

石家庄铁道学院 2008-2009 学年第 1 学期

2007 级本科班期末考试试卷 B 卷-答案

课程名称: 数据结构 任课教师: 刘立嘉、姚雄伟 考试时间: 120 分钟

学号: _____ 姓名: _____ 班级: _____

考试性质 (学生填写): 正常考试 () 缓考补考 () 重修 () 提前修读 ()

题 号	一	二	三	四	五	六	七	八	九	总 分
满 分	20	20	10	10	10	10	10	10	10	100
得 分										
改卷人										

所有答案请写在答题纸上, 写在试卷或草稿纸上无效。

一、填空题 (每空 2 分, 共 20 分)

1. 在有 n 个结点的二叉树的二叉链表中, 非空链域的个数为 $2n-(n-1)=n+1$ 。
2. 一棵有 17 个叶子结点的哈夫曼树共有 $2*17-1=33$ 个结点。
3. 有关键字序列 {45, 20, 12, 10, 8, 2} 中, 若用二分查找法查找关键字 45, 则需比较 2 次。
4. 若前序遍历二叉树的结果为序列 A、B、C, 则有 5 棵不同的二叉树可以得到这一结果。
5. 若栈的入栈序列为 ABC, 则出栈序列不可能的是 BCA。

6. 求下列广义表操作的结果:

$\text{GetHead}[\text{GetTail}[\text{GetHead}[(a, b), (c, d)]]] = \underline{b};$

$\text{GetTail}[\text{GetHead}[\text{GetTail}[(a, b), (c, d)]]] = \underline{(d)}.$

7. 中缀表达式 $A*B+C$ 的后缀形式为 $AB*C+$; 中缀表达式 $(A+B)*D+E/(F+A*D)+C$ 的后缀形式为 $AB+D*EFAD*+/+C+$ 。

8. 如果已知一棵二叉树有 20 个叶子结点, 有 10 个结点仅有左孩子, 15 个结点仅有右孩子, 则该二叉树共有 64 个结点。

$$n_0=20, n_1=10+15=25, n_2=n_0-1=19, s=n_0+n_1+n_2=64$$

二、简答题 (每题 5 分, 共 20 分)

1. 简述算法应该具有的特征。

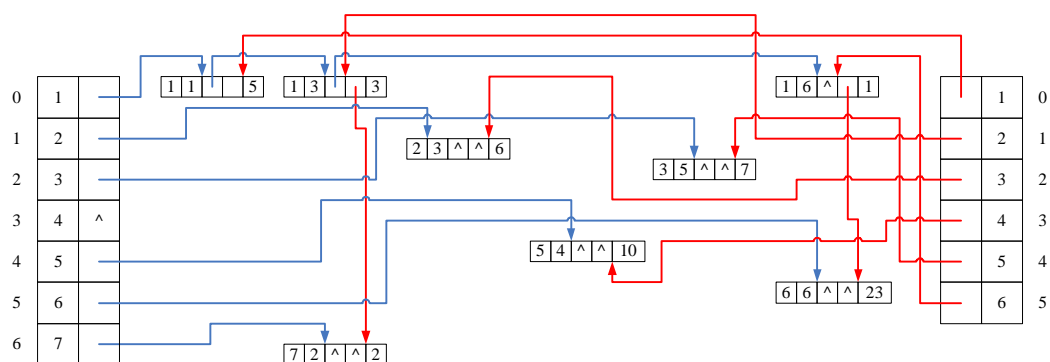
解答: 算法的基本特征包括以下几方面:

①、有穷性; ②、确定性; ③、可行性; ④、输入; ⑤、输出

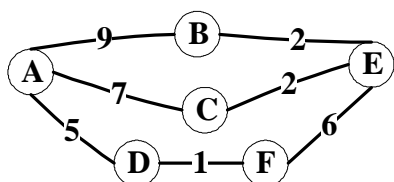
2. 请给出下面矩阵的十字链表存储结构。

5	0	3	0	0	1
0	0	6	0	0	0
0	0	0	0	7	0
0	0	0	0	0	0
0	0	0	10	0	0
0	0	0	0	0	23
0	2	0	0	0	0

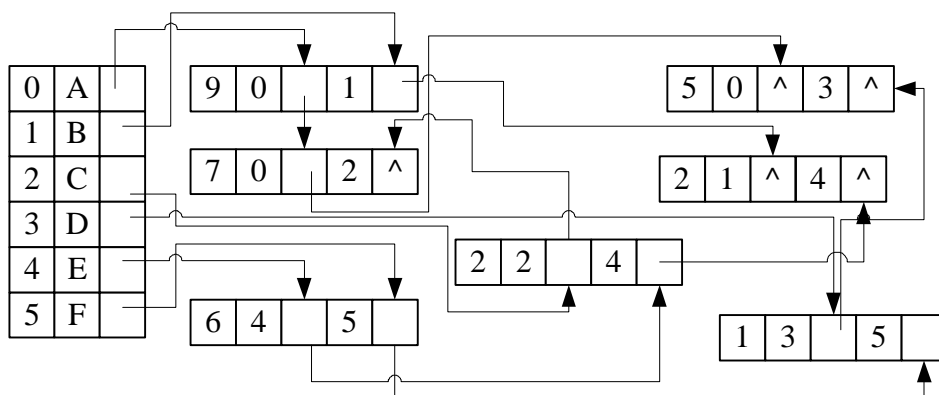
解答:



2. 请给出下面无向图的邻接多重表存储结构。

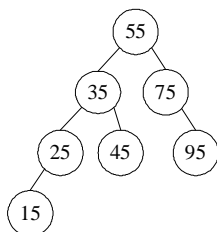
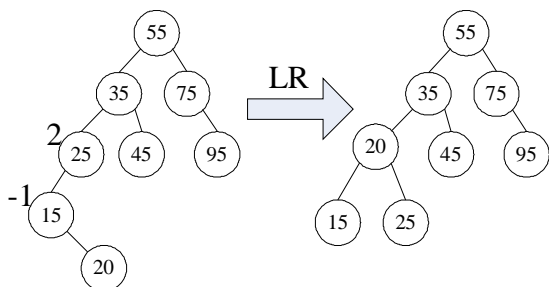


解答:



3. 下图一棵平衡二叉排序树，现向该二叉排序树中插入结点 20，请给出插入后的二叉排序树，并判断其是否失衡；若失衡请指出失衡的类型，并将其平衡化。

解答：



三、已知序列{50, 8, 90, 17, 40, 13, 01, 67, 76, 10}请给出采用快速排序法对该序列作降序排序的求解过程，快速时限定采用序列的首元素为轴。(10 分)

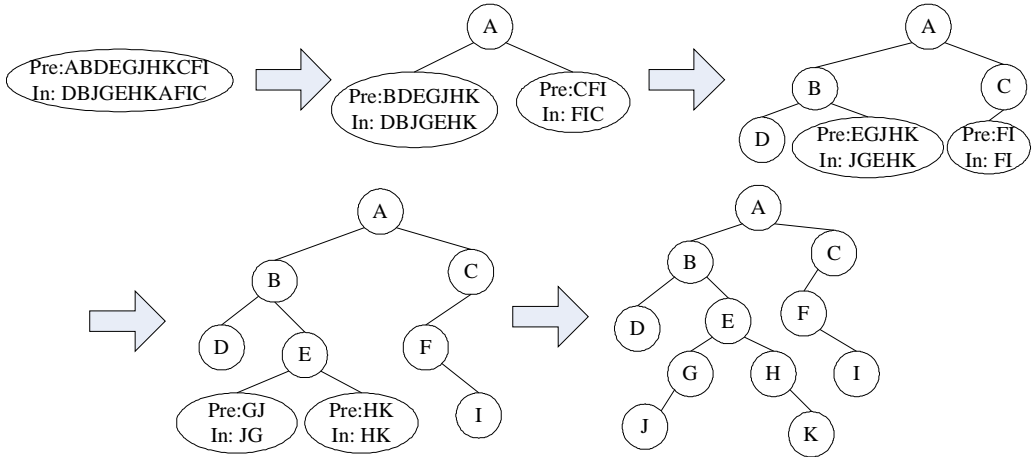
解答：

	pivot	0	1	2	3	4	5	6	7	8	9	low	high	
	50	50	8	90	17	40	13	1	67	76	10	0	9	
	50	10	8	90	17	40	13	1	67	76	10	0	9	
	50	10	8	90	17	40	13	1	67	76	10	2	9	
	50	10	8	90	17	40	13	1	67	76	90	2	6	
	50	10	8	1	17	40	13	1	67	76	90	2	6	
	50	10	8	1	17	40	13	1	67	76	90	6	6	
	50	10	8	1	17	40	13	50	67	76	90	6	6	必有
	10	10	8	1	17	40	13	50	67	76	90	0	5	
	10	10	8	1	17	40	13	50	67	76	90	0	2	
	10	1	8	1	17	40	13	50	67	76	90	0	2	
	10	1	8	10	17	40	13	50	67	76	90	2	2	
	10	1	8	10	17	40	13	50	67	76	90	2	2	必有
	1	1	8	10	17	40	13	50	67	76	90	0	1	
	1	1	8	10	17	40	13	50	67	76	90	0	0	

	17	1	8	10	17	40	13	50	67	76	90	3	5	
	17	1	8	10	13	40	13	50	67	76	90	3	5	
	17	1	8	10	13	40	13	50	67	76	90	4	5	
	17	1	8	10	13	40	40	50	67	76	90	4	5	
	17	1	8	10	13	17	40	50	67	76	90	4	5	必有
	67	1	8	10	13	17	40	50	67	76	90	7	9	
	67	1	8	10	13	17	40	50	67	76	90	7	7	
	67	1	8	10	13	17	40	50	67	76	90	7	7	
	76	1	8	10	13	17	40	50	67	76	90	8	9	
	76	1	8	10	13	17	40	50	67	76	90	8	8	
		1	8	10	13	17	40	50	67	76	90			必有

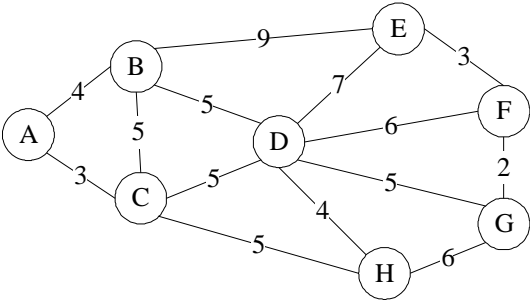
四、已知某二叉树的前序遍历 ABDEGJHKCFI，中序遍历 DBJGEHKAFIC，请画出该二叉树并求解该二叉树的后序遍历序列。要求将处理过程详细画出来。（10 分）

解答：



后序遍历: DJGKHEBIFCA

五、对于下图中的无向带权图，按 Prim 算法求最小生成树。要求将每一步的操作用表格列出，最后列出最小生成树的边集合。（10 分）

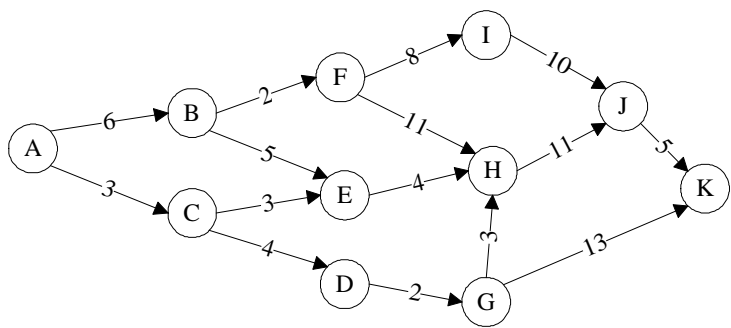


解答：

步骤	顶点 Closededge	B	C	D	E	F	G	H	U	V-U	选择
1	adj vex lowcost	A 4	A 3						{A}	{B, C, D, E, F, G, H}	C <A, C>
2	adj vex lowcost	A 4		C 5				C 5	{A, C}	{B, D, E, F, G, H}	B <A, B>
3	adj vex lowcost			C 5	B 9			C 5	{A, B, C}	{D, E, F, G, H}	D <C, D>
4	adj vex lowcost				D 7	D 6	D 5	D 4	{A, B, C, D}	{E, F, G, H}	H <D, H>
5	adj vex lowcost				D 7	D 6	D 5		{A, B, C, D, H}	{E, F, G}	G <D, G>
6	adj vex lowcost				D 7	G 2			{A, B, C, D, G, H}	{E, F}	F <G, F>
7	adj vex lowcost				F 3				{A, B, C, D, F, G, H}	{E}	E <E, F>

最小生成树的边集合： <A, C>, <A, B>, <C, D>, <D, H>, <D, G>, <G, F>, <E, F>

六、对于下图中的 AOE 网，求关键路径。要求用表格列出所有事件的最早发生时间和最迟发生时间，以及所有活动的最早和最迟开始时间，最后给出所有关键活动和关键路径。（10 分）



解答：

Vertex	Ve	VI
A	0	0
B	6	6
C	3	10
D	7	14
E	11	15
F	8	8
G	9	16
H	19	19

Activity	e	l	l-e
<A, B>	0	0	0
<A, C>	0	7	7
<B, F>	6	6	0
<B, E>	6	10	4
<C, E>	3	12	9
<C, D>	3	10	7
<F, I>	8	12	4
<F, H>	8	8	0

I	16	20
J	30	30
K	35	35

<E, H>	11	15	4
<D, G>	7	14	7
<G, H>	9	16	7
<I, J>	16	20	4
<H, J>	19	19	0
<G, K>	9	22	13
<J, K>	30	30	0

关键活动： <A, B>, <B, F>, <F, H>, <H, J>, <J, K>

关键路径： A-B-F-H-J-K

七、程序设计题（每题 10 分，共 20 分）

1. 请编写函数 `void Reverse(struct LNode *h)` 将给定的带头结点的单链表 `h`（`h` 指向头结点）原地逆置。链表结点类型为 `struct LNode`（数据域中包含一个 `struct LNode *next` 成员，用来指向下一个结点）。

例程：

`void Reverse(LNode *h)`

```
{
    LNode *p1,*p2;
    p1=h->next;
    h->next=NULL;
    while(p1){
        p2=p1;
        p1=p1->next;
        p2->next=h->next;
        h->next=p2;
    }
}
```

2. 请设计函数 `int getBranchNode(TNode *T)` 返回二叉树 `T` 的中的分支结点(即非叶结点)的数目,参数 `T` 指向树根。`TNode` 结构体表示二叉树结点,包含两个指针域 `TNode *left` 和 `TNode *right`, 分别指向左右子树。

例程：

`int getBranchNode(TNode *T)`

```
{
    int l,r;
    if(T->left==NULL && T->right==NULL) return 0;
```

```
if(T->left) l=getBranchNode(T->left);  
if(T->right) r=getBranchNode(T->right);  
return (l+r+1) ;  
}
```