

华中师范大学 2012-2013 学年第 1 学期 期末考试试卷 (B 卷)

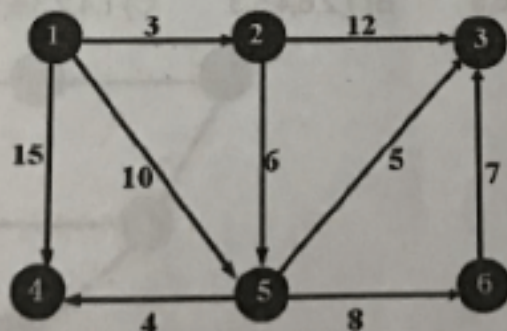
课程名称 数据结构 课程编号 84810007 任课教师 魏开平、王敬华、沈显君

题型	填空题	单选题	计算题	编程题		总分
分值	30	10	30	20		100
得分						

得分	评阅人

一、填空题。(每个空格 2 分, 共 30 分)

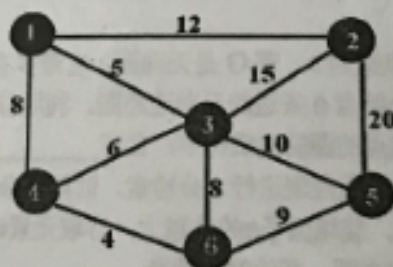
- 在线性表的顺序存储中, 元素之间的逻辑关系是通过_____决定的。
- 对于栈, 只能在_____插入和删除元素。
- 向顺序栈中插入新的元素分三步, 第一步进行栈满的判断; 第二步是修改栈顶指针; 第三步是把新元素赋给_____。
- 已知广义表 $A=(a,b)$, $B=(A,A)$, $\text{GetHead}(\text{GetHead}(\text{GetTail}(B)))=$ _____。
- 设有两个串 p 和 q , 求 q 在 p 中首次出现的位置的运算叫_____。
- 二维数组 A 的每个元素是六个字符组成的串, 行下标的范围从 0-9, 列下标的范围从 0-19, 存放 A 至少需要_____字节。
- 对于一个具有 n 个结点的二叉树, 当它为一棵_____二叉树时具有最小高度。
- 对于一棵具有 n 个结点的二叉树, 当进行二叉链式存储时, 其二叉链表中的空指针域的总数为_____个。
- 设图 G 的结点数为 n , 若 G 是无向图, 它最多有_____条边。
- 对于含有 n 个结点 e 条边的无向连通图, 利用普里姆算法生成最小生成树的时间复杂度为_____。
- 在具有 n 个结点的图的生成树中, 含有_____条边。
- 对于长度为 n 的线性表进行二分检索, 时间复杂度为_____。
- 在散列存储中, 装填因子 α 的值越大, 存取元素时发生冲突的可能性就_____。
- 若 G 是有向完全图, 其边的总数是_____。
- 如下图所示, 则从结点 1 到图中每个结点的最短路径长度之和为_____。



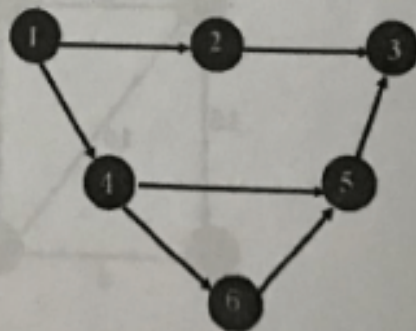
得分	评阅人

二、单选题。(每小题1分,共10分)

16. 设单链表中指针 p 指向结点 m , 若要删除 m 之后的结点(若存在), 则需修改指针的操作为_____。
 A) $p \rightarrow next = p \rightarrow next \rightarrow next$ B) $p = p \rightarrow next$
 C) $p = p \rightarrow next \rightarrow next$ D) $p \rightarrow next = p$
17. 一个中缀算术表达式为 $1+(3-x)y$, 则其对应的后缀表达式为_____。
 A) $13+x-y^*$ B) $13x+-y^*$ C) $13x-y^*+$ D) $13xy-+^*$
18. 在具有 n 个单元的顺序存储的循环队列中, 假定 $front$ 和 $rear$ 分别为队头指针和队尾指针, 则队列为空的条件为_____。
 A) $rear \% n = front$ B) $front \% n + 1 = rear$
 C) $rear \% n - 1 = front$ D) $rear \% n + 1 = front$
19. 在一个链队列中, 假定 $front$ 和 $rear$ 分别为队头和队尾指针, 则插入 s 指针所指向的结点的操作是_____。
 A) $front \rightarrow next = s; front = s$ B) $s \rightarrow next = rear; rear = s$
 C) $rear \rightarrow next = s; rear = s$ D) $s \rightarrow next = front; front = s$
20. 假定在一棵二叉树中, 双分支结点数为 15 个, 单分支结点数为 16 个, 则叶子结点数为_____个。
 A) 15 B) 16 C) 17 D) 45
21. 在一棵具有 k 层的满二叉树中, 结点总数为_____。
 A) $(2^k - 1)/2$ B) $2^k - 1$ C) $(2^k - 1)/3$ D) 2^k
22. 任何一棵二叉树的叶结点在先序、中序和后序遍历序列中的相对次序_____。
 A) 不发生改变 B) 发生改变 C) 不能确定 D) 以上都不对
23. 如果要求一个线性表既能较快的检索, 又能适应动态变化的要求, 则宜采用的检索方法为_____。
 A) 分块检索 B) 顺序检索 C) 折半检索 D) 基于属性检索
24. 已知一个图如下所示, 在该图的最小生成树中各条边上权值之和为_____。
 A) 31 B) 38 C) 36 D) 43



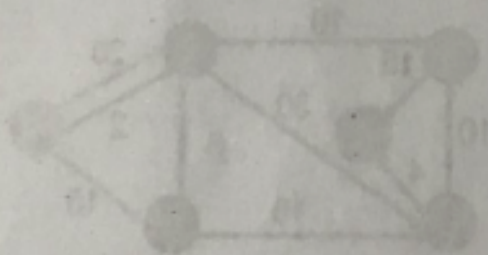
25. 已知一个图如下所示, 则由该图得到的一种拓扑序列为_____。
 A) 1,2,3,4,6,5 B) 1,2,6,4,5,3 C) 1,4,2,3,6,5 D) 1,4,6,5,2,3



得分	评阅人

三. 计算题。(每小题 8 分, 共 40 分)

26. 假定用于通信的电文由 8 个字母 A,B,C,D,E,F,G,H 组成, 各字母在电文中出现得概率为 5%, 22%, 4%, 7%, 9%, 15%, 30%, 8%, 试为这 8 个字母设计赫夫曼编码。
27. 已知一组数据的输入顺序为: 6, 3, 5, 7, 8, 0, 1, 4, 9, 试为该序列建立二叉查找树, 写出该二叉查找树, 并求出该树的平均查找长度。

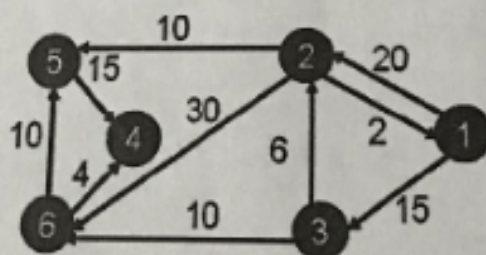


28. 选取散列函数 $H(\text{key}) = (3 * \text{key}) \% 11$, 用线性探测法处理冲突, 对关键码序列 [22, 41, 53, 08, 46, 30, 01, 31, 66] 构造一个哈希(Hash)表, 散列地址空间为 0~10。

29. 以关键字序列(256,301,751,129,037,863,742,694,076,438)为例, 分别写出执行堆排序算法的各趟排序结束时, 关键字序列的状态。

30. 如下所示的带权有向图 G,

- (1) 画出图 G 的邻接矩阵;
- (2) 分别给出从结点 1 出发按深度和广度优先搜索遍历 G 所得的结点序列;
- (3) 用 Dijkstra 算法求从结点 1 到其它各结点的最短路径 (要求写出最短路径值和相对应的路径)。



31. 设 h 为带表头结点的双向链表的头指针, 请编写一个删除表中数据域值为 x 的所有结点的算法。
注: 链表结点类型如下

```

struct Node {
    int data;
    struct Node *next;

```

```

};

```

算法函数说明: void del_x(Node *h, int x);

32. 试写一个判别给定二叉树是否为二叉排序树的算法，设此二叉树以二叉链表作存储结构，且树中结点的关键字均不同。

注：二叉树结点类型如下

```
struct Bitree {  
    int data;  
    struct Bitree *lchild, *rchild;  
};
```

算法函数说明：int IsBSTree(Bitree *T);

一、填空题

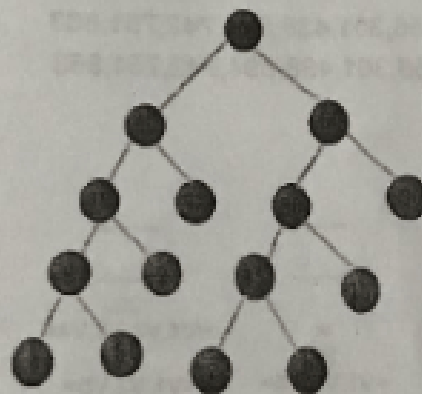
1. 物理地址
2. 栈顶
3. 栈顶指针所指的位置
4. a
5. 模式匹配
6. $10 \times 20 \times 6 = 1200$
7. 完全二叉树
8. $n+1$
9. $n \times (n-1)/2$
10. $O(n \log(n))$
11. $n-1$
12. $O(\log(n))$
13. 越大
14. $n \times (n-1)$
15. $11+3+9+14+18=55$

二、单选题

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

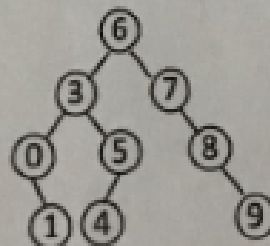
三、计算题

1. 解



A: 0001 B: 01 C: 0000 D: 1000 E: 001 F: 101 G: 11 H: 1001

2. 解



$$ASL = (1 + 2 \times 2 + 3 \times 3 + 4 \times 3) / 9 = 26/9$$

3. 解

由题意知, $m=11$ (刚好为素数)

则 $(22*3)\%11=0$
 $(53*3)\%11=5$
 $(46*3)\%11=8$
 $(01*3)\%11=3$
 $(66*3)\%11=9$

$(41*3)\%11=2$
 $(08*3)\%11=2$
 $(30*3)\%11=8$
 $(31*3)\%11=5$

22	66	41	8	30	53	46	1	31		
0	1	2	3	4	5	6	7	8	9	10
1		3	4,7							

4. 解

堆排序(大根堆)的中间过程如下:

初始: 256,301,751,129,037,863,742,694,076,438
 第1趟: 863,694,751,301,438,256,742,129,076,037
 第2趟: 751,694,742,301,438,256,037,129,076,863
 第3趟: 751,694,742,301,438,256,037,129,076,863
 第4趟: 742,694,256,301,438,076,037,129,751,863
 第5趟: 694,438,256,301,129,076,037,742,751,863
 第6趟: 438,301,256,037,129,076,694,742,751,863
 第7趟: 301,129,256,037,076,438,694,742,751,863
 第8趟: 256,129,076,037,301,438,694,742,751,863
 第9趟: 129,037,076,256,301,438,694,742,751,863
 第10趟: 076,037,129,256,301,438,694,742,751,863
 第11趟: 037,076,129,256,301,438,694,742,751,863

5. 解

20	20			
<V1,V2>	<V1,V2>			
15				
<V1,V3>				
			29	
			<V1,V3,V6,V4>	
		30	30	30
		<V1,V2,V5>	<V1,V2,V5>	<V1,V2,V5>
	25	25		
	<V1,V3,V6>	<V1,V3,V6>		
V3:15	V2:20	V6:25	V4:29	V5:30
<V1,V3>	<V1,V2>	<V1,V3,V6>	<V1,V3,V6,V4>	<V1,V2,V5>

四、编程题

1. 解

```
void del_x( Node *h, int x )
{
    Node *p,*q;
    p=h->next;
    while(p!=h) {
        if (p->data==x) {
```

```

        q=p->next;
        p->prior->next=p->next;
        p->next->prior=p->prior;
        free(p);
        p=q;
    }
    else p=p->next;
}
}

```

2. 解

```

int last=0, flag=1;
int IsBSTree(Bitree *T)
{
    if ( T->lchild&&flag ) Is_BSTree(T->lchild);
    if ( T->data<last ) flag=0;    //与其中序前驱相比较
    last = T->data;
    if ( T->rchild&&flag ) Is_BSTree(T->rchild);
    return flag;
}

```