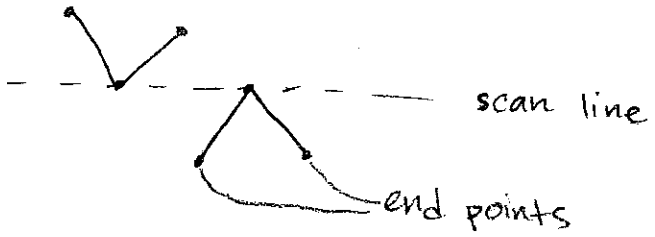
# 1

a) case 1:




- If both end points are above or below the scan line, then count as 2 intersections (this case is handled by default).

b) case 2:

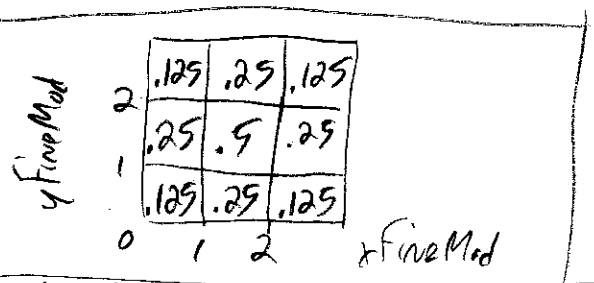

- If one end point is above scan line, then count 1 intersection.

Done by making one line shorter

(2) For both PolyLineBRES and Circle Arc Mdpt Algorithms:

Use A pixel MASK for intensity Sum within subpixel matrix

i.e. 3x3 subpixel matrix:



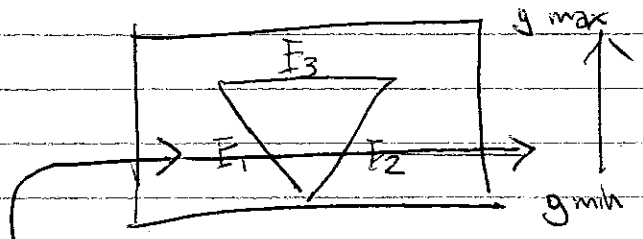

```
intensitySum = 0;
if (xFineMod == 1) && (yFineMod == 1) // center pixel
    intensitySum += 0.5;
else if (abs(xFineMod - yFineMod) == 1)
    intensitySum += 0.25;
else intensitySum += 0.125; // corner pixels
```

intensitySum is then plugged into the Alpha value of the painted (coarse) pixel;

Ikai Lan
Marvin Gouw

3. The "sorted edge table" allows for quick building of the "active list," which is a dynamic list of all the current edges which need to be considered in drawing the current scan line. The table is sorted by the y values of the edges.

Visually, this means:




If we are currently drawing this scan line, the "active list" of edges we must consider is $\{E_1, E_2\}$, which is a subset of the sorted $\{E_1, E_2, E_3\}$, the "sorted edge table," which is sorted incrementally by y (scan lines are drawn from the bottom up).

(4) Describe how the flood fill algorithm work.

specify the interiorColor and fillColor. If some given point x, y has the interior color, then fill it with the desired fill color.

Starting from point x, y, do 4 recursive call with point (x+1, y), (x-1, y), (x, y+1), (x, y-1).

The function call will be:

floodFill4 ( int x, int y, int fillColor,
int interiorColor )

5. EXPLAIN HOW TO GET THE CURRENT PIXEL COLOR USING OpenGL QUERY FUNCTIONS. SIMILARLY, EXPLAIN HOW TO GET THE CURRENT MODEL VIEW STACK SIZE.

```
// GET CURRENT PIXEL COLOR
gl.GetFloatv(GL_CURRENT_COLOR, colorValues);

// GET CURRENT MODEL VIEW STACK SIZE.
gl.GetIntegerv(GL_MODELVIEW_STACK_DEPTH, numMats);
```

CS116 MT2

GABRIEL LADEN

PR#6 2D SCALE, FACTOR 5, ABOUT POINT (1,2)

STEP 1: $M = T(1,2) \cdot S(5,5) \cdot T^{-1}(1,2)$

STEP 2: $M = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{pmatrix}$

STEP 3: $M = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 5 & 0 & -5 \\ 0 & 5 & -10 \\ 0 & 0 & 1 \end{pmatrix}$

STEP 4: $M = \begin{pmatrix} 5 & 0 & -4 \\ 0 & 5 & -8 \\ 0 & 0 & 1 \end{pmatrix}$

Jordan
James Wong

7.a) Two matrix operations that do not commute:
Non-uniform scaling and Rotation
$S_{(x,y)}$ $R(\phi)$
$\therefore S_{(x,y)} \cdot R(\phi) \neq R(\phi) \cdot S_{(x,y)}$

Example: $S_{(2,1)} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ $R(45) = \begin{bmatrix} \cos 45 & -\sin 45 & 0 \\ \sin 45 & \cos 45 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

b) $(1,2,1)$ and $(-4,-8,-4)$ represent the same point in 2D homogeneous coord, since $h = -4$, $(1,2,1) = -4(-1,-2,-1)$

8. To undo a rotation about $(2,1)$ by angle of $45°$.
First, Translate $(2,1)$

$$T_{(2,1)} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Then, Rotate $(-45°)$

$$R(-45) = \begin{bmatrix} -\cos 45 & \sin 45 & 0 \\ -\sin 45 & -\cos 45 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Finally, Translate $(-2,-1)$

$$T_{(-2,-1)} = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

So, all combined $T_{(2,1)} \cdot R(-45) \cdot T_{(-2,-1)}$

$$\begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -\cos 45 & \sin 45 & 0 \\ -\sin 45 & -\cos 45 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -\cos 45 & \sin 45 & 2 \\ -\sin 45 & -\cos 45 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -\cos 45 & \sin 45 & 2\cos 45 - \sin 45 + 2 \\ -\sin 45 & -\cos 45 & 2\sin 45 + \cos 45 + 1 \\ 0 & 0 & 1 \end{bmatrix}$$

(9)

$$R(\theta) = T^{-1} \cdot R_x^{-1}(\alpha) \cdot R_y^{-1}(\beta) \cdot R_z(\theta) \cdot R_y(\beta) \cdot R_x(\alpha) \cdot T$$

$$\sin\alpha = \frac{b}{d} \quad , \quad \sin\beta = -a$$

where $\alpha$ is the amount we need to rotate onto the x-z plane and $\beta$ is the amount we need to rotate onto the y-z plane.

(10) $(\sqrt{\frac{1}{2}}, \sqrt{\frac{1}{2}}, 0)$ by $45^0$

$$P' = qPq^{-1} \qquad q^{-1} = \frac{q^*}{\|q\|^2}$$

where

$$q = \cos\left(\frac{45}{2}\right) + \sin\left(\frac{45}{2}\right)\frac{\sqrt{2}}{2}\hat{i} + \sin\left(\frac{45}{2}\right)\frac{\sqrt{2}}{2}\hat{j}$$

$$q^* = \cos\left(\frac{45}{2}\right) - \sin\left(\frac{45}{2}\right)\frac{\sqrt{2}}{2}\hat{i} - \sin\left(\frac{45}{2}\right)\frac{\sqrt{2}}{2}\hat{j}$$

$$q^{-1} = \frac{q^*}{\left[\cos^2\frac{45}{2} + \sin^2\left(\frac{45}{2}\right)\frac{1}{2} + \sin^2\left(\frac{45}{2}\right)\frac{1}{2}\right]}$$

$$= \frac{q^*}{1} = q^* = \cos\left(\frac{45}{2}\right) - \sin\left(\frac{45}{2}\right)\frac{\sqrt{2}}{2}\hat{i} - \sin\left(\frac{45}{2}\right)\frac{\sqrt{2}}{2}\hat{j}$$