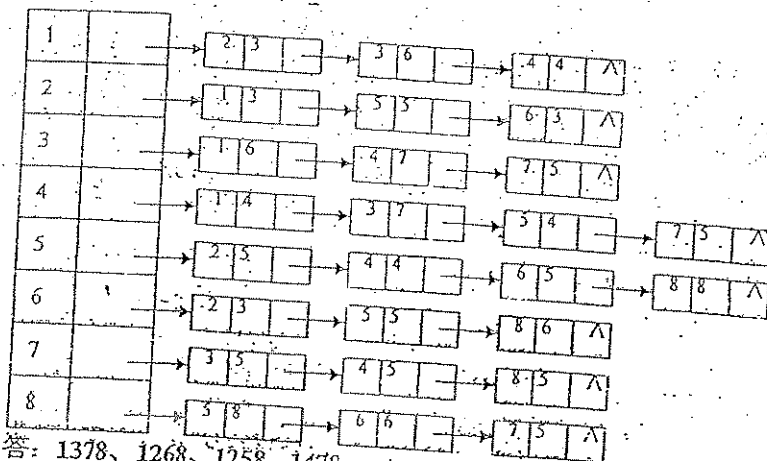
5、(9%) 答: 编码: A: 1010 B: 00 C: 10000 D: 1001
E: 11 F: 10001 G: 01 H: 1011

$$WPL = \sum_{i=1}^8 W_i L_i = [(2+3) \cdot 5 + (6+7+10) \cdot 4 + (32+19+21) \cdot 2] / 100 = 2.61$$

哈夫曼树如右图:



6. (共 13%) (1)

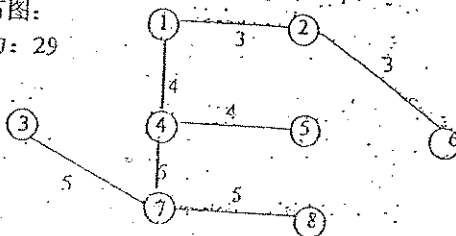


(2) 答: 1378, 1268, 1258, 1478

(3) 答: 从顶点 1 出发的深度优先搜索顶点序列: 12543786
从顶点 1 出发的广度优先搜索顶点序列: 12345678

(4) 答: 最小代价生成树如右图:

最小代价生成树的代价为: 29
该树是唯一的



7. (7%)

(1) 答: 该工程的 AOV 网为:

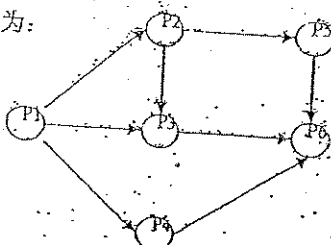
(3) 工程 P 的四种可能的施工顺序为:

(4) P1, P2, P3, P4, P5, P6

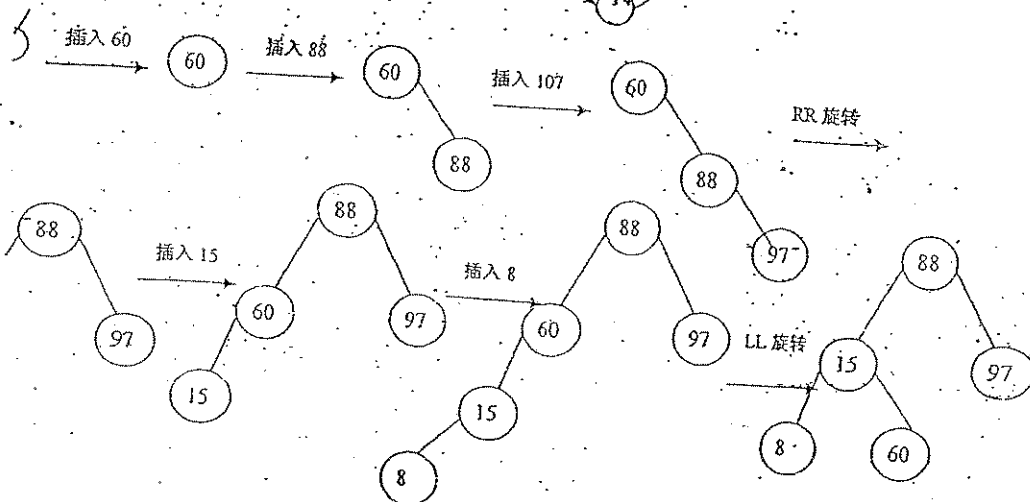
P1, P2, P5, P3, P4, P6

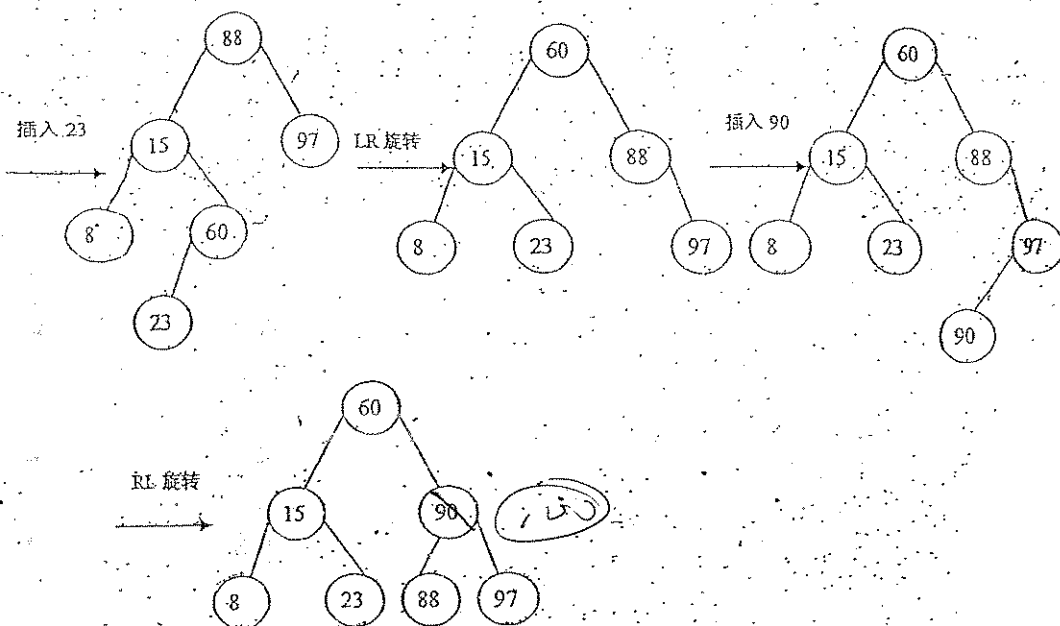
P1, P4, P2, P3, P5, P6

P1, P4, P2, P5, P3, P6



8. (6%)





9. (6%)

i	0	1	2	3	4	5	6	7	8	9	10	11	12
ht[i]	78	14	27	41	56	66	67		30		50	80	24

10. (6%)

冒泡排序法

第一趟: [55, 12, 71, 4, 70, 51,] 98
 第二趟: [12, 55, 4, 70, 51,] 71, 98
 第三趟: [12, 4, 55, 51,] 70, 71, 98
 第四趟: [4, 12, 51,] 55, 70, 71, 98
 第五趟: 4, 12, 51, 55, 70, 71, 98

二路合并排序法

第一趟: [55, 71], [12, 98], [4, 70], [51]
 第二趟: [12, 55, 71, 98], [4, 51, 70]
 第三趟: 4, 12, 51, 55, 70, 71, 98

二、算法填空: (8%)

(1) bool (2) p=p->rchild (3) return false (4) r=s (5) p->element=s->element

(5) t=p->rchild (7) p=root (8) delete p

三、算法设计 (10%)

template <class T>

void SingleList<T>::Merge(SingleList<T> &r)

{ Node<T> *p1=first, *q1=r.first, *p=p1->link, *q=q1->link;

while (q!=r.first)

{ while (p->data<q->data)

{ p1=p; p=p->link;

} if (p->data==q->data)

{ p1=p; p=p->link;

q1->link=q->link;

delete q;

q=q1->link;

}

else

{

q1->link=q->link;

q->link=p;

p1->link=q;

p1=p1->link;

q=q1->link;

}

}

45

《数据结构 B》期末试卷 (B 卷)

院 (系、专业) _____ 班级 _____ 学号 _____ 姓名 _____

得分

一、填空题 (20 分, 请注意: 答案写在下面的表格空白中, 否则无效!)

题号	1	2	3	4	5	6	7	8	9	10
答案			线性							

1. 数据结构从逻辑上分为 线性 和非线性结构。
2. 若长度为 n 的线性表采用顺序存储结构, 插入一个新元素的算法的时间复杂度为 $O(n)$ 。
3. 解决计算机与打印机之间速度不匹配问题, 需要设置一个数据缓冲区。该缓冲区遵守先进先出的原则。缓冲区应采用 队列 结构。
4. 一个 5×5 的对称矩阵, 压缩后的存储容量为 15。
5. 已知一棵二叉树有 24 个结点, 则其二叉链表结构中共有 25 个空链域。
6. 对有序表 (21, 22, 28, 31, 35, 40, 45, 56, 70) 进行对半搜索, 第一次比较时与关键字 35 进行比较。
7. 建立 10, 50, 30, 90, 40 为输入时的二叉搜索树, 其高度为 4。
8. 具有相同散列函数值的关键字值, 对该散列函数而言称为 同义词。
9. 10 个顶点的连通图至少有 9 条边。
10. 两路合并排序的空间复杂度为 $O(n \log_2 n)$ 。

得分

二、选择题 (20 分, 请注意: 答案写在下面的表格空白中, 否则无效!)

题号	1	2	3	4	5	6	7	8	9	10
答案										

1. 设使用某算法对 n 个元素进行处理, 所需的时间是 $T(n) = 100n \log_2 n + 200n + 2000$, 则该算法的渐近时间复杂度为 D。
A. $O(1)$ B. $O(n)$ C. $O(200n)$ D. $O(n \log_2 n)$
2. 单链表中 p 结点的后面插入新结点 s , 应执行语句序列 C。
A. $p \rightarrow next = s \rightarrow next; s = p \rightarrow next;$ B. $p \rightarrow next = s \rightarrow next; s \rightarrow next = p \rightarrow next;$
C. $s \rightarrow next = p \rightarrow next; p \rightarrow next = s;$ D. $p \rightarrow next = s; s \rightarrow next = p \rightarrow next;$
3. 在初始为空的堆栈中依次插入元素 f, e, d, b, b, a 以后, 连续进行了 3 次删除操作, 此时栈顶元素是 B。
A. c B. d C. b D. e
4. 有一个 10 阶对称矩阵 A , 采用压缩存储方式 (以行序为主序存储, 且 $A[0][0] = 0$), 则 $A[8][5]$ 的地址是 C。
A. 39 B. 40 C. 41 D. 42
5. 将一棵有 100 个结点的完全二叉树从根这一层开始, 每一层上从左到右依次对结点进行编号, 根结点的编号为 0, 则编号为 50 的结点的右孩子编号为 D。
A. 不存在 B. 100 C. 101 D. 102
6. 采用二分搜索法在线性表上查找的条件是线性表有序且 B。
A. 使用链式存储结构 B. 使用顺序存储结构

C. 使用索引结构

D. 使用散列结构

7. 已知一棵由关键字集合 {18, 43, 27, 77, 44, 36, 39} 所构造的二叉搜索树, 对该树进行中序遍历得到的结果为

A. 树形未定, 无法确定

B. 18, 43, 27, 77, 44, 36, 39

C. 18, 27, 36, 39, 43, 44, 77

D. 77, 44, 43, 39, 36, 27, 18

8. 采用线性探查法解决冲突, 容易产生

A. 二级聚集

B. 均匀聚集

C. 基本聚集

D. 三级聚集

9. 具有 10 顶点的无向图最多有 条边。

A. 0

B. 9

C. 10

D. 45

10. 对以下关键字序列用快速排序进行排序, 哪一种情况速度最慢?

A. {3, 7, 15, 19, 21, 23, 28}

B. {23, 21, 28, 15, 19, 3, 7}

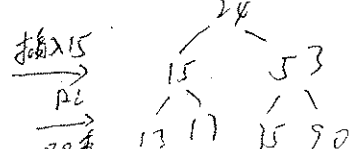
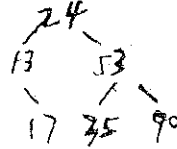
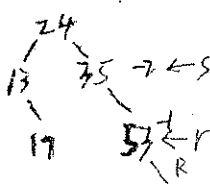
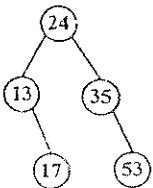
C. {19, 7, 15, 28, 23, 21, 3}

D. {19, 23, 3, 15, 7, 21, 28}

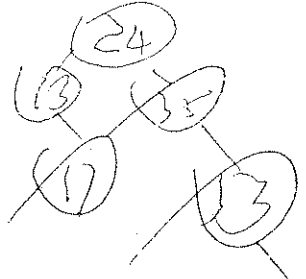
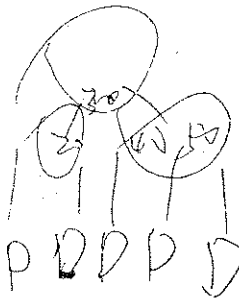
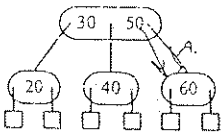
得分

三、简答题 (40 分)

1. 在 AVL 树中插入 90, 请写出树的变化过程。



2. 3 阶 B 树, 画出删除 60 后得 B 树。

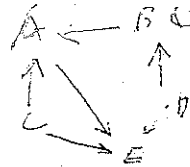


3. 如下的散列表为空，设散列函数为 $h(\text{key}) = \text{key} \% 11$ ，处理冲突的方法为二次探查法。试画出在下列散列表上插入关键字 5, 37, 14, 26 后散列表的情况。

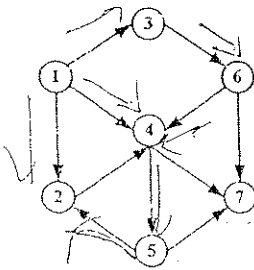
0	1	2	3	4	5	6	7	8	9	10

4. 设某项工程由下图所示的工序组成。若各工序以流水方式进行（即串行进行）。给出该工程的全部合理的工程流程。（注：紧前工序是指若有工序 A 和 B，工序 B 必须在工序 A 完工后才能开工，则工序 A 是工序 B 的紧前工序。）

工序	紧前工序
A	B, C
B	D
C	无
D	无
E	A, C, D



5. 写出下图中顶点 1 到顶点 2 的所有简单路径。



① 1 2
② 1 4 5 2 ✓
③ 1 3 6 4 5 2 ✓

得分

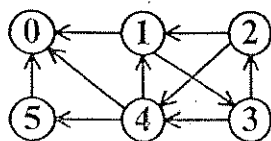
四、程序阅读题（10 分）

1. 请阅读下面的代码，回答有关问题：

```
template<class T>
int LinkedGraph<T>::fun(int v){
    if ((v<0)||((v>n-1))) return -1;
    int out = 0;
    ENode<T> *p = a[v];
    while (p){
        out++;
        p = p->nextarc;
    }
    return out;
}
```

问题：

- （1）请问以上程序代码的功能是什么？
- （2）已知有向图如下图所示，针对该图运行 $\text{fun}(1)$ ，函数返回值是多少？



解答:

(1)

(2)

得分

五、编程题 (10 分)

1. 设二叉树以二叉链表存储, 编写完成删除一棵二叉树, 并释放所有的结点空间的算法。

```
template <class T>
bool BinaryTree<T>::Clear()
{

}

template <class T>
bool BinaryTree<T>::Clear(BTNode<T>* t)
{
```

自觉遵守考试规则, 诚信考试, 绝不作弊

《数据结构 B》期末试卷 (B 卷)

院 (系、专业) _____ 班级 _____ 学号 _____ 姓名 _____

一、填空题 (20 分, 共 10 题)

数据一般分为数值数据和 非数值数据

存储结构上讲, 线性表的表示可分为 顺序 和 链接 两类。

设顺序表长度为 2012, 则删除第 2000 ($i=0, 1, \dots, n$) 个元素需要移动 12 个元素。

后缀表达式 $20 \ 10 \ 5 \ * \ + \ 49 \ 7 \ / \ -$ 的值为 63。

一棵二叉树中, 若叶结点的个数为 1643, 则度为 2 的结点个数为 1642。
5 阶 B 树的高度为 2 时, 树中元素个数最少为 1。

7. 若有向图的拓扑排序不能输出所有的顶点, 则该有向图存在 回路。

8. 有 100 个结点的完全二叉树的高度是 10。

9. 10 个顶点的无向图中有 4 条边, 若采用邻接矩阵表示, 则矩阵中的非 0 元素个数是 8 个。

10. 在数据表有序时, 快速排序算法的时间复杂度是 $O(n \log n)$ 。

二、选择题 (20 分, 共 10 题)

1. 若用单链表来表示 队列, 则选用 C 最好。

A. 只带尾指针的非循环链表

B. 只带尾指针的循环链表

C. 只带头指针的非循环链表

D. 只带头指针的循环链表

2. 设有一个二维数组 $A[m][n]$ 按行优先顺序存储, 假设 $A[0][0]$ 的地址是 644, $A[2][2]$ 的地址是 676, 每个元素占 1 个单元, 则 $A[4][5]$ 的地址是 C。

A. 672

B. 626

C. 709

D. 724

3. 在对非空二叉树进行中序遍历的序列中, 根结点右边 D。

A. 只有左子树上的部分结点

B. 只有左子树上的全部结点

C. 只有右子树上的部分结点

D. 只有右子树上的全部结点

4. 具有 10 个顶点的连通图的深度优先搜索生成树, 其边的数目为 A。

A. 9

B. 10

C. 11

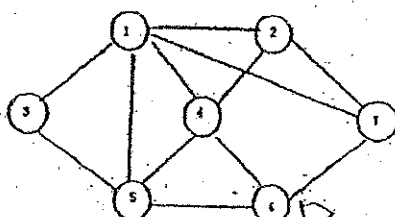
D. 8

下列说法错误的是 C。

A. 已知先序和中序遍历序列能唯一确定一棵二叉树。

B. 已知后序和中序遍历序列能唯一确定一棵二叉树。

- C. 已知先序和后序遍历序列能唯一确定一棵二叉树
 D. 已知先序遍历序列能唯一确定一棵(二叉搜索树) 若以中序遍历排列, 则呈递增排列
6. 下列排序算法中, 一趟排序后一定能确定某个元素最终位置的算法是 B.
 A. 直接插入排序 B. 快速排序 C. 两路归并排序 D. 以上均不对
7. 散列表采用线性探查法解决冲突会出现 C 现象。
 A. 二次聚集 B. 探查失败 C. 假溢出 D. 基本聚集 改进方法: 二次探查法
8. 下图给出由 7 个顶点组成的无向图。从顶点 1 出发, 对它进行广度优先遍历得到的顶点序列是 B.
 A. 1534267 B. 1726453 C. 1354276 D. 1247653



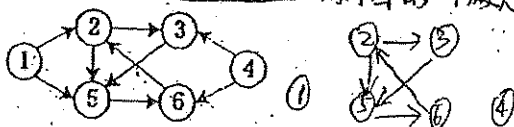
9. 一个 n 个顶点的有向完全图, 其边的个数为 B.
 A. n B. $n(n-1)$ C. n^2 D. $n(n-1)/2$
10. 适用于折半查找的表的存储方式及元素排列要求为 D.
 A. 链接方式存储, 元素无序 B. 链接方式存储, 元素有序
 C. 顺序方式存储, 元素无序 D. 顺序方式存储, 元素有序

(假溢出现象)

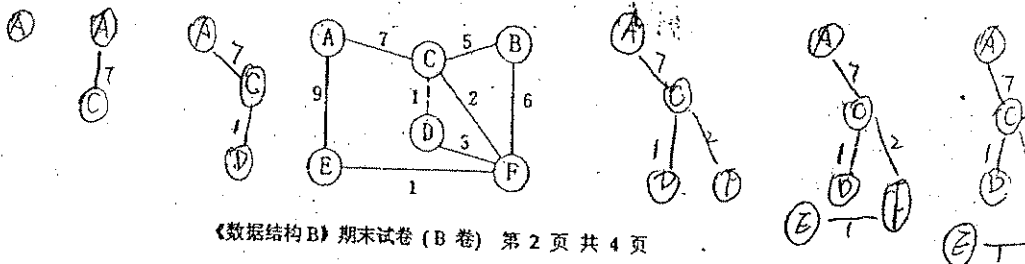
对半搜索 = 适用对象: 有序顺序表

三、简答题 (40 分, 共 5 题)

1. 设有向图如下图所示, 请画出该图所有的强连通分量。= 有向图的一个极大强连通子图。



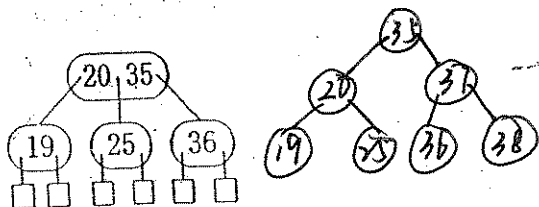
2. 使用普里姆 (Prim) 算法以 A 为源点, 构造下图的最小代价生成树, 要求画出构造过程中各步的结果。



$$\lceil 3/2 \rceil = 2$$

max=2

- ③ 下图是一个 3 阶 B-树，请画出插入 37 之后的 B-树，再画出在此基础上插入 38 后的 B 树。



- ④ 设字符集合 $S=\{A, B, C, D, E\}$ ，各字符的使用频率为 $W=\{5, 7, 8, 9, 10\}$ 。

- (1) 画出哈夫曼树；(构建新树时，新树根的左子树根的权值小于等于右子树根的权值)
(2) 求该哈夫曼树的带权路径长度。

$$(2) 10 \times 5 + 9 \times 7 + 8 \times 8 + 7 \times 9 + 5 \times 10 = 177$$

5. 表长为 11 的散列表采用双散列法解决冲突，散列函数 $h_1(key)=key \% 11$ ， $h_2(key)=key \% 9 + 1$ 。已知散列表目前如下表所示，请问在该散列表中再依次插入关键字 70、80、19、41 后，散列表的情况如何？

下标	0	1	2	3	4	5	6	7	8	9	10
关键字		12		25	70						

key	70	80	19	41
$h_1(key)$	4	3	8	8
$h_2(key)$	8	9	2	6

四、程序填空题 (10 分，共 1 题)

以下程序是对半搜索的迭代算法，请填写完整。

```

template <class T>
ResultCode ListSet<T>::Search(T &x) const
{
    int m, low=0, high= (1) ;
    while ( (2) )
    {
        (3)
        if (x<l[m]) high=m-1;
        else
        {
            if (x>l[m]) (4) ;
            else
            {
                x=l[m]; (5) ;
            }
        }
    }
    return NotPresent;
}

```

五、编程题 (10 分, 共 1 题)

1. 在单链表类 `SingleList` 中增加一个成员函数 `Reverse()`, 其功能是将头指针 `first` 指向的单链表逆置。若原链表为空, 则返回 `false`, 否则返回 `true`。模板结点类 `Node` 的定义如下:

```
template <class T>
class Node
{
    T data;
    Node<T> *link;
    friend class SingleList<T>
};
```

函数原型为:

```
template <class T>
bool SingleList<T>::Reverse()
```

自觉遵守考场规则, 诚信考试, 绝不作弊

南京邮电大学 2008/2009 学年第二学期

《数据结构 B》期末 试卷 (B)

院(系) _____ 班级 _____ 学号 _____ 姓名 _____

题号	一	二	三	四	总分
得分					

请考生注意:

1. 答案请写在答题纸上, 写在试卷上一律无效!
2. 考试完毕, 请将答题纸和试卷交给监考老师, 不得带出考场!

得分

一、填空题 (每小题 2 分, 共 20 分)

1. 根据数据结构中数据元素之间关系的不同特征, 可划分为四种基本逻辑结构: 集合结构、线性结构、树形结构和图结构。
2. 线性表的存储结构有顺序存储和链接存储两种, 其中可以随机存取的是 顺序 存储。
3. 38 个结点的二叉树高度至少为 6。 $2^{h-1} = n$
4. 在一棵 15 阶的 B-树上, 每个结点所包含的关键字数目最多为 14 个。
5. AOV 网络中的领先关系是一种拟序关系, 它具有 传递 性和反自反性。
6. 一个待散列存储的元素序列 (25, 63, 50, 42, 32, 90) 中, 若选用 $h(key) = key \% 9$ 作为散列函数, 则元素 18 的同义词共有 2 个。
7. 10 个顶点的有向图中, 最多有 90 条边。 10×9
8. 堆栈和队列都是特殊的线性结构, 堆栈只在栈顶进行插入操作, 队列只在 队尾 进行插入操作。
9. 二叉搜索树的 中序 遍历序列是一个按关键字递增排列的有序序列。
10. 后缀表达式 $5 \cdot 3 \cdot 2 * 3 + 3 / +$ 的值为 8。

自觉遵守考场纪律, 诚信考试, 绝不作弊

55

得分

二、单项选择题 (每小题 2 分, 共 20 分)

1. 分析下面一段程序

```
for (i=1; i<n; i++)
{
    y=y+1;
    for (j=0; j<=(2*n); j++)
        x=x+1;
}
```

该段程序的时间复杂度为 (B)。

- A. $O(\log_2 n)$ B. $O(n^2)$ C. $O(n)$ D. $O(n \log_2 n)$

2. 设线性表 $L=(a_0, a_1, \dots, a_{n-1})$, 下列说法正确的是 (D)。

- A. 每个元素都有一个直接前驱和直接后继
B. 线性表中至少有一个元素 \times 可有空表
C. 表中元素的排列顺序必须是由小到大或由大到小 \times
D. 除第一个元素和最后一个元素外, 其余每个元素都有一个且仅有一个直接前驱和直接后继。

3. 堆栈中, 栈底至栈顶依次存放元素 A、B、C、D, 在第 5 个元素 E 入栈前, 栈中元素可以出栈, 则出栈序列可能的是 (C)。

- A. ABCDE \times B. DBCEA C. DCBEA D. CDABE \times

4. 已知顺序表中每个元素占 2 个存储单元, 第一个元素 a_0 在内存中的存储地址是 100, 则表中元素 a_5 在内存中的存储地址为 (B)。

- A. 112 B. 110 C. 120 D. 140

5. 设森林中有 3 棵树, 其中第 1、第 2 和第 3 棵树的结点个数分别为 n_1 、 n_2 、 n_3 , 则与森林对应的二叉树中根结点的右子树上的结点个数是 (D)。

- A. n_1 B. $n_1 + n_2$ C. n_3 D. $n_2 + n_3$

6. 在一非空二叉树的中序遍历序列中, 根结点的右边 (A)。

- A. 只有右子树上的所有结点
B. 只有右子树上的部分结点
C. 只有左子树上的部分结点
D. 只有左子树上的所有结点

7. 二叉搜索树中, 最小元素的左子树 (), 它的右子树 ()。 A

- A. 一定为空, 不一定为空
B. 不一定为空, 一定不为空
C. 一定不为空, 不一定为空
D. 不一定为空, 不一定为空

直接插入排序
 $\text{while}(\text{使用})$
 $\text{temp} < A[j]$
 $\text{temp} <= A[j]$

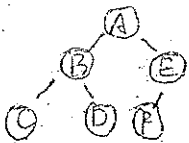
8. 有向图 G 用邻接矩阵 a 存储, 则顶点 i 的入度等于 a 中 (B)。
- A. 第 i 行非 0 的元素个数 B. 第 i 列非 0 的元素个数
 C. 第 i 行和第 i 列非 0 元素个数之和 D. 第 i 行和第 i 列非 0 元素个数之积
9. AOV 网中存在两个顶点 i 和 j, 若 i 领先 j, 选项 (C) 肯定是错误的。
- A. i, j 之间存在一条有向边 B. AOV 网的拓扑序列中 i 在 j 之前
 C. j 到 i 存在一条有向路径 D. i 到 j 存在一条有向路径
10. 快速排序在最坏情况下, 渐近时间复杂度为 (D)。
- A. $O(n)$ B. $O(\log_2 n)$ C. $O(n \log_2 n)$ D. $O(n^2)$

得分

三、简答题 (每小题 8 分, 共 40 分)

1. 设一个散列表的长度 $M=11$, 其下标从 0 到 10, 散列函数是 $h(\text{key}) = \text{key} \% 11$ 。在空的散列表中依次插入关键字 14、25、36、47。分别采用线性探查法和二次探查法解决冲突, 请写出建立起来的散列表。

下标	0	1	2	3	4	5	6	7	8	9	10
线性探查法				14	25	36	47				
二次探查法											



先: ABCDEF
 中: CBDAFE

2. 图 1 是一棵完全二叉树的顺序存储结构表示, 请完成:

- (1). 画出对应的二叉树;
 (2). 写出该二叉树的先序遍历序列和中序遍历序列。

结点	A	B	E	C	D	F
下标	0	1	2	3	4	5

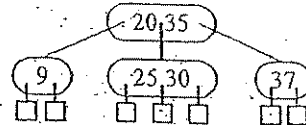


图 1

图 2

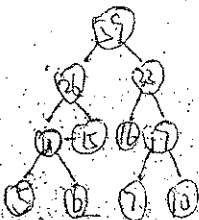
3. 图 2 是一棵 3 阶 B-树, 请完成:

- (1) 分别画出向该树中依次插入关键字 10, 15 后相应的 B-树;
 (2) 分别画出从原 B-树中 (没有插入关键字 10, 15) 依次删除关键字 20, 35 后的 B-树。



4. 设字符集 $D=\{A, B, C, D, E, F\}$, 各字符使用频度 $W=\{5, 6, 7, 10, 15, 16\}$, 请完成:

- (1) 画出以 W 为权值构建的哈夫曼树; (哈夫曼树中结点用权值表示。构建新二叉树时, 新二叉树根的左子树根的权值小于等于右子树根的权值)
 (2) 求该哈夫曼树的加权路径长度。



$$(5+6) \times 3 + 15 \times 2 + 16 \times 2 + (7+6) \times 3 = 146$$

33 30 22 51

直接插入排序

最小时(或同时)满足条件
是 $temp < A[j-1]$ 稳定
是 $temp < A[j-1]$ 不稳定

时间复杂度

对于搜索算法成功时, 数据值之间比较次数不超过 $2 \log n + 1$. 对于初始搜索:
需作 $2 \log n$ 或 $2 \log n + 1$ 次比较.
对于初始搜索失败时, 平均时间复杂度为 $O(\log n)$.
[查表时间: $O(\log n)$]
5. 无向连通加权图如图3所示, 请完成:
(1) 使用普里姆算法以0为源点构建该图的最小代价生成树, 要求画出每一步的结果.
(2) 计算该生成树的代价.

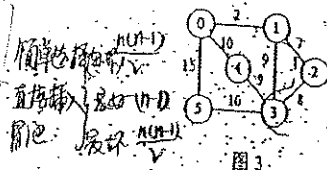
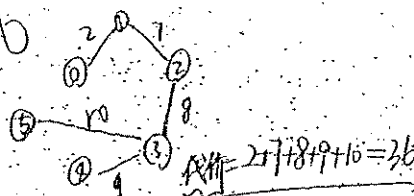


图3



最小代价生成树
由上一步骤得到

四、编程题 (每小题10分, 共20分)

1. 在SingleList中增加一个成员函数, 将单链表中数据域值最小的那个结点放到单链表的最前面.

得分

```
template <class T>
void SingleList<T>::MoveMinToFirst()
```

```
Node<T> * p = head; int min = 0;
while (p) {
    if (p->data < min) {
        min = p->data;
        p = p->next;
    }
}
```

2. 在二叉树类BinaryTree中, 编写算法实现交换所有结点左右孩子.

```
template <class T>
void BinaryTree<T>::Exchange(BTNode<T> * t) {
```

```
template <class T>
```

```
void BinaryTree<T>::Exchange(BTNode<T> * t)
```

```
{
    BTNode<T> * temp;
    if (t) {
        change(t->lchild);
        change(t->rchild);
        temp = t->lchild;
        t->lchild = t->rchild;
        t->rchild = temp;
    }
}
```

```
Node<T> * r = p->next;
p->next = r->next;
head->next = r;
r->next = head->next;
p->next
```

10

37



35 后的

0, 15, 16}

新二叉树时

(数据结构B) 试卷(B) 第4页共4页

自觉遵守考试规则 诚信考试 绝不作弊

南京邮电大学 2008/2009 学年第一学期

《数据结构 B》期末试卷 (B 卷)

学院 _____ 班级 _____ 学号 _____ 姓名 _____

注意事项:

1. 试卷和答题纸上都必须填写专业、班级、学号和姓名;
2. 全部试题解答都必须写在答题纸上, 写在试卷上无效;
3. 答题必须使用钢笔、圆珠笔或签字笔, 铅笔答题无效;
4. 考试完毕, 请将试卷、答题纸和草稿纸均交给监考教师, 不得带出考场。

一、判断题 (每小题 2 分, 共 10 分) (请回答“√”或“×”)

1. 设有元素入栈的次序为: a, b, c , 则不可能的出栈次序是 c, a, b . (√)
2. 访问线性表的第 i 个元素的时间同 i 的大小有关. (√)
3. 若一棵二叉树根的右子树为空树, 则其对应的森林中只有一棵树. (√)
4. 在采用线性探查法解决冲突建立散列表时, 会发生基本聚集现象. (×)
5. 在具有 n 个顶点的有向图中, 顶点的度最大可以是 $n-1$. (×)

二、选择题 (每小题 2 分, 共 10 分)

1. 设有顺序表 $L = (a_0, a_1, \dots, a_{n-1})$, 并假定在任何一个元素之后以及在第一个元素之前插入的概率相同, 则进行一次插入操作平均移动元素的次数是 (B).
A. n B. $n/2$ C. $n(n+1)/2$ D. n^2
2. 设以 S 和 X 分别表示入栈和出栈操作, 假定栈的初态和终态均为空, 则下面操作序列中 (A) 是合法的。
A. $SXSXSXSX$ B. $SXSXSXSX$ C. $SSXSXSXSX$ D. $XSSXSXSX$
3. 设 a, x 和 y 是二叉树 B 中的三个结点, x 是 a 的左孩子, y 是 x 的右孩子, T 是与 B 对应的树; 在 T 中, y 是 a 的 (A).
A. 孩子 B. 兄弟 C. 双亲 D. 后裔但非孩子
4. 设有关键字值序列 $(12, 22, 32, 42, 52, 62, 72, 82)$, 现采用对半搜索方法查找关键字 $k=62$ 的元素, 则在查找过程中, x 将与序列中 () 关键字值进行比较。
A. 52, 62 B. 42, 52, 62 C. 42, 62 D. 52, 42, 62
5. 将序列中第一个元素作为一个有序序列, 然后将剩下的 $n-1$ 个元素按关键字值大小依次插入该有序序列, 每插入一个元素后依然保持该序列有序, 这种排序算法称为 ()。
A. 简单选择排序 B. 两路合并排序 C. 直接插入排序 D. 快速排序

$6 \quad n$
 $x \quad n$
 $1 \quad n$
 $0 \quad n$

$[(7 \times 2) + 3] / 3 + 7$
 $n \times (n-1)$

$3 \times 2 = 6$
 $7 \times 4 + 3$
 $3 \times 2 = 6$
 $7 \times 3 + 1$
 $7 \times 3 + 1$

填空题 (每空 2 分, 共 10 分)

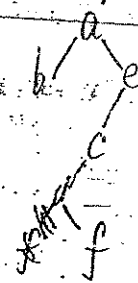
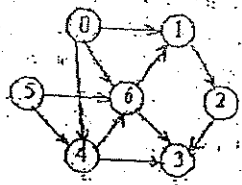
设操作数是 1 位十进制数, 则后缀表达式 $7 \ 3 \ 2 \ * \ 3 \ + \ 3 \ / \ +$ 的值为 10。
 在双向链表中指定的结点之前插入一个新结点需修改的指针数是 2 个。
 设有二维数组 $A[5][7]$ 以行主序顺序存储, 设第一个元素的起始地址为 102, 每个元素占两个单元, 则元素 $A[3][3]$ 的存储地址是 158。
 一棵二叉树中, 已知度为 3 的结点数等于度为 2 的结点数, 且树中叶结点的数目为 10, 则度为 2 的结点数目为 10。
 具有 n 个顶点的无向图最多有 $n(n-1)/2$ 条边。

简答题 (每题 4 分, 共 16 分)

画出如图 1 所示的稀疏矩阵的行三元组表表示, 计算对该稀疏矩阵执行矩阵快速转置, 数组 $k[]$ 的值。
 对一棵二叉树的先序和中序遍历的结点次序分别为 (a, b, e, c, d, f) 和 (a, d, f, c, e) , 则画出该二叉树, 并写出该二叉树作后序遍历的结点次序。
 画出图 2 所示的有向图的邻接矩阵存储结构。

row col val

0	1	2
2	0	4
2	2	1
3	2	5
4	3	7



$A[i][j] = [x \times n + y] \times 2 + 10$
 $A[1][2]$
 $(3 \times 7 + 3) \times 2 + 10 = 48 + 10 = 58$

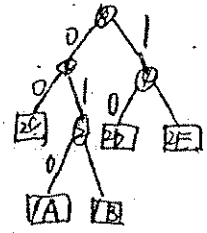
图 1

图 2

解答题 (每题 8 分, 共 32 分)

设有字符集 $S = \{A, B, C, D, E\}$, 已经构造哈夫曼树, 并得到了字符集 S 的各字符的编码为:

字符	A	B	C	D	E
编码	010	011	00	10	11



画出该哈夫曼树形。
 现有码文 010011000101001111, 请译出原电文。
A B C A D A B E

有向图如图 2 所示,

画出从顶点 0 开始的广度优先搜索该图的生成树 (或森林)。设顶点 0 的三个邻接点在邻接表中的次序为 1, 4, 6。

(2) 给出所有可能的拓扑序列，什么情况下拓扑排序算法不能输出图中全部顶点。

3. 在图 3 所示的 4 阶 B-树上，

(1) 画出插入 70 后的 B-树；

(2) 画出删除 22 后的 B-树（删除运算仍在图 3 所示的原 B-树上执行）。

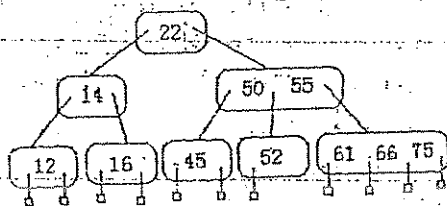


图 3

4. 使用两路合并排序算法对元素序列 (23, 43, 36, 30, 20, 54, 76, 28) 进行排序。

(1) 写出每趟排序后的结果；

(2) 给出两路合并排序的最坏情况的渐近时间复杂度；

(3) 给出除存储原元素序列的空间外，两路合并排序所需的附加空间复杂度。

六、算法填空题（每空 2 分，共 8 分）

1. 补充完整下列在带头结点的单链表中插入新元素的函数 insert(i, x)。x 插入在第 i 个元素之后。若 i=-1，则将新元素 x 插在最前面。若插入成功，则返回 true，否则返回 false。

```
template<class T>
bool HeaderList<T>::Insert(int i, T x)
{
    if (i<-1 || i>n-1) {
        cout<< "Out of Bounds"<<endl; return false;
    }
    Node<T> *p= ① ;
    for (int j=0; j<=i; j++) p=p->link;
    Node<T> *q=new Node<T>; q->element=x;
    ② ; p->link=q;
    n++; return true;
}
```

2. 补充完整下列二叉树后序遍历的递归函数

```
template<class T>
void BinaryTree<T>::PostOrder(void (*Visit)(T&x), BTreeNode*&t)
```

```

if (t) {
    PostOrder ③ ;
    PostOrder ④ ;
    Visit (t->element);
}

```

七、算法阅读题 (6 分)

设有二叉树类上的递归函数 X 如下,

```

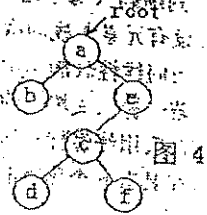
template <class T>
BTNode<T>* BinaryTree<T>::X(BTNode<T>* t)
{
    if (!t) return NULL;
    BTNode<T>* q=new BTNode<T>(t->element);
    q->LChild=X(t->LChild); q->RChild=X(t->RChild);
    return q;
}

```

(1) 说明以上程序的功能。

(2) 设二叉树如图 4 所示, 在调用函数中, 以语句

BTNode<T>* p=X(root); 调用函数 X 的运行结果是什么?



八、算法设计题 (8 分)

试编写二叉树类上定义的递归函数 Y, 该函数判断一棵二叉树中是否扩充二叉树。扩充二叉树也称 2-树。若该二叉树是扩充二叉树, 则函数返回 true, 否则返回 false。

设二叉树采用二叉链表存储, 每个结点的类型为 BTNode<T>, 结点有三个域: element, LChild 和 RChild。参数 t 是指向一棵二叉树根的指针。函数原型如下:

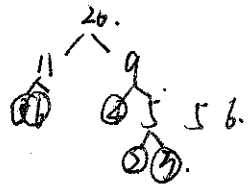
```

template <class T>
bool BinaryTree<T>::Y(BTNode<T>* t)

```

试卷结束

数据结构试卷 (六)



一、选择题(30 分)

1. 设一组权值集合 $W=\{2, 3, 4, 5, 6\}$, 则由该权值集合构造的哈夫曼树中带权路径长度之和为 ().
 (A) 20 (B) 30 (C) 40 (D) 45
 $(5+6+4) \times 2 + 5 \times 3 = 30 + 15 = 45$
2. 执行一趟快速排序能够得到的序列是 (). A.
 (A) [41, 12, 34, 45, 27] [59] [72, 63]
 (B) [45, 34, 12, 41] [59] [72, 63, 27]
 (C) [63, 12, 34, 45, 27] [59] [41, 72]
 (D) [12, 27, 45, 41] [59] [34, 63, 72]
3. 设一条单链表的头指针变量为 head 且该链表没有头结点, 则其判空条件是 (A).
 (A) head==0 (B) head->next==0
 (C) head->next==head (D) head!=0
4. 时间复杂度不受数据初始状态影响而恒为 $O(n \log_2 n)$ 的是 (A).
 (A) 堆排序 (B) 冒泡排序 (C) 希尔排序 (D) 快速排序
5. 设二叉树的先序遍历序列和后序遍历序列正好相反, 则该二叉树满足的条件是 (D).
 (A) 空或只有一个结点 (B) 高度等于其结点数
 (C) 任一结点无左孩子 (D) 任一结点无右孩子
6. 一趟排序结束后不一定能够选出一个元素放在其最终位置上的是 (). D.
 (A) 堆排序 (B) 冒泡排序 (C) 快速排序 (D) 希尔排序
7. 设某棵三叉树中有 40 个结点, 则该三叉树的最小高度为 (B).
 (A) 3 (B) 4 (C) 5 (D) 6
8. 顺序查找不论在顺序线性表中还是在链式线性表中的时间复杂度为 (A).
 (A) $O(n)$ (B) $O(n^2)$ (C) $O(n^{1/2})$ (D) $O(\log_2 n)$
9. 二路归并排序的时间复杂度为 (C).
 (A) $O(n)$ (B) $O(n^2)$ (C) $O(n \log_2 n)$ (D) $O(\log_2 n)$
10. 深度为 k 的完全二叉树中最少有 (B) 个结点.
 (A) $2^{k-1}-1$ (B) 2^{k-1} (C) $2^{k-1}+1$ (D) 2^k-1
11. 设指针变量 front 表示链式队列的队头指针, 指针变量 rear 表示链式队列的队尾指针, 指针变量 s 指向将要入队列的结点 X, 则入队列的操作序列为 (C).
 (A) front->next=s; front=s; (B) s->next=rear; rear=s;
 (C) rear->next=s; rear=s; (D) s->next=front; front=s;
12. 设某无向图中有 n 个顶点 e 条边, 则建立该图邻接表的时间复杂度为 ().
 (A) $O(n+e)$ (B) $O(n^2)$ (C) $O(ne)$ (D) $O(n^3)$
13. 设某哈夫曼树中有 199 个结点, 则该哈夫曼树中有 (B) 个叶子结点.
 (A) 99 (B) 100 (C) 101 (D) 102
14. 设二叉排序树上有 n 个结点, 则在二叉排序树上查找结点的平均时间复杂度为 (D).
 (A) $O(n)$ (B) $O(n^2)$ (C) $O(n \log_2 n)$ (D) $O(\log_2 n)$
15. 设用邻接矩阵 A 表示有向图 G 的存储结构, 则有向图 G 中顶点 i 的入度为 (B).
 (A) 第 i 行非 0 元素的个数之和 (B) 第 i 列非 0 元素的个数之和
 (C) 第 i 行 0 元素的个数之和 (D) 第 i 列 0 元素的个数之和

二、判断题(20 分)

1. 调用一次深度优先遍历可以访问到图中的所有顶点。 (X)
2. 分块查找的平均查找长度不仅与索引表的长度有关, 而且与块的长度有关。 ()
3. 冒泡排序在初始关键字序列为逆序的情况下执行的交换次数最多。 (✓)
4. 满二叉树一定是完全二叉树, 完全二叉树不一定是满二叉树。 (✓)
5. 设一棵二叉树的先序序列和后序序列, 则能够唯一确定出该二叉树的形状。 (X)
6. 层次遍历初始堆可以得到一个有序的序列。 (X)
7. 设一棵树 T 可以转化成二叉树 BT, 则二叉树 BT 中一定没有右子树。 (✓)
8. 线性表的顺序存储结构比链式存储结构更好。 (X)

9. 中序遍历二叉排序树可以得到一个有序的序列。(✓)
10. 快速排序是排序算法中平均性能最好的一种排序。(✓)

三、填空题(30分)

1. for($i=1, t=1, s=0; i \leq n; i++$) { $t=t*i; s=s+t;$ } 的时间复杂度为 $O(n)$ 。
2. 设指针变量 p 指向单链表中结点 A, 指针变量 s 指向被插入的新结点 X, 则进行插入操作的语句序列为 $s \rightarrow next = p \rightarrow next; p \rightarrow next = s$ (设结点的指针域为 next)。
3. 设有向图 G 的二元组形式表示为 $G = (D, R)$, $D = \{1, 2, 3, 4, 5\}$, $R = \{r\}$, $r = \{ \langle 1, 2 \rangle, \langle 2, 4 \rangle, \langle 4, 5 \rangle, \langle 1, 3 \rangle, \langle 3, 2 \rangle, \langle 3, 5 \rangle \}$, 则给出该图的一种拓扑排序序列 $1, 3, 2, 4, 5$ 。
4. 设图 G 中有 n 个顶点, 则该无向图中每个顶点的度数最多是 $n-1$ 。
5. 设二叉树中度数为 0 的结点数为 50, 度数为 1 的结点数为 30, 则该二叉树中总共有 129 个结点数。
6. 设 F 和 R 分别表示顺序循环队列的头指针和尾指针, 则判断该循环队列为空的条件为 $F == R$ 。
7. 设二叉树中结点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 $p \rightarrow lchild == 0 \text{ \& } p \rightarrow rchild == 0$ 。
8. 简单选择排序和直接插入排序算法的平均时间复杂度为 $O(n^2)$ 。
9. 快速排序算法的空间复杂度平均情况下为 $O(\log_2 n)$, 最坏的情况下为 $O(n)$ 。
10. 散列表中解决冲突的两种方法是 $开放定址法$ 和 $链地址法$ 。

四、算法设计题(20分)

1. 设计在顺序有序表中实现二分查找的算法。
2. 设计判断二叉树是否为二叉排序树的算法。
3. 在链式存储结构上设计直接插入排序算法。

数据结构试卷(六) 参考答案

一、选择题

1. D 2. A 3. A 4. A 5. D
6. D 7. B 8. A 9. C 10. B
11. C 12. A 13. E 14. D 15. B

二、判断题

1. 错 2. 对 3. 对 4. 对 5. 错
6. 错 7. 对 8. 错 9. 对 10. 对

三、填空题

1. 1. $O(n)$
2. 2. $s \rightarrow next = p \rightarrow next; p \rightarrow next = s$
3. 3. (1, 3, 2, 4, 5)
4. 4. $n-1$
5. 5. 129
6. 6. $F == R$
7. 7. $p \rightarrow lchild == 0 \text{ \& } p \rightarrow rchild == 0$
8. 8. $O(n^2)$
9. 9. $O(n \log_2 n), O(n)$
10. 10. 开放定址法, 链地址法

四、算法设计题

1. 1. 设计在顺序有序表中实现二分查找的算法。
struct record {int key; int others;};
int biseach(struct record r[], int k)


```

{
    int low=0,mid,high=n-1;
    while(low<=high)
    {
        mid=(low+high)/2;
        if(r[mid].key==k) return(mid+1); else if(r[mid].key>k) high=mid-1; else low=mid+1;
    }
    return(0);
}

```

2. 2. 设计判断二叉树是否为二叉排序树的算法。

```

int minnum=-32768,flag=1;
typedef struct node{int key; struct node *lchild,*rchild;}bitree;
void inorder(bitree *bt)
{
    if (bt!=0) {inorder(bt->lchild); if(minnum>bt->key)flag=0;
    minnum=bt->key;inorder(bt->rchild);}
}

```

3. 3. 在链式存储结构上设计直接插入排序算法

```

void straightinsertsort(lklist *&head)
{
    lklist *s,*p,*q; int t;
    if (head==0 || head->next==0) return;
    else for(q=head,p=head->next;p!=0;p=q->next)
    {
        for(s=head;s!=q->next;s=s->next) if (s->data>p->data) break;
        if(s==q->next)q=p;
        else {q->next=p->next; p->next=s->next; s->next=p;
        t=p->data;p->data=s->data;s->data=t;}
    }
}

```

数据结构试卷（七）

一、选择题(30分)

1. 设某无向图有 n 个顶点，则该无向图的邻接表中有 (C) 个表头结点。
(A) $2n$ (B) n (C) $n/2$ (D) $n(n-1)$
2. 设无向图 G 中有 n 个顶点，则该无向图的最小生成树上有 (B) 条边。
(A) n (B) $n-1$ (C) $2n$ (D) $2n-1$
3. 设一组初始记录关键字序列为(60, 80, 55, 40, 42, 85)，则以第一个关键字 45 为基准而得到的一趟快速排序结果是 (C)。
(A) 40, 42, 60, 55, 80, 85 (B) 42, 45, 55, 60, 85, 80
(C) 42, 40, 55, 60, 80, 85 (D) 42, 40, 60, 85, 55, 80
4. () 二叉排序树可以得到一个从小到大的有序序列 (B)。
(A) 先序遍历 (B) 中序遍历 (C) 后序遍历 (D) 层次遍历
5. 设按照从上到下、从左到右的顺序从 1 开始对完全二叉树进行顺序编号，则编号为 i 结点的左孩子结点的编号为 (A)。
(A) $2i+1$ (B) $2i$ (C) $i/2$ (D) $2i-1$
6. 程序段 $s=i=0$; do { $i=i+1$; $s=s+i$; }while($i \leq n$); 的时间复杂度为 (A)。
(A) $O(n)$ (B) $O(n \log_2 n)$ (C) $O(n^2)$ (D) $O(n^3/2)$
7. 设带有头结点的单向循环链表的头指针变量为 head，则其判空条件是 (A)。
(A) head=0 (B) head->next==0
(C) head->next==head (D) head!=0
8. 设某棵二叉树的高度为 10，则该二叉树上叶子结点最多有 (C)。
(A) 20 (B) 256 (C) 512 (D) 1024

9. 设一组初始记录关键字序列为(13, 18, 24, 35, 47, 50, 62, 83, 90, 115, 134), 则利用二分法查找关键字 90 需要比较的关键字数为 (B)。
(A) 1 (B) 2 (C) 3 (D) 4
10. 设指针变量 top 指向当前链式栈的栈顶, 则删除栈顶元素的操作序列为 (D)。
(A) top=top+1; (B) top=top-1;
(C) top->next=top; (D) top=top->next;

二、判断题(20 分)

1. 不论是入队列操作还是入栈操作, 在顺序存储结构上都需要考虑“溢出”情况。 (✓)
2. 当向二叉排序树中插入一个结点, 则该结点一定成为叶子结点。 (✓)
3. 设某堆中有 n 个结点, 则在该堆中插入一个新结点的时间复杂度为 $O(\log_2 n)$ 。 (X)
4. 完全二叉树中的叶子结点只能在最后两层中出现。 (✓)
5. 哈夫曼树中没有度数为 1 的结点。 (✓)
6. 对连通图进行深度优先遍历可以访问到该图中的所有顶点。 (✓)
7. 先序遍历一棵二叉排序树得到的结点序列不一定是有序序列。 (✓)
8. 由树转化成二叉树, 该二叉树的右子树不一定为空。 (X)
9. 线性表中的所有元素都有一个前驱元素和后继元素。 (X)
10. 带权无向图的最小生成树是唯一的。 (X)

三、填空题(30 分)

1. 设指针变量 p 指向双向链表中的结点 A, 指针变量 s 指向被插入的结点 X, 则在结点 A 的后面插入结点 X 的操作序列为 $s \rightarrow left = p; s \rightarrow right = p \rightarrow right; p \rightarrow right = s;$ (设结点中的两个指针域分别为 left 和 right)。
2. 设完全有向图中有 n 个顶点, 则该完全有向图中共有 $n(n-1)$ 条有向边; 设完全无向图中有 n 个顶点, 则该完全无向图中共有 $\frac{n(n-1)}{2}$ 条无向边。
3. 设关键字序列为 (K_1, K_2, \dots, K_n) , 则用筛选法建初始堆必须从第 $\frac{n}{2}$ 个元素开始进行筛选。

$$= n_0 + n_1 + n_2 + n_3$$

$$= n_0 - 1 = n_0 + 2n_1 + 3n_2 + 4n_3$$

$$= \frac{n_0 - n_2 - 1}{2}$$

$$= \frac{10 - 2 - 1}{2}$$

$$= \frac{7 - 1}{2}$$

$$= \frac{3 - 1}{2}$$

$$= \frac{1 - 1}{2}$$

$$= \frac{0 - 1}{2}$$

$$= \frac{-1 - 1}{2}$$

$$= \frac{-2 - 1}{2}$$

$$= \frac{-3 - 1}{2}$$

$$= \frac{-4 - 1}{2}$$

$$= \frac{-5 - 1}{2}$$

$$= \frac{-6 - 1}{2}$$

$$= \frac{-7 - 1}{2}$$

$$= \frac{-8 - 1}{2}$$

$$= \frac{-9 - 1}{2}$$

解决散列表冲突的两种方法是 开放地址法 和 拉链法。

设一棵三叉树中有 50 个度数为 0 的结点, 21 个度数为 2 的结点, 则该二叉树中度数为 3 的结点数有 14 个。

高度为 h 的完全二叉树中最少有 2 个结点, 最多有 2^h 个结点。

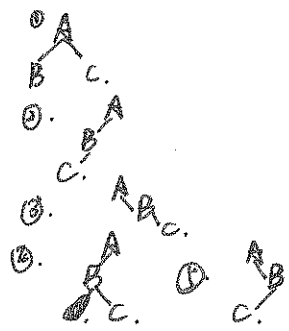
设有一组初始关键字序列为 (24, 35, 12, 27, 18, 26), 则第 3 趟直接插入排序结束后的结果是 12, 18, 24, 26, 27, 35。

设有一组初始关键字序列为 (24, 35, 12, 27, 18, 26), 则第 3 趟简单选择排序结束后的结果是 12, 18, 24, 26, 27, 35。

设一棵二叉树的前序序列为 ABC, 则有 3 种不同的二叉树可以得到这种序列。

下面程序段的功能是实现一趟快速排序, 请在下划线处填上正确的语句。

```
struct record {int key; datatype others;};
void quickpass(struct record r[], int s, int t, int &i)
{
    int j=t; struct record x=r[s]; i=s;
    while(i<j)
    {
        while(i<j && r[j].key>x.key) j=j-1; if(i<j) {r[i]=r[j]; i=i+1;}
        while(r[i].key<x.key) i=i+1; if(i<j) {r[j]=r[i]; j=j-1;}
    }
    r[i]=x;
}
```



四、算法设计题(20 分)

1. 设计在链式结构上实现简单选择排序算法。
2. 设计在顺序存储结构上实现求子串算法。

3. 3. 设计求结点在二叉排序树中层次的算法。

数据结构试卷（七）

一、选择题

- | | | | | |
|------|------|------|------|-------|
| 1. B | 2. B | 3. C | 4. B | 5. B |
| 6. A | 7. C | 8. C | 9. B | 10. D |

二、判断题

- | | | | | |
|------|------|------|------|-------|
| 1. 对 | 2. 对 | 3. 对 | 4. 对 | 5. 对 |
| 6. 对 | 7. 对 | 8. 错 | 9. 错 | 10. 错 |

三、填空题

1. 1. $s \rightarrow \text{left} = p, p \rightarrow \text{right}$
2. 2. $n(n-1), n(n-1)/2$
3. 3. $n/2$
4. 4. 开放定址法, 链地址法
5. 5. 14
6. 6. $2^{h-1}, 2^h - 1$
7. 7. (12, 24, 35, 27, 18, 26)
8. 8. (12, 18, 24, 27, 35, 26)
9. 9. 5
10. 10. $i < j \ \&\& \ r[i].\text{key} < x.\text{key}, r[i] = x$

四、算法设计题

1. 1. 设计在链式结构上实现简单选择排序算法。

```
void simpleselectsortlklst(lklist *&head)
{
    lklist *p,*q,*s; int min,t;
    if(head==0 || head->next==0) return;
    for(q=head; q!=0; q=q->next)
    {
        min=q->data; s=q;
        for(p=q->next; p!=0; p=p->next) if(min>p->data){min=p->data; s=p;}
        if(s!=q){t=s->data; s->data=q->data; q->data=t;}
    }
}
```
2. 2. 设计在顺序存储结构上实现求子串算法。

```
void substring(char s[], long start, long count, char t[])
{
    long i,j,length=strlen(s);
    if (start<1 || start>length) printf("The copy position is wrong");
    else if (start+count-1>length) printf("Too characters to be copied");
    else { for(i=start-1; j=0; i<start+count-1; i++,j++) t[j]=s[i]; t[j]='\0'; }
}
```
3. 3. 设计求结点在二叉排序树中层次的算法。

```
int lev=0;
typedef struct node{int key; struct node *lchild,*rchild;}bitree;
void level(bitree *bt,int x)
{
    if (bt!=0)
    {lev++; if (bt->key==x) return; else if (bt->key>x) level(bt->lchild,x); else
level(bt->rchild,x);}
```

数据结构试卷 (八)

一、选择题(30 分)

1. 1. 字符串的长度是指 (C)。
(A) 串中不同字符的个数 (B) 串中不同字母的个数
(C) 串中所含字符的个数 (D) 串中不同数字的个数
2. 2. 建立一个长度为 n 的有序单链表的时间复杂度为 (A)。
(A) $O(n)$ (B) $O(1)$ (C) $O(n^2)$ (D) $O(\log_2 n)$
3. 3. 两个字符串相等的充要条件是 (B)。
(A) 两个字符串的长度相等 (B) 两个字符串中对应位置上的字符相等
(C) 同时具备(A)和(B)两个条件 (D) 以上答案都不对
4. 4. 设某散列表的长度为 100, 散列函数 $H(k)=k \% P$, 则 P 通常情况下最好选择 (A)。
(A) 99 (B) 97 (C) 91 (D) 93
5. 5. 在二叉排序树中插入一个关键字值的平均时间复杂度为 (B)。
(A) $O(n)$ (B) $O(\log_2 n)$ (C) $O(n \log_2 n)$ (D) $O(n^2)$
6. 6. 设一个顺序有序表 $A[1:14]$ 中有 14 个元素, 则采用二分法查找元素 $A[4]$ 的过程中比较元素的顺序为 ()。
(A) $A[1], A[2], A[3], A[4]$ (B) $A[1], A[14], A[7], A[4]$
(C) $A[7], A[3], A[5], A[4]$ (D) $A[7], A[5], A[3], A[4]$
7. 7. 设一棵完全二叉树中有 65 个结点, 则该完全二叉树的深度为 ()。
(A) 8 (B) 7 (C) 6 (D) 5
8. 8. 设一棵三叉树中有 2 个度数为 1 的结点, 2 个度数为 2 的结点, 2 个度数为 3 的结点, 则该三叉链权中有 (C) 个度数为 0 的结点。
(A) 5 (B) 6 (C) 7 (D) 8
9. 9. 设无向图 G 中的边的集合 $E=\{(a, b), (a, e), (a, c), (b, e), (e, d), (d, f), (f, c)\}$, 则从顶点 a 出发进行深度优先遍历可以得到的一种顶点序列为 ()。
(A) aedfcb (B) acfebd (C) aebcfd (D) aedfbc
10. 10. 队列是一种 () 的线性表。
(A) 先进先出 (B) 先进后出 (C) 只能插入 (D) 只能删除

二、判断题(20 分)

1. 1. 如果两个关键字的值不等但哈希函数值相等, 则称这两个关键字为同义词。()
2. 2. 设初始记录关键字基本有序, 则快速排序算法的时间复杂度为 $O(n \log_2 n)$ 。()
3. 3. 分块查找的基本思想是首先在索引表中进行查找, 以便确定给定的关键字可能存在的块号, 然后再在相应的块内进行顺序查找。()
4. 4. 二维数组和多维数组均不是特殊的线性结构。()
5. 5. 向二叉排序树中插入一个结点需要比较的次数可能大于该二叉树的高度。()
6. 6. 如果某个有向图的邻接表中第 i 条单链表为空, 则第 i 个顶点的出度为零。()
7. 7. 非空的双向循环链表中任何结点的前驱指针均不为空。()
8. 8. 不论线性表采用顺序存储结构还是链式存储结构, 删除值为 X 的结点的时间复杂度均为 $O(n)$ 。()
9. 9. 图的深度优先遍历算法中需要设置一个标志数组, 以便区分图中的每个顶点是否被访问过。()
10. 10. 稀疏矩阵的压缩存储可以用一个三元组表来表示稀疏矩阵中的非 0 元素。()

三、填空题(30 分)

1. 1. 设一组初始记录关键字序列为(49, 38, 65, 97, 76, 13, 27, 50), 则以 $d=4$ 为增量的一趟希尔排序结束后的结果为_____。

2. 2. 下面程序段的功能是实现在二叉排序树中插入一个新结点, 请在下划线处填上正确的内容。

```
typedef struct node {int data; struct node *lchild; struct node *rchild;} bitree;
void bstinsert(bitree *&t, int k)
{
    if (t==0) { _____; t->data=k; t->lchild=t->rchild=0; }
    else if (t->data>k) bstinsert(t->lchild, k); else _____;
}
```

3. 3. 设指针变量 p 指向单链表中结点 A, 指针变量 s 指向被插入的结点 X, 则在结点 A 的后面插入结点 X 需要执行的语句序列: s->next=p->next; _____;。
4. 4. 设指针变量 head 指向双向链表中的头结点, 指针变量 p 指向双向链表中的第一个结点, 则指针变量 p 和指针变量 head 之间的关系是 p=_____和 head=_____ (设结点中的两个指针域分别为 llink 和 rlink)。
5. 5. 设某棵二叉树的中序遍历序列为 ABCD, 后序遍历序列为 BADC, 则其前序遍历序列为_____。
6. 6. 完全二叉树中第 5 层上最少有_____个结点, 最多有_____个结点。
7. 7. 设有向图中不存在有向边 $\langle V_i, V_j \rangle$, 则其对应的邻接矩阵 A 中的数组元素 A[i][j] 的值等于_____。
8. 8. 设一组初始记录关键字序列为 (49, 38, 65, 97, 76, 13, 27, 50), 则第 4 趟直接选择排序结束后的结果为_____。
9. 9. 设连通图 G 中有 n 个顶点 e 条边, 则对应的最小生成树上有_____条边。
10. 10. 设有一组初始记录关键字序列为 (50, 16, 23, 68, 94, 70, 73), 则将它们调整成初始堆只需把 16 与_____相互交换即可。

四、算法设计题(20 分)

1. 1. 设计一个在链式存储结构上统计二叉树中结点个数的算法。
2. 2. 设计一个算法将无向图的邻接矩阵转为对应邻接表的算法。

数据结构试卷(八) 参考答案

一、选择题

1. C 2. C 3. C 4. B 5. B
6. C 7. B 8. C 9. A 10. A

二、判断题

1. 对 2. 错 3. 对 4. 错 5. 错
6. 对 7. 对 8. 对 9. 对 10. 对

三、填空题

1. 1. (49, 13, 27, 50, 76, 38, 65, 97)
2. 2. t=(bitree *)malloc(sizeof(bitree)), bstinsert(t->rchild, k)
3. 3. p->next=s
4. 4. head->rlink, p->llink
5. 5. CABD
6. 6. 1, 16
7. 7. 0
8. 8. (13, 27, 38, 50, 76, 49, 65, 97)
9. 9. n-1
10. 10. 50

四、算法设计题

1. 设计一个在链式存储结构上统计二叉树中结点个数的算法。

```
void countnode(bitree *bt,int &count)
{
    if(bt!=0)
    {count++; countnode(bt->lchild,count); countnode(bt->rchild,count);}
}
```
2. 设计一个算法将无向图的邻接矩阵转为对应邻接表的算法。

```
typedef struct {int vertex[m]; int edge[m][m];}gadjmatrix;
typedef struct node1 {int info;int adjvertex; struct node1 *nextarc;}glinklistnode;
typedef struct node2 {int vertexinfo;glinklistnode *firstarc;}glinkheadnode;
void adjmatrixtoadjlist(gadjmatrix g1[],glinkheadnode g2[])
{
    int i,j; glinklistnode *p;
    for(i=0;i<=n-1;i++) g2[i].firstarc=0;
    for(i=0;i<=n-1;i++) for(j=0;j<=n-1;j++)
    if (g1.edge[i][j]==1)
    {
        p=(glinklistnode *)malloc(sizeof(glinklistnode));p->adjvertex=j;
        p->nextarc=g[i].firstarc; g[i].firstarc=p;
        p=(glinklistnode *)malloc(sizeof(glinklistnode));p->adjvertex=i;
        p->nextarc=g[j].firstarc; g[j].firstarc=p;
    }
}
```

数据结构试卷（九）

一、选择题(30分)

1. 下列程序段的时间复杂度为 (A) $O(n^2)$.

```
for(i=0; i<m; i++) for(j=0; j<t; j++) c[i][j]=0;
for(i=0; i<m; i++) for(j=0; j<t; j++) for(k=0; k<n; k++) c[i][j]=c[i][j]+a[i][k]*b[k][j];
```

(A) $O(m*n*t)$ (B) $O(m+n+t)$ (C) $O(m+n*t)$ (D) $O(m*t+n)$
2. 设顺序线性表中有 n 个数据元素，则删除表中第 i 个元素需要移动 (A) $n-i$ 个元素。

(A) $n-i$ (B) $n+1-i$ (C) $n-1-i$ (D) i
3. 设 F 是由 T_1 、 T_2 和 T_3 三棵树组成的森林，与 F 对应的二叉树为 B ， T_1 、 T_2 和 T_3 的结点数分别为 N_1 、 N_2 和 N_3 ，则二叉树 B 的根结点的左子树的结点数为 (A) N_1-1 。

(A) N_1-1 (B) N_2-1 (C) N_2+N_3 (D) N_1+N_3
4. 利用直接插入排序法的思想建立一个有序线性表的时间复杂度为 (C) $O(n^2)$ 。

(A) $O(n)$ (B) $O(n\log_2 n)$ (C) $O(n^2)$ (D) $O(\log_2 n)$
5. 设指针变量 p 指向双向链表中结点 A ，指针变量 s 指向被插入的结点 X ，则在结点 A 的后面插入结点 X 的操作序列为 (B) $p->right=s; s->left=p; p->right->left=s; s->right=p->right;$ 。

(A) $p->right=s; s->left=p; p->right->left=s; s->right=p->right;$
 (B) $s->left=p; s->right=p->right; p->right=s; p->right->left=s;$
 (C) $p->right=s; p->right->left=s; s->left=p; s->right=p->right;$
 (D) $s->left=p; s->right=p->right; p->right->left=s; p->right=s;$
6. 下列各种排序算法中平均时间复杂度为 $O(n^2)$ 是 (D) 冒泡排序。

(A) 快速排序 (B) 堆排序 (C) 归并排序 (D) 冒泡排序
7. 设输入序列 $1、2、3、\dots、n$ 经过栈作用后，输出序列中的第一个元素是 n ，则输出序列中的第 i 个输出元素是 (C) $n-i$ 。

(A) $n-i$ (B) $n-1-i$ (C) $n+1-i$ (D) 不能确定
8. 设散列表中有 m 个存储单元，散列函数 $H(key)=key \% p$ ，则 p 最好选择 (B) 小于等于 m 的最大素数。

(A) 小于等于 m 的最大奇数 (B) 小于等于 m 的最大素数
 (C) 小于等于 m 的最大偶数 (D) 小于等于 m 的最大合数

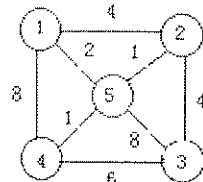
$$n_0 = n_2 + 2n_1 + 1 = 1 + 4 + 1 = 6$$

$$n_0 = n_1 + n_2 + 1 = 1 + 2 + 2 = 5$$

9. 设在一棵度数为 3 的树中，度数为 3 的结点数有 2 个，度数为 2 的结点数有 1 个，度数为 1 的结点数有 2 个，那么度数为 0 的结点数有 (C) 个。
 (A) 4 (B) 5 (C) 6 (D) 7
10. 设完全无向图中有 n 个顶点，则该完全无向图中有 (A) 条边。
 (A) $n(n-1)/2$ (B) $n(n-1)$ (C) $n(n+1)/2$ (D) $(n-1)/2$
11. 设顺序表的长度为 n ，则顺序查找的平均比较次数为 (C)。
 (A) n (B) $n/2$ (C) $(n+1)/2$ (D) $(n-1)/2$
12. 设有序表中的元素为 (13, 18, 24, 35, 47, 50, 62)，则在其中利用二分法查找值为 24 的元素需要经过 (C) 次比较。
 (A) 1 (B) 2 (C) 3 (D) 4
13. 设顺序线性表的长度为 30，分成 5 块，每块 6 个元素，如果采用分块查找，则其平均查找长度为 (B)。
 (A) 6 (B) 11 (C) 5 (D) 6.5
14. 设有向无环图 G 中的有向边集合 $E = \{ \langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 1, 4 \rangle \}$ ，则下列属于该有向图 G 的一种拓扑排序序列的是 (A)。
 (A) 1, 2, 3, 4 (B) 2, 3, 4, 1 (C) 1, 4, 2, 3 (D) 1, 2, 4, 3
15. 设有一组初始记录关键字序列为 (34, 76, 45, 18, 26, 54, 92)，则由这组记录关键字生成的二叉排序树的深度为 (A)。
 (A) 4 (B) 5 (C) 6 (D) 7

二、填空题(30 分)

1. 1. 设指针 p 指向单链表中结点 A ，指针 s 指向被插入的结点 X 。则在结点 A 的前面插入结点 X 时的操作序列为：
 1) $s \rightarrow \text{next} = p \rightarrow \text{next}$; 2) $p \rightarrow \text{next} = s$; 3) $t = p \rightarrow \text{data}$;
 4) $p \rightarrow \text{data} = s \rightarrow \text{data}$; 5) $s \rightarrow \text{data} = t$;
2. 2. 设某棵完全二叉树中有 100 个结点，则该二叉树中有 50 个叶子结点。
3. 3. 设某顺序循环队列中有 m 个元素，且规定队头指针 F 指向队头元素的前一个位置，队尾指针 R 指向队尾元素的当前位置，则该循环队列中最多存储 $m-1$ 个元素。
4. 4. 对一组初始关键字序列 (40, 50, 95, 20, 15, 70, 60, 45, 10) 进行冒泡排序，则第一趟需要进行相邻记录的比较的次数为 8，在整个排序过程中最多需要进行 8 趟排序才可以完成。
5. 5. 在堆排序和快速排序中，如果从平均情况下排序的速度最快的角度来考虑应最好选择 快速 排序，如果从节省存储空间的角度来考虑则最好选择 堆 排序。
6. 6. 设一组初始记录关键字序列为 (20, 12, 42, 31, 18, 14, 28)，则根据这些记录关键字构造的二叉排序树的平均查找长度是 19/7。
7. 7. 设一棵二叉树的中序遍历序列为 BDCA，后序遍历序列为 DBAC，则这棵二叉树的前序序列为 CBDA。
8. 8. 设用于通信的电文仅由 8 个字母组成，字母在电文中出现的频率分别为 19, 32, 21, 10，根据这些频率作为权值构造哈夫曼树，则这棵哈夫曼树的高度为 6。
9. 9. 设一组记录关键字序列为 (80, 70, 33, 65, 24, 56, 48)，则用筛选法建成的初始堆为 80。
10. 10. 设无向图 G (如右图所示)，则其最小生成树上所有边的权值之和为 8。



三、判断题(20 分)

1. 1. 有向图的邻接表和逆邻接表中表结点的个数不一定相等。 (X) F
2. 2. 对链表进行插入和删除操作时不必移动链表中结点。 (X) T
3. 3. 子串 "ABC" 在主串 "AABCBABCD" 中的位置为 2。 (X) T
4. 4. 若一个叶子结点是某二叉树的中序遍历序列的最后一个结点，则它必是该二叉树的先序遍历序列中的最后一个结点。 (X) F

- O(n log n)*
5. 5. 希尔排序算法的时间复杂度为 $O(n^2)$ 。(X) F
 6. 6. 用邻接矩阵作为图的存储结构时, 则其所占用的存储空间与图中顶点数无关而与图中边数有关。(X) F
 7. 7. 中序遍历一棵二叉排序树可以得到一个有序的序列。(V) T
 8. 8. 入栈操作和入队列操作在链式存储结构上实现时不需要考虑栈溢出的情况。(V) T
 9. 9. 顺序表查找指的是在顺序存储结构上进行查找。(X) F
 10. 10. 堆是完全二叉树, 完全二叉树不一定是堆。() T

五、算法设计题(20 分)

1. 1. 设计计算二叉树中所有结点值之和的算法。
2. 2. 设计将所有奇数移到所有偶数之前的算法。
3. 3. 设计判断单链表中元素是否是递增的算法。

数据结构试卷(九) 参考答案

一、选择题

1. A 2. A 3. A 4. C 5. D
6. D 7. C 8. B 9. C 10. A
11. C 12. C 13. D 14. A 15. A

二、填空题

1. 1. $p \rightarrow next, s \rightarrow data$
2. 2. 50
3. 3. $m-1$
4. 4. 6, 8
5. 5. 快速, 堆
6. 6. $i/7$
7. 7. CBDA
8. 8. 6
9. 9. {24, 65, 33, 80, 70, 56, 48}
10. 10. 8

三、判断题

1. 错 2. 对 3. 对 4. 对 5. 错
6. 错 7. 对 8. 对 9. 错 10. 对

四、算法设计题

1. 1. 设计计算二叉树中所有结点值之和的算法。

```
void sum(bitree *bt, int &s)
{
    if (bt != 0) {s = s + bt->data; sum(bt->lchild, s); sum(bt->rchild, s);}
}
```
2. 2. 设计将所有奇数移到所有偶数之前的算法。

```
void quickpass(int r[], int s, int t)
{
    int i = s, j = t, x = r[s];
    while (i < j)
    {
        while (i < j && r[j] % 2 == 0) j = j - 1; if (i < j) {r[i] = r[j]; i = i + 1;}
        while (i < j && r[i] % 2 == 1) i = i + 1; if (i < j) {r[j] = r[i]; j = j - 1;}
    }
    r[i] = x;
}
```


3. 设计判断单链表中元素是否是递增的算法。

```
int isriselk(lklist *head)
```

```
{
    if(head==0||head->next==0) return(1);else
    for(q=head,p=head->next;p!=0;q=p,p=p->next)if(q->data>p->data) return(0);
    return(1);
}
```

数据结构试卷 (十)

$$\frac{n(n+1)}{2} = n$$

一、选择题 (24 分)

1. 下列程序段的时间复杂度为 (B).

$$S = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

```
i=0, s=0; while (s<n) {s=s+i; i++;}
```

- (A) $O(n^{1/2})$ (B) $O(n^{1/3})$ (C) $O(n)$ (D) $O(n^2)$

2. 设某链表中最常用的操作是在链表的尾部插入或删除元素, 则选用下列 (C) 存储方式最节省运算时间。

- (A) 单向链表 (B) 单向循环链表
(C) 双向链表 (D) 双向循环链表

3. 设指针 q 指向单链表中结点 A, 指针 p 指向单链表中结点 A 的后继结点 B, 指针 s 指向被插入的结点 X, 则在结点 A 和结点 B 插入结点 X 的操作序列为 (A)。

- (A) $s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s;$ (B) $q \rightarrow \text{next} = s; s \rightarrow \text{next} = p;$
(C) $p \rightarrow \text{next} = s \rightarrow \text{next}; s \rightarrow \text{next} = p;$ (D) $p \rightarrow \text{next} = s; s \rightarrow \text{next} = q;$

4. 设输入序列为 1、2、3、4、5、6, 则通过栈的作用后可以得到的输出序列为 (B)。

- (A) 5, 3, 4, 6, 1, 2 (B) 3, 2, 5, 6, 4, 1

- (C) 3, 1, 2, 5, 4, 6 (D) 1, 5, 4, 6, 2, 3

5. 设有一个 10 阶的下三角矩阵 A (包括对角线), 按照从上到下、从左到右的顺序存储到连续的 55 个存储单元中, 每个数组元素占 1 个字节的存储空间, 则 $A[5][4]$ 地址与 $A[0][0]$ 的地址之差为 ()。

- (A) 10 (B) 19 (C) 28 (D) 55

6. 设一棵 m 叉树中有 N_1 个度数为 1 的结点, N_2 个度数为 2 的结点, ..., N_m 个度数为 m 的结点, 则该树中共有 (D) 个叶子结点。

$$n_0 = n_1 + 1$$

- (A) $\sum_{i=1}^m (i-1)N_i$ (B) $\sum_{i=1}^m N_i$ (C) $\sum_{i=2}^m N_i$ (D) $1 + \sum_{i=2}^m (i-1)N_i$

7. 二叉排序树中左子树上所有结点的值均 (A) 根结点的值。

- (A) $<$ (B) $>$ (C) $=$ (D) $!=$

8. 设一组权值集合 $W = \{15, 3, 14, 2, 8, 9, 16, 17\}$, 要求根据这些权值集合构造一棵哈夫曼树, 则这棵哈夫曼树的带权路径长度为 ()。

- (A) 129 (B) 219 (C) 189 (D) 229

9. 设有 n 个关键字具有相同的 Hash 函数值, 则用线性探测法把这 n 个关键字映射到 HASH 表中需要做 () 次线性探测。

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

10. 设某棵二叉树中只有度数为 0 和度数为 2 的结点且度数为 0 的结点数为 n, 则这棵二叉树中共有 (C) 个结点。

- (A) 2n (B) n+1 (C) 2n-1 (D) 2n+1

11. 设一组初始记录关键字的长度为 8, 则最多经过 (B) 趟插入排序可以得到有序序列。

- (A) 6 (B) 7 (C) 8 (D) 9

12. 设一组初始记录关键字序列为 (Q, H, C, Y, P, A, M, S, R, D, F, X), 则按字母升序的第一趟冒泡排序结束后的结果是 ()。

- (A) F, H, C, D, P, A, M, Q, R, S, Y, X
(B) P, A, C, S, Q, D, F, X, R, H, M, Y
(C) A, D, C, R, F, Q, M, S, Y, P, H, X

(D) H, C, Q, P, A, M, S, R, D, F, X, Y

二、填空题(48分, 其中最后两小题各6分)

1. 1. 设需要对5个不同的记录关键字进行排序, 则至少需要比较 10 次, 至多需要比较 31 次。
2. 2. 快速排序算法的平均时间复杂度为 $O(n^2)$, 直接插入排序算法的平均时间复杂度为 $O(n \log_2 n)$ 。
3. 3. 设二叉排序树的高度为 h , 则在该树中查找关键字 key 最多需要比较 h 次。
4. 4. 设在长度为20的有序表中进行二分查找, 则比较一次查找成功的结点数有 1 个, 比较两次查找成功的结点数有 2 个。
5. 5. 设一棵 m 叉树的结点数 n , 用多重链表表示其存储结构, 则该树中有 $m \times n - n$ 个空指针域。
6. 6. 设指针变量 p 指向单链表中结点 A , 则删除结点 A 的语句序列为:
 $q = p \rightarrow next; p \rightarrow data = q \rightarrow data; p \rightarrow next = q$; free(q);
7. 7. 数据结构从逻辑上划分为三种基本类型: 线性、树形 和 图形。
8. 8. 设无向图 G 中有 n 个顶点 e 条边, 则用邻接矩阵作为图的存储结构进行深度优先或广度优先遍历的时间复杂度为 $O(n^2)$; 用邻接表作为图的存储结构进行深度优先或广度优先遍历的时间复杂度为 $O(n + e)$ 。
9. 9. 设散列表的长度为8, 散列函数 $H(k) = k \% 7$, 用线性探测法解决冲突, 则根据一组初始关键字序列(8, 15, 16, 22, 30, 32)构造出的散列表的平均查找长度是 1.5。
10. 10. 设一组初始关键字序列为(38, 65, 97, 76, 13, 27, 10), 则第3趟冒泡排序结束后的结果为 10, 13, 27, 38, 65, 76, 97。
11. 11. 设一组初始关键字序列为(38, 65, 97, 76, 13, 27, 10), 则第3趟简单选择排序后的结果为 10, 13, 27, 38, 65, 76, 97。
12. 12. 设有向图 G 中的有向边的集合 $E = \{ \langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 1, 4 \rangle, \langle 4, 5 \rangle, \langle 5, 3 \rangle, \langle 4, 6 \rangle, \langle 6, 5 \rangle \}$, 则该图的一个拓扑序列为 1 2 4 6 5 3。
13. 13. 下面程序段的功能是建立二叉树的算法, 请在下划线处填上正确的内容。
typedef struct node {int data; struct node *lchild; struct node *rchild;} bitree;
void createbitree(bitree *&bt)
{
scanf("%c", &ch);
if(ch == '#') return; else
{ bt = (bitree *)malloc(sizeof(bitree)); bt->data = ch; createbitree(bt->rchild);
}
}
14. 14. 下面程序段的功能是利用从尾部插入的方法建立单链表的算法, 请在下划线处填上正确的内容。
typedef struct node {int data; struct node *next;} llist;
void llistcreate(llist * &head)
{
for (i = 1; i <= n; i++)
{
p = (llist *)malloc(sizeof(llist)); scanf("%d", &(p->data)); p->next = 0;
if (i == 1) head = p; else {q->next = p; p = q;}
}
}

三、算法设计题(22分)

1. 1. 设计在链式存储结构上合并排序的算法。
2. 2. 设计在二叉排序树上查找结点 X 的算法。
3. 3. 设关键字序列 $(k_1, k_2, \dots, k_{n-1})$ 是堆, 设计算法将关键字序列 $(k_1, k_2, \dots, k_{n-1}, x)$ 调整为堆。

数据结构试卷(十) 参考答案

一、选择题

1. A 2. D 3. B 4. B 5. B 6. D
7. A 8. D 9. D 10. C 11. B 12. D

二、填空题

1. 1. 4, 10
2. 2. $O(n\log_2 n)$, $O(n^2)$
3. 3. n
4. 4. 1, 2
5. 5. $n(m-1)+1$
6. 6. $q \rightarrow \text{next}$
7. 7. 线性结构, 树型结构, 图型结构
8. 8. $O(n^2)$, $O(n+e)$
9. 9. $8/3$
10. 10. (38, 13, 27, 10, 65, 76, 97)
11. 11. (10, 13, 27, 76, 65, 97, 38)
12. 12. 124653
13. 13. struct node *rchild, bt=0, createbitree(bt->lchild)
14. 14. lklist, q=p

三、算法设计题

1. 1. 设计在链式存储结构上合并排序的算法。
void mergelklist(lklist *ha, lklist *hb, lklist *&hc)
{
 lklist *s=hc=0;
 while(ha!=0 && hb!=0)
 if(ha->data<hb->data){if(s==0) hc=s=ha; else {s->next=ha; s=ha;}; ha=ha->next;}
 else {if(s==0) hc=s=hb; else {s->next=hb; s=hb;}; hb=hb->next;}
 if(ha==0) s->next=hb; else s->next=ha;
}
2. 2. 设计在二叉排序树上查找结点 X 的算法。
bitree *bstsearch1(bitree *t, int key)
{
 bitree *p=t;
 while(p!=0) if (p->key==key) return(p); else if (p->key>key) p=p->lchild; else p=p->rchild;
 return(0);
}
3. 3. 设关键字序列(k_1, k_2, \dots, k_{n-1})是堆, 设计算法将关键字序列($k_1, k_2, \dots, k_{n-1}, x$)调整为堆。
void adjustheap(int r[], int n)
{
 int j=n, i=j/2, temp=r[j-1];
 while (i>1) if (temp>=r[i-1]) break; else {r[j-1]=r[i-1]; j=i; i=i/2;}
 r[j-1]=temp;
}

