姓名

班级

学号

密

封

线

计算机图形学试题 **B(**卷一**)**答案

（适用班级：0520541、542、551）

2005/2006 学年第二学期期末试题

题号	一	二	三	四	五	六	七	总分
得分								
评卷人								

一)用 Bresenham 算法计算（0，0）到（5，2）的直线并将结果填写下表。（15 分）

坐标 x	坐标 y	判别式 d
0	0	1
1	0	-3
2	1	3
3	1	-1
4	2	3
5	2	-1

二）如图 1 所示多边形，若采用 ET 边表算法进行填充，试写出该多边形的 ET 表和当扫描线 Y=3 时的有效边表（AET 表）。（15 分）

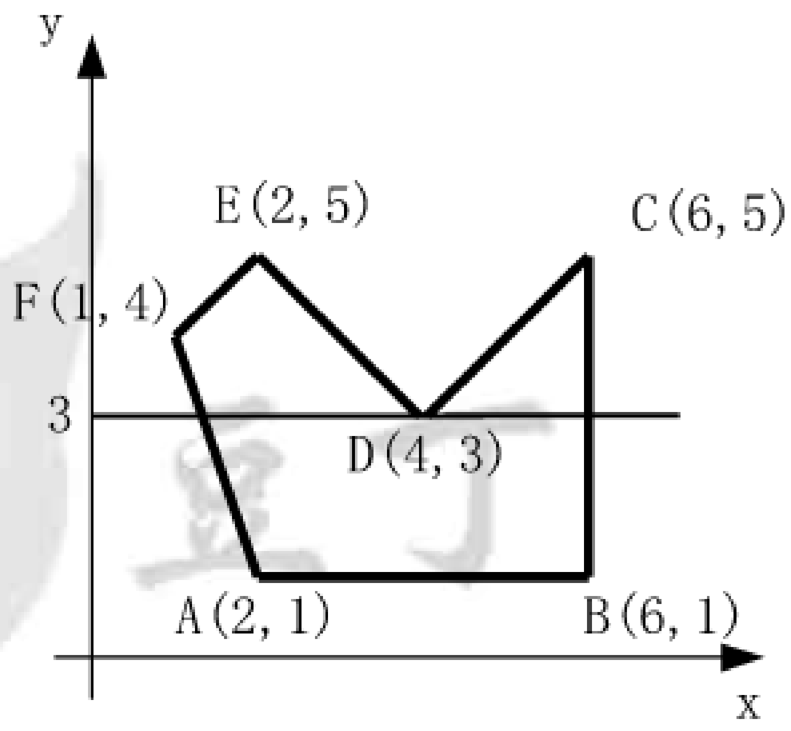
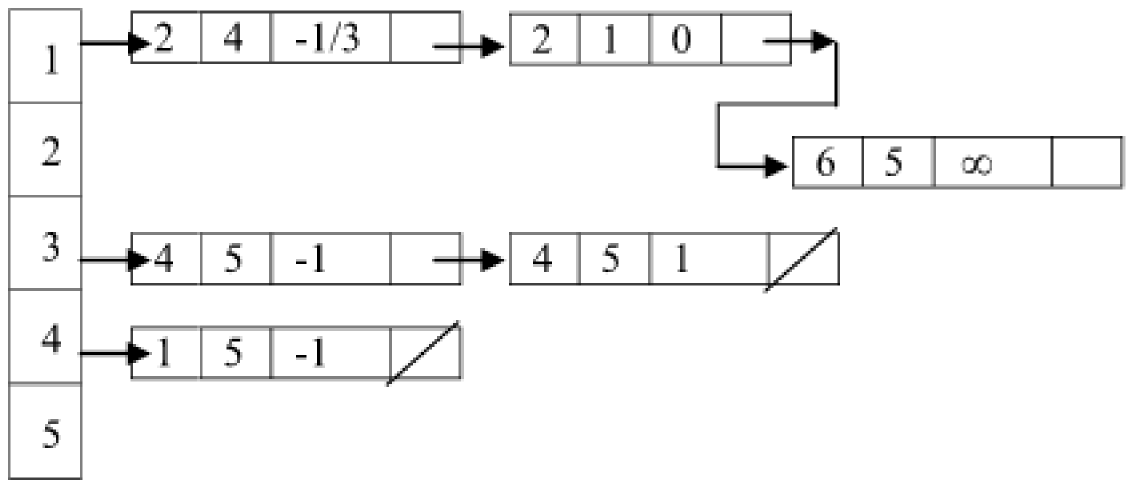
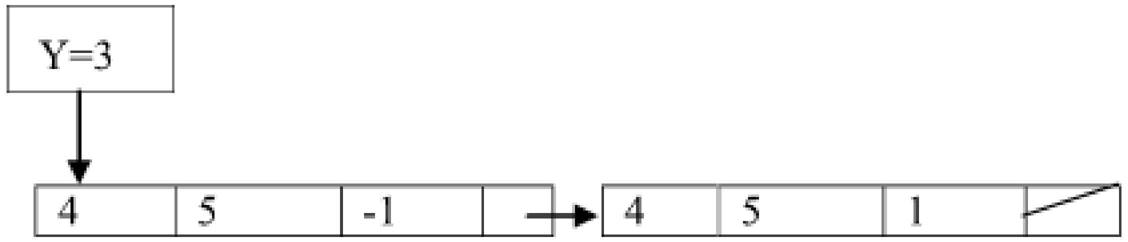


图 1

ET 表



AET 表



三）用梁友栋算法裁减如图 2 线段 AB，A、B 点的坐标分别为(3,3)、(-2,-1) 裁剪窗口为 wxl=0，wxr=2，wyb=0，wyt=2。(10 分)

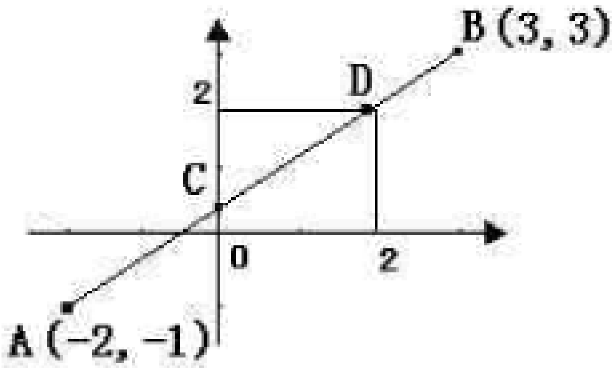


图 2

解：以 A（3，3）为起点，B(-2，-1)为终点

所以有 $x1=3$，$y1=3$，$x2=-2$，$y2=-1$，$wxl=0$，$wxr=2$，$wyb=0$，$wyt=2$

构造直线参数方程：

$x=x1+u(x2-x1) \quad (0<=u<=1)$

$y=y1+u(y2-y1)$

把 $x1=3$，$y1=3$，$x2=-2$，$y2=-1$ 代入得

$x=3-5u$

$y=3-4u$

计算各个 p 和 q 值有：

$p1=x1-x2=5 \quad q1=x1-wxl=3$

$p2=x2-x1=-5 \quad q2=wxr-x1=-1$

$p3=y1-y2=4 \quad q3=y1-wyb=3$

$p4=y2-y1=-4 \quad q4=wyt-y1=-1$

根据，$uk=qk/pk$ 算出

$pk<0$ 时：$u2=1/5$ $u4=1/4$

$pk>0$ 时：$u1=3/5$ $u3=3/4$

$umax=MAX(0,u2,u4)=MAX(0,1/5,1/4)=1/4$ （取最大值）

$umin=MIN(u1,u3,1)=MIN(3/5,3/4,1)=3/5$ （取最小值）

由于 $umax<umin$ ，故此直线 AB 有一部分在裁减窗口内，

$pk<0$ 时，将 $umax=1/4$ 代入直线参数方程

$x=x1+u(x2-x1)$

$x=3+1/4*(-5)=3-5/4=7/4$

$y=y1+u(y2-y1)$

$y=3+1/4*(-4)=2$

求出直线在窗口内部分的端点 C(7/4,2)

pk>0 时，将 umin=3/5 代入直线参数方程

x=x1+u(x2-x1)

x=3+3/5*(-5)=0

y=y1+u(y2-y1)

y=3+3/5*(-4)=3/5

求出直线在窗口内部分的端点 D(0,3/5)。

所以，直线在窗口内部分的端点为 C(7/4,2)，D(0,3/5)。

四）如图 3 所示三角形 ABC，将其关于 A 点逆时针旋转 900，写出其变换矩阵。（10 分）

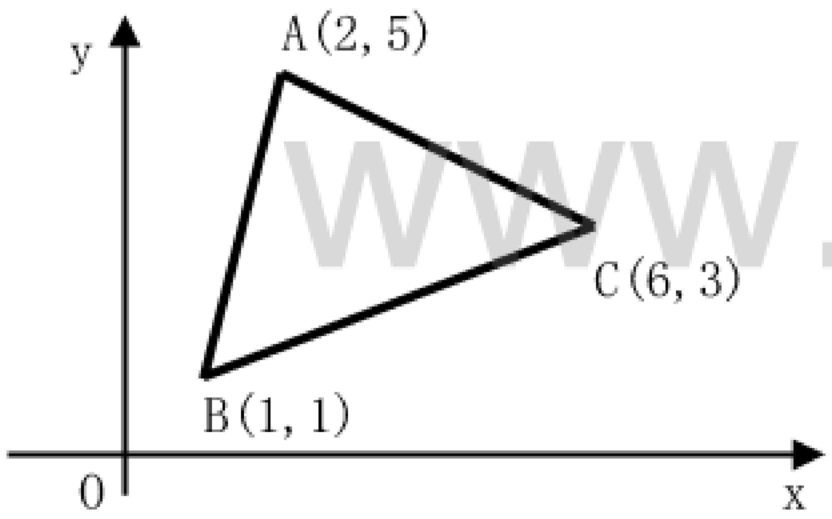


图 3

解：1)三角形绕 B 点(2,5)旋转 θ 的变换矩阵

T=Tt * TR * Tt-1

平移到坐标原点	旋转角度 θ	反平移回原来位置
1 0 0 0 1 0 -2 -5 1	cosθ sinθ 0 -sinθ cosθ 0 0 0 1	1 0 0 0 1 0 2 5 1

2)三角形绕 B 点顺时针旋转 90 度的变换矩阵，θ=-90°

T=Tt * TR * Tt-1

平移到坐标原点	旋转角度 θ	反平移回原来位置
1 0 0 0 1 0 -2 -5 1	cos90° -sin90° 0 sin90° cos90° 0 0 0 1	1 0 0 0 1 0 2 5 1

变换过程：三角形 ABC 的规范化齐次坐标(x,y,1) * 3 阶二维变换阵

P=P * T

得到三角形 ABC 变换后的规范化齐次坐标(x',y',1)

顶点	x y 1
A	4.6 2 1
B	2 5 1
C	0 -1 1

可以写出顶点坐标：A'(4.6,2) B'(2,5) C'(0,-1)

五）将图 4 中的空间四面体关于 E 点整体放大两倍，写出变换矩阵以及变换后图形各点的规范化齐次坐标。（20 分）

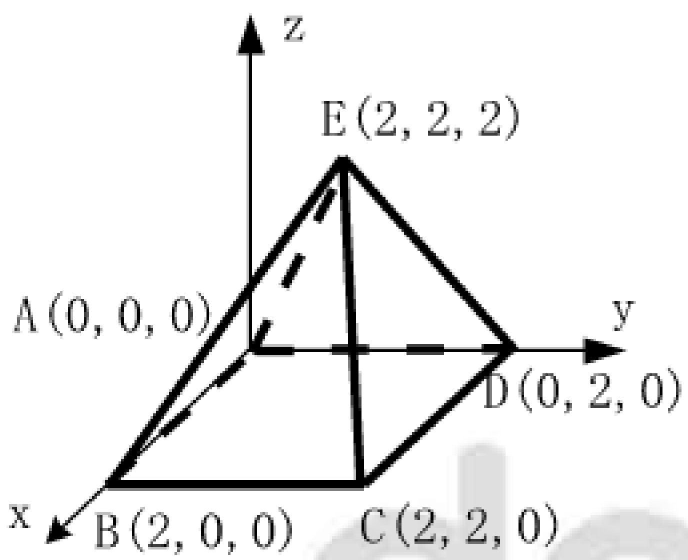


图 4

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \bullet T_s = [x \quad y \quad z \quad 1] \bullet \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix} =$$

$$[2x \quad 2y \quad 2z \quad 1]$$

A(0,0,0) B(4,0,0) C(4,4,0) D(0,4,0) E(4,4,4)

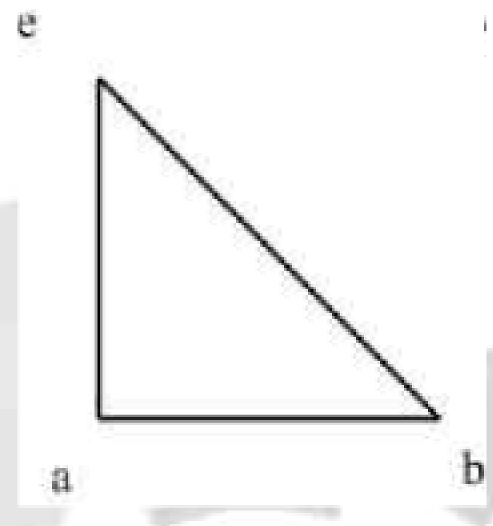
六）试作出图 4 中三维形体 ABCDE 的三视图（平移矢量均为 1）。要求写清变换过程，并画出生成的三视图。（10 分）

主视图：

将 xOz 面（又称 V 面）做垂直投影，的到主视图。由投影变换前后三维形体上点到主视图上的点的关系，次投影变换矩阵应为：

$$T_V = T_{XOZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

变换后点 A(0，0，0) B(2,0,0) C(2,0,0) D(0,0,0) E(2,0,2)



2 俯视图

三维形体向 xOy 面（又称 H 面）作垂直投影的到俯视图，其投影变换矩阵应为：

$$T_{XOY} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

为了使俯视图与主视图都画在一个平面内，就要使 H 面绕 x 轴旋转负 90^0 即应有一个旋转变换，其变换矩阵为：

$$T_{Rx} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-90^0) & \sin(-90^0) & 0 \\ 0 & -\sin(-90^0) & \cos(-90^0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

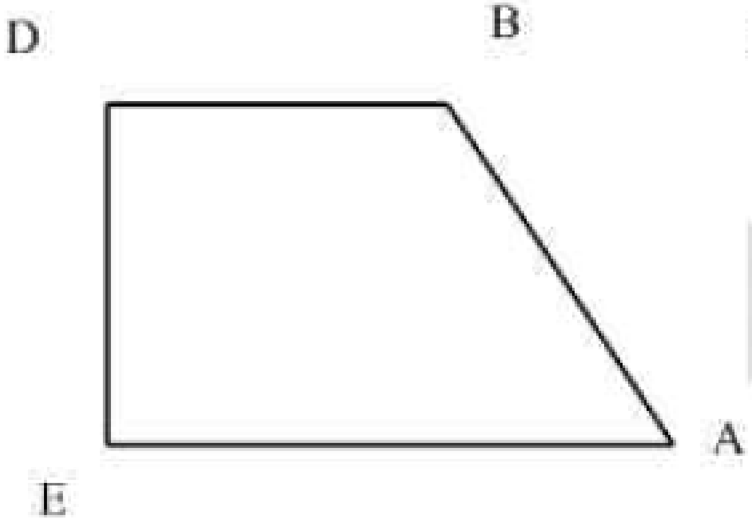
为了使主视图和俯视图间有一定的间距，还要使 H 面沿 z 方向平移一段距离-z_0，其变换矩阵为：

$$T_{XOY}=\begin{bmatrix}1&0&0&0\\0&0&-1&0\\0&0&0&0\\0&0&-z_0&1\end{bmatrix}$$

于是，俯视图的投影变换矩阵为上面三个矩阵的乘积

变换后点 A(0,0,0)　B(2,0,-1)　C(2,0,-3)

D(0,0,-3)　E(2,0,-3)



3 侧视图

获得侧视图是将三维形体往 yOz 面（侧 W）作垂直投影，所以侧视图的投影变换矩阵为：

$$T_{yOz}=\begin{bmatrix}0&0&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

为使侧视图与主视图也在一个主平面内，就要使 W 面绕 z 轴正转 90^0，其旋转变换矩阵为：

$$T_{Rz}=\begin{bmatrix}\cos(90^0)&\sin(90^0)&0&0\\-\sin(90^0)&\cos(90^0)&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}=\begin{bmatrix}0&0&0&0\\-1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

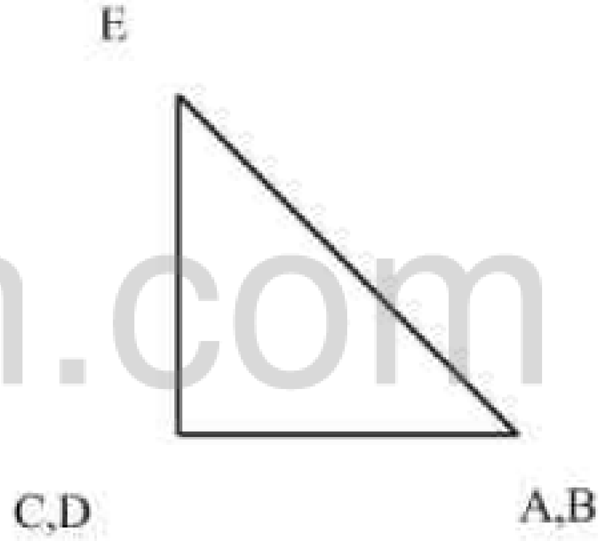
为了使主视图和侧视图有一定的间距，还要使 W 面沿-x 方向平移一段 x_0 该平移矩阵为：

$$T_{tx}\begin{bmatrix}0&0&0&0\\-1&0&0&0\\0&0&1&0\\-x_0&0&0&1\end{bmatrix}$$

于是，侧视图的投影变换矩阵为上面三个变换矩阵的连乘积：

变换后的点:A(-1,0,0) B(-1,0,0)　C(-3,0,0)　D(-3,0,0)

E(-3,0,2)



七）用编码裁剪算法裁剪线段 P1(0,2)，P2(3,3)，裁剪窗口为 wxl=1，wxr=4，wyb=1，wyt=4，要求写出：（20）

1)窗口边界划分的 9 个区间的编码原则；

首先对直线段的端点进行编码，即对直线段的任一端点(x,y),根据其坐标所在的区域，赋予一个四位的二进制码 D3D2D1D0

若 x<wxl,则 D0=1,否则 D0=0;

若 x>wxr,则 D1=1,否则 D1=0;

若 y<wyb,则 D_2=1,否则 D_2=0;

若 y>wyt,则 D_3=1,否则 D_3=0;

2)线段端点的编码；

code1=0001

code2=0000

3)裁剪的主要步骤；

(1)输入直线的两端点坐标：p1(x1,y1),p2(x2,y2),以及窗口的四边界坐标：wyt,wyb,wxl 和 wxr

(2)对 p1,p2 进行编码：点 p1 的编码为 code1,点 p2 的编码为 code2.

(3)若 code1|code2=0，对直线应减取之，转(6)；否则，若 code1&code2 ≠ 0，对直线段可简弃之，转(7)；当上述两条均不满足时，进行步骤(4).

(4)确保 p1 在窗口外部：若 p1 在窗口内，则交换 p1 和 p2 的坐标值和编码。

(5)按左,右，下，上的顺序检查编码并要求出直线段与窗口边界的交点，用该交点的坐标值替换 p1 的坐标值。也即在交点，假定为 S,S 处把线段一分为二，并去调 p1S 这一段(考虑 p1 是窗口外的一点,因此可以去掉 p1S) 转(2).

(6)用直线扫描转换算法画出当前的 p1p2.

(7)算法结束。

4) 裁剪的输出结果。