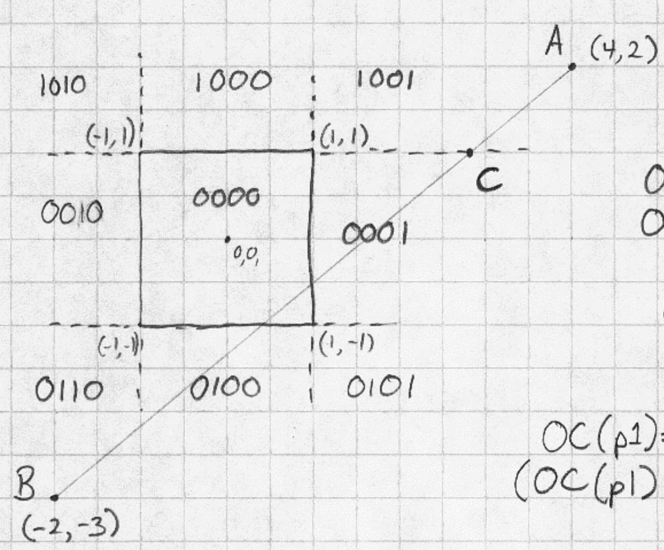
CPSC HW4

Q1)



OC(A) = 1001

OC(B) = 0110

1001 & 0110 = 1111

boolean AND

OC(p1)==0 && OC(p2)==0 → accept

(OC(p1) & OC(p2)) != 0 → reject

bitwise AND

Step a) intersect w/ top edge: point C

line: $(-2,-3) + t(6,5)$

$C = [?, 1]$ top edge, $y = 1$

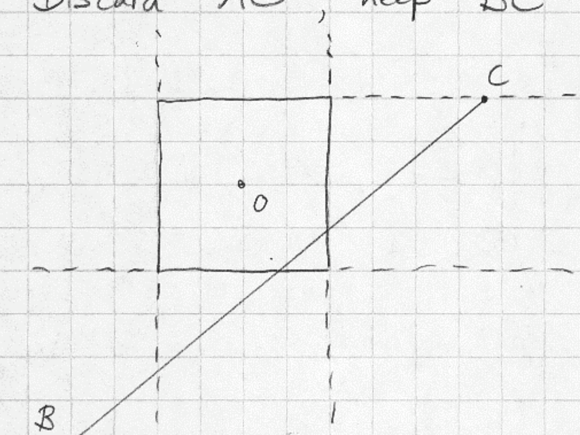
$1 = -3 + 5t_c \qquad t_c = \frac{4}{5}$

$$C_x = -2 + \frac{4}{5} \cdot 6$$
$$= -2 + \frac{24}{5}$$
$$= 2.8$$

points on a line are "inside" that line

$C = [2.8 \ \ 1] \qquad OC(C) = 0001$

Discard AC, keep BC

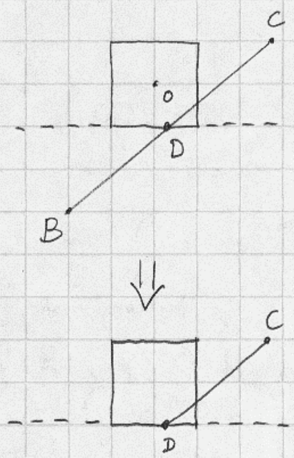


over →

b) $B = [-2, -3]$ $OC(B) = 0110$
$C = [2.8, 1]$ $OC(C) = 0001$ } cannot trivially accept or reject.

bottom edge: $y = -1$

Drawing is smaller now to save space



line: $(-2,-3) + t(4.8, 4)$

point D $[?, -1]$

$$-3 + 4t = -1$$
$$t = \frac{2}{4} = \frac{1}{2}$$

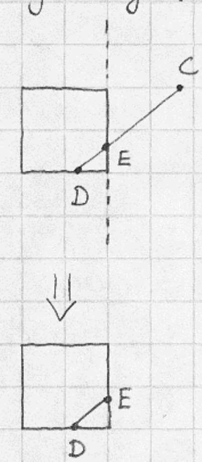
$$D_x = -2 + 4.8t$$
$$= 0.4$$

$D = [0.4, -1]$ $OC(D) = 0000$

Discard BD, keep DC

c) $C = [2.8, 1]$ $OC(C) = 0001$
$D = [0.4, -1]$ $OC(D) = 0000$ } cannot trivially accept or reject

right edge: $x = 1$



line $(0.4, -1) + t(2.8, 1)$

point E $[1, ?]$

$$0.4 + 2.8t = 1$$
$$t = 0.2143$$

$$E_y = -1 + t$$
$$= -0.7857$$

$E = [1, -0.7857]$ $OC(E) = 0000$

Discard EC, keep DE

$(OC(D) = 0000)$ && $(OC(E) = 0000)$ ACCEPT

d) The line was accepted in step 3, no further steps necessary.

Q2) Sutherland Hodgeman Algorithm:

- iterate around vertices of polygon:

4 cases:

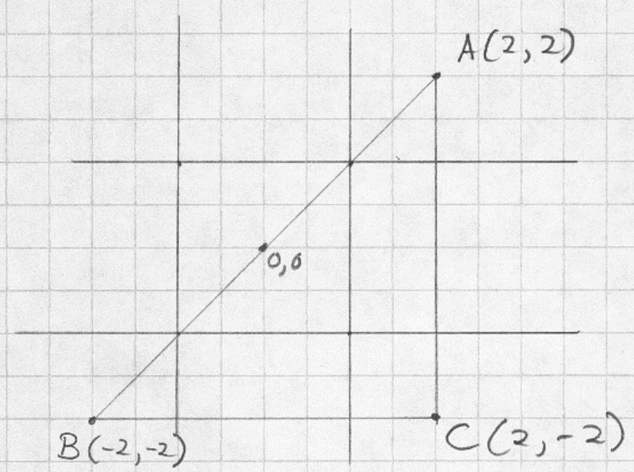
(1) - prev vertex inside clip box
- current vertex inside clip box } → add current vertex to output

(2) - prev vertex outside
- curr vertex inside } → add intersection on prev-current line to output
→ add current vertex to output

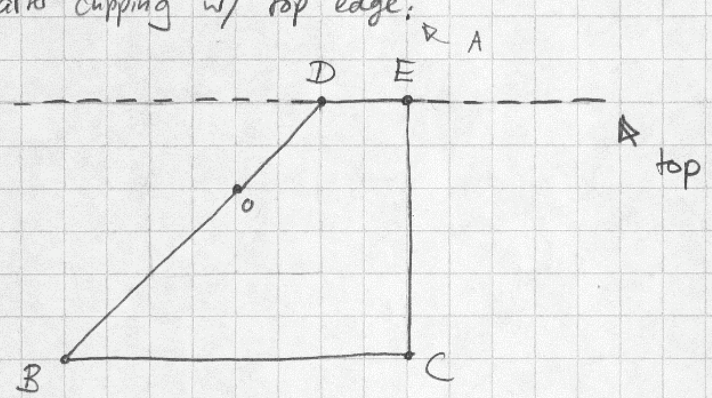
(3) - prev vertex inside
- curr vertex outside } → add intersection on prev-current line to output

(4) - prev vertex outside
- curr vertex ~~inside~~ out } → add nothing

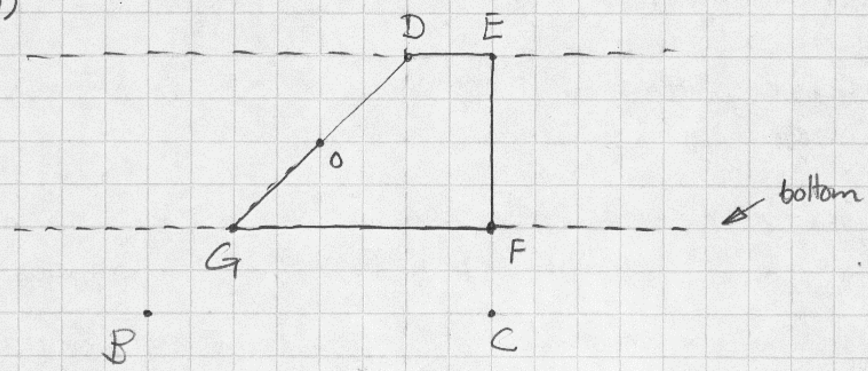
initially



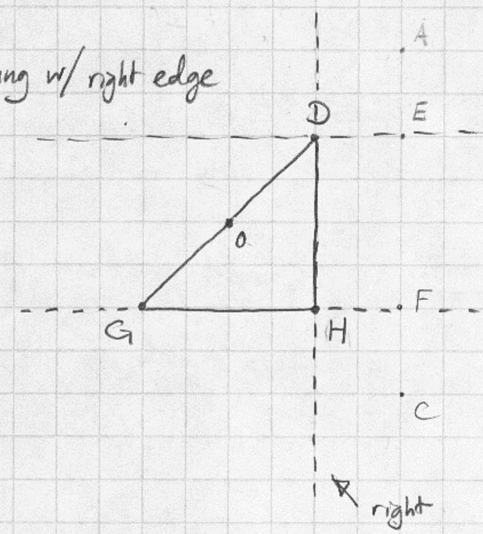
a) after clipping w/ top edge:



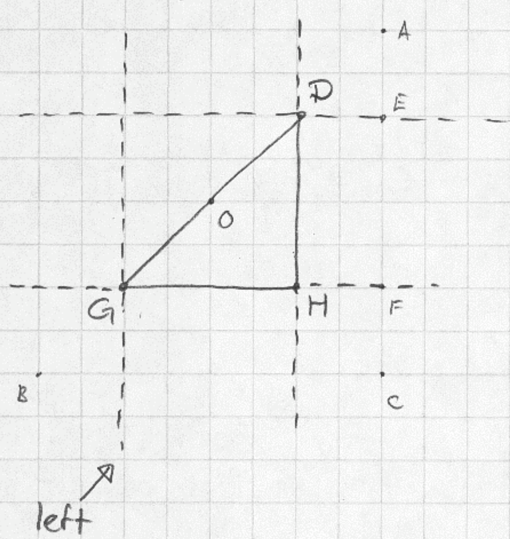
b) clipping w/ bottom edge



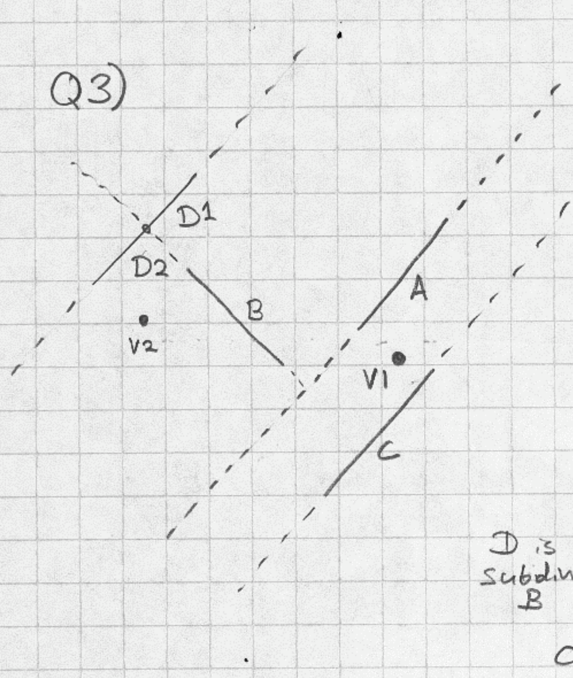
c) clipping w/ right edge



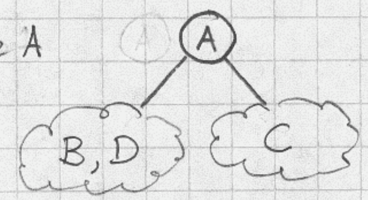
d) clipping w/ left edge (no effect)



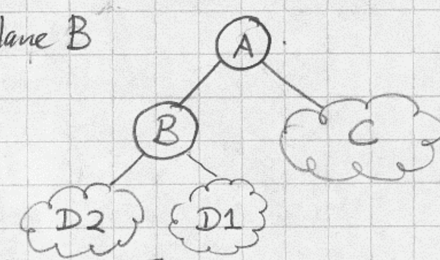
Q3)



a) Plane A

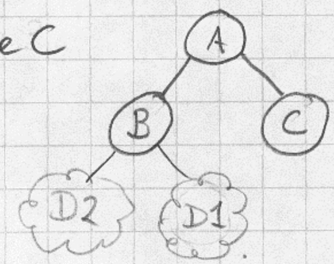


b) Plane B

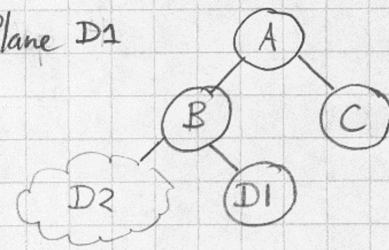


D is subdivided by B

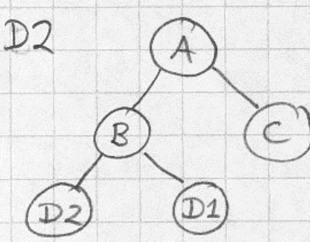
c) Plane C



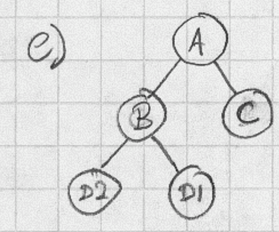
d) Plane D1



Plane D2



e)



brackets indicate unexpanded

start with node A: [A]

V1 is on the labelled side of node A, so do left child first, then A, then right child:

[B] A [C]

expand node B:

V1 is on the labelled side of B so do left child of B first

[D2] B [D1] A [C]

expand ~~ed~~ node D2:

D2 has no children

D2 B [D1] A [C]

expand node D1:

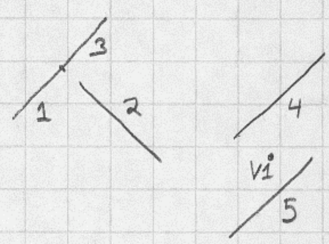
D1 has no children

D2 B D1 A [C]

expand node C:

C has no children

D2 B D1 A C



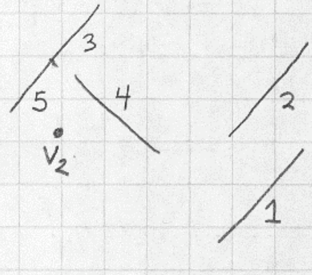
result first D2, B, D1, A, C

f) Same method as e), but with V_2

[C] A [B]

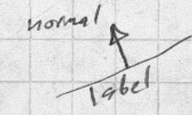
C A [D1] B [D2]

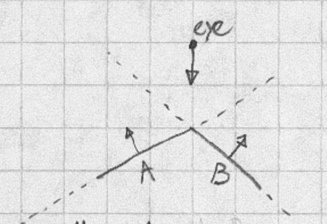
C, A, D1, B, D2



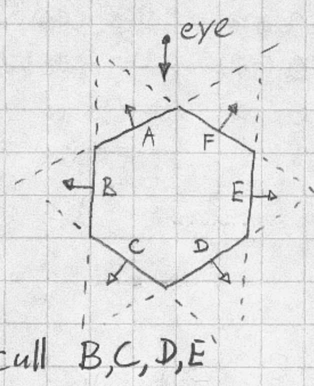
Q4) One or more of the polygons will be subdivided along the plane of another, resulting in two halves, resolving the conflict.

Q5) Assumption: positive normal on opposite side to label
Algorithm: cull if view point is on negative side of plane

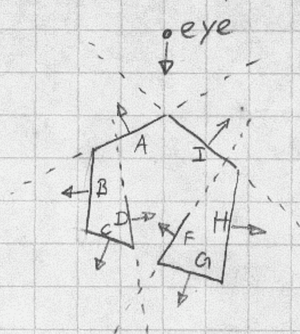




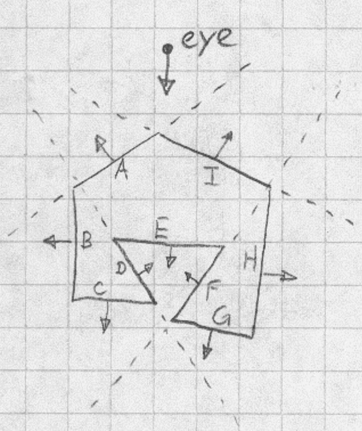
cull nothing



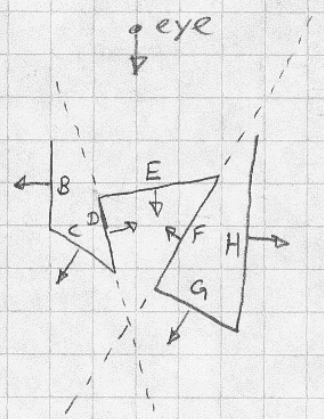
cull B,C,D,E



cull B,C,G,H
(or nothing)*



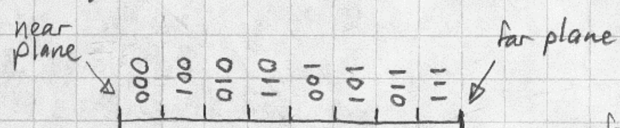
cull B,C,E,G,H



cull B,C,E,F,G,H
(or nothing)*

* backface culling ~~may~~ be disabled for non-closed objects → depends on implementation.

Q6) 3 bits → $2^3 = 8$ bins



DCS: 0.0   1.0
VCS: -0.1   -50

formula from class notes

$$z_{NDCS} = \frac{f+n}{f-n} + \frac{1}{z_{vcs}}\left(\frac{2\cdot f\cdot n}{f-n}\right)$$

$$z_{DCS} = \frac{1}{2} + \frac{1}{2} z_{NDCS}$$

$$z_{DCS} = \frac{1}{2} + \frac{1}{2}\left(\frac{f+n}{f-n} + \frac{1}{z_{vcs}}\left(\frac{2\cdot f\cdot n}{f-n}\right)\right)$$

$$\rightarrow z_{vcs} z_{DCS} = z_{vcs}\left(\frac{1}{2} + \frac{1}{2}\frac{f+n}{f-n}\right) + \frac{1}{2}\frac{2fn}{f-n}$$

$$z_{vcs} = \frac{-fn}{f-n} \Big/ \left(z_{DCS} - \frac{1}{2} - \frac{1}{2}\frac{f+n}{f-n}\right)$$

$$z_{vcs} = \frac{-fn}{(f-n)z_{DCS} - \frac{1}{2}(f-n) - \frac{1}{2}(f+n)}$$

$$z_{vcs} = \frac{-fn}{(f-n)z_{DCS} - f}$$

Over →

$$Z_{VCS} = \frac{-fn}{(f-n)Z_{DCS} - f}$$

$f = -50 \quad n = -0.1$

$$Z_{VCS} = \frac{-5}{(-49.9)V_{DCS} + 50}$$

bin #	z-buffer value	Z_{DCS}	Z_{VCS}	
		1.0000	-50.000	← far ☑
7	111	0.875	-0.789	
6	110	0.750	-0.398	
5	101	0.625	-0.266	
4	100	0.500	-0.200	
3	011	0.375	-0.160	
2	010	0.250	-0.133	
1	001	0.125	-0.114	
0	000	0	-0.100	← near ☑

↑ 8 bins   ↑ 9 planes

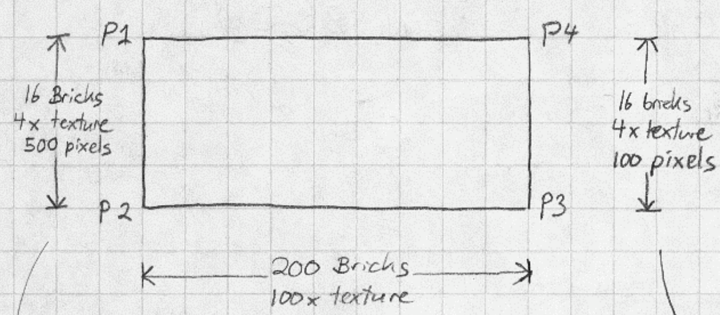
note nonlinearity & lack of precision in range -1.0 → -50.0 in VCS

↓

definitely want more than 3 bits in z-buffer.

Q7) Wall:

a)



NOTE: original texture is 128x128 px

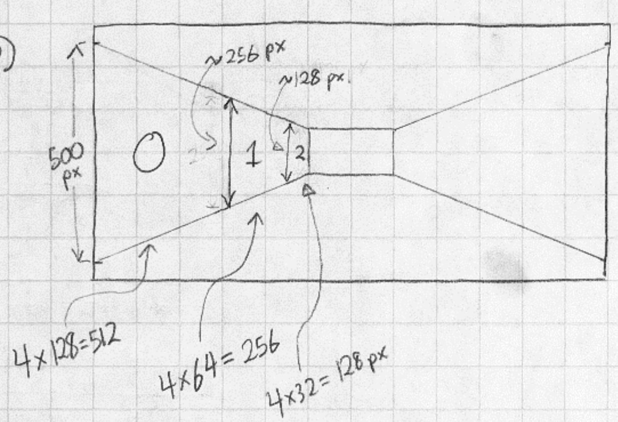
P1P2: 4 repeats = 4x128 texels

$$\Rightarrow \frac{4 \times 128}{500_{\text{pixels}}} = 1.024 \ \frac{\text{texels}}{\text{pixel}}$$

P3P4: 4 repeats = 4x128 texels

$$\Rightarrow \frac{4 \times 128}{100_{\text{pixels}}} = 5.120 \ \frac{\text{texels}}{\text{pixel}}$$

b)



The next MIP-map image is used when the effective scale drops by ½. Eg, level 2 is used when the wall is 4x64 = 256 pixels high.