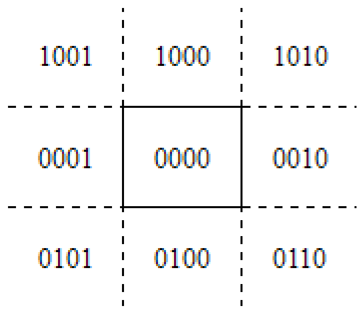
Interactive Computer Graphics Midterm Exam, April 28, 2022

2.(10%) When using the Cohen-Sutherland line clipping algorithm, how do we check the outcodes to see if a line can be trivially accepted or rejected?

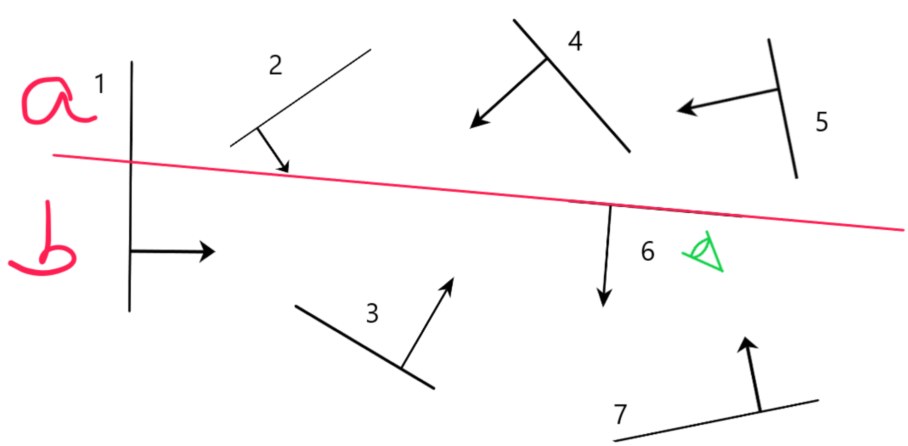
Ans :



設線段端點 P1、P2，檢查其 Outcode C1、C2。
若 C1 = C2 = 0，則為視窗內之線段，不做更動。(Trivially accept)
若 C1 ^ C2 != 0，則皆在視窗外，捨棄該線段。(Trivially reject)

3.(10%) BSP Tree

A. Construct the Binary Space Partitioning (BSP) tree of the model in Figure below. Please use the node "6" as the root. Split any line segment as you wish, and mark them as 1.a, 1.b,1.c, etc. Please choose smaller numbers/alphabets as the sub-tree root node.



B. From the BSP tree in (a), derive the display sequence in terms of the given viewing position in this figure.

Ans :

A.

```
          6
        /    \
     1.a     1.b
        \      \
         2       3
          \       \
           4        7
          /
        5
```

B.

1.a -> 2 -> 5 -> 4 -> 6 -> 1.b -> 3 -> 7

(Here I suppose 1.b is in front of the eye, if you suppose it as behind the eye is also acceptable)

4.(10%) Painter’s Algorithm

A. "86 -Eighty Six- " is a famous animation during 2021-2022. It talks about war, AI,racism, self-identity and a classic boy-meets-girl story. Here is a picture of the main character in the story. What is the drawing order in the picture below, if we use painter’s algorithm?

B.

Painter’s algorithm is the easiest way to solve visibility problems in the computer graphic area. However, it has some obvious disadvantages. Try to draw a simple picture which the painter's algorithm cannot deal with and explain the reason.

Ans:

A.

Multiple answers. Here is just one example.

3 -> 1 -> 6 -> 8 -> 5 -> 4 -> 2 ->7

B.

A picture like above cannot be shown with painter's algorithm.
Because they are overlapping each other, painter's algorithm cannot decide which one is on the top.

5. Ray intersection (10%)
Consider the 3D real vector space R3. A triangle surface S has three vertices: (6, 8, 10), (7, 9, 6), and (8, 6, 8). There is a light ray shooting from (1, 2, 3) to (2, 3, 4) (and continues going on). Will the ray hit the triangle S? Please answer yes or no and provide your reasoning and calculation.

Ans.

Yes.
Let the three points be a = (6, 8, 10), b = (7, 9, 6), c = (8, 6, 8). The plane spanned by the three points is
$5x + 3y + 2z = 74$
. The ray is given by
$(1 + t, 2 + t, 3 + t), t \geq 0$
. If we solve t for intersection of the ray with the plane, we get t = 5.7, and hence the intersection is at k = (5.7, 6.7, 7.7). To check whether the point is in the triangle, we create three vectors
$ak = a - k$
$bk = b - k$
$ck = c - k$

, and we see if cross(ak, bk), cross(bk, ck), cross(ck, ak) are in the same direction. With some calculations we can verify the three cross terms are indeed in the same direction, and hence the intersection point k is in the triangle.

6. Bezier curve (10%)

Recall the Bezier curve mentioned in the lecture. Bezier function is defined as follows:

$$B\acute{e}zier(n,t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \cdot w_i$$

A cubic Bezier curve is defined with four points (n = 3 hence four terms) by substituting wi above with their coordinates with t ∈ [0, 1] (B´ezier interval). Let L(t) be the cubic Bezier curve in R2 defined by the four points (0, 0), (1, 1), (2, −1), (3, 1/2) in order. This curve L(t) is equivalent to the curve y = f(x) in R2 . How many unique real x are there for f(x) = 0? Choose an answer below and provide your reasoning and calculation.

Ans.

The answer is (D).
The parametric y-coordinate is
y = 0 · 1 · (1 − t)^3 + 1 · 3 · (1 − t)^2 · t + (−1) · 3 · (1 − t) · t^2 + (1/2) · t^3 = (13/2) t^3 − 9 t^2 + 3 t
.Observe that y > 0 when t = 0.5, y < 0 when t = 0.8 and y > 0 when t = 1. Also, there is a trivial solution: y = 0 when t = 0. This means that, since the curve is cubic, there are three ts in [0, 1] causing y = 0.

7. Error diffusion dithering (10%)
The following is the pseudo-code of a version of Floyd-Steinberg error diffusion dithering for an n×n matrix with some specific diffusing pattern. Here, the index i is from top to down and the index j is from left to right.

Algorithm 1 Floyd-Steinberg error diffusion dithering

$A \leftarrow n \times n$ pixel matrix
$q \leftarrow$ quantization number
(pretending to) pad A at right and bottom to $(n+1) \times (n+1)$ matrix so the following won't err.
for i from 0 to $n-1$ **do**
    for j from 0 to $n-1$ **do**
        old_pixel $\leftarrow A[i][j]$
        new_pixel $\leftarrow floor(\text{old\_pixel}/q) \cdot q$
        err $\leftarrow$ old_pixel $-$ new_pixel
        $A[i][j] \leftarrow$ new_pixel
        $A[i+1][j] \leftarrow A[i+1][j] + floor((2/5) \cdot \text{err})$
        $A[i][j+1] \leftarrow A[i][j+1] + floor((2/5) \cdot \text{err})$
        $A[i+1][j+1] \leftarrow A[i+1][j+1] + floor((1/5) \cdot \text{err})$
    end for
end for
return A ignoring the padding part

Given a matrix

$$\begin{bmatrix} 19 & 48 \\ 31 & 18 \end{bmatrix}$$

and the quantization number q = 7, calculate and show the final matrix after running the algorithm.

Ans.

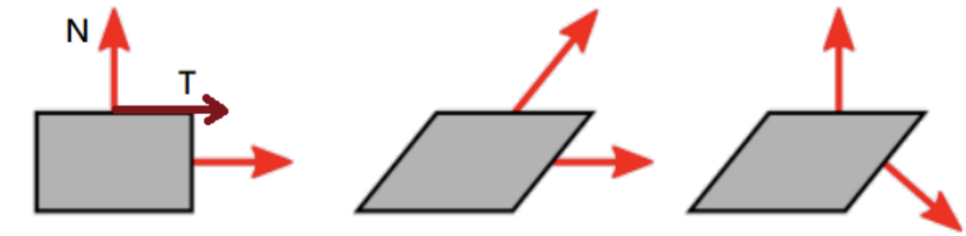
The resulting matrix is

$$\begin{bmatrix} 14 & 49 \\ 28 & 21 \end{bmatrix}$$

.

8. Transformation of normal vectors (10%)

In homework1, when you modify the upper left 3x3 in a 4x4 transformation matrix to a normal vector, most situations work; however, it may go wrong when the transformation matrix contains shear operations such as in Figure (2) given below.



Figures 1,2,and 3.

1. original object with normal vectors shown in red	2. same transform is applied to object and normal vectors	3. transformed object with the correct normal vectors

Now consider a special case of 3x3 inside a 4x4 matrix M which only has shear operations. How do you modify M to make it correctly transform a normal vector shown in Figure (3)? (10%)

Ans.

假設 N 所須乘上正確的 Matrix 為 Q, 則 $(QN)T \cdot (MT) = 0$

$NTQT \cdot (MT) = 0$

$QT \cdot M = I$ or $QT \cdot M = O$ (false)

$QT = M-1$

$Q = (M-1)T$

9. Rendering Shadows (10%)

How do you modify homework1 to make it render fake shadows like the image below? (10%) (Hint: similar to Z-buffer)

(*Note that you cannot use ray-object intersection test, it costs too high to use in realtime rendering.)

Ans.

Step1: 首先以光源當作 camera 看向整個場景，並將各個物體的 depth 輸出成 depth map，並將此時的 transform matrix 記錄下來

Step2: 以正常的 camera 看向整個場景，接著計算頂點套用前一步驟得到的 transform matrix 的 depth，與 depth map 的數值進行比較，如果較遠，則該處為陰影。