**BCA 6th SEM**

**COMPUTER GRAPHICS**

**HOME ASSIGNMENT**

1. Define BRESENHAM'S line drawing algorithm?

Bresenham's line algorithm is a [line drawing algorithm](https://en.m.wikipedia.org/wiki/Line_drawing_algorithm) that determines the points of an n-dimensional [raster](https://en.m.wikipedia.org/wiki/Raster_graphics) that should be selected in order to form a close approximation to a [straight line between two points](https://en.m.wikipedia.org/wiki/Straight_line_between_two_points). It is commonly used to draw [line primitives](https://en.m.wikipedia.org/wiki/Line_primitives) in a [bitmap image](https://en.m.wikipedia.org/wiki/Bitmap_image) (e.g. on a [computer screen](https://en.m.wikipedia.org/wiki/Computer_screen)), as it uses only integer addition, subtraction and [bit shifting](https://en.m.wikipedia.org/wiki/Bitwise_operation), all of which are very cheap operations in standard [computer architectures](https://en.m.wikipedia.org/wiki/Computer_architecture). It is an [incremental error algorithm](https://en.m.wikipedia.org/wiki/Incremental_error_algorithm). It is one of the earliest algorithms developed in the field of [computer graphics](https://en.m.wikipedia.org/wiki/Computer_graphics).

2. Define mid-point circle drawing algorithm?

The mid-point circle drawing algorithm is an algorithm used to determine the points needed for rasterizing a circle.We use the mid-point algorithm to calculate all the perimeter points of the circle in the first octant and then print them along with their mirror points in the other octants. This will work because a circle is symmetric about it’s centre.The algorithm is very similar to the [Mid-Point Line Generation Algorithm](https://www.geeksforgeeks.org/mid-point-line-generation-algorithm/amp/). Here, only the boundary condition is different.For any given pixel (x, y), the next pixel to be plotted is either (x, y+1) or (x-1, y+1). This can be decided by following the steps below.

1. Find the mid-point p of the two possible pixels i.e (x-0.5, y+1)
2. If p lies inside or on the circle perimeter, we plot the pixel (x, y+1), otherwise if it’s outside we plot the pixel (x-1, y+1)

3.Advantages and disadvantages of DDA line drawing algorithm?

Advantages of DDA algorithm;

a) Faster than the direct use of line equation and it does not need any floating point multiplication.

Disadvantages of DDA algorithm;

a) Floating point Addition is still needed

b) Precession loss is possible because of rounding of the points.

c) The algorithm is orientation dependent.

4.Define BRESENHAM'S circle drawing algorithm?

Rounding-off in DDA is time consuming.We cannot display a continuous arc on the raster display. Instead, we have to choose the nearest pixel position to complete the arc.From the following illustration, you can see that we have put the pixel at X,YX,Y location and now need to decide where to put the next pixel − at N X+1,YX+1,Y or at S X+1,Y−1X+1,Y−1.

This can be decided by the decision parameter d.

If d <= 0, then NX+1,YX+1,Y is to be chosen as next pixel.

If d > 0, then SX+1,Y−1X+1,Y−1 is to be chosen as the next pixel.

Algorithm

Step 1 − Get the coordinates of the center of the circle and radius, and store them in x, y, and R respectively. Set P=0 and Q=R.

Step 2 − Set decision parameter D = 3 – 2R.

Step 3 − Repeat through step-8 while P ≤ Q.

Step 4 − Call Draw Circle X,Y,P,QX,Y,P,Q.

Step 5 − Increment the value of P.

Step 6 − If D < 0 then D = D + 4P + 6.

Step 7 − Else Set R = R - 1, D = D + 4P−QP−Q + 10.

Step 8 − Call Draw Circle X,Y,P,QX,Y,P,Q.

Define Cohen-Sutherland line clipping algorithm?

This algorithm uses the clipping window as shown in the following figure. The minimum coordinate for the clipping region is (XWmin,YWmin)(XWmin,YWmin) and the maximum coordinate for the clipping region is (XWmax,YWmax)(XWmax,YWmax).We will use 4-bits to divide the entire region. These 4 bits represent the Top, Bottom, Right, and Left of the region as shown in the following figure. Here, the TOP and LEFT bit is set to 1 because it is the TOP-LEFT corner.

There are 3 possibilities for the line −

* Line can be completely inside the window ThislineshouldbeacceptedThislineshouldbeaccepted.
* Line can be completely outside of the window ThislinewillbecompletelyremovedfromtheregionThislinewillbecompletelyremovedfromtheregion.
* Line can be partially inside the window WewillfindintersectionpointanddrawonlythatportionoflinethatisinsideregionWewillfindintersectionpointanddrawonlythatportionoflinethatisinsideregion.

Algorithm

Step 1 − Assign a region code for each endpoints.

Step 2 − If both endpoints have a region code 0000 then accept this line.

Step 3 − Else, perform the logical ANDoperation for both region codes.

Step 3.1 − If the result is not 0000, then reject the line.

Step 3.2 − Else you need clipping.

Step 3.2.1 − Choose an endpoint of the line that is outside the window.

Step 3.2.2 − Find the intersection point at the window boundary baseonregioncodebaseonregioncode.

Step 3.2.3 − Replace endpoint with the intersection point and update the region code.

Step 3.2.4 − Repeat step 2 until we find a clipped line either trivially accepted or trivially rejected.

Step 4 − Repeat step 1 for other lines.