**复习题（一）**

**一．填空题（每空1分，共15分）**

1. 一个算法的效率可分为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_效率和\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_效率。
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_是被限定为只能在表的一端进行插入运算，在表的另一端进行删除运算的线性表。
3. 设**S**=“A;/document/Mary.doc”，则strlen(**S**)= \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_， “/”的字符定位的位置为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
4. 设数组a[1…60, 1…70]的基地址为2048，每个元素占2个存储单元，若以列序为主序顺序存储，则元素a[32,58]的存储地址为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
5. 一棵深度为6的满二叉树有\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_个分支结点和\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_个叶子。
6. 用5个权值{3, 2, 4, 5, 1}构造的哈夫曼（Huffman）树的带权路径长度是 。
7. 设有一稀疏图***G***，则***G***采用 存储较省空间。
8. 快速排序算法是对 算法的一种改进。
9. 在数据的存放无规律而言的线性表中进行检索的最佳方法是 。
10. 大多数排序算法都有两个基本的操作： 和 。
11. 设要将序列（Q, H, C, Y, P, A, M, S, R, D, F, X）中的关键码按字母序的升序重新排列，则：快速排序一趟扫描的结果是 。
12. **选择题（每题2分，共30分）**

（ ）1．数据在计算机存储器内表示时，物理地址与逻辑地址相同并且是连续的，称之为：

（A）存储结构 （B）逻辑结构 （C）顺序存储结构 （D）链式存储结构

（ ）2. 向一个有127个元素的顺序表中插入一个新元素并保持原来顺序不变，平均要移动 个元素

（A）8 （B）63.5 （C）63 （D）7

（ ）3. 链接存储的存储结构所占存储空间：

1. 分两部分，一部分存放结点值，另一部分存放表示结点间关系的指针
2. 只有一部分，存放结点值
3. 只有一部分，存储表示结点间关系的指针
4. 分两部分，一部分存放结点值，另一部分存放结点所占单元数

（ ）4. 设*a*1、*a*2、*a*3为3个结点，整数**P0**，3，4代表地址，则如下的链式存储结构称为

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **P0** |  | **3** |  |  | **4** |  |  |
| **P0** | **🡪** | *a*1 | **3** | **🡪** | *a*2 | **4** | **🡪** | *a*3 | **0** |

（Ａ）循环链表 （Ｂ）单链表 （Ｃ）双向循环链表 （Ｄ）双向链表

（ ）5．双向循环链表的每个结点中包括两个指针next和previous，分别指向该结点的后继和前驱结点。现要删除指针***p***所指向的结点，下面的操作序列中哪一个是正确的？

（A）***p***-next-〉previous = ***p***->previous; ***p***->previous-〉next = ***p***->next；

（B）***p***->next-〉previous = ***p***->next; ***p***->previous-〉next = ***p***->previous；

（C）***p***->previous-〉next = ***p***->previous; ***p***->next-〉previous = ***p***->next；

（D）***p***->priou-〉next-〉next = ***p***-next; ***p***->next-〉previous = ***p***->previous；

（ ）6. 若已知一个栈的入栈序列是1，2，3，…，*n*，其输出序列为*p*1，*p*2，*p*3，…，*pn*，若*p*1=*n*，则*pi*为：

（Ａ）*i* （Ｂ）*n*=*i*  （Ｃ）*n*-*i*+1 （Ｄ）不确定

）7. 数组Q[*n*]用来表示一个循环队列，*f*为当前队列头元素的前一位置，*r*为队尾元素的位置，假定队列中元素的个数小于*n*，计算队列中元素的公式为：

（Ａ）*r*－*f* （Ｂ）（*n*＋*f*－*r*）% *n* （Ｃ）*n*＋*r*－*f* （Ｄ）（*n*＋*r*－*f*）% *n*

（ ）8. 设串***s*1**=’ABCDEFG’，***s*2**=’PQRST’，函数con(*x*,*y*)返回*x*和*y*串的连接串，subs(*s*, *i*, *j*)返回串*s*的从序号*i*开始的*j*个字符组成的子串，len(*s*)返回串*s*的长度，则con(subs(***s*1**, 2, len(***s*2**)), subs(***s*1**, len(***s*2**), 2))的结果串是：

（Ａ）BCDEF （Ｂ）BCDEFG （Ｃ）BCPQRST （Ｄ）BCDEFEF

（ ）9. 设矩阵**A**是一个对称矩阵，为了节省存储，将其下三角部分（如右图所示）按行序存放在一维数组B[1, n(n-1)/2 ]中，对下三角部分中任一元素ai,j（i≤j），在一维数组B中下标*k*的值是：



（Ａ）i(i-1)/2+j-1 （Ｂ）i(i-1)/2+j

（Ｃ）i(i+1)/2+j-1 （Ｄ）i(i+1)/2+j

（ ）10. 具有n(n>0)个结点的完全二叉树的深度为 。

（Ａ） ⎡log2(n)⎤ （Ｂ） ⎣ log2(n)⎦ （Ｃ） ⎣ log2(n) ⎦+1 （Ｄ） ⎡log2(n)+1⎤

（ ）11. 深度优先遍历类似于二叉树的

（Ａ）先序遍历 （B）中序遍历 （C）后序遍历 （D）层次遍历

（ ）12. 已知图的邻接矩阵如右图，根据算法，则从顶点0出发，按深度优先遍历的结点序列是：

（A）0 2 4 3 1 5 6 （B）0 1 3 5 6 4 2

（C）0 4 2 3 1 6 5 （D）0 1 3 4 2 5 6

（ ）13. 设哈希表长度为11，哈希函数为H（key）=key mod 11。表中已有4个元素H（15）=4；H（38）=5；H（61）=6；H（84）=7其余地址为空，若用二次探测再散列处理冲突，关键字为49的元素的地址是\_\_\_\_\_\_\_\_\_\_。



（A）3 （B）5 （C）8 （D）9

（ ）14. 任何一个无向连通图的最小生成树

（Ａ）只有一棵 （B）一棵或多棵 （C）一定有多棵 （D）可能不存在

（ ）15. 下述几种排序方法中，要求内存最大的是

（Ａ）插入排序 （B）快速排序 （C）归并排序 （D）选择排序

1. **判断题（每题2分，共20分）**

（ ）1. 链表的删除算法很简单，因为当删除链中某个结点后，计算机会自动地将后续的各个单元向前移动。

（ ）2. 链表的物理存储结构具有同链表一样的顺序。

（ ）3. 顺序表结构适宜于进行顺序存取，而链表适宜于进行随机存取。

（ ）4. 线性表在顺序存储时，逻辑上相邻的元素未必在存储的物理位置次序上相邻。

（  ）5. 栈是一种对所有插入、删除操作限于在表的一端进行的线性表，是一种后进先出型结构。

（  ）6. 栈和队列是一种非线性数据结构。

（ ）7. 二叉树中每个结点的两棵子树的高度差等于1。

（  ）8. 对于不同的使用者，一个表结构既可以是栈，也可以是队列，也可以是线性表。

（ ）9. 二叉树中所有结点个数是2*k*-1-1，其中*k*是树的深度。

（  ）10. 用二叉链表法（link-rlink）存储包含*n*个结点的二叉树，结点的2*n*个指针区域中有*n*+1个为空指针。

1. **简答题（三题，共12分）**

**1. 给定如图所示二叉树T，请画出与其对应的中序线索二叉树。（5分）**

28

1. 33

40 60 08 54

55

**解答：**

**2. 假定对有序表：（3，4，5，7，24，30，42，54，63，72，87，95）进行折半查找，试回答下列问题：（7分）**

1. **画出描述折半查找过程的判定树；**
2. **若查找元素54，需依次与哪些元素比较？**
3. **假定每个元素的查找概率相等，求查找成功时的平均查找长度。**

**解答：**

1. **算法理解题：（共13分）**

**1. 设如下图所示的二叉树B的存储结构为二叉链表，root为根指针，结点结构为：(lchild,data,rchild)。其中lchild，rchild分别为指向左右孩子的指针，data为字符型，root为根指针，对下列二叉树B，执行下列算法traversal(root)，试指出其输出结果（4分）；**

C的结点类型定义如下：

struct node

{char data;

struct node \***lchild,** rchild;

};

C算法如下：

void traversal(struct node \*root)

{if (root)

{printf(“%c”, root->data);

traversal(root->lchild);

printf(“%c”, root->data);

traversal(root->rchild);

}

}

A

B D

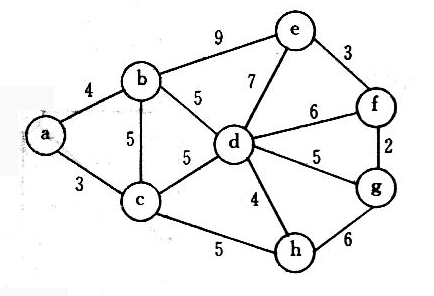
C F G

E

**二叉树B**

**解答：**

**2. 请对下图的无向带权图，写出它的邻接矩阵，并按普里姆算法求其最小生成树。（9分）**



1. **算法设计题（10分）**

**编写按层次顺序（同一层自左至右）遍历二叉树的算法。（10分）**

**解答：**

**复习题（一）答案**

**一、填空题(每空1分，共15分)**

1. 时间；空间
2. 队列
3. 20；3
4. 8950
5. 31；32
6. 33
7. 邻接表
8. 起泡
9. 顺序查找（线性查找）
10. 比较；移动
11. F H C D P A M Q R S Y X
12. **单项选择题(每空2分，共30分，多选漏选均不得分)**

1. C 2. B 3. A 4.B 5.A

6. C 7. D 8. D 9.B 10.C

11. A 12. D 13. A 14.A 15.C

**三、判断题(每题2分，共20分)**

1. × 2. × 3. × 4. × 5.**√**

6. × 7. × 8. **√** 9. × 10. **√**

**四、简答题(共12分，意思正确给分)**

**1. 给定如图所示二叉树T，请画出与其对应的中序线索二叉树。（5分）**

28

1. 33

40 60 08 54

55

**解答**：要遵循中序遍历的轨迹来画出每个前驱和后继。

中序遍历序列：55 40 25 60 28 08 33 54

**28**

**25**

**40**

**5555**

**60**

**33**

**08**

**54**

**28**

**NIL**

**NIL**

1. **33**

**40 60 08 54**

**55**

**3. 假定对有序表：（3，4，5，7，24，30，42，54，63，72，87，95）进行折半查找，试回答下列问题：（7分）**

1. **画出描述折半查找过程的判定树；（3分）**
2. **若查找元素54，需依次与哪些元素比较？（2分）**
3. **假定每个元素的查找概率相等，求查找成功时的平均查找长度。（2分）**

**解答：**

1. 先画出判定树如下（注：mid=⎣(1+12)/2⎦=6）：

30

5 63

3 7 42 87

4 24 54 72 95

1. 查找元素54，需依次与30, 63, 42, 54 等元素比较；
2. 求ASL之前，需要统计每个元素的查找次数。判定树的前3层共查找1＋2×2＋4×3=17次；但最后一层未满，不能用8×4，只能用5×4=20次，所以ASL＝1/12（17＋20）＝37/12≈3.08。

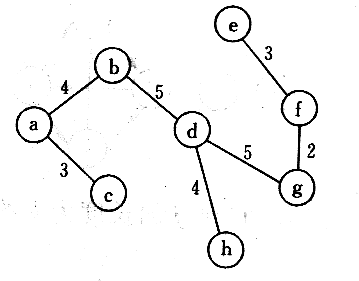
**五、算法理解题：（共13分）**

**1. 解答**（4分）：

这是“先根再左再根再右”，比前序遍历多打印各结点一次，输出结果为：A B C C E E B A D F F D G G；

**2. 解答**：设起点为a。可以直接由原始图画出最小生成树，而且最小生成树只有一种（类）。（9分）

邻接矩阵为：



**最小生成树→**

PRIM算法（横向变化）：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **V** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **U** | **V-U** |
| Vex  **lowcost** | **a**  **4** | **a**  **3** | **a**  **∞** | **a**  **∞** | **a**  **∞** | **a**  **∞** | **a**  **∞** | **{a}** | **{b,c,d,e,f,g,h}** |
| Vex  **lowcost** | **a**  **4** | **0** | **c**  **5** | **a**  **∞** | **a**  **∞** | **a**  **∞** | **c**  **5** | **{a,c}** | **{b, d,e,f,g,h}** |
| Vex  **lowcost** | **0** | **0** | **c**  **5** | **b**  **9** | **a**  **∞** | **a**  **∞** | **c**  **5** | **{a,c,b}** | **{d,e,f,g,h}** |
| Vex  **lowcost** | **0** | **0** | **0** | **d**  **7** | **d**  **6** | **d**  **5** | **d**  **4** | **{a,c,b,d }** | **{e,f,g,h}** |
| Vex  **lowcost** | **0** | **0** | **0** | **d**  **7** | **d**  **6** | **d**  **5** | **0** | **{a,c,b,d ,h }** | **{e,f,g }** |
| Vex  **lowcost** | **0** | **0** | **0** | **d**  **7** | **g**  **2** | **0** | **0** | **{a,c,b,d ,h ,g}** | **{ f,e }** |
| Vex  **lowcost** | **0** | **0** | **0** | **f**  **3** | **0** | **0** | **0** | **{a,c,b,d ,h ,g, f }** | **{e }** |
| Vex  **lowcost** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **{a,c,b,d ,h ,g, f, e }** | **{ }** |

**六、算法设计题（10分）**

**编写按层次顺序（同一层自左至右）遍历二叉树的算法。（10分）**

**解答**：既然要求从上到下，从左到右，则利用队列存放各子树结点的指针是个好办法。这是一个循环算法，用while语句不断循环，直到队空之后自然退出该函数。技巧之处：当根结点入队后，会自然使得左、右孩子结点入队，而左孩子出队时又会立即使得它的左右孩子结点入队，……以此产生了按层次输出的效果。

level(liuyu\*T)

/\* liuyu \*T,\*p,\*q[100]; 假设max已知\*/

{int f,r;

f=0; r=0; /\*置空队\*/

r=(r+1)%max;

q[r]=T; /\*根结点进队\*/

while(f!=r) /\*队列不空\*/

{f=(f+1%max);

p=q[f]; /\*出队\*/

printf("%d",p->data); /\*打印根结点\*/

if(p->lchild){r=(r+1)%max; q[r]=p->lchild;} /\*若左子树不空，则左子树进队\*/

if(p->rchild){r=(r+1)%max; q[r]=p->rchild;} /\*若右子树不空，则右子树进队\*/

}

return(0);

}

**方法二：**

void LayerOrder(Bitree T)//层序遍历二叉树

{

InitQueue(Q); //建立工作队列

EnQueue(Q,T);

while(!QueueEmpty(Q))

{

DeQueue(Q,p);

visit(p);

if(p->lchild) EnQueue(Q,p->lchild);

if(p->rchild) EnQueue(Q,p->rchild);

}

}//LayerOrder

**复习题（二）**

1. **填空题（每空1分，共15分）**
2. 数据结构被形式地定义为(**D**，**R**)，其中**D**是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_的有限集合，**R**是**D**上的\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_有限集合。
3. 在*n*个结点的单链表中要删除已知结点\*p，需找到它的\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_的地址，其时间复杂度为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
4. 栈中元素的进出原则是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
5. 若*n*为主串长，*m*为子串长，则串的古典（朴素）匹配算法最坏的情况下需要比较字符的总次数为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
6. 假设有二维数组**A**6×8，每个元素用相邻的6个字节存储，存储器按字节编址。已知A的起始存储位置（基地址）为1000，则末尾元素**A**57的第一个字节地址为 ；若按行存储时，元素**A**14的第一个字节地址为 。
7. 一棵具有257个结点的完全二叉树，它的深度为 。
8. 一棵含有*n*个结点的*k*叉树，可能达到的最大深度为 。
9. *n*个顶点*e*条边的图，若采用邻接表存储，则空间复杂度为 。
10. 设有一稠密图***G***，则***G***采用 存储较省空间。
11. 线性有序表(a1, a2, a3, …, a256)是从小到大排列的，对一个给定的值*k*，用二分法检索表中与*k*相等的元素，在查找不成功的情况下，最多需要检索 次。
12. 设要将序列(Q, H, C, Y, P, A, M, S, R, D, F, X)中的关键码按字母序的升序重新排列，则：冒泡排序一趟扫描的结果是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
13. 在堆排序、快速排序和归并排序中，若只从最坏情况下最快并且要节省内存考虑，则应选取\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_方法。
14. **选择题（每题2分，共30分）**

（ ）1. 算法分析的两个主要方面是：

（A）空间复杂性和时间复杂性 （B）正确性和简明性

（C）可读性和文档性 （D）数据复杂性和程序复杂性

（ ）2. 在*n*个结点的顺序表中，算法的时间复杂度是O(1)的操作是：

（A）访问第*i*个结点（1≤*i*≤*n*）和求第*i*个结点的直接前驱（2≤*i*≤*n*）

1. 在第*i*个结点后插入一个新结点（1≤*i*≤*n*）
2. 删除第*i*个结点（1≤*i*≤*n*）
3. 将*n*个结点从小到大排序

（ ）3．双向循环链表的每个结点中包括两个指针next和previous，分别指向该结点的后继和前驱结点。现要删除指针***p***所指向的结点，下面的操作序列中哪一个是正确的？

（A）***p***-next-〉previous = ***p*** ->previous; ***p*** ->previous-〉next = ***p*** ->next；

（B）***p*** ->next-〉previous = ***p*** ->next; ***p*** ->previous-〉next = ***p*** ->previous；

（C）***p*** ->previous-〉next = ***p*** ->previous; ***p*** ->next-〉previous = ***p*** ->next；

（D）***p*** ->previous-〉next-〉next = ***p*** -next; ***p*** ->next-〉previous = ***p*** -> previous；

（ ）4. 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

（ ）5. 若已知一个栈的入栈序列是1，2，3，…，*n*，其输出序列为*p*1，*p*2，*p*3，…，*pn*，若*p*1=*n*，则*pi*为

（Ａ）*i* （Ｂ）*n*=*i*  （Ｃ）*n*-*i*+1 （Ｄ）不确定

（ ）6. 数组Q[*n*]用来表示一个循环队列，*f*为当前队列头元素的前一位置，*r*为队尾元素的位置，假定队列中元素的个数小于*n*，计算队列中元素的公式为

（Ａ）*r*－*f* （Ｂ）（*n*＋*f*－*r*）% *n* （Ｃ）*n*＋*r*－*f* （Ｄ）（*n*＋*r*－*f*）% *n*

（ ）7. 判定一个栈ST（最多元素为*m*）为空的条件是

（Ａ）ST->top<>0 （Ｂ）ST->top=0 （Ｃ）ST->top<>*m* （Ｄ）ST->top=*m*

（ ）8. 判定一个队列QU（最多元素为*m*）为满队列的条件是

（Ａ）QU->rear － QU->front = = *m*  （Ｂ）QU->rear － QU->front －1= = *m*

（Ｃ）QU->front = = QU->rear （Ｄ）QU->front = = QU->rear+1

（ ）9. 设串***s*1**=’ABCDEFG’，***s*2**=’PQRST’，函数con(x,y)返回x和y串的连接串，subs(s, i, j)返回串s的从序号i开始的j个字符组成的子串，len(s)返回串s的长度，则con(subs(***s*1**, 2, len(***s*2**)), subs(***s*1**, len(***s*2**), 2))的结果串是：

（Ａ）BCDEF （Ｂ）BCDEFG （Ｃ）BCPQRST （Ｄ）BCDEFEF

（ ）10. 具有12个结点的完全二叉树有个度为2的结点。

（Ａ）4 （Ｂ）5 （Ｃ）6 （Ｄ）7

（ ）11. 具有*n*(*n*>0)个结点的完全二叉树的深度为 。

（Ａ） ⎡log2(*n*)⎤ （Ｂ） ⎣ log2(*n*)⎦ （Ｃ） ⎣ log2(*n*) ⎦+1 （Ｄ） ⎡log2(*n*)+1⎤

（ ）12. 已知图的邻接矩阵如右图，根据算法，则从顶点0出发，按深度优先遍历的结点序列是：

（A）0 2 4 3 1 5 6 （B）0 1 3 5 6 4 2

（C）0 4 2 3 1 6 5 （D）0 1 3 4 2 5 6

（ ）13. 设哈希表长度为11，哈希函数为H（key）=key mod 11。表中已有4个元素 H（15）=4； H（38）=5；H（61）=6； H（84）=7其余地址为空，若用二次探测再散列处理冲突，关键字为49的元素的地址是\_\_\_\_\_\_\_\_\_\_。



（A）3 （B）5 （C）8 （D）9

（ ）14. 假设有60行70列的二维数组a[1…60, 1…70]以列序为主序顺序存储，其基地址为10000，每个元素占2个存储单元，那么第32行第58列的元素a[32,58]的存储地址为 。（无第0行第0列元素）

（Ａ）16902 （Ｂ）16904 （Ｃ）14454 （Ｄ）答案A, B, C均不对

（ ）15. 广度优先遍历类似于二叉树的

（Ａ）先序遍历 （B）中序遍历 （C）后序遍历 （D）层次遍历

1. **判断题（每题2分，共20分）**

（ ）1. 链表的删除算法很简单，因为当删除链中某个结点后，计算机会自动地将后续的各个单元向前移动。

（ ）2. 线性表在物理存储空间中也一定是连续的。

（ ）3. 顺序存储方式只能用于存储线性结构。

（  ）4. 对于不同的使用者，一个表结构既可以是栈，也可以是队列，也可以是线性表。

（ ）5. 若二叉树用二叉链表作存贮结构，则在*n*个结点的二叉树链表中只有*n*-1个非空指针域。

（ ）6. 二叉树中每个结点的两棵子树是有序的。

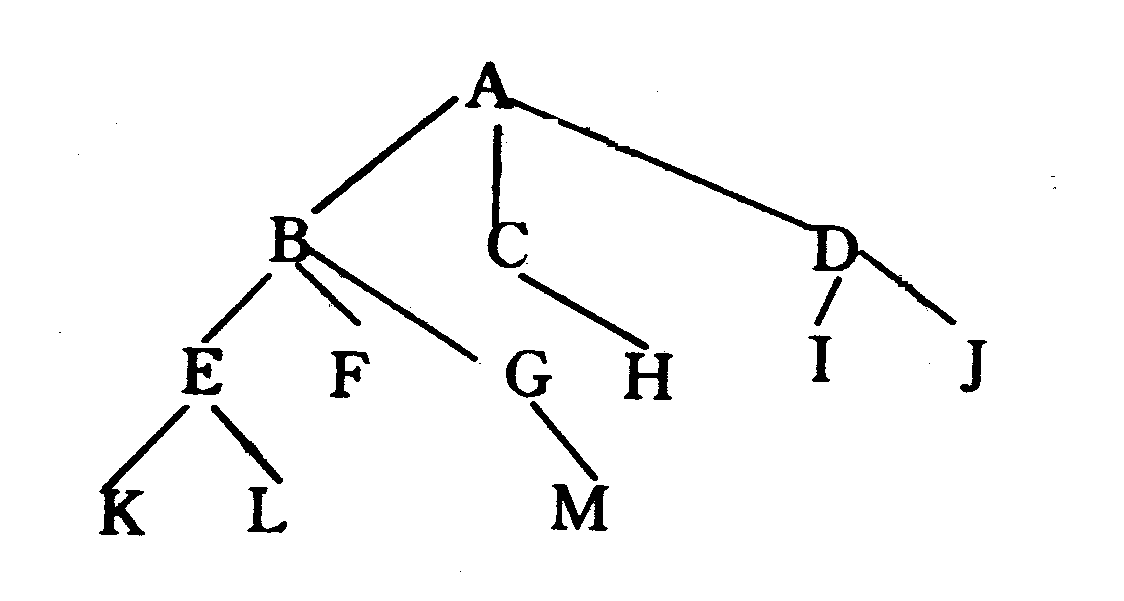
（ ）7. 二叉树中每个结点的关键字值大于其左非空子树（若存在的话）所有结点的关键字值，且小于其右非空子树（若存在的话）所有结点的关键字值。

（ ）8. 二叉树中所有结点，如果不存在非空左子树，则不存在非空右子树。

（ ）9. 用二叉链表法（link-rlink）存储包含*n*个结点的二叉树，结点的2*n*个指针区域中有*n*+1个为空指针。

（ ）10.在一棵二叉树中，如果叶子结点的个数为*n*0，度为2的结点的个数为*n*2，则*n*0=*n*2+1。

1. **简答题（共12分）**
2. **试比较顺序存储结构和链式存储结构的优缺点。在什么情况下用顺序表比链表好？（5分）**
3. **把如图所示的树转化成二叉树。（7分）**



1. **算法理解题（共13分）**

**1、阅读算法，写出该算法的结果(4分)**

main()

{ SqStackTp sq; int i; char ch;

InitStack(&sq);

For(ch=’A’;ch<=’A’+6;ch++)

{ Push(&sq,ch); printf(“%c”,ch);

}

printf(“\n”);

while(!EmptyStack(sq))

{ Pop(&sq,&ch);

printf(“&c”,ch);

} printf(“\n”);

}

**解答：**

1. **设哈希（Hash）表的地址范围为0～17，哈希函数为：H（K）＝K MOD 16。K为关键字，用线性探测法再散列法处理冲突，输入关键字序列：（10，24，32，17，31，30，46，47，40，63，49）造出Hash表，试回答下列问题：（9分）**
2. **画出哈希表的示意图；（4分）**
3. **若查找关键字63，需要依次与哪些关键字进行比较？（2分）**
4. **假定每个关键字的查找概率相等，求查找成功时的平均查找长度。（3分）**
5. **算法设计题（10分）**

**请编写一个递归算法，求二叉树的深度。（10分）**

**解答：**

**答案**

1. **填空题(每空1分，共15分)**
2. 数据元素；关系

2前驱结点；O(*n*)

3后进先出

4(n-m+1)\*m

5.1282；1072

6.9

*7.n*

8．O(*n*+*e*)

9.邻接矩阵

10.8

11.H C Q P A M S R D F X Y

1. 堆排序

**二单项选择题(每空2分，共30分，多选漏选均不得分)**

1. A 2. A 3. A 4.D 5.C

6. D 7. B 8. A 9.D 10.B

11. C 12. D 13. A 14.A 15.D

**三、判断题(每题2分，共20分)**

1. × 2. × 3. × 4. **√** 5. **√**

6. **√** 7. × 8. × 9. **√** 10.**√**

**四、简答题(共12分，意思正确给分)**

**1. 试比较顺序存储结构和链式存储结构的优缺点。在什么情况下用顺序表比链表好？（5分）**

**解答：**

① 顺序存储时，相邻数据元素的存放地址也相邻（逻辑与物理统一）；要求内存中可用存储单元的地址必须是连续的。

优点：存储密度大，存储空间利用率高。

缺点：插入或删除元素时不方便。

②链式存储时，相邻数据元素可随意存放，但所占存储空间分两部分，一部分存放结点值，另一部分存放表示结点间关系的指针。

优点：插入或删除元素时很方便，使用灵活。

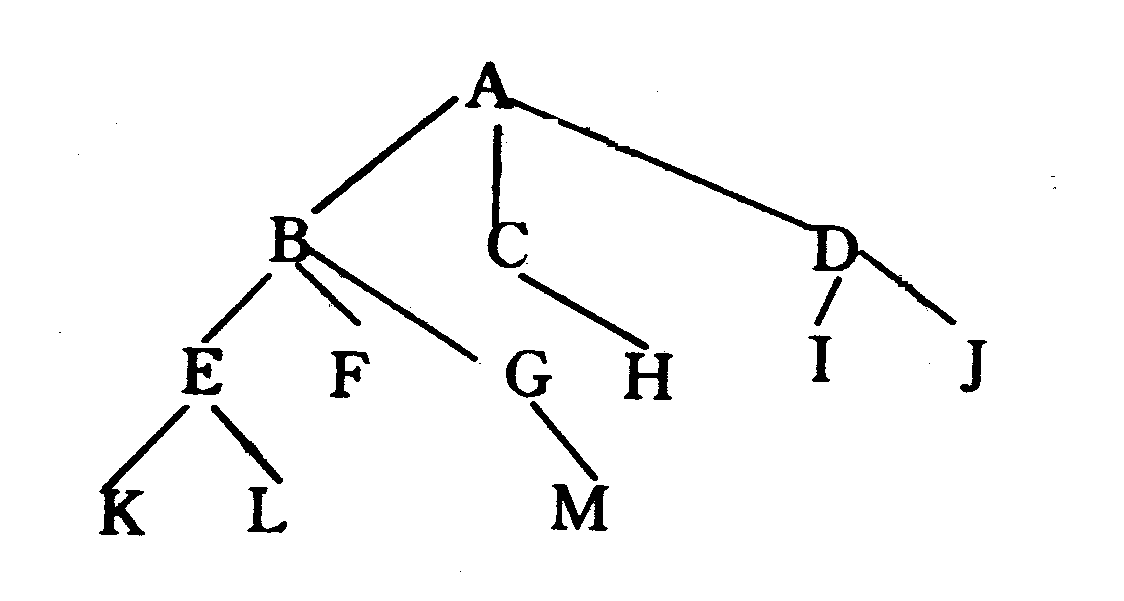
缺点：存储密度小（<1），存储空间利用率低。

顺序表适宜于做查找这样的静态操作；链表宜于做插入、删除这样的动态操作。

若线性表的长度变化不大，且其主要操作是查找，则采用顺序表；

若线性表的长度变化较大，且其主要操作是插入、删除操作，则采用链表。

**2.把如图所示的树转化成二叉树。（7分）**



**解答：**注意全部兄弟之间都要连线（包括度为2的兄弟）,并注意原有连线结点一律归入左子树，新添连线结点一律归入右子树。

**A**

**B**

**E C**

# K F H D

**L G I**

**M J**

**五、算法理解题：（共13分）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 顶点 | 1 | 2 | 3 | 4 | 5 | 6 |
| 入度 |  |  |  |  |  |  |
| 出度 |  |  |  |  |  |  |

**1、阅读算法，写出该算法的结果(4分)**

**解答：**

ABCDEFG（2分）

GFEDCBA（2分）

**2.设哈希（Hash）表的地址范围为0～17，哈希函数为：H（K）＝K MOD 16。K为关键字，用线性探测法再散列法处理冲突，输入关键字序列：（10，24，32，17，31，30，46，47，40，63，49）造出Hash表，试回答下列问题：（9分）**

1. **画出哈希表的示意图；（4分）**
2. **若查找关键字63，需要依次与哪些关键字进行比较？（2分）**
3. **假定每个关键字的查找概率相等，求查找成功时的平均查找长度。（3分）**

**解答：**

（1）画表如下：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 32 | 17 | 63 | 49 |  |  |  |  | 24 | 40 | 10 |  |  |  | 30 | 31 | 46 | 47 |

（2）查找63，首先要与H(63)=63%16=15号单元内容比较，即63 vs 31，no；然后顺移，与46,47,32,17,63相比，一共比较了6次！

（3）ASL=1/11（6＋2＋3×3）＝17/11=1.5454545454≈1.55

**六、算法设计题（两题共10分）**

**请编写一个递归算法，求二叉树的深度。（10分）**

**解答：**

**int** Depth (BiTree T )**{** // 返回二叉树的深度

**if** ( **!**T ) depthval = 0;

**else {**

m = Depth( T->**l**child );

n= Depth( T->**r**child );

**depthval = 1 + (m > n ? m : n);**

**}**

**return** depthval;

**}**