

石家庄铁道学院 2009-2010 学年第 1 学期

2008 级本科班期末考试试卷（B）答案

课程名称：数据结构 B 任课教师：武守秋、邸书灵 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	总分
满 分	20	20	45	15				100
得 分								
阅卷人								

一、单项选择题（每小题 2 分，共 20 分）

1. 执行下面程序段时，执行 S 语句的次数为 A。

```
for ( int i = 1; i <= n; i++ )  
    for ( int j = 1; j <= n; j++ )  
        S;
```

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

2. 用链表表示线性表的优点是 C。

(A)便于随机存取 (B)花费的存储空间比顺序表少
(C)便于插入与删除 (D)数据元素的物理顺序与逻辑顺序相同

3. 对于只在表的首、尾两端进行插入操作的线性表，宜采用的存储结构为 C。

(A) 顺序表 (B) 用头指针表示的循环单链表
(C) 用尾指针表示的循环单链表 (D) 单链表

4. 一个栈的输入序列为 12345，则下列序列中是栈的输出序列的是 A。

(A)23415 (B)54132 (C)31245 (D)14253

5. 队列的插入操作在 B 进行。

(A)队头 (B)队尾 (C)任意位置 (D)指定位置

6. 设循环队列中数组的下标范围是 $1 \sim n$ ，其头尾指针分别为 f 和 r，则其元素个数为 D。

(A)r-f (B)r-f+1 (C) (r-f) mod n +1 (D) (r-f+n) mod n

7. 数组 A[1..5,1..6]的每个元素占 5 个单元，将其按行优先顺序存储在起始地址为 1000 的连续的内存单元中，则元素 A[5,5]的地址为 A。

(A)1140 (B)1145 (C)1120 (D)1125

8. 设森林 F 中有 4 棵树，第 1、2、3、4 棵树的结点个数分别为 n_1 、 n_2 、 n_3 、 n_4 ，当把森林 F 转换成一棵二叉树后，其根结点的左子树中有 A 个结点。

(A) n_1-1 (B) n_2 (C) n_3 (D) n_4

9. 有 n 个顶点的强连通有向图 G 至少有 C 条弧。

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

10. 路径长度 A 的路径是关键路径。

(A)最长 (B)最短

二、填空题（每空 2 分，共 20 分）

1. 算法具有输入、输出、确定性、（有穷性）和可执行性等特性。

2. 在顺序表中插入或删除一个元素，需要平均移动一半元素，具体移动的元素个数与插入或删除的位置有关。插入时平均次数 $n/2$ ，删除时平均次数 $((n-1)/2)$ 。

3. 设有一个顺序栈 S ，元素 $s_1, s_2, s_3, s_4, s_5, s_6$ 依次进栈，如果 6 个元素的出栈顺序为 $s_2, s_3, s_4, s_6, s_5, s_1$ ，则顺序栈的容量至少应为多少？（3）

4. 设 $s = \text{'I AM A STUDENT'}$, $t = \text{'GOOD'}$, $q = \text{'WORKER'}$ 。SubString(sub1, s, 1, 7) 操作的结果：sub1 = ('I AM A')。

5. 假设有 6 行 8 列的二维数组 A ，每个元素占用 6 个字节，存储器按字节编址。已知 A 的基地址为 1000，数组 A 共占用多少字节（288）。

6. 在一棵二叉树中，假定度为 2 的结点有 5 个，度为 1 的结点有 6 个，则叶子结点数有（6）个。

7. 设二叉树有 n 个结点且根结点处于第 1 层，则其高度为（不确定）。

8. 有向图的极大强连通子图被称为有向图的（强连通分量）

9. 一个连通图的（生成树）是一个极小连通子图，含有图中全部顶点，但是只有构成一棵树的 $n-1$ 条边。

10. 在赫夫曼编码中，若编码长度只允许小于等于 4，则除了已对两个字符编码为 0 和 10 外，还可以最多对（4）个字符编码。

三、简答题（共 45 分）

1. 设 n 为正整数，分析下列各程序段中加下划线的语句的程序步数。

```
for (int i = 1; i <= n; i++)  
    for (int j = 1; j <= n; j++) {  
        c[i][j] = 0.0;  
        for (int k = 1; k <= n; k++)  
            c[i][j] = c[i][j] + a[i][k] * b[k][j];  
    }
```

【答案】 $\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n 1 = n^3$

2. 下面是一个在链表中删除结点的算法，请填上相应语句。

Status ListDelete_L(LinkList L, int i, ElemType &e)

{

// 删除以 L 为头指针(带头结点)的单链表中第 i 个结点

$p = L;$ $j = 0;$

```

while (p->next && j < i-1) // 寻找第 i 个结点，并令 p 指向其前趋
{ p=p->next; ++j; }
if (!(p->next) || j > i-1) return ERROR; // 删除位置不合理
q = p->next; p->next = q->next; // 删除并释放结点
e = q->data; free(q);
return OK;
} // ListDelete_L

```

3. 下面两个算法分别是在循环队列中插入新的元素以及删除元素，请填上相应语句。

---循环队列—队列的顺序存储结构

```

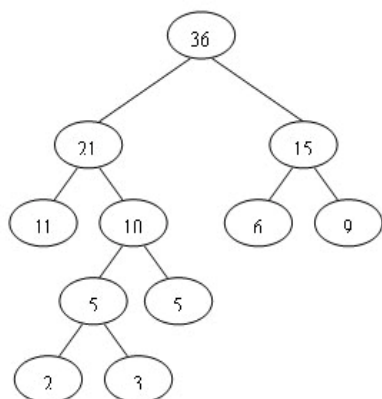
#define MAXQSIZE 100 //最大队列长度
typedef struct {
    QElemType *base; // 初始化的动态分配存储空间
    int front; // 头指针，若队列不空，指向队列头元素
    int rear; // 尾指针，若队列不空，指向队列尾元素的下一个位置
} SqQueue;

Status EnQueue (SqQueue &Q, QElemType e) {
// 插入元素 e 为 Q 的新的队尾元素
    if (((Q.rear+1) % MAXQSIZE == Q.front))
        return ERROR; //队列满
    Q.base[Q.rear] = e;
    Q.rear = (Q.rear+1) % MAXQSIZE;
    return OK;
}

Status DeQueue (SqQueue &Q, QElemType &e) { // 若队列不空，则删除 Q 的队头元素，
// 用 e 返回其值，并返回 OK; 否则返回 ERROR
    if (Q.front == Q.rear) return ERROR;
    e = Q.base[Q.front];
    Q.front = (Q.front+1) % MAXQSIZE;
    return OK;
}

```

4. 已知叶子结点值 2, 3, 5, 6, 9, 11, 构造哈夫曼树，计算其带权路径长度。



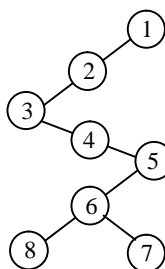
$$WPL=11\times 2+6\times 2+9\times 2+5\times 3+2\times 4+3\times 4=87$$

5. 二叉树的先序序列为 1, 2, 3, 4, 5, 6, 8, 7; 中序序列为 3, 4, 8, 6, 7, 5, 2, 1, 请画出这棵二叉树。

【解答】

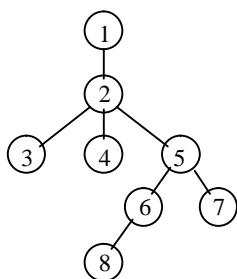
|

二叉树

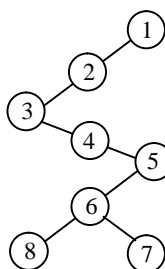


6. 请画出右图所示的树所对应的二叉树。

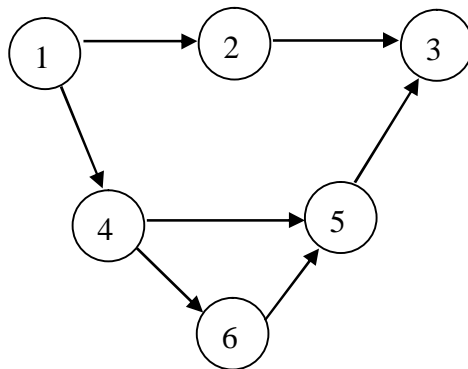
【解答】



对应二叉树

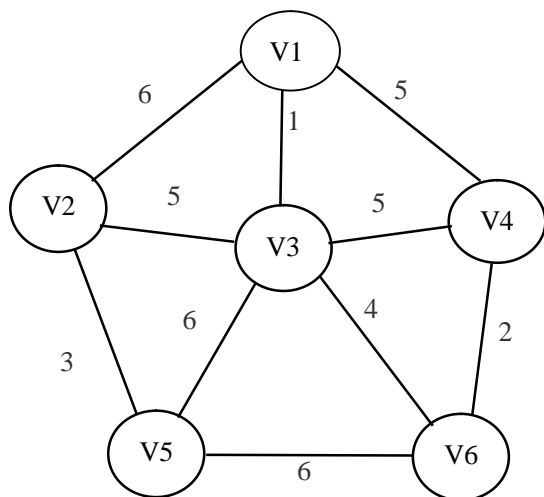


7. 已知一个有向图如下所示, 写出由该图得到的所有拓扑序列。(5 分)



v1 v4 v6 v2 v5 v3

8. 请用普里姆算法构造一棵最小生成树。



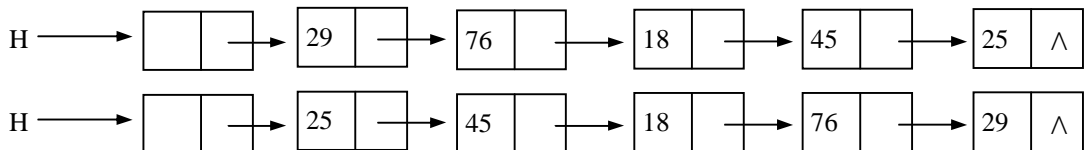
	0 (v1)	1 (v2)	2 (v3)	3 (v4)	4 (v5)	5 (v6)	U	V-U	k
adjvex lowcost		V1 6	V1 1	V1 5			{v1}	{v2,v3,v4,v5,v6}	2
adjvex lowcost		V3 5		V1 5	V3 6	V3 4	{v1,v3}	{v2,v4,v5,v6}	5
adjvex lowcost		V3 5		V6 2	V3 6		{v1,v3,v6}	{v2,v4,v5}	3
adjvex lowcost		V3 5			V3 6		{v1,v3,v6,v4}	{v2,v5}	1

adjvex				V2		{v1,v3,v6,v4,v2}	{v5}	4
lowcost		0	0	0	3	0		
adjvex						{v1,v3,v6,v4,v2,v5}	{ }	
lowcost		0	0	0	0	0		

四、算法设计（共 10 分）

已知单链表 H，写一算法将其倒置。即实现如图的操作。

(a)为倒置前，(b)为倒置后。（有头结点）



void reverse (Linklist H)

```
{ LNode *p;
  p=H->next; //p 指向第一个结点
  H->next=NULL; //将原链表置为空表 H
  while (p)
  { q=p; //q 始终指向当前插入结点
    p=p->next; //p 指向下一插入结点
    q->next=H->next; //将当前结点插到头结点的后面
    H->next=q;
  }
}
```

2. 称正读和反读都相同的字符序列为“回文”，例如，“abccddcba”、“qwerewq”是回文，“ashgash”不是回文。是写一个算法判断读入的一个以‘@’为结束符的字符序列是否为回文。

Boolean IsPalindrome()

```
{
  InitStack(S);
  InitQueue(Q);
  while((c=read())!='@')
  {
    Push(S,c);EnQueue(Q,c);
  }
  while(!StackEmpty(S)&&!QueueEmpty(Q))
  {
    if(Pop(S)!=DeQueue(Q)) return false;
  }
}
```