Scan Conversion of Lines

Dr. Scott Schaefer

TAMU Computer Science

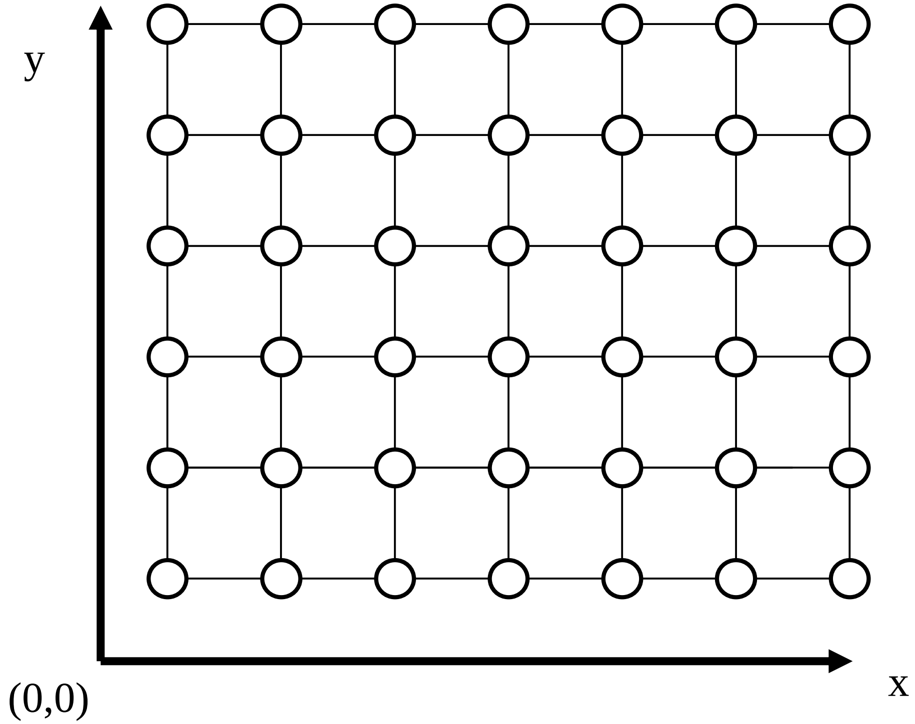
Displays – Cathode Ray Tube

Displays – Liquid Crystal Displays

Displays – Light-Emitting Diode

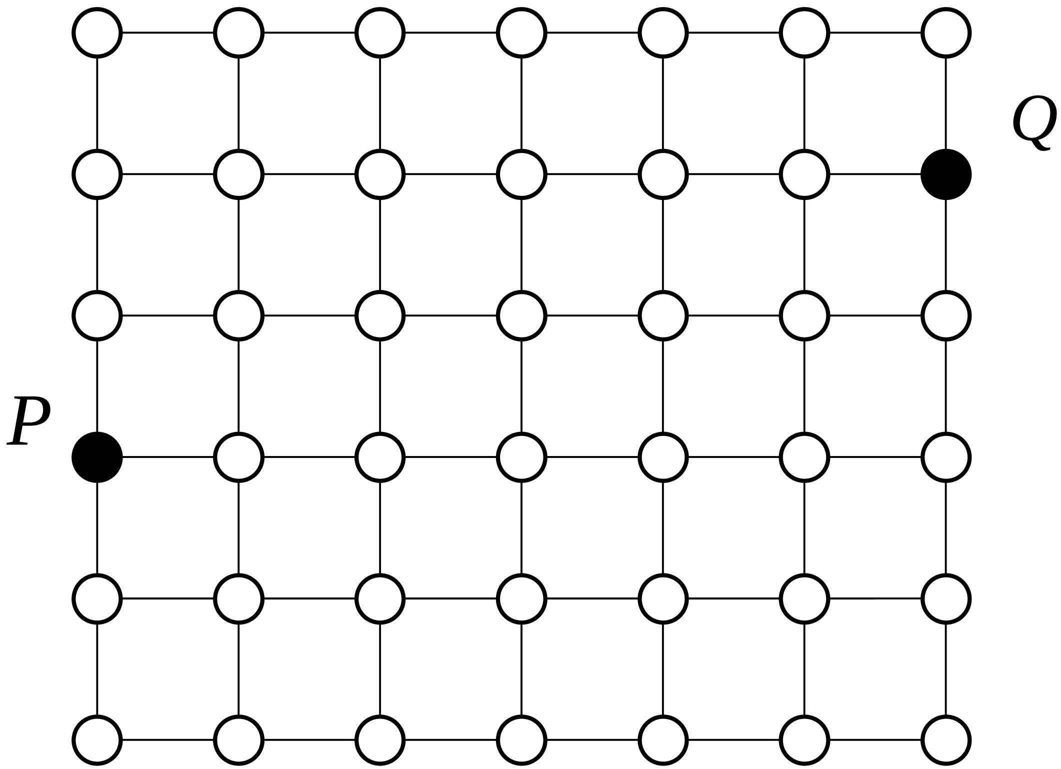
Displays – Pixels

- Pixel: the smallest element of picture
  - Integer position (i,j)
  - Color information (r,g,b)



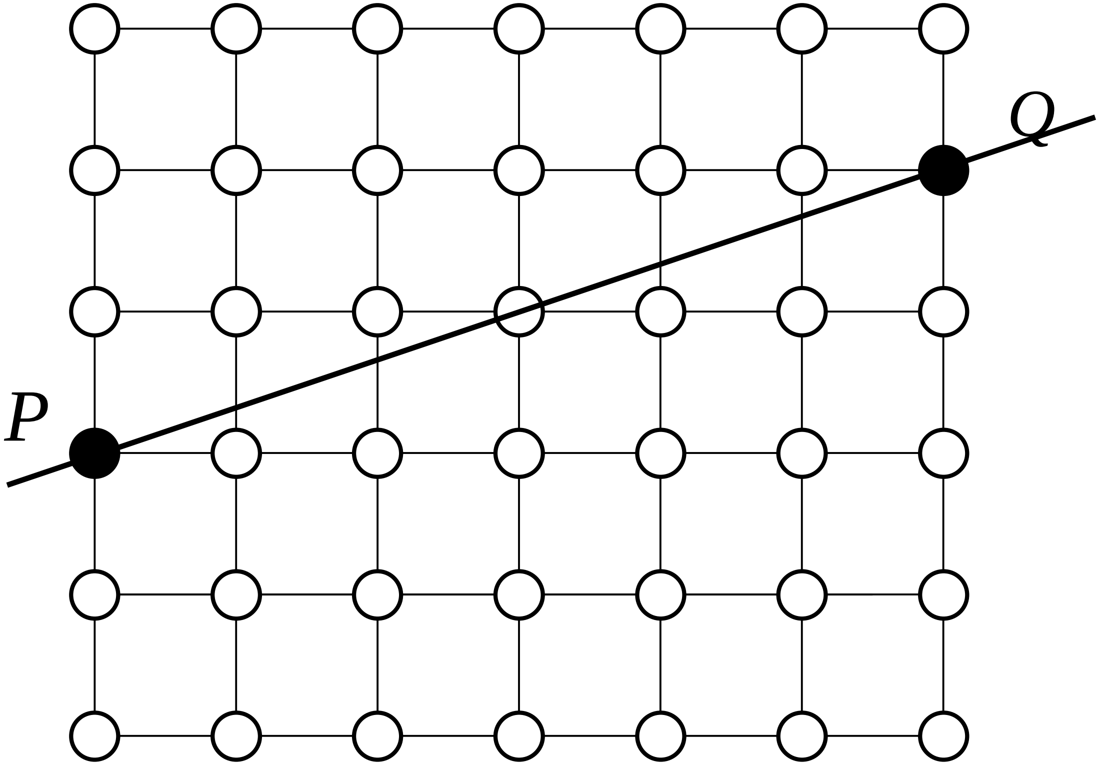
Problem

- Given two points (P, Q) on the screen (with integer coordinates) determine which pixels should be drawn to display a unit width line



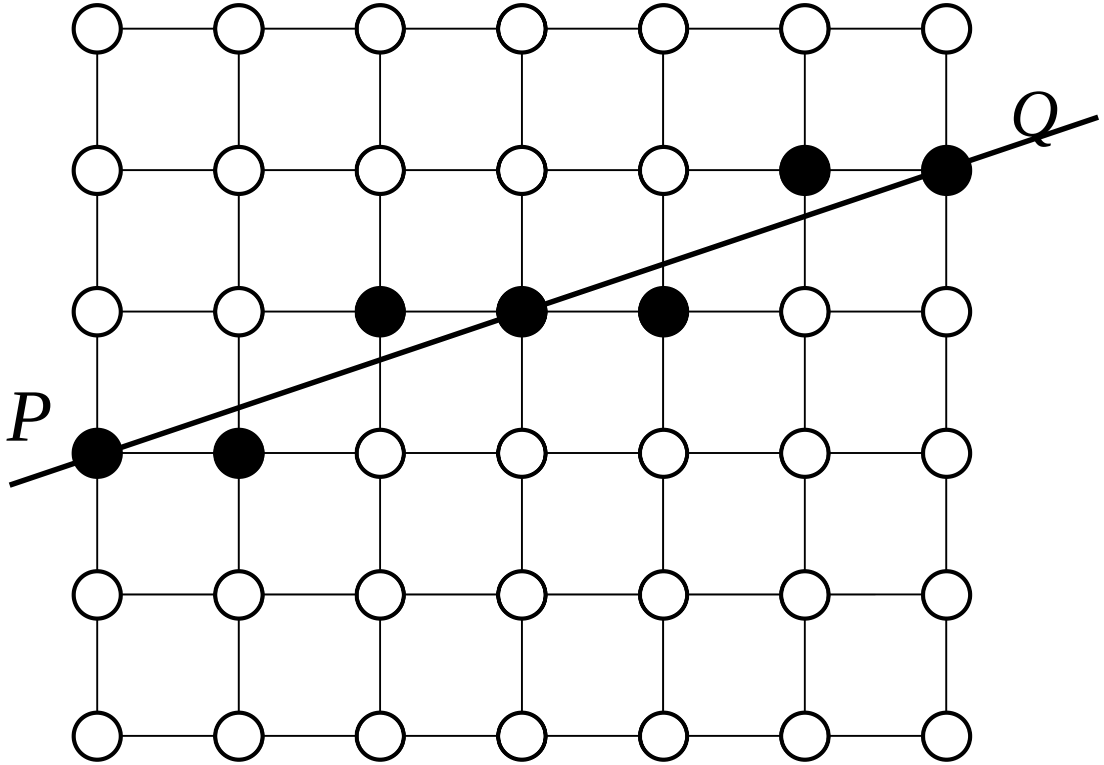
Problem

- Given two points (P, Q) on the screen (with integer coordinates) determine which pixels should be drawn to display a unit width line

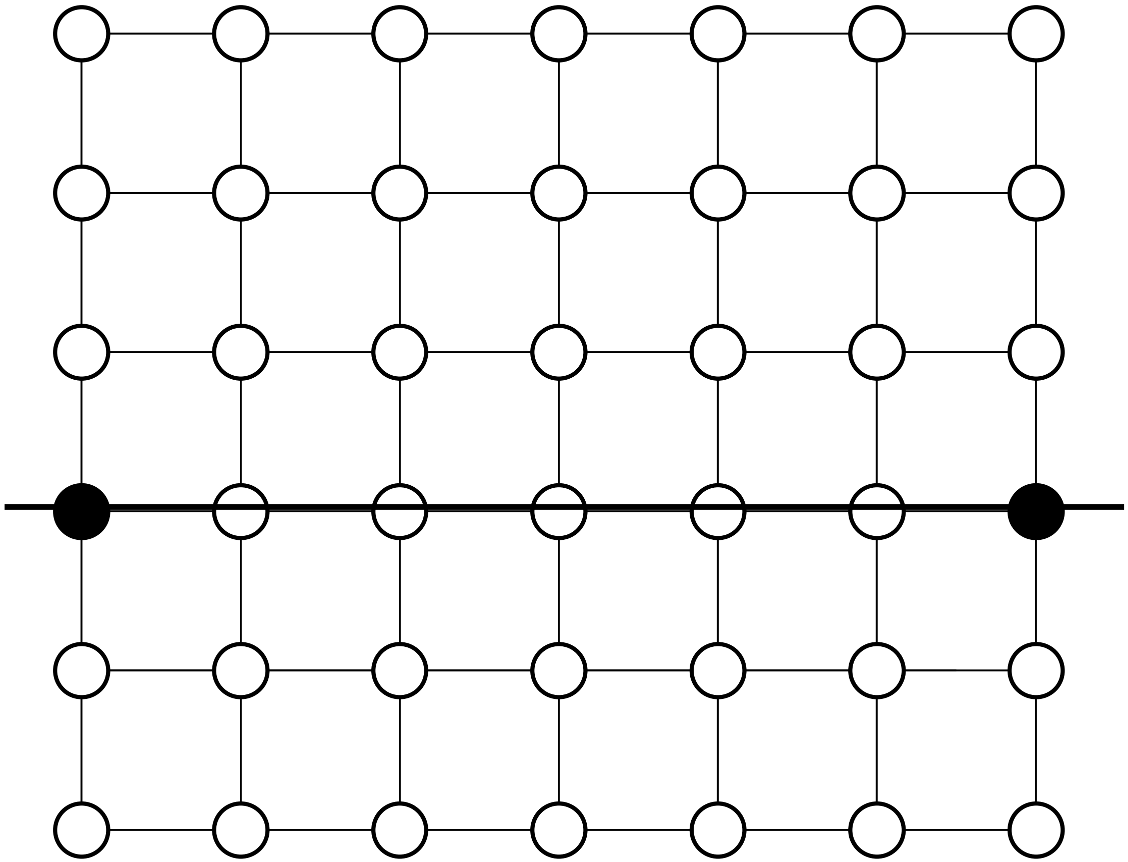


Problem

- Given two points (P, Q) on the screen (with integer coordinates) determine which pixels should be drawn to display a unit width line

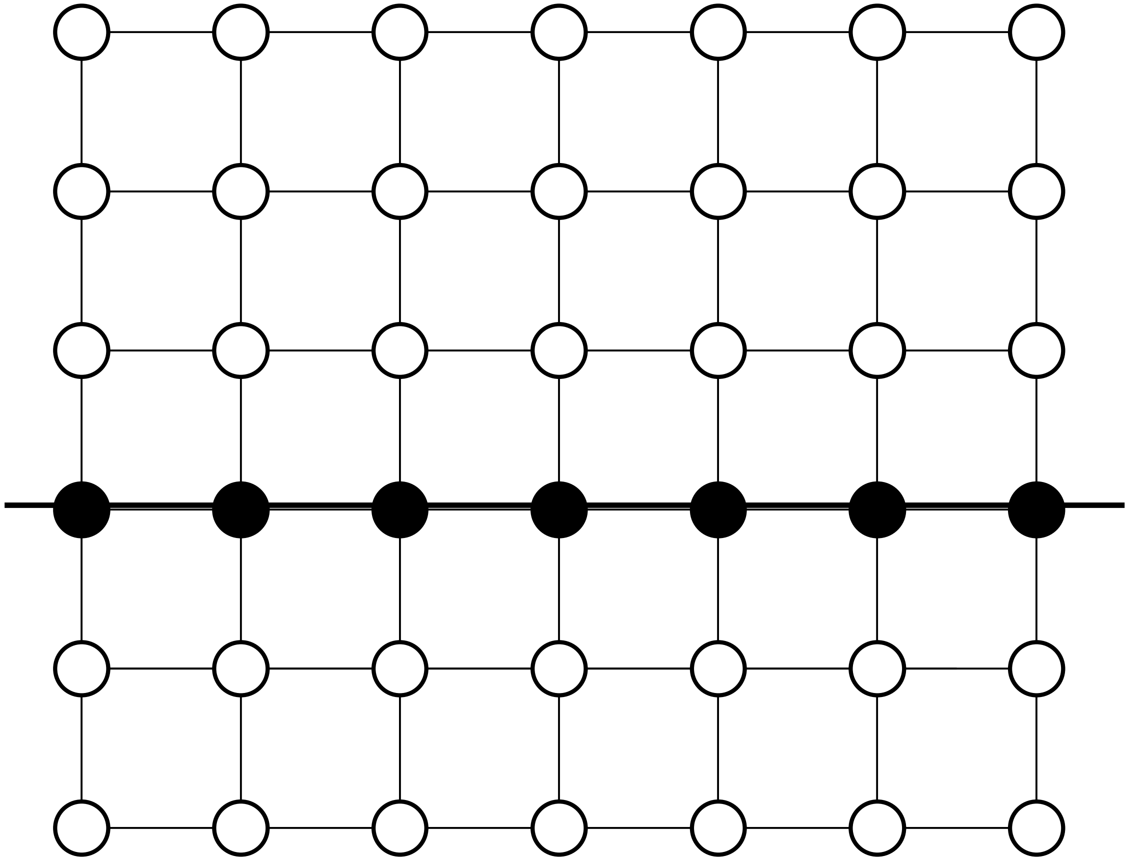


Special Lines - Horizontal

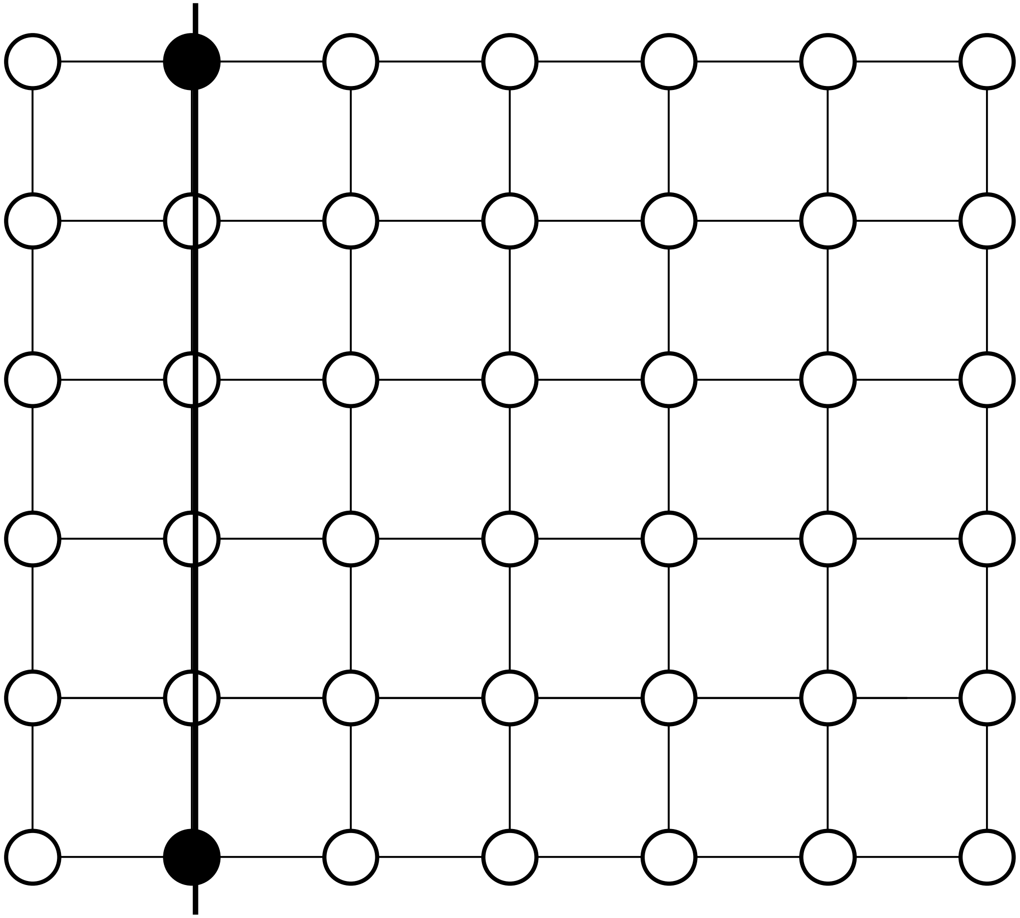


Special Lines - Horizontal

Increment *x* by 1, keep *y* constant

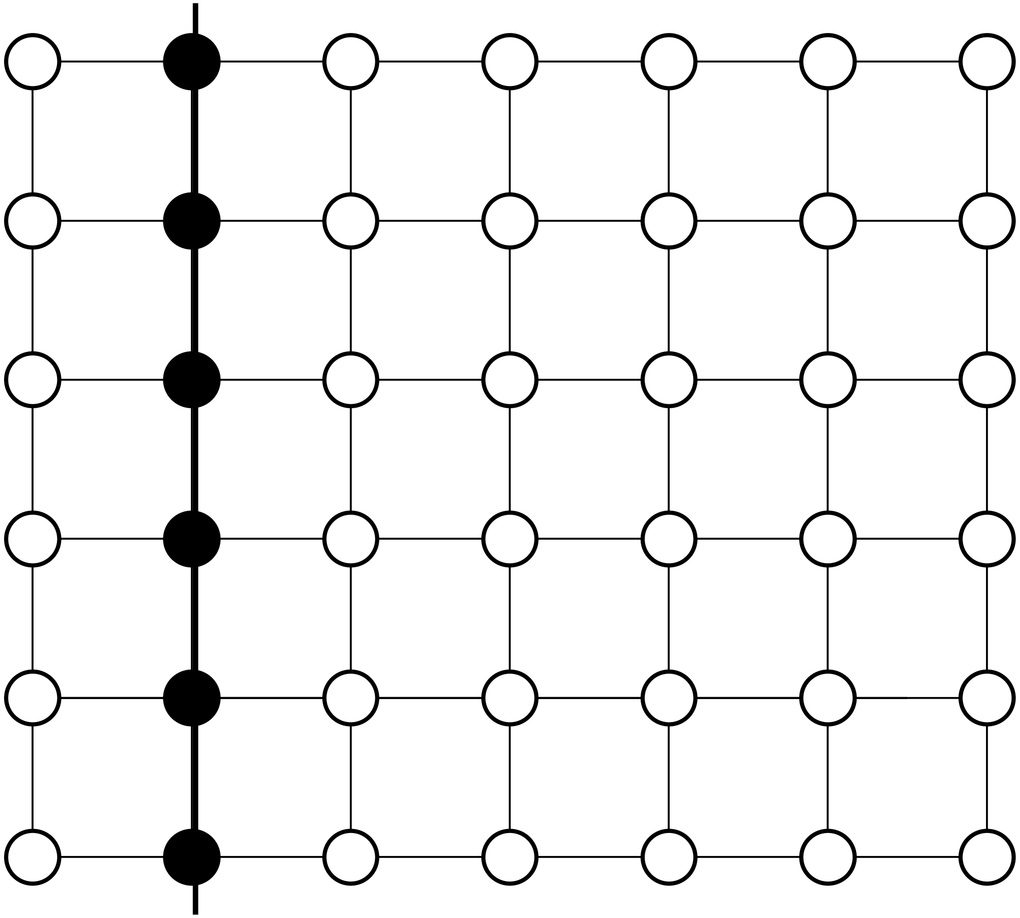


Special Lines - Vertical

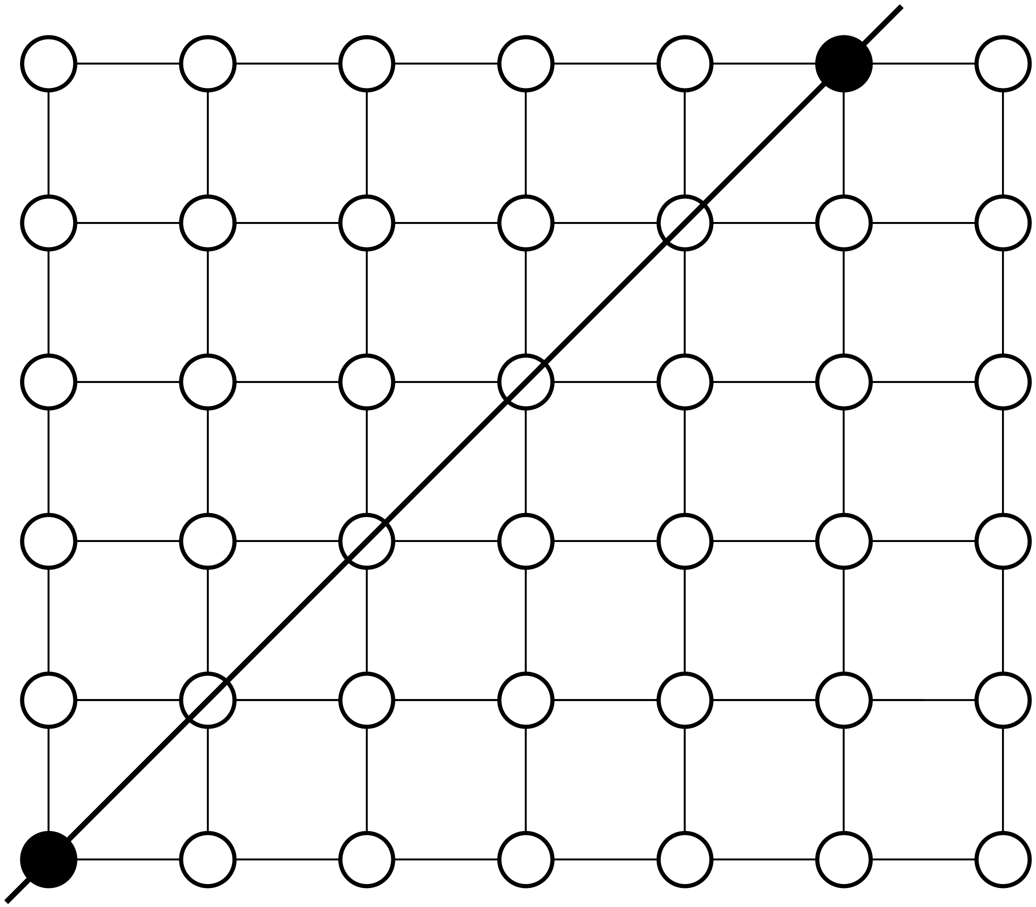


Special Lines - Vertical

Keep x constant, increment y by 1

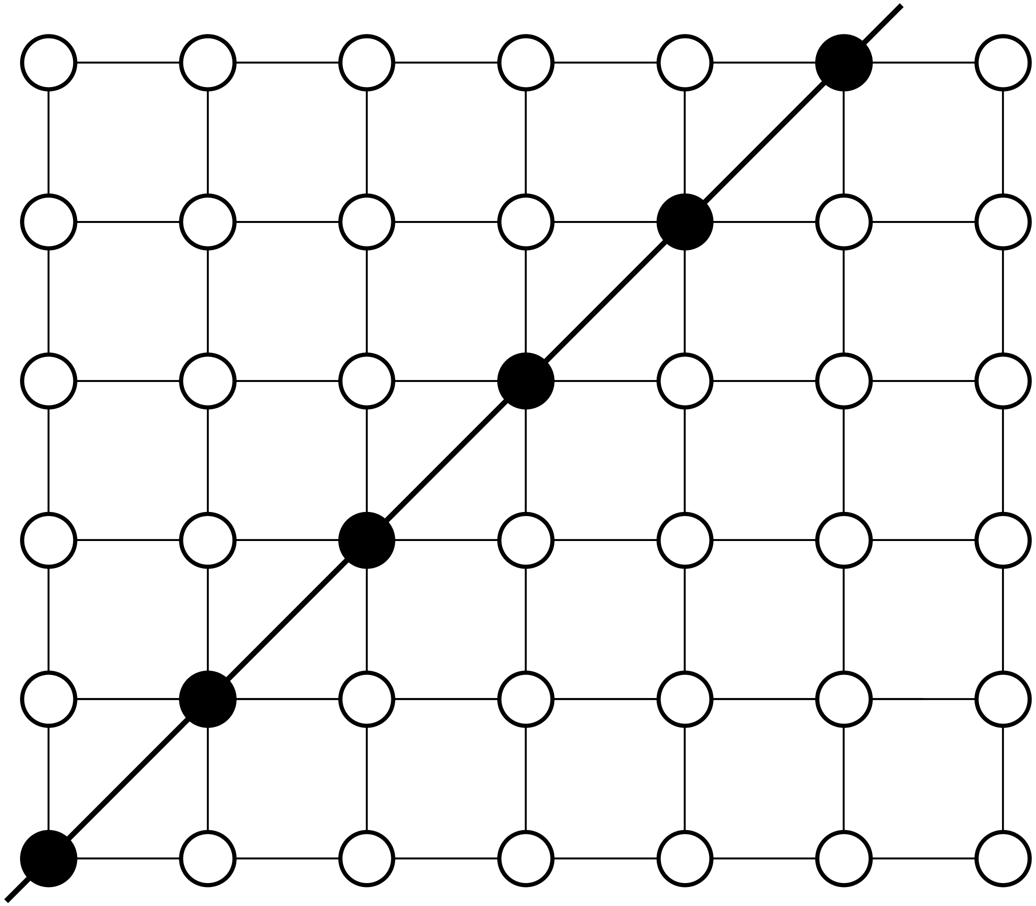


Special Lines - Diagonal

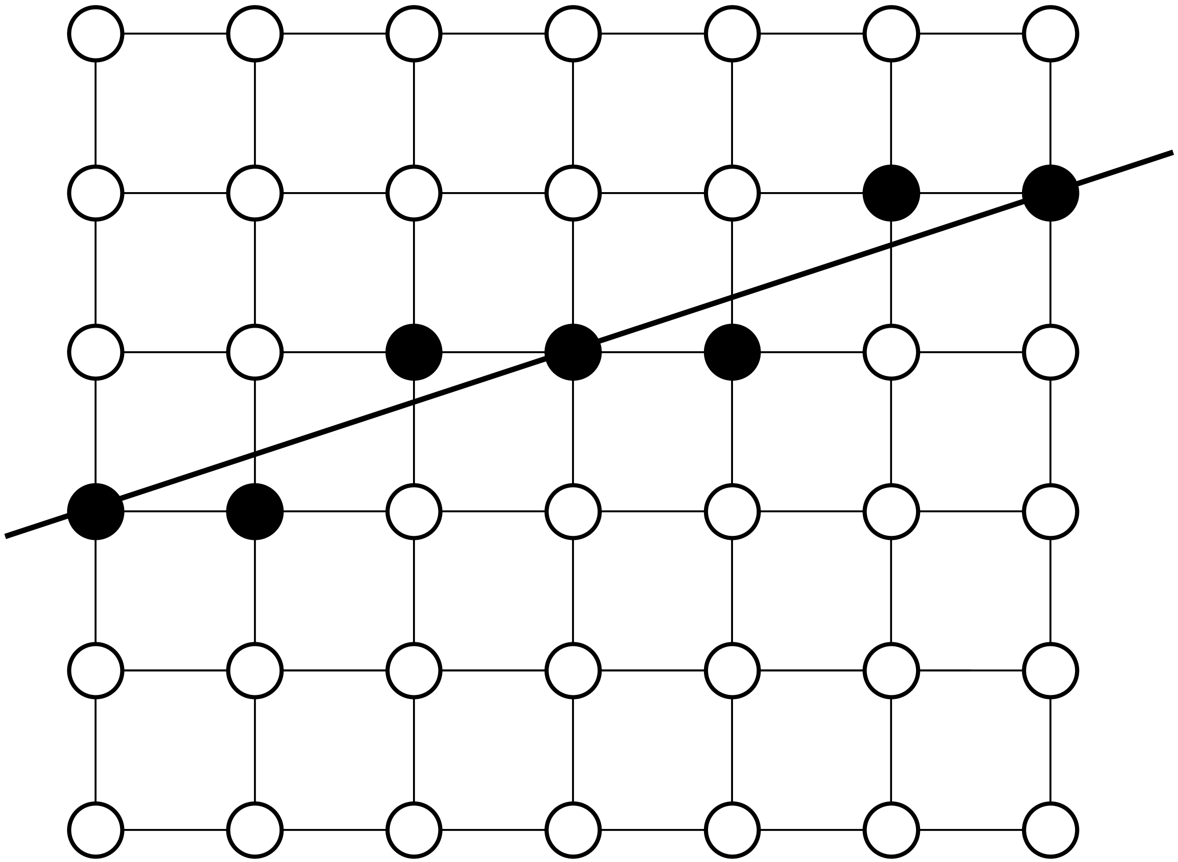


Special Lines - Diagonal

Increment x by 1, increment y by 1



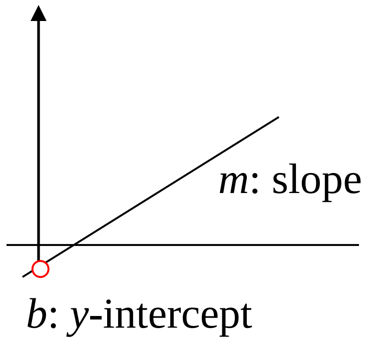
How about Arbitrary Lines?



Arbitrary Lines

Slope-intercept equation for a line:

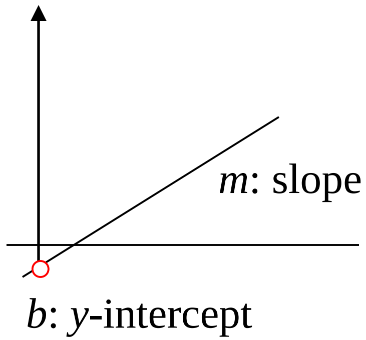
$$y = mx + b$$



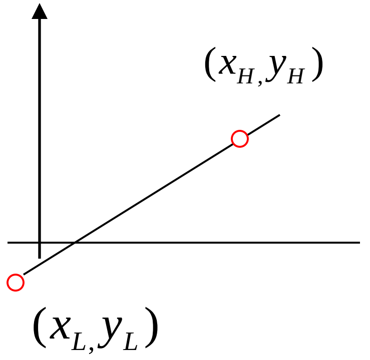
Arbitrary Lines

Slope-intercept equation for a line:

$$y = mx + b$$



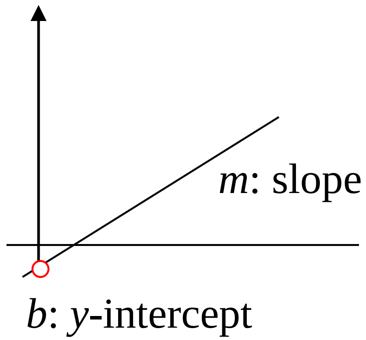
How can we compute the equation from (x_L, y_L), (x_H, y_H)?



Arbitrary Lines

Slope-intercept equation for a line:

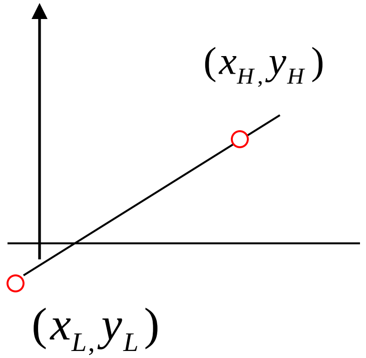
$$y = mx + b$$



How can we compute the equation from (x_L, y_L), (x_H, y_H)?

$$m = \frac{y_H - y_L}{x_H - x_L}$$

$$b = y_L - mx_L$$



Arbitrary Lines

$$y = mx + b$$

- Assume $0 < m \leq 1$
- Other lines by swapping x, y or negating

- Take steps in x and determine where to fill y

Simple Algorithm

- Start from (x_L, y_L) and draw to (x_H, y_H) where $x_L < x_H$

$(x_0, y_0) = (x_L, y_L)$

For ($i = 0$; $i <= x_H - x_L$; i++)

  DrawPixel (x_i, Round (y_i))

  $x_{i+1} = x_i + 1$

  $y_{i+1} = m\, x_{i+1} + b$

Simple Algorithm

- Start from (x_L, y_L) and draw to (x_H, y_H) where $x_L < x_H$

$(x_0, y_0) = (x_L, y_L)$

For ($i = 0$; $i <= x_H - x_L$; i++)

DrawPixel (x_i, Round (y_i))

$x_{i+1} = x_i + 1$

$y_{i+1} = m (x_i + 1) + b$

Simple Algorithm

- Start from (x_L, y_L) and draw to (x_H, y_H) where $x_L < x_H$

$(x_0, y_0) = (x_L, y_L)$

For ($i = 0$; $i <= x_H - x_L$; i++)

DrawPixel (x_i, Round (y_i))

$x_{i+1} = x_i + 1$

$y_{i+1} = m\, x_i + m + b$

Simple Algorithm

- Start from (x_L, y_L) and draw to (x_H, y_H) where $x_L < x_H$

$(x_0, y_0) = (x_L, y_L)$

For ($i = 0$; $i <= x_H - x_L$; i++)

DrawPixel (x_i, Round (y_i))

$x_{i+1} = x_i + 1$

$y_{i+1} = m\, x_i + b + m$

Simple Algorithm

- Start from (x_L, y_L) and draw to (x_H, y_H) where $x_L < x_H$

$(x_0, y_0) = (x_L, y_L)$

For ($i = 0$; $i <= x_H - x_L$; i++)

DrawPixel (x_i, Round (y_i))

$x_{i+1} = x_i + 1$

$y_{i+1} = \boxed{m\, x_i + b} + m$

Simple Algorithm

- Start from (x_L, y_L) and draw to (x_H, y_H) where $x_L < x_H$

$(x_0, y_0) = (x_L, y_L)$

For ($i = 0$; $i <= x_H - x_L$; i++)

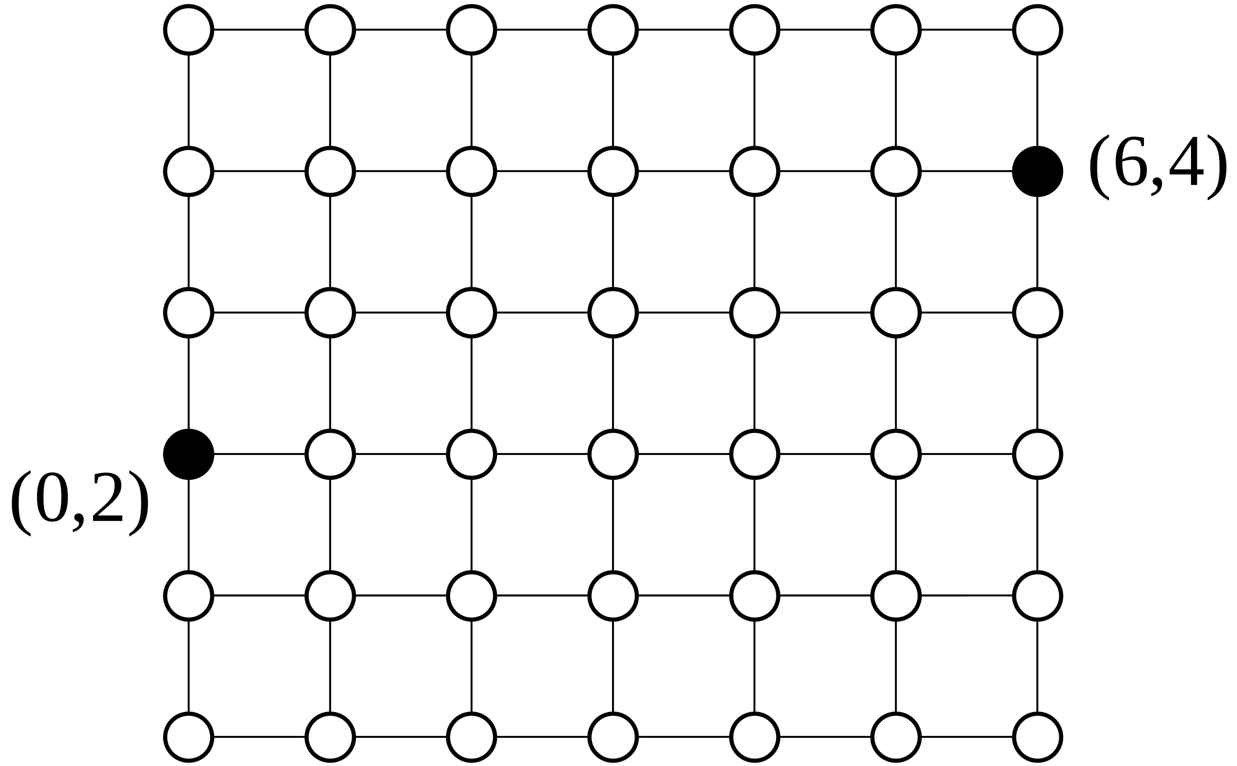
DrawPixel (x_i, Round (y_i))

$x_{i+1} = x_i + 1$

$y_{i+1} = y_i + m$

Simple Algorithm - Example

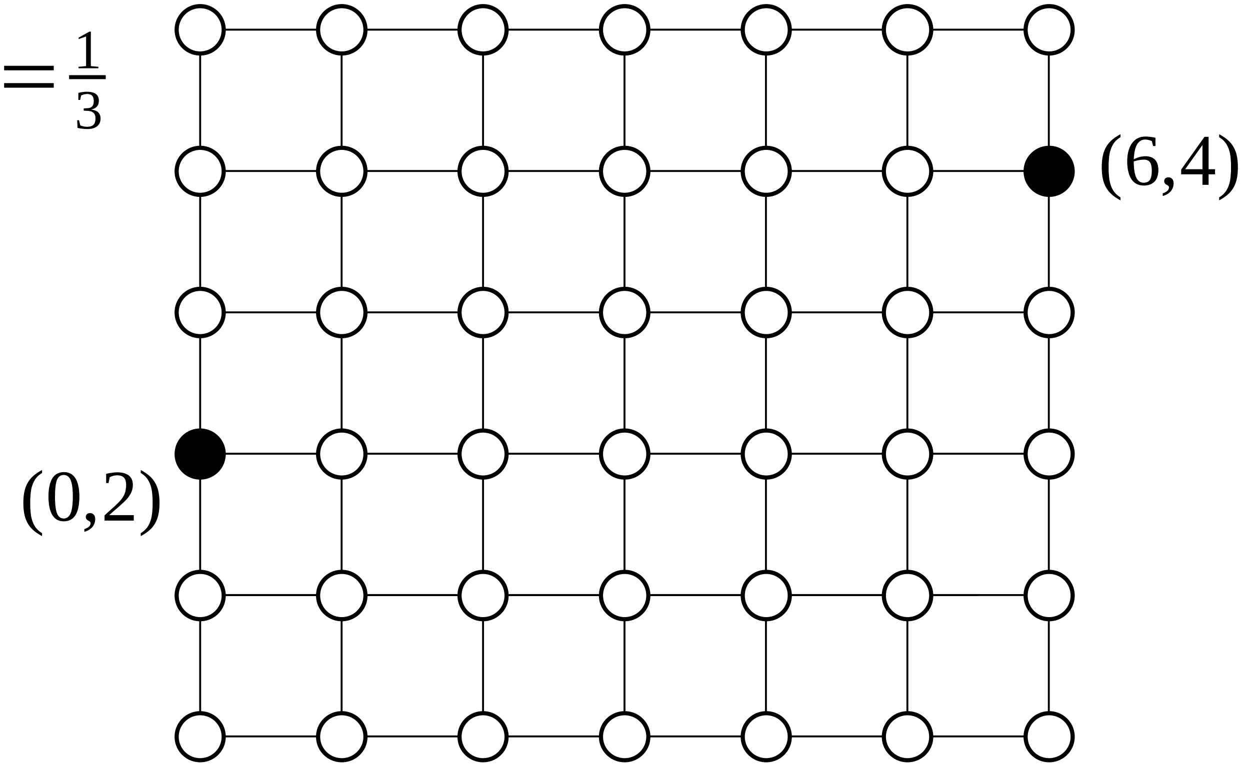
$$y = mx + b$$



Simple Algorithm - Example

$$y = mx + b$$

$$m = \frac{y_H - y_L}{x_H - x_L} = \frac{4-2}{6-0} = \frac{1}{3}$$



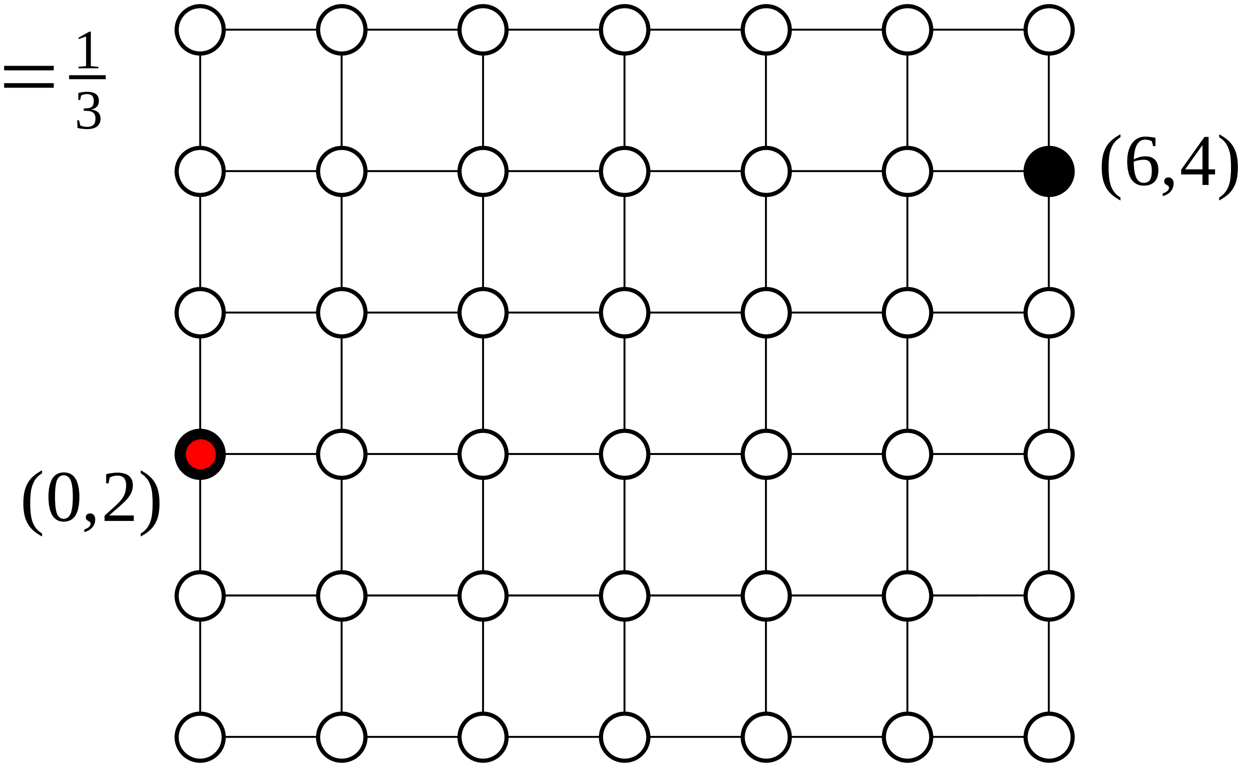
Simple Algorithm - Example

$$y = mx + b$$

$$m = \frac{y_H - y_L}{x_H - x_L} = \frac{4-2}{6-0} = \frac{1}{3}$$

$$x = 0$$

$$y = 2$$



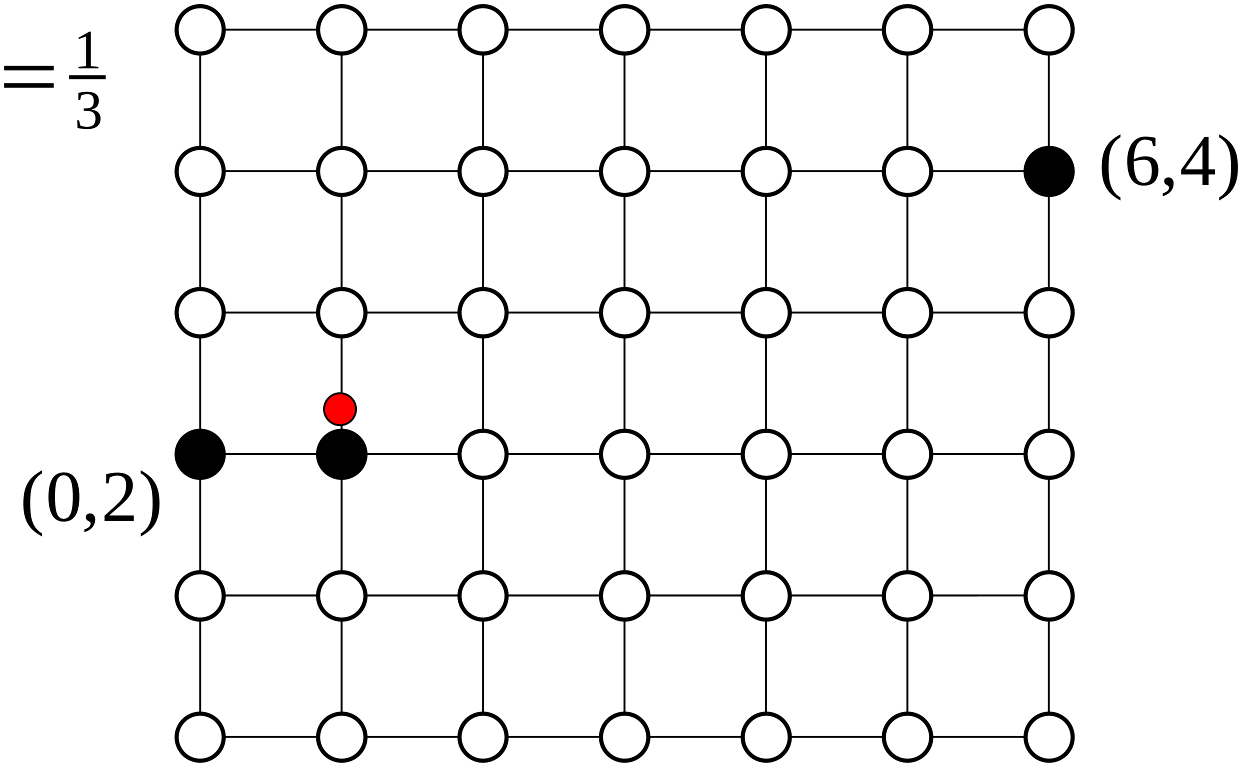
Simple Algorithm - Example

$$y = mx + b$$

$$m = \frac{y_H - y_L}{x_H - x_L} = \frac{4-2}{6-0} = \frac{1}{3}$$

$$x = 1$$

$$y = \frac{7}{3}$$



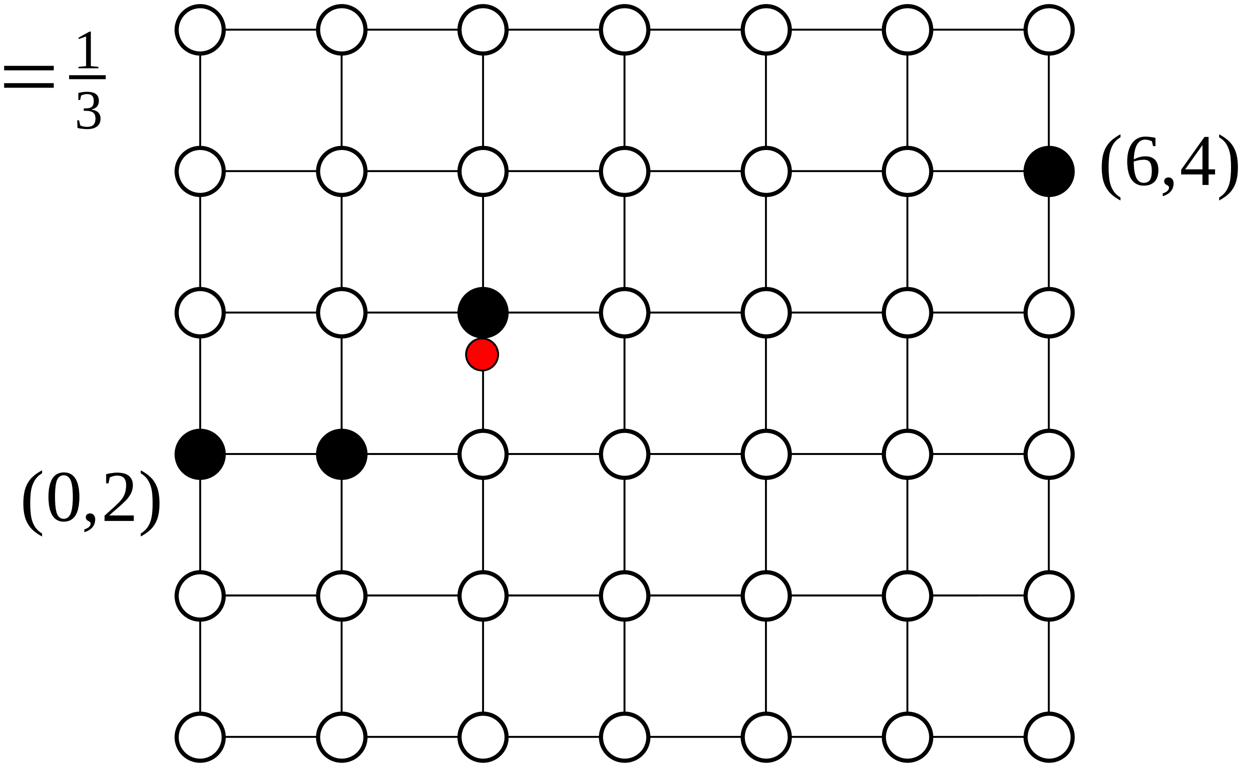
Simple Algorithm - Example

$$y = mx + b$$

$$m = \frac{y_H - y_L}{x_H - x_L} = \frac{4-2}{6-0} = \frac{1}{3}$$

$$x = 2$$

$$y = \frac{8}{3}$$



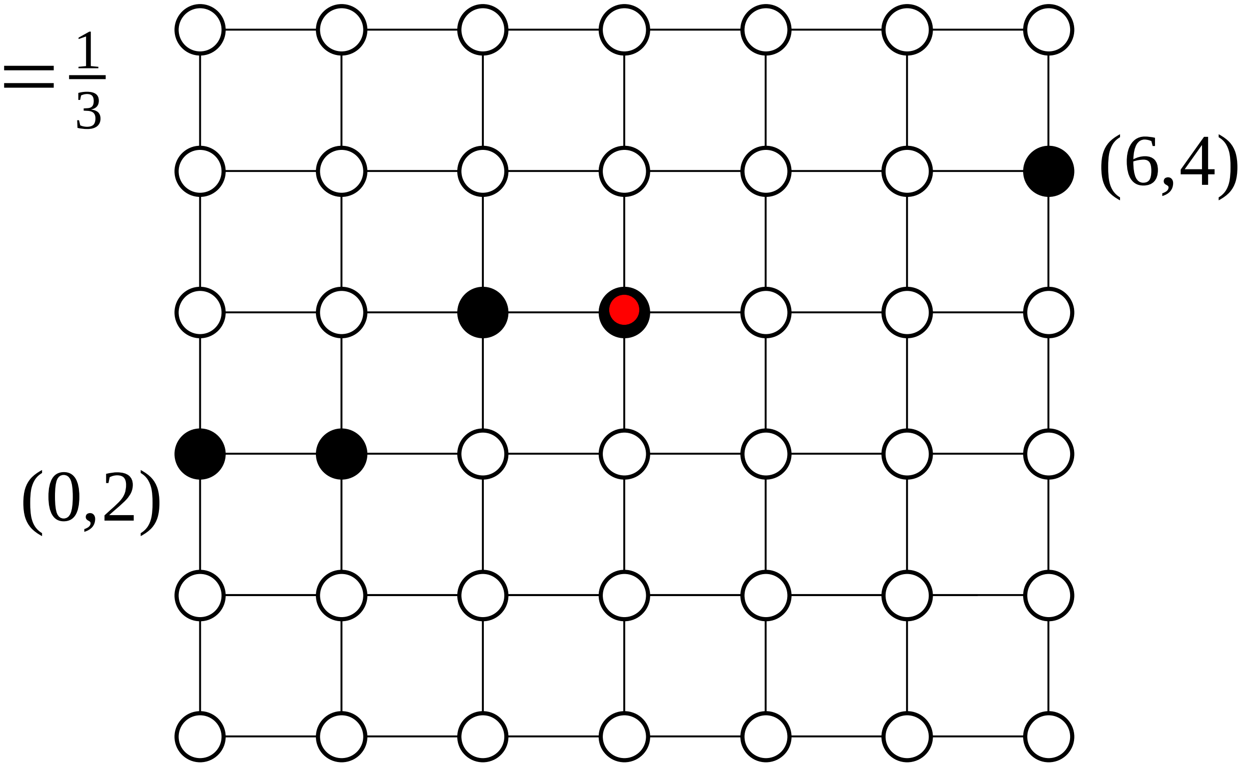
Simple Algorithm - Example

$$y = mx + b$$

$$m = \frac{y_H - y_L}{x_H - x_L} = \frac{4-2}{6-0} = \frac{1}{3}$$

$$x = 3$$

$$y = 3$$



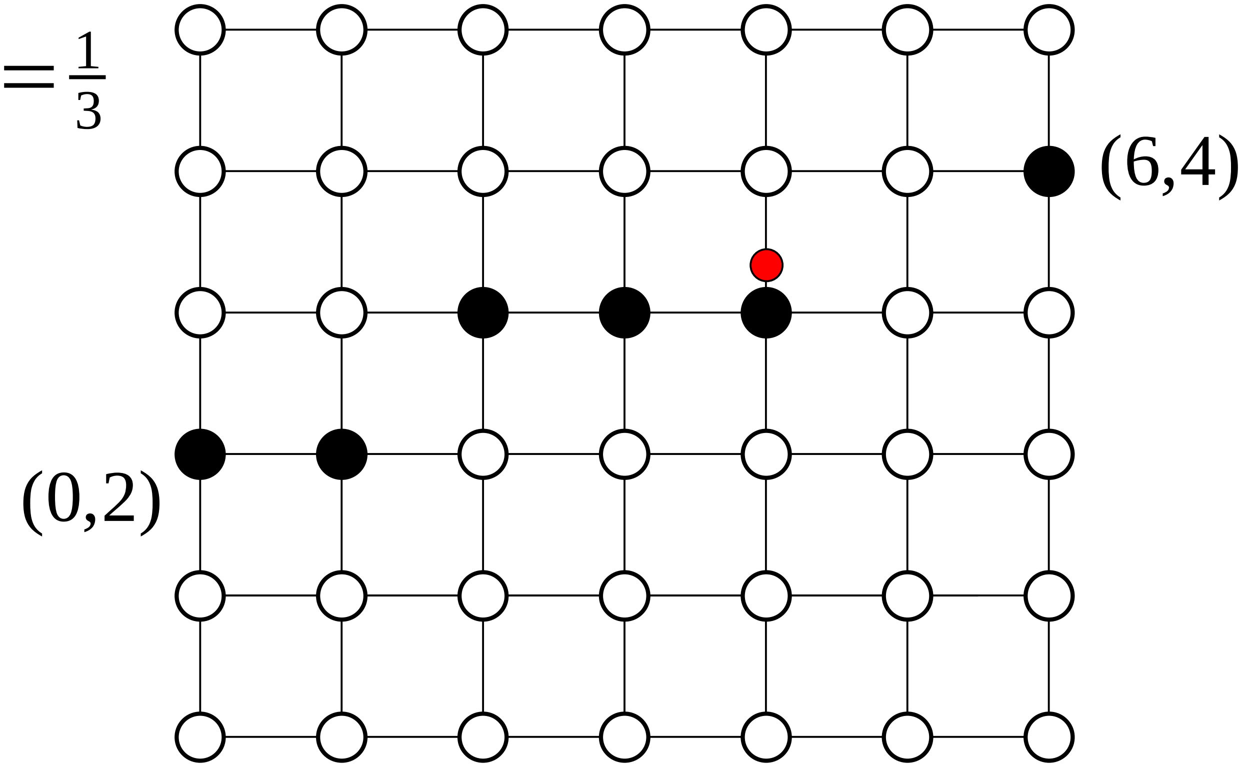
Simple Algorithm - Example

$$y = mx + b$$

$$m = \frac{y_H - y_L}{x_H - x_L} = \frac{4-2}{6-0} = \frac{1}{3}$$

$$x = 4$$

$$y = \frac{10}{3}$$



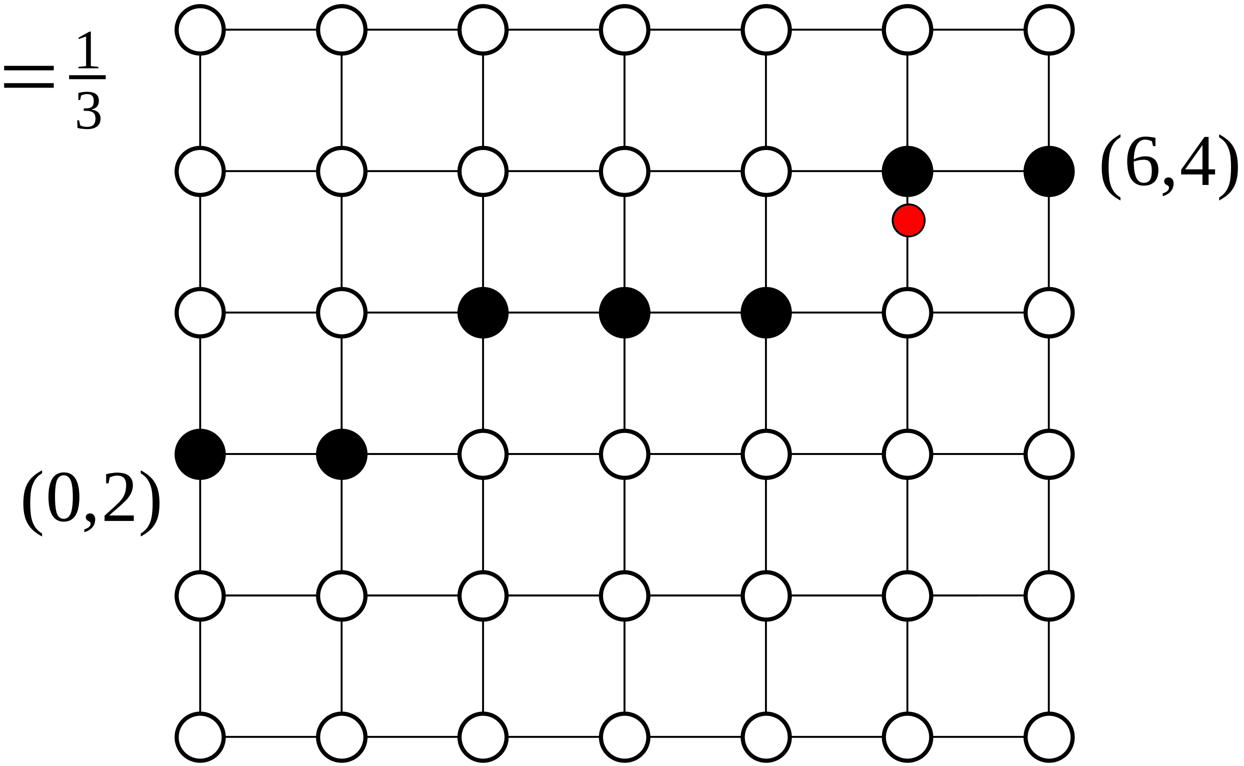
Simple Algorithm - Example

$$y = mx + b$$

$$m = \frac{y_H - y_L}{x_H - x_L} = \frac{4-2}{6-0} = \frac{1}{3}$$

$$x = 5$$

$$y = \frac{11}{3}$$



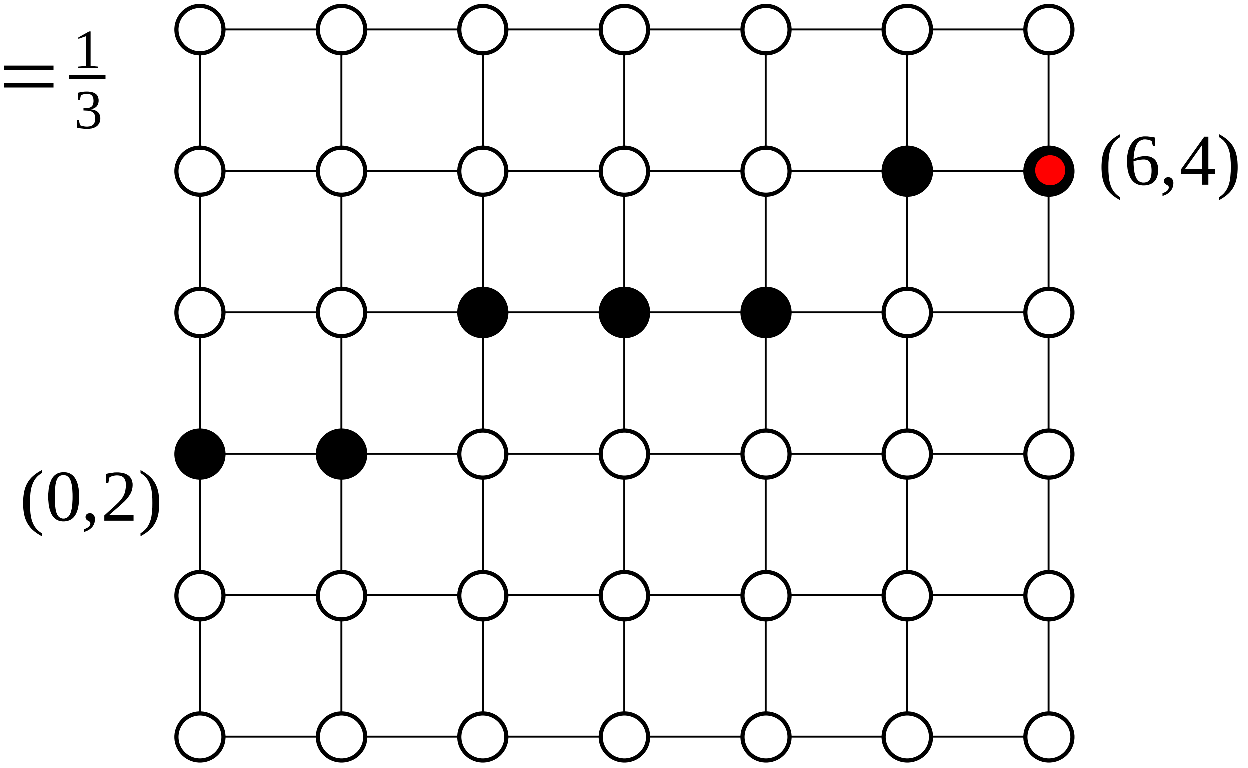
Simple Algorithm - Example

$$y = mx + b$$

$$m = \frac{y_H - y_L}{x_H - x_L} = \frac{4-2}{6-0} = \frac{1}{3}$$

$$x = 6$$

$$y = 4$$



Simple Algorithm - Problems

- Floating point operations
- Rounding
- Can we do better?

$(x_0, y_0) = (x_L, y_L)$

For ($i = 0$; $i <= x_H - x_L$; i++)

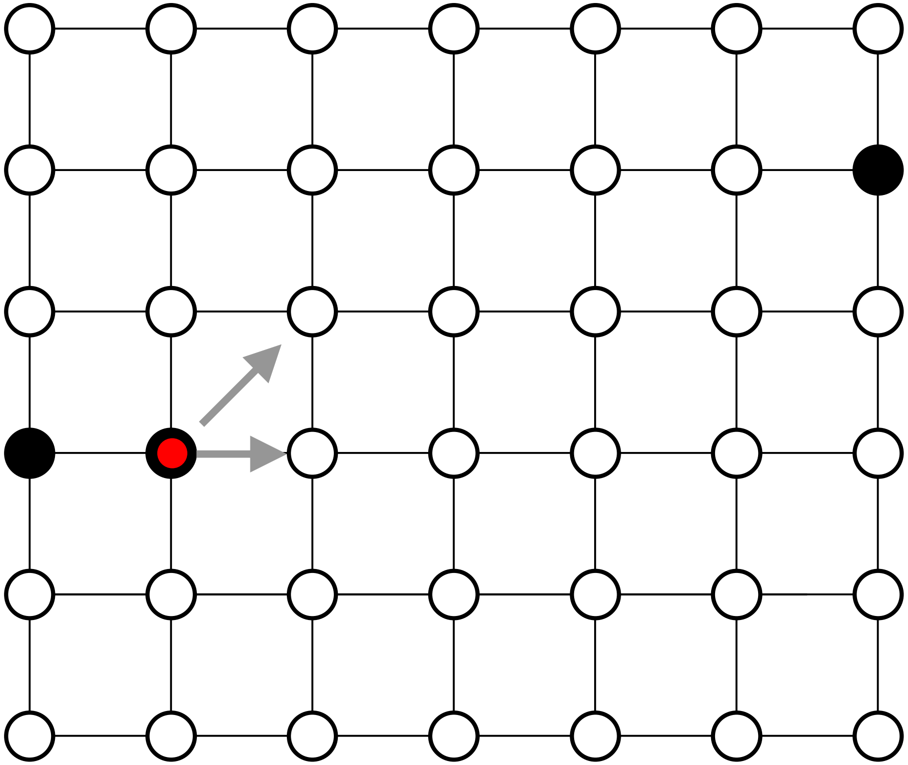
  DrawPixel (x_i, Round (y_i))

  $x_{i+1} = x_i + 1$

  $y_{i+1} = y_i + m$

Midpoint Algorithm

- Given a point just drawn, determine whether we move E or NE on next step



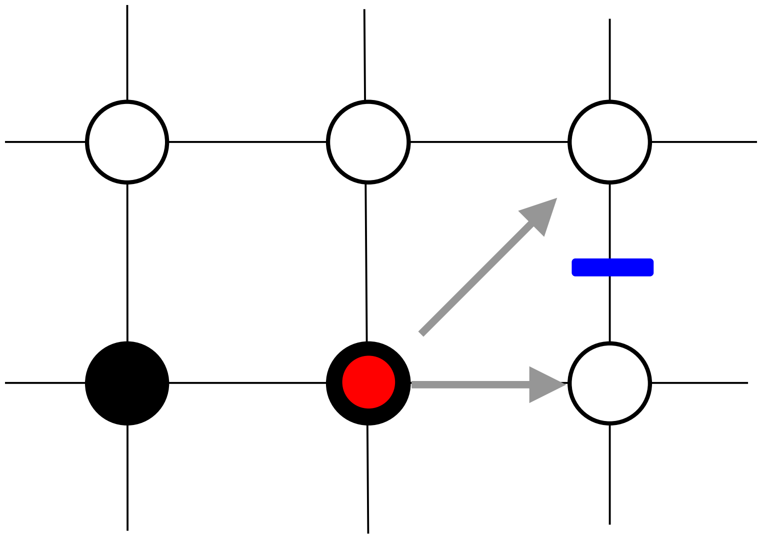
Midpoint Algorithm

- Given a point just drawn, determine whether we move E or NE on next step

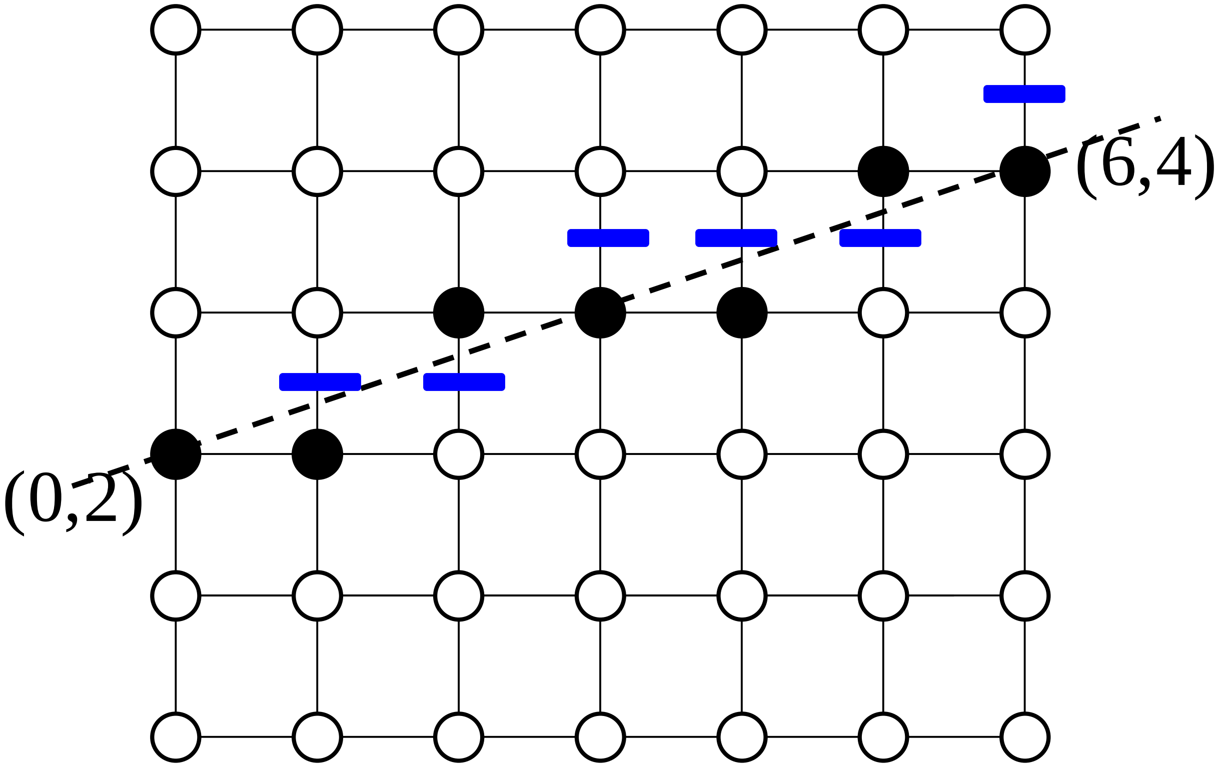
- Is the line above or below $(x+1, y+\frac{1}{2})$?

  Below: move E

  Above: move NE



Midpoint Algorithm



Midpoint Algorithm – Implicit Forms

- How do we tell if a point is above or below the line?

$$y = mx + b$$

Midpoint Algorithm – Implicit Forms

- How do we tell if a point is above or below the line?

$$y = \frac{y_H - y_L}{x_H - x_L} x + b$$

Midpoint Algorithm – Implicit Forms

- How do we tell if a point is above or below the line?

$$(x_H - x_L)y = (\frac{y_H - y_L}{x_H - x_L}x + b)(x_H - x_L)$$

Midpoint Algorithm – Implicit Forms

- How do we tell if a point is above or below the line?

$$(x_H - x_L)y = (y_H - y_L)x + (x_H - x_L)b$$

Midpoint Algorithm – Implicit Forms

- How do we tell if a point is above or below the line?
  $(x_H - x_L)y + (y_L - y_H)x + (x_L - x_H)b = 0$

Midpoint Algorithm – Implicit Forms

- How do we tell if a point is above or below the line?

$$(x_H - x_L)y + (y_L - y_H)x + (x_L - x_H)b = 0$$

$$f(x,y) = cx + dy + e$$

$$c = y_L - y_H \qquad d = x_H - x_L \qquad e = b(x_L - x_H)$$

Midpoint Algorithm – Implicit Forms

- How do we tell if a point is above or below the line?

$$(x_H - x_L)y + (y_L - y_H)x + (x_L - x_H)b = 0$$

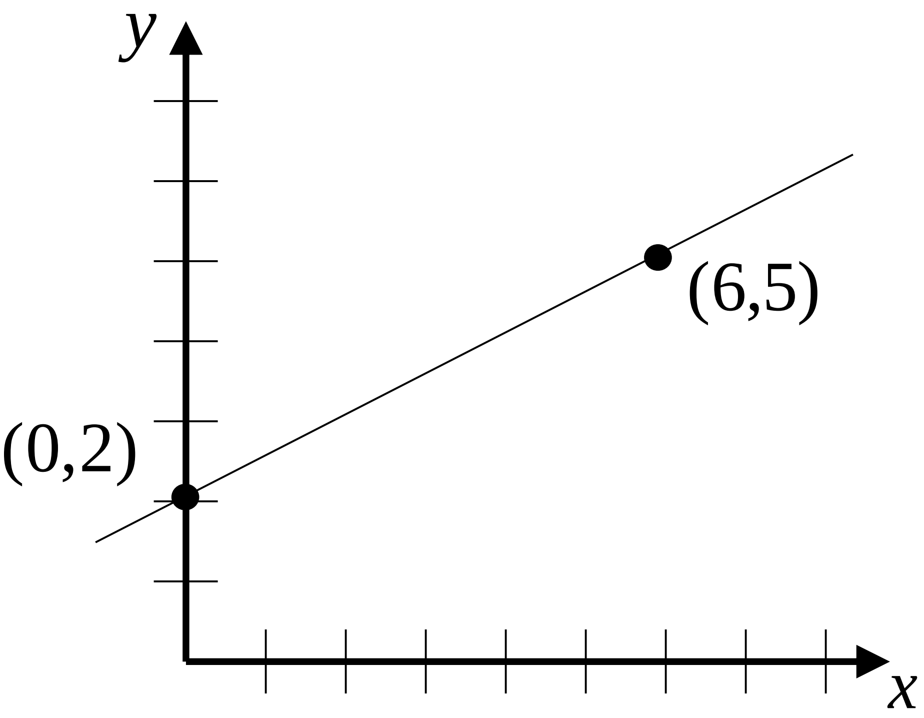
$$f(x, y) = cx + dy + e$$

$$c = y_L - y_H \qquad d = x_H - x_L \qquad e = b(x_L - x_H)$$

Properties	
$f(x,y)=0$	(x,y) on the line
$f(x,y)<0$	(x,y) below the line
$f(x,y)>0$	(x,y) above the line

Midpoint Algorithm – Implicit Forms

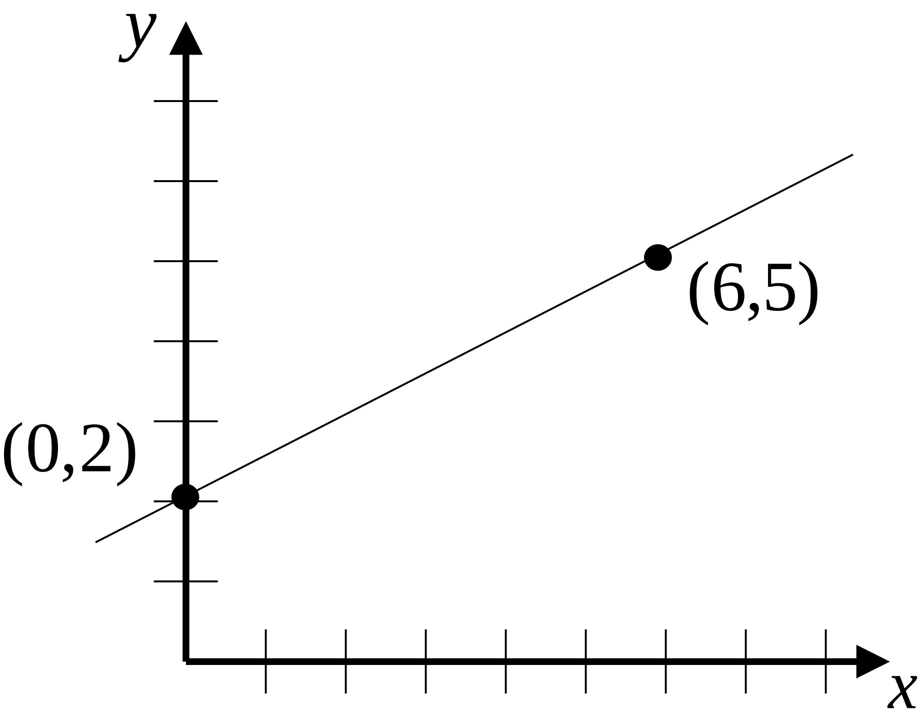
$$y = \frac{3}{6}x + 2$$



Midpoint Algorithm – Implicit Forms

$$y = \frac{3}{6}x + 2$$

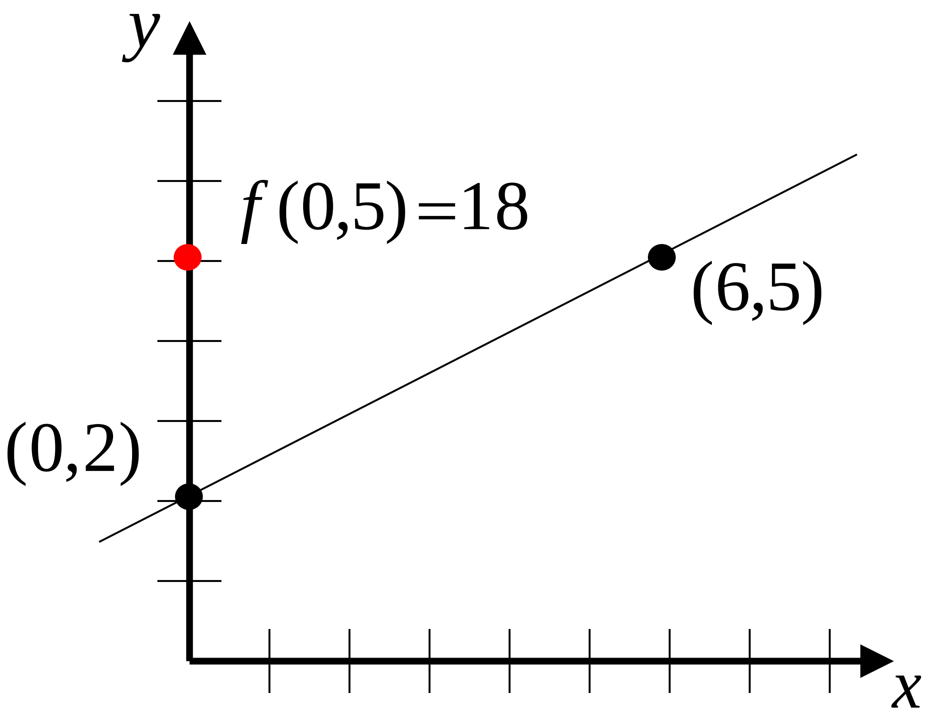
$$f(x, y) = 6y - 3x - 12$$



Midpoint Algorithm – Implicit Forms

$$y = \frac{3}{6}x + 2$$

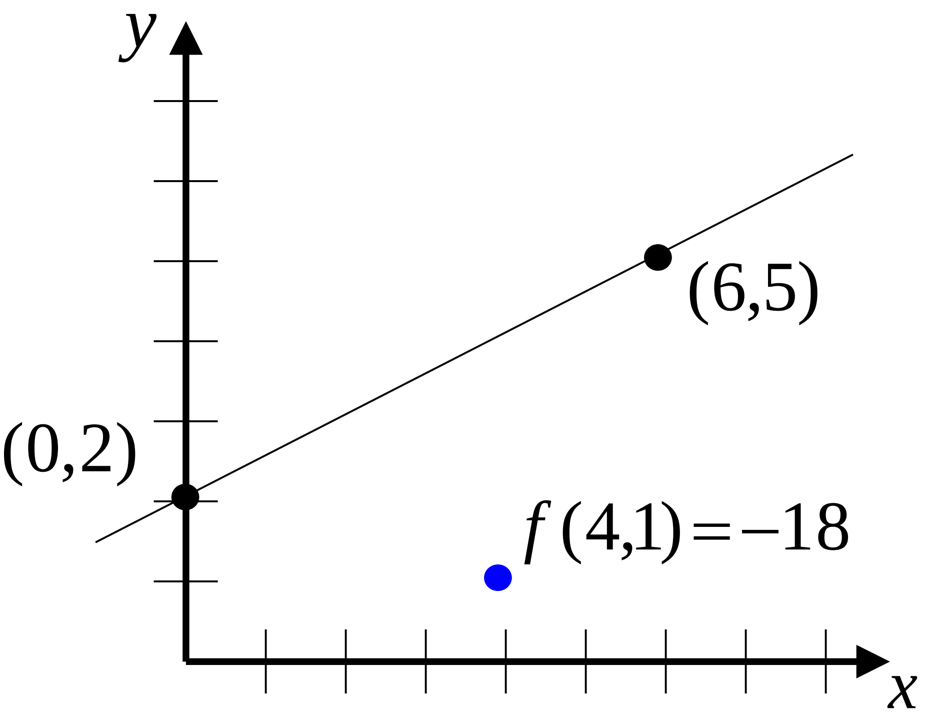
$$f(x, y) = 6y - 3x - 12$$



Midpoint Algorithm – Implicit Forms

$$y = \frac{3}{6}x + 2$$

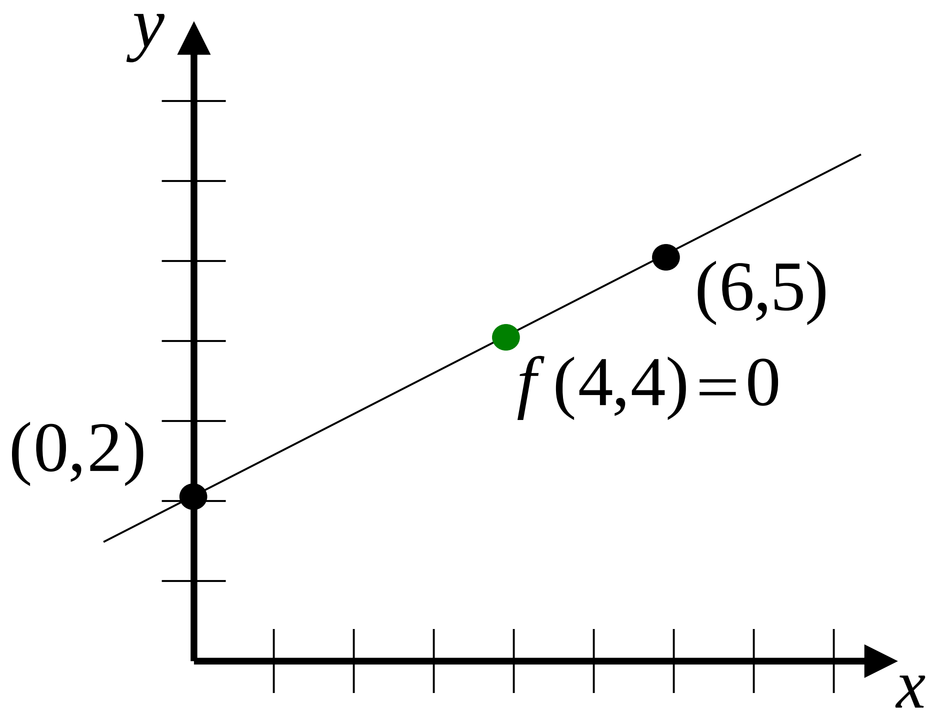
$$f(x, y) = 6y - 3x - 12$$



Midpoint Algorithm – Implicit Forms

$$y = \frac{3}{6}x + 2$$

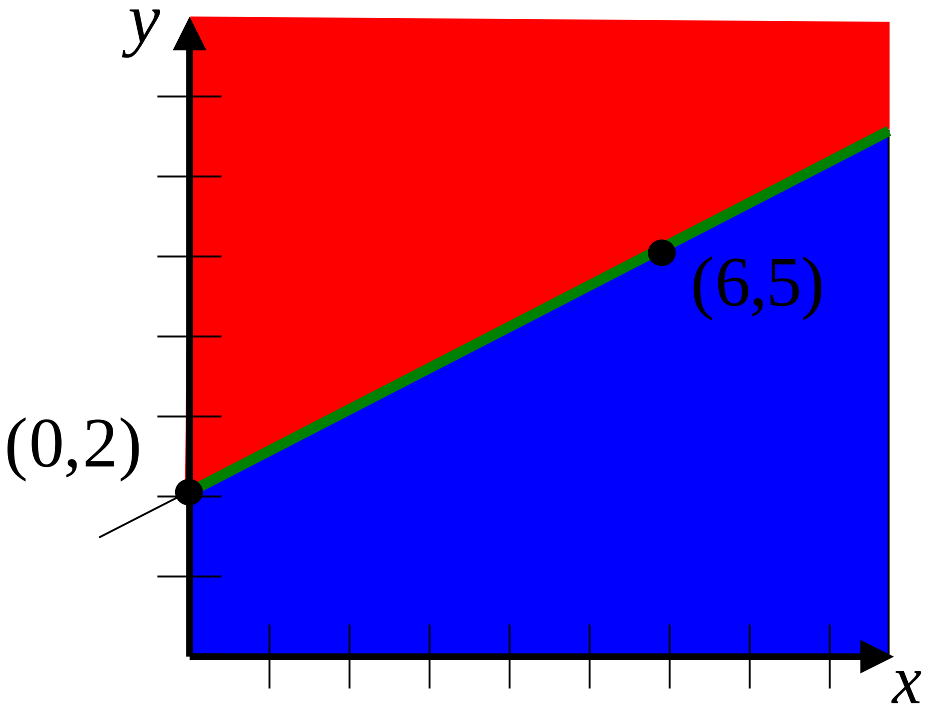
$$f(x, y) = 6y - 3x - 12$$



Midpoint Algorithm – Implicit Forms

$$y = \frac{3}{6}x + 2$$

$$f(x,y) = 6y - 3x - 12$$

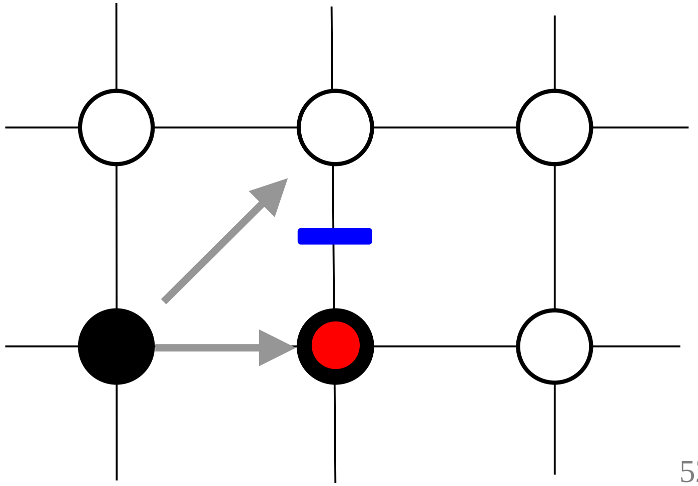


Midpoint Algorithm

- Need value of $f(x+1, y+\frac{1}{2})$ to decide E or NE
- Build incremental algorithm
- Assume we have value of $f(x+1, y+\frac{1}{2})$

  Find value of $f(x+2, y+\frac{1}{2})$ if E chosen

  Find value of $f(x+2, y+\frac{3}{2})$ if NE chosen

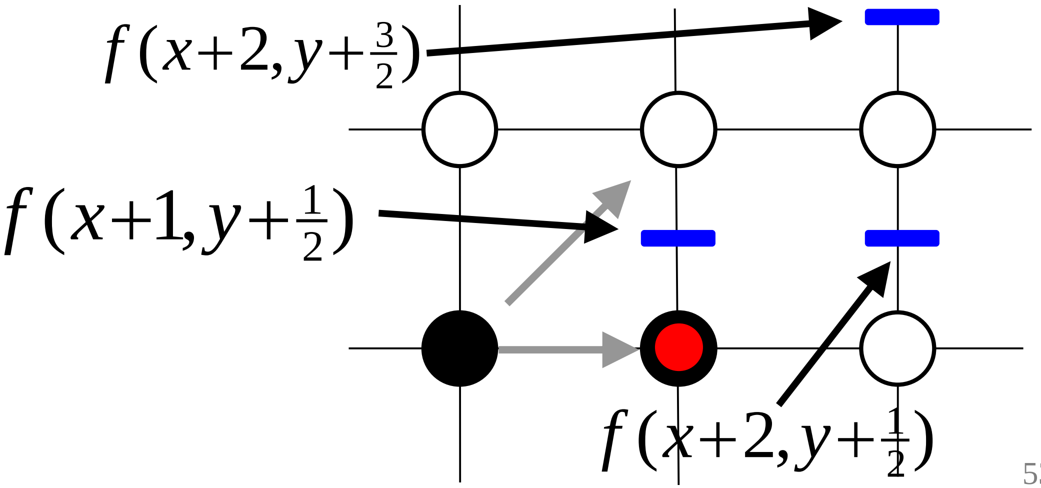


Midpoint Algorithm

- Need value of $f(x+1, y+\frac{1}{2})$ to decide E or NE
- Build incremental algorithm
- Assume we have value of $f(x+1, y+\frac{1}{2})$

  Find value of $f(x+2, y+\frac{1}{2})$ if E chosen

  Find value of $f(x+2, y+\frac{3}{2})$ if NE chosen



Midpoint Algorithm

If E was chosen, find $f(x+2, y+\frac{1}{2})$

$$f(x+2, y+\tfrac{1}{2}) = c(x+2) + d(y+\tfrac{1}{2}) + e$$

Midpoint Algorithm

If E was chosen, find $f(x+2, y+\frac{1}{2})$

$$f(x+2, y+\tfrac{1}{2}) = c(x+2) + d(y+\tfrac{1}{2}) + e$$

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

Midpoint Algorithm

If NE was chosen, find $f(x+2, y+\frac{3}{2})$

$$f(x+2, y+\tfrac{3}{2}) = c(x+2) + d(y+\tfrac{3}{2}) + e$$

Midpoint Algorithm

If NE was chosen, find $f(x+2, y+\frac{3}{2})$

$$f(x+2, y+\tfrac{3}{2}) = c(x+2) + d(y+\tfrac{3}{2}) + e$$

$$f(x+2, y+\tfrac{3}{2}) = c + d + f(x+1, y+\tfrac{1}{2})$$

Midpoint Algorithm

What about starting value?

$$f(x_L+1, y_L+\tfrac{1}{2}) = c(x_L+1) + d(y_L+\tfrac{1}{2}) + e$$

Midpoint Algorithm

What about starting value?

$$f(x_L+1, y_L+\tfrac{1}{2}) = c(x_L+1) + d(y_L+\tfrac{1}{2}) + e$$

$$f(x_L+1, y_L+\tfrac{1}{2}) = f(x_L, y_L) + c + \tfrac{1}{2}d$$

Midpoint Algorithm

What about starting value?

$$f(x_L+1, y_L+\tfrac{1}{2}) = c(x_L+1) + d(y_L+\tfrac{1}{2}) + e$$

$$f(x_L+1, y_L+\tfrac{1}{2}) = \cancel{f(x_L, y_L)} + c + \tfrac{1}{2}d$$

(x_L, y_L) is on the line!

Midpoint Algorithm

What about starting value?

$$f(x_L + 1, y_L + \tfrac{1}{2}) = c(x_L + 1) + d(y_L + \tfrac{1}{2}) + e$$

$$f(x_L + 1, y_L + \tfrac{1}{2}) = c + \tfrac{1}{2}d$$

Midpoint Algorithm

What about starting value?

$$f(x_L+1, y_L+\tfrac{1}{2}) = c(x_L+1) + d(y_L+\tfrac{1}{2}) + e$$

$$f(x_L+1, y_L+\tfrac{1}{2}) = 2c + d$$

Multiplying coefficients by 2 doesn't change sign of f

Midpoint Algorithm – Summary

$x=x_L$ $\qquad y=y_L$

$d=x_H - x_L \quad c=y_L - y_H$

$sum=2c+d$

```
draw(x,y)
while ( x < x_H)
    if ( sum < 0 )
        sum += 2d
        y++
    x++
    sum += 2c
    draw(x,y)
```

Midpoint Algorithm – Summary

$x = x_L$ $\quad\quad$ $y = y_L$

$d = x_H - x_L$ $\quad$ $c = y_L - y_H$

sum=2c+d //initial value

draw(x,y)

while ($x < x_H$)

    if (*sum* < 0)

        sum += *2d*

        y++

    x++

    sum += *2c*

    draw(x,y)

Midpoint Algorithm – Summary

```
x=x_L              y=y_L
d=x_H - x_L    c=y_L - y_H
sum=2c+d //initial value
draw(x,y)
while ( x < x_H) // iterate until reaching the end point
    if ( sum < 0 )
        sum += 2d
        y++
    x++
    sum += 2c
    draw(x,y)
```

Midpoint Algorithm – Summary

$x=x_L$ $y=y_L$

$d=x_H - x_L$ $c=y_L - y_H$

sum=2c+d //initial value

draw(x,y)

while ($x < x_H$) // *iterate until reaching the end point*

  if (*sum* < 0) // *below the line and choose NE*

    sum += 2d

    y++

  x++

  sum += *2c*

  draw(x,y)

Midpoint Algorithm – Example

$x = x_L$ $y = y_L$

$d = x_H - x_L$ $c = y_L - y_H$

$sum = 2c + d$

draw(x,y)

while ($x < x_H$)

    if ($sum < 0$)

        sum += 2d

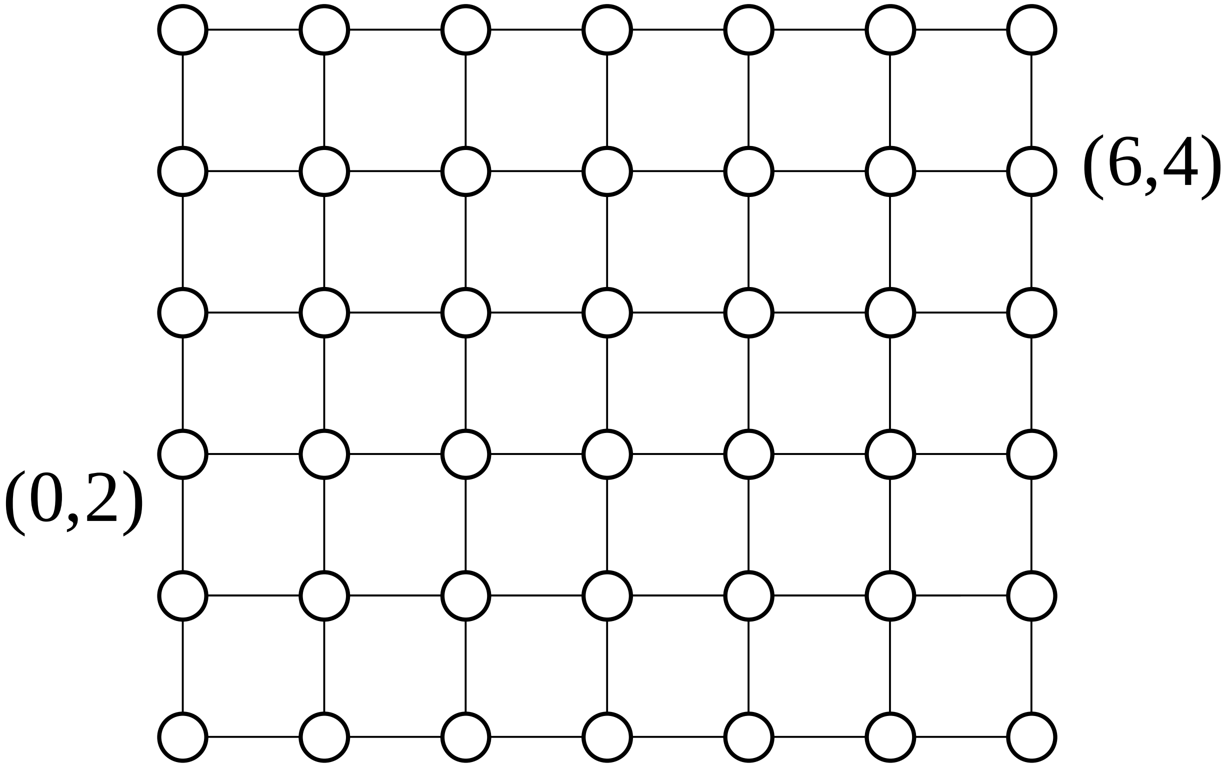
        y++

    x++

    sum += 2c

    draw(x,y)

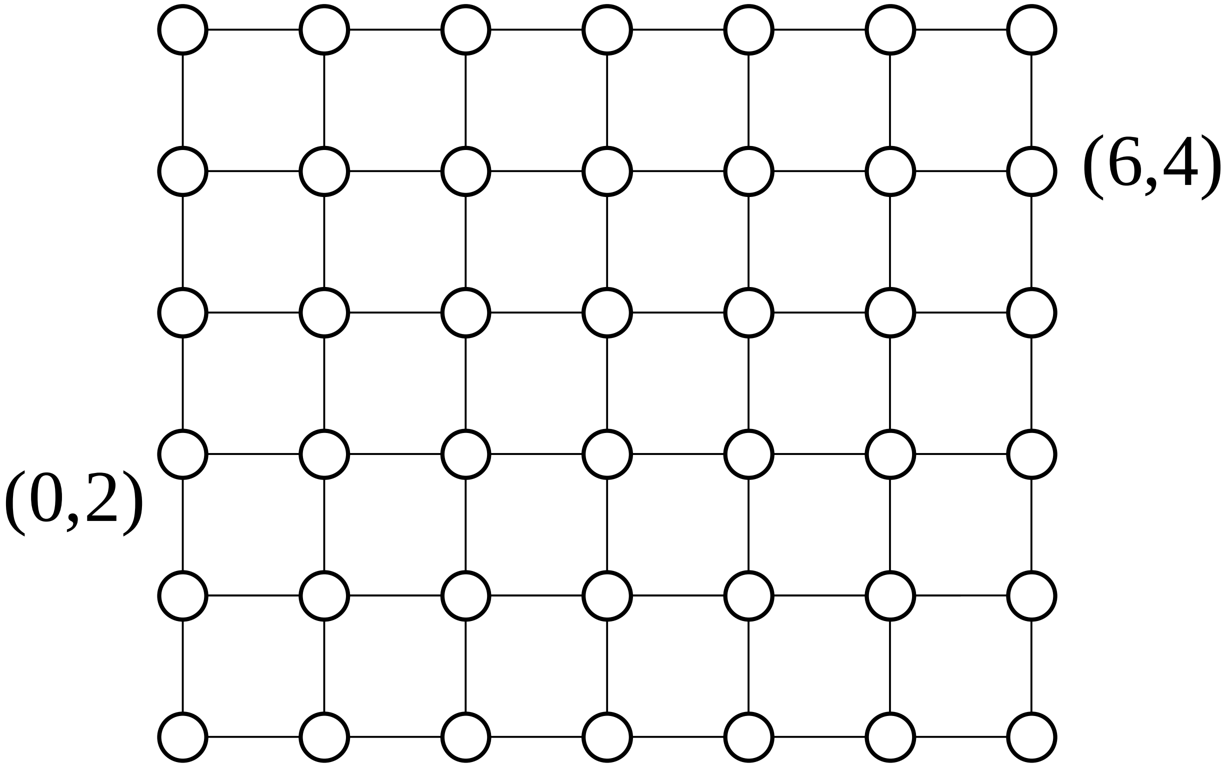
$x = 0$ $y = 2$ $sum = 2$



Midpoint Algorithm – Example

x=x_L y=y_L
d=6 c=-2
sum=2c+d
draw(x,y)
while (x < x_H)
    if (sum < 0)
        sum += 2d
        y++
    x++
    sum += 2c
    draw(x,y)

$x = 0 \quad y = 2 \quad sum = 2$



Midpoint Algorithm – Example

$x = x_L$ $y = y_L$

$d = 6$ $c = -2$

$sum = 2c + d$

draw(x,y)

while ($x < x_H$)

    if ($sum < 0$)

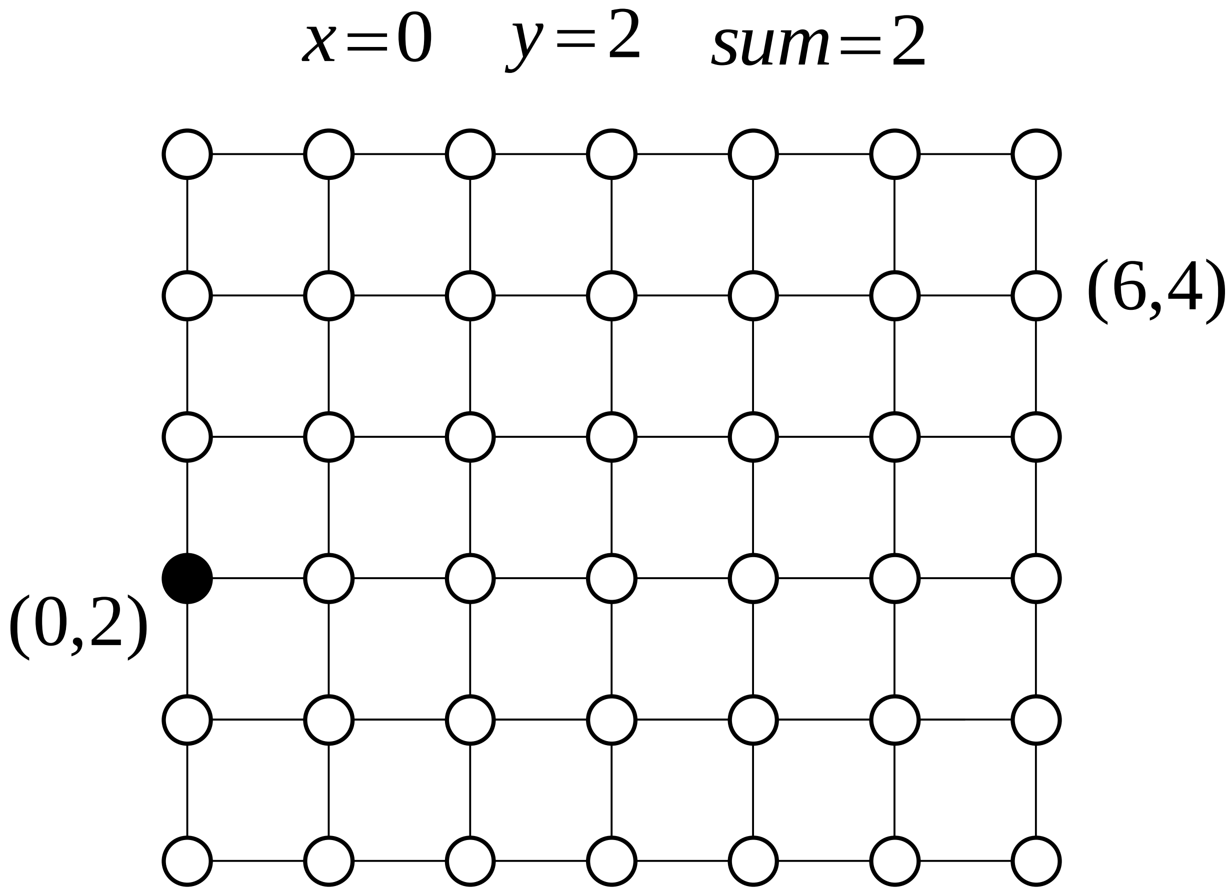
        sum += 2d

        y++

    x++

    sum += 2c

    draw(x,y)



Midpoint Algorithm – Example

$x = x_L$ $y = y_L$

$d = 6$ $c = -2$

$sum = 2c + d$

draw(x,y)

while ($x < x_H$)

    if ($sum < 0$)

        sum += $2d$

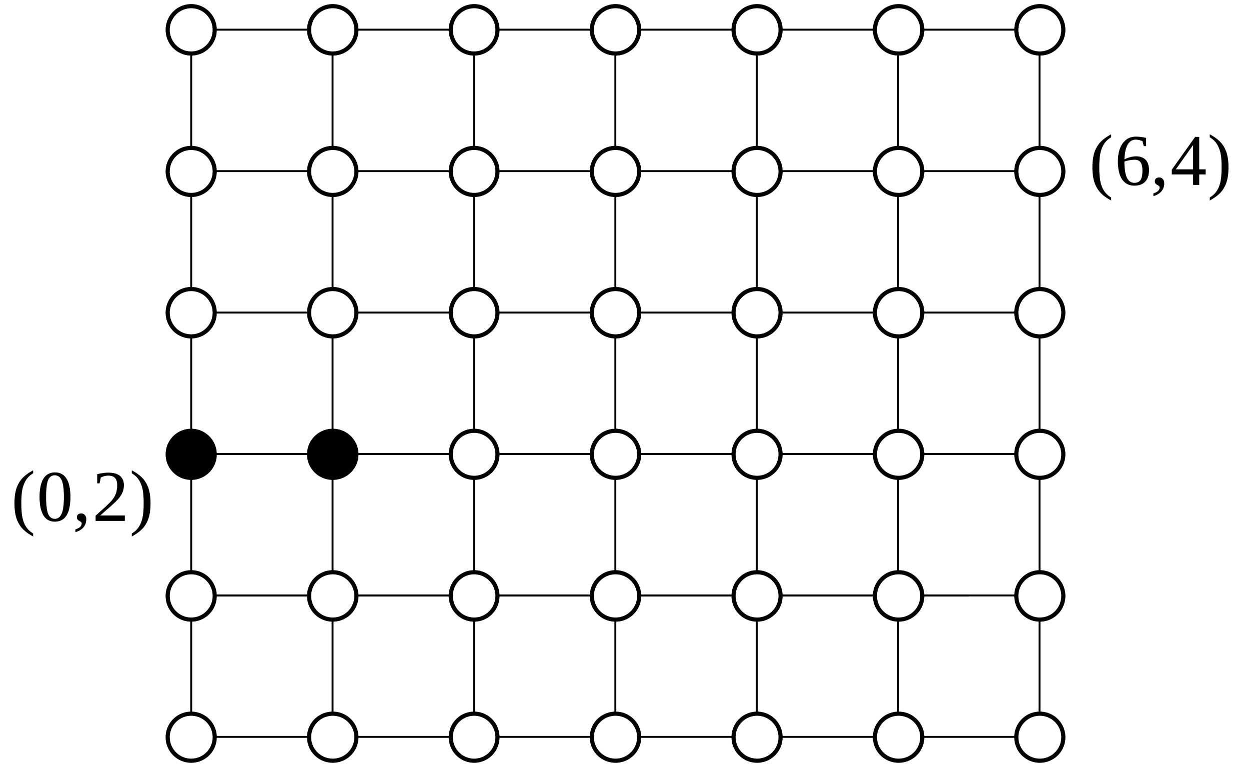
        y++

    x++

    sum += $2c$

    draw(x,y)

$x = 1$ $y = 2$ $sum = -2$



Midpoint Algorithm – Example

x=x_L y=y_L

d=6 c=-2

sum=2c+d

draw(x,y)

while ($x < x_H$)

if (sum < 0)

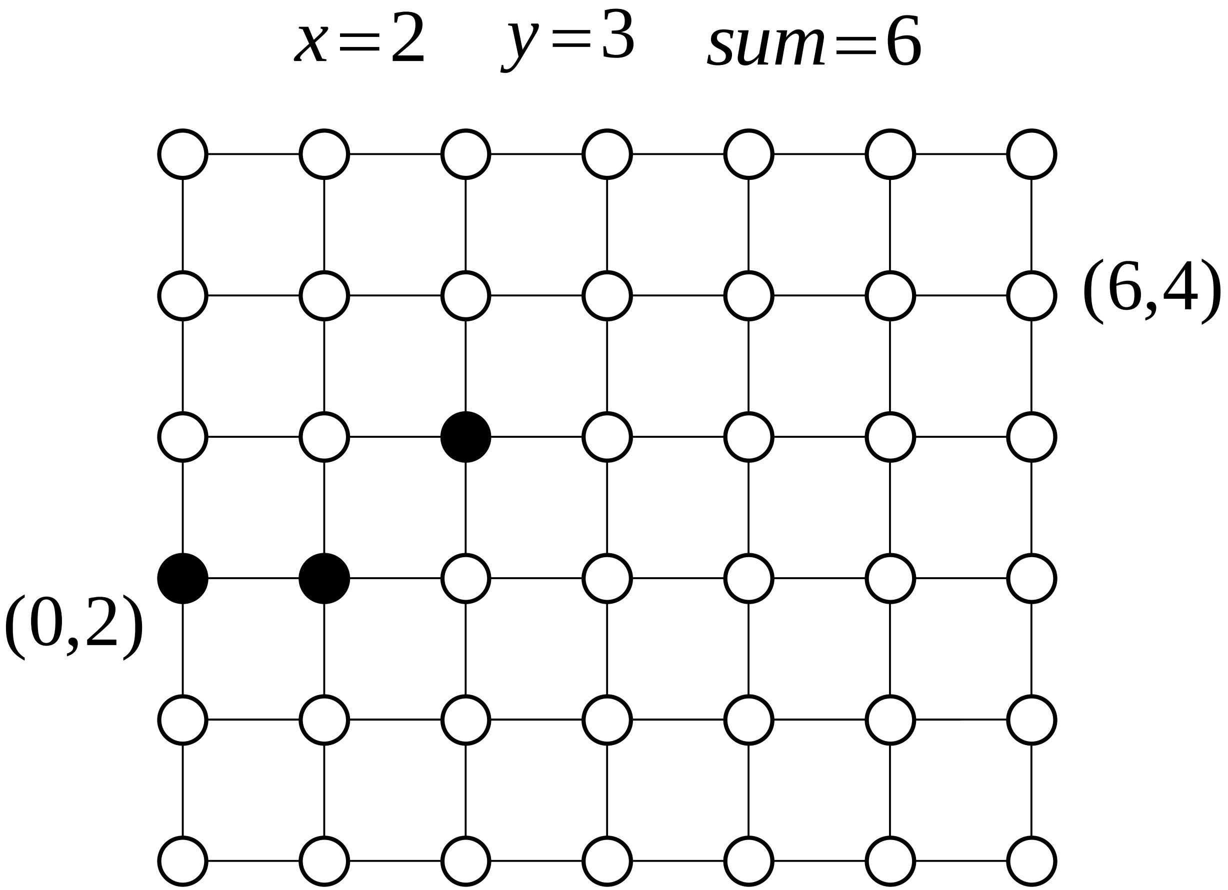
sum += 2d

y++

x++

sum += 2c

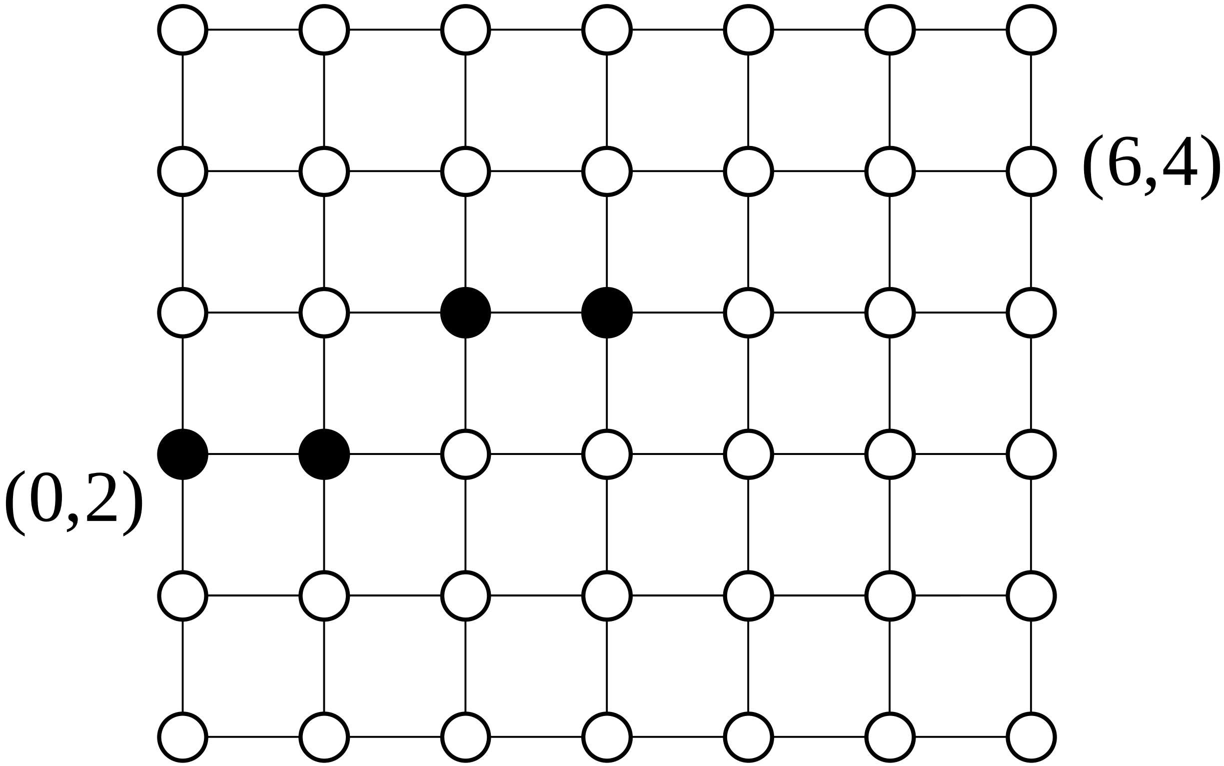
draw(x,y)



Midpoint Algorithm – Example

```
x=x_L          y=y_L
d=6   c=-2
sum=2c+d
draw(x,y)
while ( x < x_H)
    if ( sum < 0 )
        sum += 2d
        y++
    x++
    sum += 2c
    draw(x,y)
```

$x = 3 \quad y = 3 \quad sum = 2$



Midpoint Algorithm – Example

$x=x_L$ $y=y_L$

$d=6$ $c=-2$

$sum=2c+d$

draw(x,y)

while ($x < x_H$)

    if ($sum < 0$)

        sum += 2d

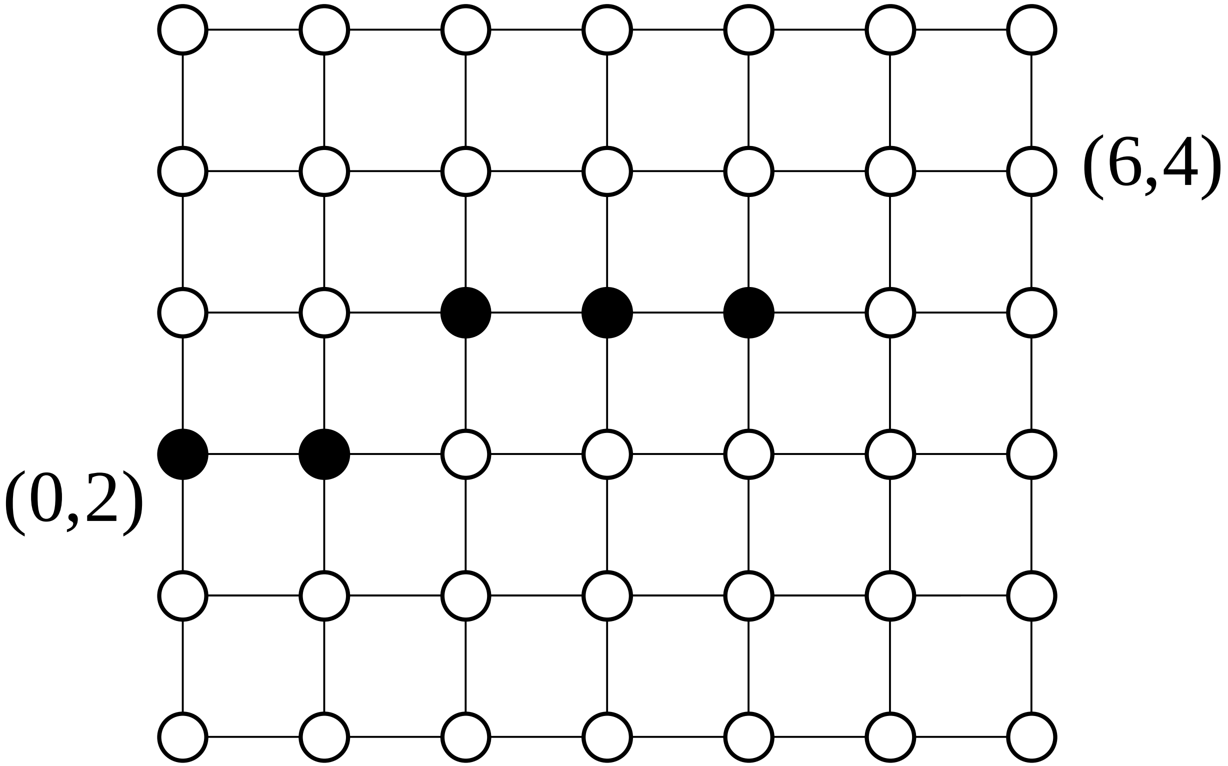
        y++

    x++

    sum += 2c

    draw(x,y)

$x=4$ $y=3$ $sum=-2$



Midpoint Algorithm – Example

$x = x_L$ $\qquad y = y_L$

$d = 6 \quad c = -2$

$sum = 2c + d$

draw(x,y)

while ($x < x_H$)

    if ($sum < 0$)

        sum += $2d$

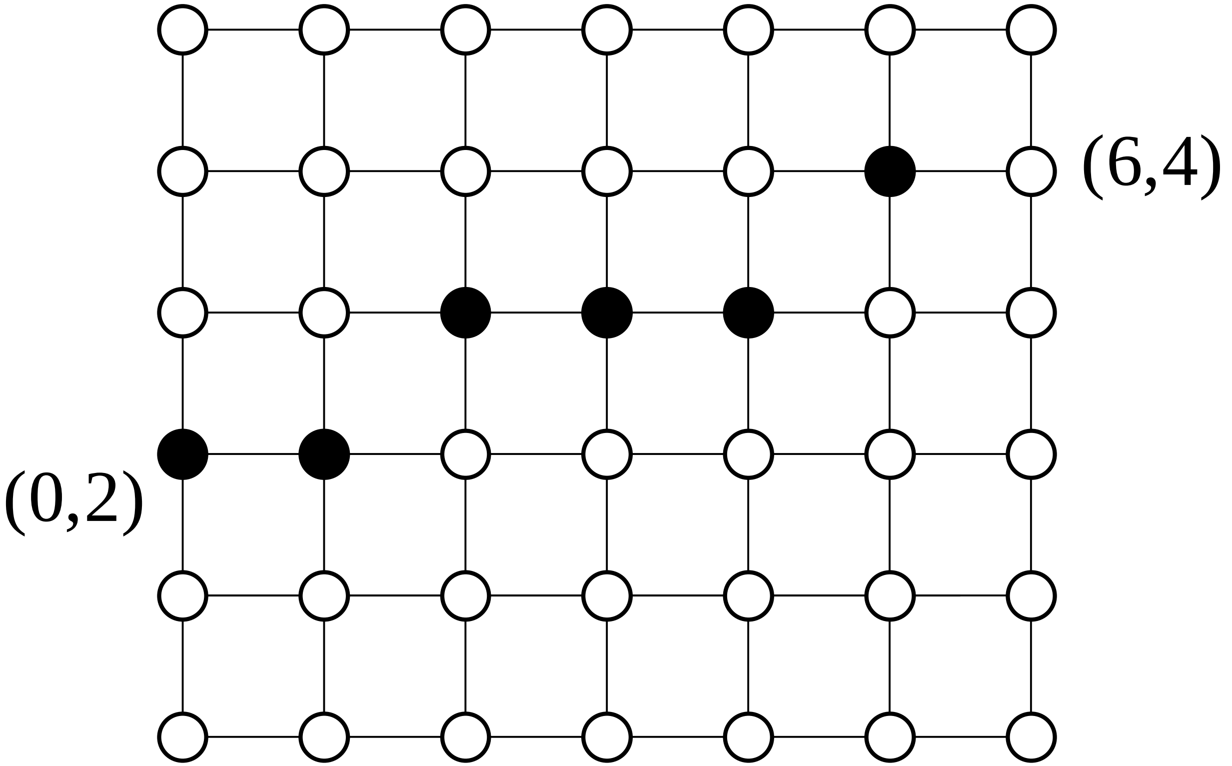
        y++

    x++

    sum += $2c$

    draw(x,y)

$x = 5 \quad y = 4 \quad sum = 6$



Midpoint Algorithm – Example

$x = x_L$ $\qquad y = y_L$

$d = 6 \quad c = -2$

$sum = 2c + d$

draw(x,y)

while ($x < x_H$)

  if ($sum < 0$)

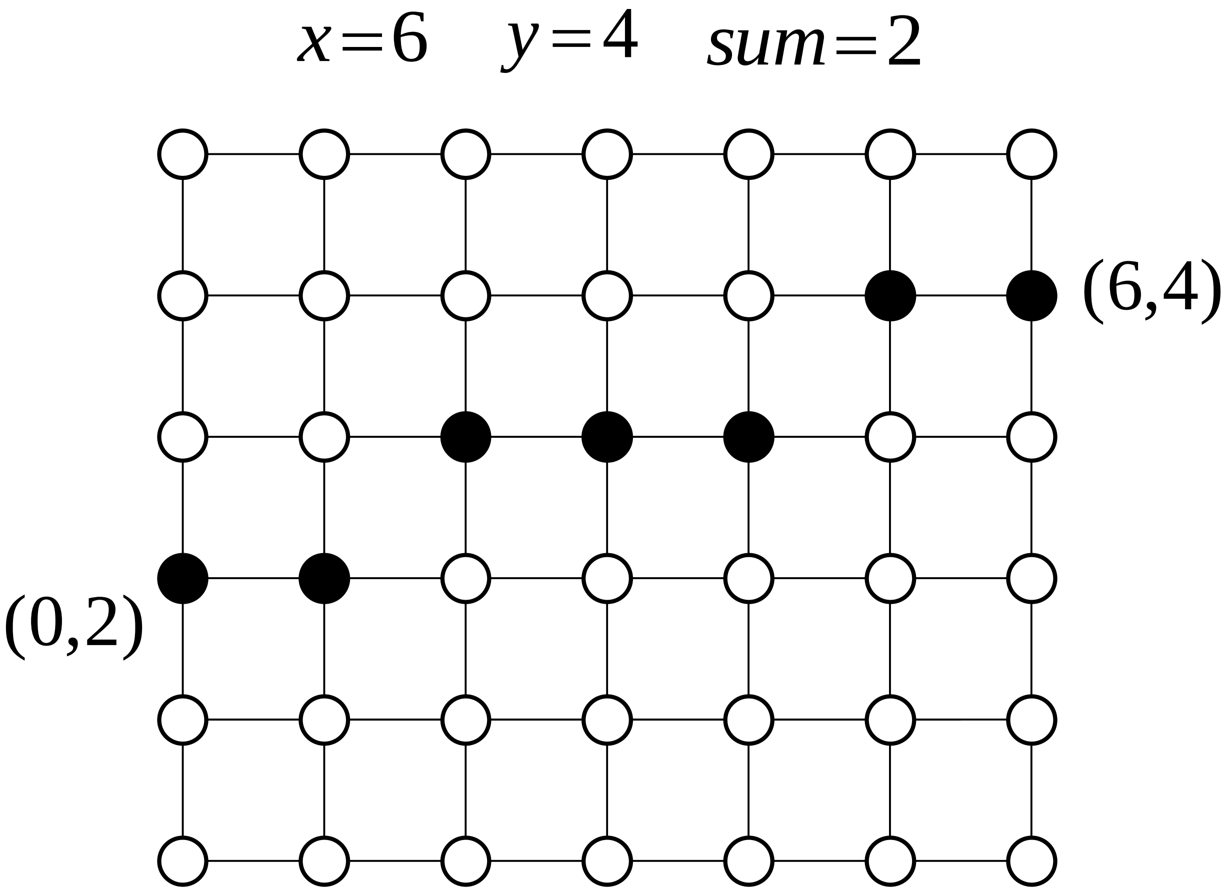
    sum += 2d

    y++

  x++

  sum += 2c

  draw(x,y)



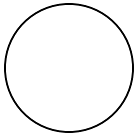
Midpoint Algorithm

- Only integer operations
- Exactly the same as the more commonly found Bresenham's line drawing algorithm
- Extends to other types of shapes (circles)

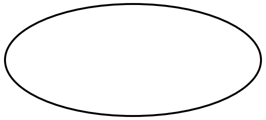
Circle, ellipse, etc…

- Need implicit forms

  - circle: $(x - x_0)^2 + (y - y_0)^2 - r^2 = 0$



  - ellipse: $Ax^2 + By^2 + Cxy + Dx + Ey + F = 0$



- Equations for incremental calculations
- Initial positions
- More details in section 6.4-6.5 of the textbook

OpenGL: Drawing Lines

```
glBegin (GL_LINES);
    glVertex2f (p1.x, p1.y);
    glVertex2f (p2.x, p2.y);
    glVertex2f (p3.x, p3.y);
    glVertex2f (p4.x, p4.y);
    glVertex2f (p5.x, p5.y);
    glVertex2f (p6.x, p6.y);
glEnd();
```