Q. Discuss the 8-way symmetry for drawing a circle.

→ Consider a circle is being drawn with its center at orgin (0,0) and radius = R.

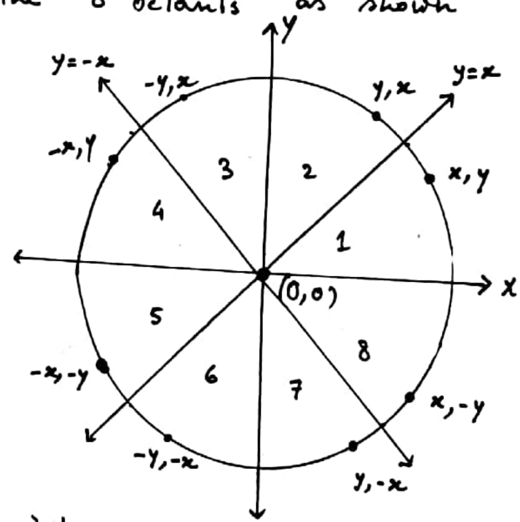
The equation of this circle is: $x^2+y^2 = R^2$

If some point $P(x_p, y_p)$ is on the circle (perimeter of the circle) then it must satisfy the above equation of this circle.

Moreover, if $(x_p, y_p)$ is a solution of the equation $x^2+y^2=R^2$, then so are $(y_p, x_p)$, $(-x_p, y_p)$, $(-y_p, x_p)$, $(x_p, -y_p)$, $(y_p, -x_p)$, $(-x_p, -y_p)$, and $(-y_p, -x_p)$. If one of these eight points, collectively represented as $(\pm x_p, \pm y_p)$ and $(\pm y_p, \pm x_p)$, is on the circle then all of them must be on the circles and creates a 8-way symmetry.

Thus if we can figure out one point on the circle, then we have actually got 8 points on the circle.

Moreover, there points are spreaded out over the 8 octants as shown in the figure.




The term symmetry refers to the reflection symmetry created by the lines $y=0$ (x-axis), $x=0$ (y-axis), $y=x$, and $y=-x$.

Thus, it suffices to draw the circle only for one octant & use the 8-way symmetry to obtain points for the other 7 octants.

NB By equation of a circle we mean equation of the locus of the point which moves along the perimeter of that circle.

Q. Why scan conversion algorithm for circles calculates the points only for the 2nd octant?

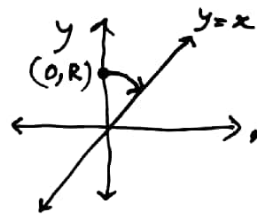
→ Since a circle exhibits 8-way symmetry, it is sufficient to draw the circles only for a single octant. Using symmetry we can obtain corresponding points for other 7 octants.

The choice of 2nd octant as the target (not 1 or other octants) is just for convenience, Starting at the top most point (0, R) in 2nd octant we have a steady increase in $x$ (while $y$ might decrease sometimes) and the expressions for decision parameters have nice forms.

Q. Describe the circle drawing algorithm by midpoint analysis method.

Suppose we have a circle centered at the origin (0,0) having radius equal to R.

→ We will rasterize the circle only for the 2nd octant. We will start at the top most point (0, R) and continue upto the $y=x$ line (end of 2nd octant).




In the 2nd octant if we draw a point $P(x, y)$

Then next point can be either the East point $E(x+1, y)$ or the south-east point $SE(x+1, y-1)$.

We define a fn. $f(x,y)$ as $f(x,y) = x^2+y^2-R^2$


Therefore, $f(x,y)=0$ implies $(x,y)$ is on the circle (perimeter)

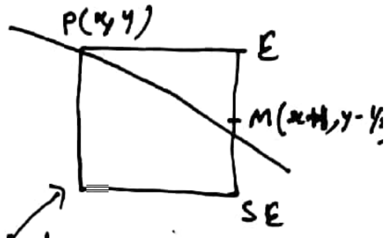
$f(x,y)<0$ implies $(x,y)$ is inside the circle

$f(x,y)>0$ " .... outside ....

NB notice that $x^2+y^2$ is squared distance of the point $(x,y)$ from the origin and $R^2$ is squared distance of the perimeter of the circle from its center located at the origin. Thus $f(x,y)$ is a difference of (squared) distances. So, $f(x,y)=0$ implies $x^2+y^2=R^2$ which is the circle eqn.

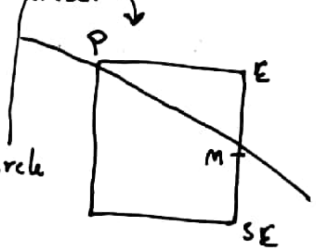
Whereas, $f(x,y)<0$ implies $x^2+y^2<R^2$ thus $(x,y)$ point is more closer to the origin than the radius i.e. $(x,y)$ is inside the circle. Similarly $f(x,y)>0$ implies $(x,y)$ is outside the circle.

Let us denote ~~We call~~ the midle point of the line segment joining E and SE as M.



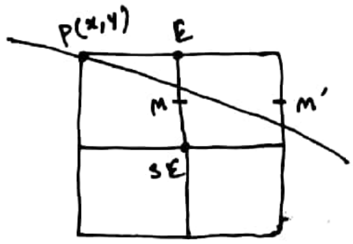

We define a decision parameter d as f(M).

If $d = f(M) < 0$, then M is inside the circle
thus point E is more closer to the circle.




and if $d = f(M) > 0$, then M is outside the circle
thus point SE is more closer to the circle

If the next point is choosen to be the point E, then next decision will be taken at point $M'(x+2, y-\frac{1}{2})$




We define,

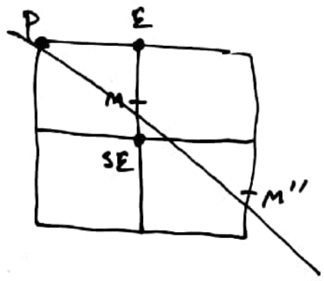
$$\Delta d_E = d_{new} - d_{old} = f(M') - f(M)$$

$$= f(x+2, y-\tfrac{1}{2}) - f(x+1, y-\tfrac{1}{2})$$

$$= (x+2)^2 + \cancel{(y-\tfrac{1}{2})^2} - \cancel{R^2} - \left[(x+1)^2 + \cancel{(y-\tfrac{1}{2})^2} - \cancel{R^2}\right]$$

$$= (x+2)^2 - (x+1)^2 = (x+2+x+1)(x+2-(x+1)) = 2x+3$$

Similarly, if the next point is choosen as the point SE, then next decision will be taken at point $M''(x+2, y-\frac{3}{2})$




We define

$$\Delta d_{SE} = f(M'') - f(M) = f(x+2, y-\tfrac{3}{2}) - f(x+1, y-\tfrac{1}{2})$$

$$= (x+2)^2 + (y-\tfrac{3}{2})^2 - R^2 - \left[(x+1)^2 + (y-\tfrac{1}{2})^2 - R^2\right]$$

$$= (x+2)^2 - (x+1)^2 + (y-\tfrac{3}{2})^2 - (y-\tfrac{1}{2})^2 = (2x+3)\cdot 1 + (2y-2)\cdot(-1)$$

$$= 2(x-y) + 5$$

At the begining we have, $x=0$, $y=R$ thus

$$d_{start} = f(0+1, R-\tfrac{1}{2}) = 1^2 + (R-\tfrac{1}{2})^2 - R^2 = \tfrac{5}{4} - R$$

Since $x, y$ are all integers and so are $\Delta d_E$ and $\Delta d_{SE}$, we can ignore fraction $\frac{1}{4}$ and simply take $d_{start} = 1 - R$.

NB Unlike line algorithm, we are not multiplying here by 4.

Suppose $h = d_{start} - 1/4 = (5/4 - R) - 1/4 = 1 - R$, therefore
$d < 0$ implies $h < -1/4$
$d > 0$ implies $h > -1/4$

As $x, y$ are integer, $\Delta d_E$ and $\Delta d_{SE}$ will also be integers. Thus, the decision parameter is updated by integral values only.

So, ~~just like~~ $h$ always remains an integer.

Therefore $h < -1/4$ is equivalent to saying $h < 0$
and $h > -1/4$ - - - - - - - - - - - - $h > 0$

Now we can just rename $h$ as $d_{start}$ and continue as before.

So far our analysis assumes that the circle is centered at the origin. If that is not the case, and the circle is centered at a point $(x_c, y_c)$ then we can first draw the circle with origin as the center using the above analysis and then translate the whole thing to $(x_c, y_c)$.

Instead of applying the translation after the circle is drawn, we can directly translate the ~~gen~~ points as soon as they are generated. Suppose, we have generated a point $(x, y)$ assuming $(0,0)$ as the center. Then to plot the actual translated points with the 8-way symmetry we define the following helper procedure.

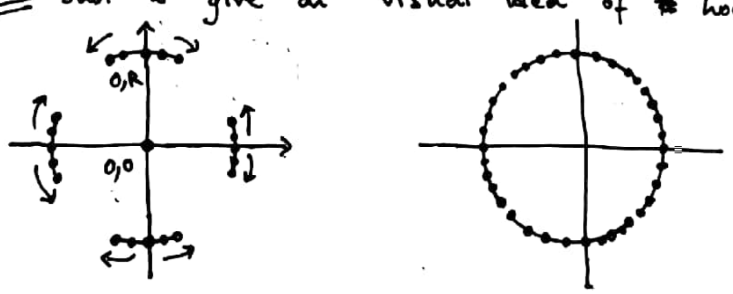
```
draw_circle(x, y, xc, yc) {
    put_pixel(x + xc, y + yc);     } 1st ~~octant~~ quadrant
    put_pixel(y + xc, x + yc);     }
    put_pixel(-x + xc, y + yc);    } 2nd ~~octant~~ "
    put_pixel(-y + xc, x + yc);    }
    put_pixel(x + xc, -y + yc);    } 4th ~~octant~~ "
    put_pixel(y + xc, -x + yc);    }
    put_pixel(-x + xc, -y + yc);   } 3rd ~~octant~~ "
    put_pixel(-y + xc, -x + yc);   }
}
```

Following the above discussion, we write the circle drawing algorithm as follows.

```
midpoint_circle_drawing_algorithm( xc, yc, R ) {
        x = 0;
        y = R;
        d = 1-R;
        draw_circle (x, y, xc, yc );
        while ( y > x ) {        // 2nd octant
                if (d ≤ 0) {     // case East
                        d = d + ΔdE;
                } else {                 // case South East
                        d = d + ΔdSE;
                        y = y - 1;
                }
                x = x + 1;   // in both case
                draw_circle (x, y, xc, yc);
        }
}
```

NB The condition of the while loop is written as $y > x$ and not $x \neq y$. Although both of these are equivalent in 2nd octant, but while dealing with integral values $x = y$ may not occur at all and the loop won't terminate in that case.

NB Just to give an visual idea of how the circle is actually generated:




As shown, the circles grows from 8 sides and eventually meet and completes the circuit.

NB The equality condition, $(d = 0)$ is merged into the East case, as it requires lesser computation than the SouthEast case.

We can improve the above algorithm by replacing the calculation of $\Delta d_E$ and $\Delta d_{SE}$ with incremental updates.

We have $\Delta d_E = 2x+3$ and $\Delta d_{SE} = 2(x-y)+5$

when we choose East, then $x$ is incremented by 1 therefore, $\begin{bmatrix} \Delta x = 1 \\ \Delta y = 0 \end{bmatrix}$

$$(\Delta d_E)_{new} = (\Delta d_E)_{old} + \overset{2\cdot\Delta x}{2} \quad \text{and} \quad (\Delta d_{SE})_{new} = (\Delta d_{SE})_{old} + \overset{2(\Delta x - \Delta y)}{2}$$

When we choose South East, then $x$ is incremented by 1 and $y$ is decremented by 1 therefore, $\begin{bmatrix} \Delta x = 1 \\ \Delta y = -1 \end{bmatrix}$

$$(\Delta d_E)_{new} = (\Delta d_E)_{old} + \underset{2\cdot\Delta x}{2} \quad \text{and} \quad (\Delta d_{SE})_{new} = (\Delta d_{SE})_{old} + \underset{2(\Delta x - \Delta y)}{4}$$

The initial values of $\Delta d_E = 2\cdot 0 + 3 = 3$
$(0, R)$

and $\Delta d_{SE} = 2(0-R)+5 = 5-2R$

The modified algorithm is as follows:

```
midpoint_circle_drawing_algorithm_ver2 ( xc, yc, R ) {
        x = 0
        y = R
        d = 1-R
        ΔdE = 3
        ΔdSE = 5-2R

        draw_circle ( x, y, xc, yc )
        while ( y > x ) {      // 2nd octant
            if ( d ≤ 0 ) {     // case East
                d = d + ΔdE
                ΔdE = ΔdE + 2
                ΔdSE = ΔdSE + 2
            } else {           // case South East
                d = d + ΔdSE
                ΔdE = ΔdE + 2
                ΔdSE = ΔdSE + 4
                y = y - 1
            }
            x = x + 1          // in both case
            draw_circle ( x, y, xc, yc )
```

10

Q. Scan convert the circle centered at (0,0) having radius = 10 using the midpoint algorithm. (Show the points only for the 2nd octant)

→

| iteration # | $d_{old}$ | case | $\Delta d_E$ $2x+3$ | $\Delta d_{SE}$ $2(x-y)+5$ | $d_{new}$ | $x$ | $y$ | point $x,y$ |
|---|---|---|---|---|---|---|---|---|
| 0 | — | — | — | — | 1-R=-9 | 0 | R=10 | 0,10 |
| 1 | -9 | ≤0, East | 2·0+3 = 3 | — | -9+3 = -6 | 0+1=1 | 10 | 1,10 |
| 2 | -6 | ≤0, East | 2·1+3 = 5 | — | -6+5 = -1 | 1+1=2 | 10 | 2,10 |
| 3 | -1 | ≤0, East | 2·2+3 = 7 | — | -1+7 = 6 | 2+1 = 3 | 10 | 3,10 |
| 4 | 6 | >0, South East | — | 2(3-10)+5 = -9 | 6+(-9) = -3 | 3+1 = 4 | 10-1 = 9 | 4,9 |
| 5 | -3 | ≤0, East | 2·4+3 = 11 | — | -3+11 = 8 | 4+1=5 | 9 | 5,9 |
| 6 | 8 | >0, South East | — | 2(5-9)+5 = -3 | 8+(-3) = 5 | 5+1=6 | 9-1=8 | 6,8 |
| 7 | 5 | >0, South East | — | 2(6-8)+5 = 1 | 5+1=6 | 6+1=7 | 8-1=7 | 7,7 |

$y \ngtr x$ stop

NB if we use the improved version (ver 2) then the table will look something like this

| iteration # | $d_{old}$ | case | $\Delta d_E$ | $\Delta d_{SE}$ | $d_{new}$ | $x$ | $y$ | point (x,y) |
|---|---|---|---|---|---|---|---|---|
| 0 | — | — | 3 | 5-2R = -15 | 1-R=-9 | 0 | R=10 | 0, 10 |
| 1 | -9 | ≤0, East | 3+2 = 5 | -15+2 = -13 | -9+3=-6 | 0+1=1 | 10 | 1, 10 |
| 2 | -6 | ≤0, East | 5+2 = 7 | -13+2 = -11 | -6+5=-1 | 1+1=2 | 10 | 2, 10 |
| 3 | -1 | ≤0, East | 7+2 = 9 | -11+2 = -9 | -1+7=6 | 2+1=3 | 10 | 3, 10 |
| 4 | 6 | >0, ~~East~~ South East | 9+2 = 11 | -9+4 = -5 | 6+(-9)=-3 | 3+1=4 | 10-1=9 | 4, 9 |

(complete this table.)

Q. Scan convert the circle centered at (1, 2) having a radius of 5 unit using the midpoint method.

→ First we generate the points assuming center is at (0, 0) for 2nd octant.

| iteration # | d | case | $\Delta d_E$ $2x+3$ | $\Delta d_{SE}$ $2(x-y)+5$ | new d | x | y |
|---|---|---|---|---|---|---|---|
| 0 | — | — | — | — | $1-R=-4$ | 0 | $R=5$ |
| 1 | $-4$ | $\le 0$, East | $2\cdot 0+3=3$ | — | $-4+3=-1$ | $0+1=1$ | 5 |
| 2 | $-1$ | $\le 0$, East | $2\cdot 1+3=5$ | — | $-1+5=4$ | $1+1=2$ | 5 |
| 3 | 4 | $>0$, SouthEast | — | $2(2-5)+5=-1$ | $4+(-1)=3$ | $2+1=3$ | $5-1=4$ |
| 4 | 3 | $>0$, South East | — | $2(3-4)+5=3$ | $3+3=6$ | $3+1=4$ | $4-1=3$ |

These points are for the 2nd octant, $y \not> x$, stop using the 8 way symmetry ~~octant~~. and translation for the center (1,2) we have.

| generated $x, y$ | (2nd octant) $x+x_c,\ y+y_c$ | (1st octant) $y+x_c,\ x+y_c$ | (3rd) $-x+x_c,\ y+y_c$ | (4th) $-y+x_c,\ x+y_c$ | (7th) $x+x_c,\ -y+y_c$ | (8th) $y+x_c,\ -x+y_c$ | (5th) $-x+x_c,\ -y+y_c$ | (6th) $-y+x_c,\ -x+y_c$ |
|---|---|---|---|---|---|---|---|---|
| 0, 5 | $0+1=1,\ 5+2=7$ | $5+1=6,\ 0+2=2$ | $-0+1=1,\ 5+2=7$ | | | | | |

[Complete this table]

NB The above example has two important contributing factors.

① it shows that, the stopping condition must be $y > x$ (and not $x \neq y$)

② some points are generated twice. More specifically points on the reflection lines $x=0$, $y=0$, $x=y$, $x+y=0$ are repeated. Moreover when we stop at $x \neq y$ (as in this case), the last two rows are the same.

NB These repeatations can be avoided by carefully injecting some condition or similar measures. Such measures makes the code a bit messy and thus left for the readers to implement on their own.

[ref: gfg]

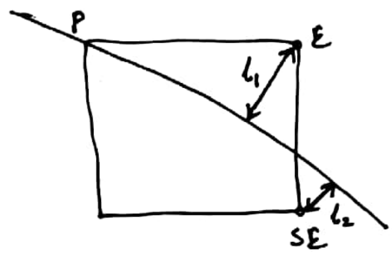
Q. Describe the Bresenham circle drawing algorithm.

→ Suppose we have a circle centered at the origin $(0,0)$ having a radius of $R$. We will rasterize the circle only for the 2nd octant. We start at $(0,R)$ and continue until $x$ becomes larger than $y$ i.e. upto the $y=x$ line.

In the 2nd octant if we draw a point $P(x,y)$ the next point can be either the east point $E(x+1,y)$ or the southeast point $SE(x+1,y-1)$.

We define, $f(x,y) = x^2+y^2-R^2$. Thus $f(E) > 0$ and $f(SE) < 0$ always.

We choose the decision parameter $d$ as $f(E)+f(SE)$.




Since $f(x,y)$ is a difference of squared distances and $f(E)$ is positive while $f(SE)$ is negetive, depending on which one of $E$ and $SE$ is more closer to the circle (perimeter) the value of $d$ will be either positive or negetive.

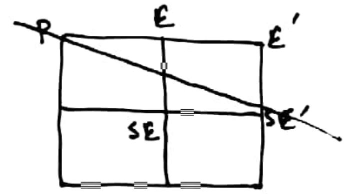
More specifically,

$d = f(E)+f(SE)$ $= 0$, when both are equidistant, we can choose either one

$< 0$, when SE is far away than E, thus choose E

$> 0$, when E is far away than SE, thus choose SE

If E is chosen as the next point then



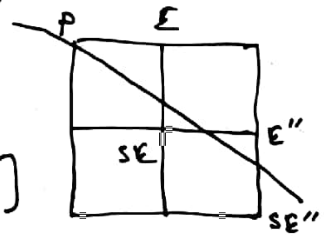

$$\Delta d_E = d_{new} - d_{old} = f(E') + f(SE') - [f(E)+f(SE)]$$

$$= f(x+2,y) + f(x+2,y-1) - [f(x+1,y) + f(x+1,y-1)]$$

$$= (x+2)^2 + \cancel{y^2} - \cancel{R^2} + (x+2)^2 + \cancel{(y-1)^2} - \cancel{R^2} - [(x+1)^2 + \cancel{y^2} - \cancel{R^2} + (x+1)^2 + \cancel{(y-1)^2} - \cancel{R^2}]$$

$$= 2[(x+2)^2 - (x+1)^2] = 2(2x+3)\cdot 1 = 4x+6$$

If SE is choosen as the next point then




$$\Delta d_{SE} = f(E'') + f(SE'') - [f(E)+f(SE)]$$

$$= f(x+2,y-1) + f(x+2,y-2) - [f(x+1,y) + f(x+1,y-1)]$$

$$= (x+2)^2 + (y-1)^2 - R^2 + (x+2)^2 + (y-2)^2 - R^2 - [(x+1)^2 + y^2 - R^2 + (x+1)^2 + (y-1)^2 - R^2]$$

$$= 2[(x+2)^2 - (x-1)^2] + (y-2)^2 - y^2 = 4x+6 + (2y-2)\cdot(-2) = 4(x-y)+10$$

The initial value of d is calculated as follows: $(x=0, y=R)$

$$
\begin{aligned}
d_{start} &= f(0+1, R) + f(0+1, R-1) \\
&= 1^2 + \cancel{R^2} - \cancel{R^2} + 1^2 + (R-1)^2 - R^2 \\
&= 2 + \cancel{R^2} - 2R + 1 - \cancel{R^2} \\
&= 3 - 2R.
\end{aligned}
$$

Thus we have the following algorithm:

```
Bresenham_circle_drawing_algorithm (xc, yc, R) {
    x = 0
    y = R
    d = 3-2R
    draw_circle (x, y, xc, yc)
    while (y > x) {            // 2nd octant
        if (d <= 0) {          // case East
            d = d + Δd_E
        } else {               // case South East
            d = d + Δd_SE
            y = y-1
        }
        x = x+1                // in both case
        draw_circle (x, y, xc, yc)
    }
}
```

> **Q.** Scan convert the circle centered at (0,0) having a radius 10 using the Bresenham algorithm.

Here the equality condition $(d=0)$ is merged into the East case as it requires less computation.

The draw_circle( ) is a helper procedure which plots the circle in all octants, using the 8-way symmetry. It also takes care of the translation required when the center is not at the origin rather at $(x_c, y_c)$.

[write draw_circle( ) ]

NB $\Delta d_E$, $\Delta d_{SE}$ can be updated incrementally as before.