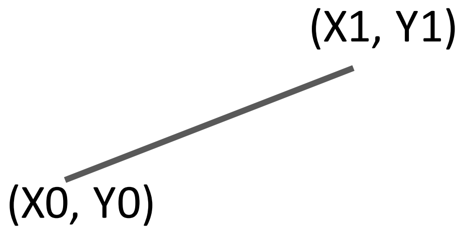
Scan Converting Lines

Bresenham's Line Drawing Algorithm

- Mohammad Imrul Jubair

The Scenario

Given,
Start point (X0, Y0)
End point (X1, Y1)

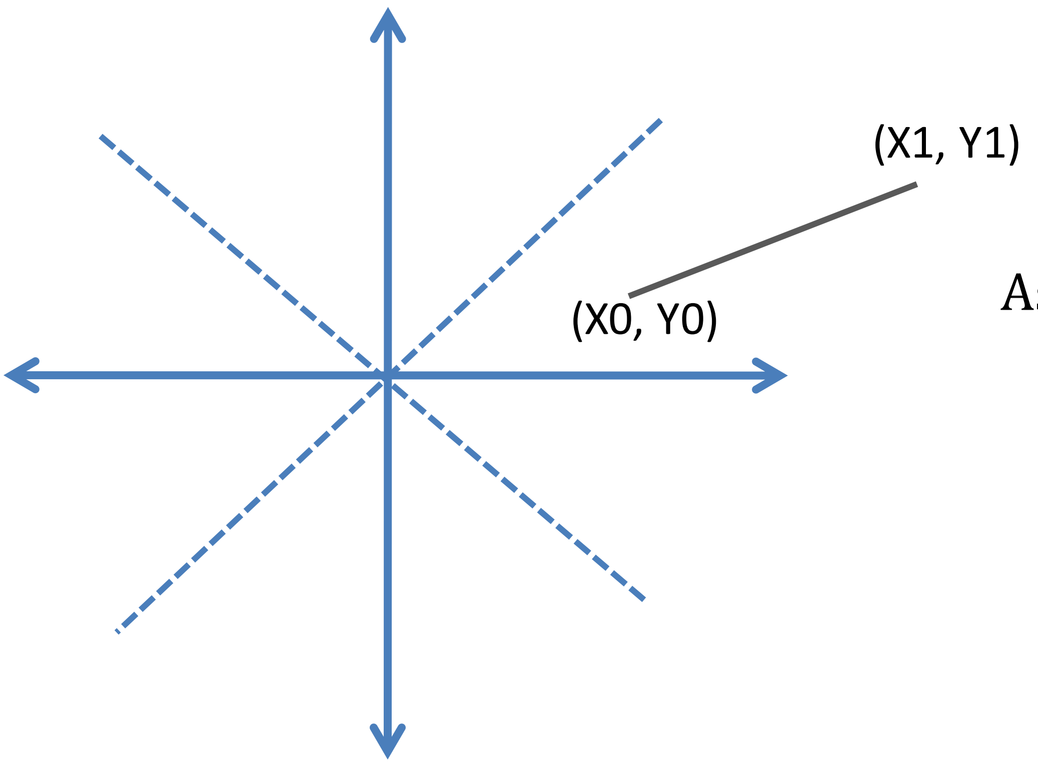


Assumption: (Only the 1st Octant for this time)

Given,

Start point (X0, Y0)

End point (X1, Y1)



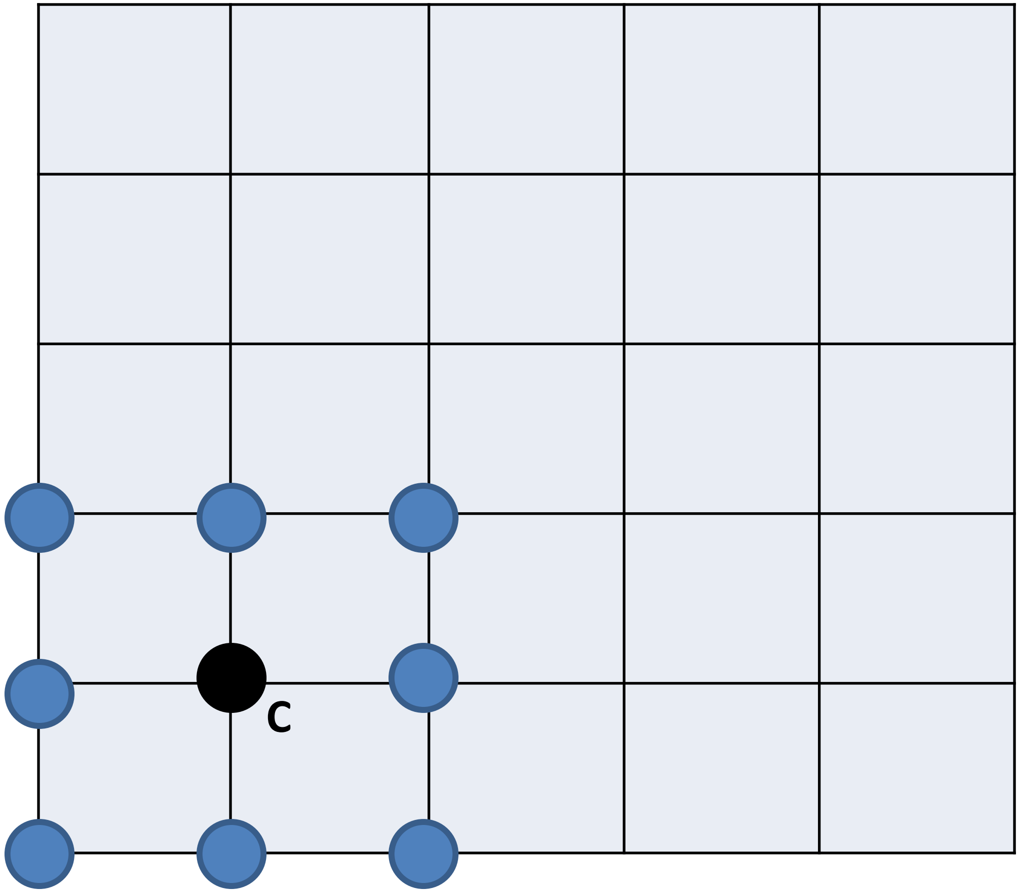
Assume,

X1 > X0

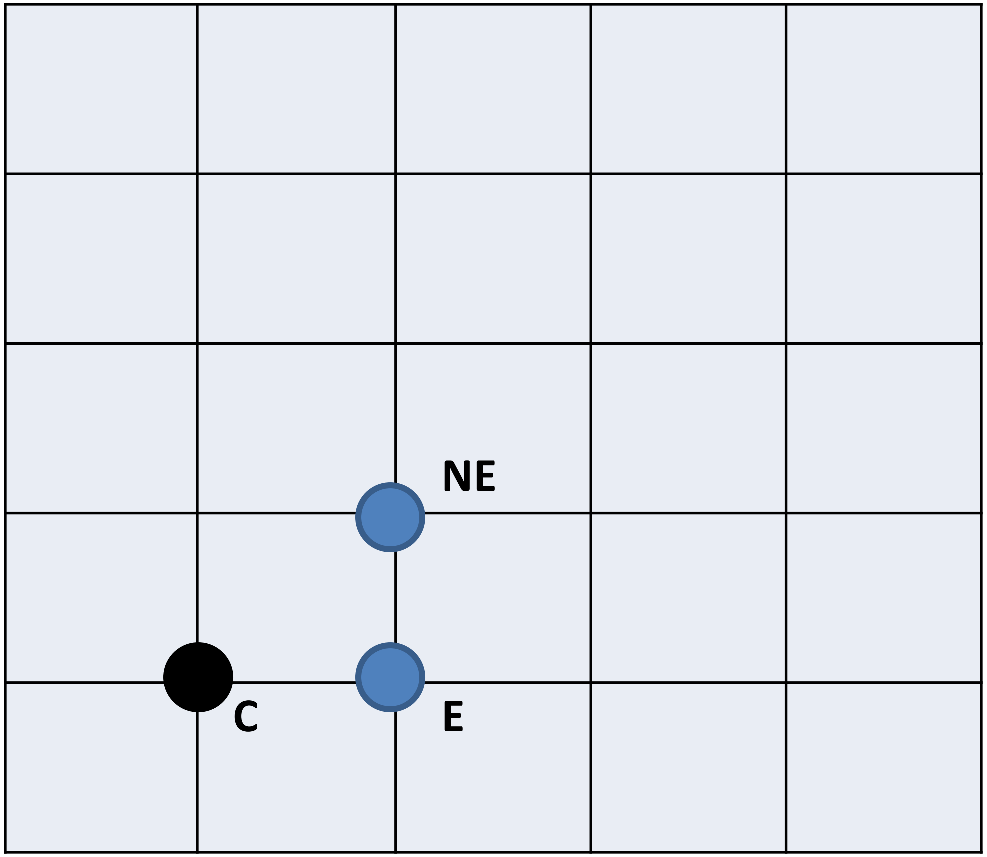
And,

Slope, m < 1

Bresenham's Mid Point Criteria: How it works

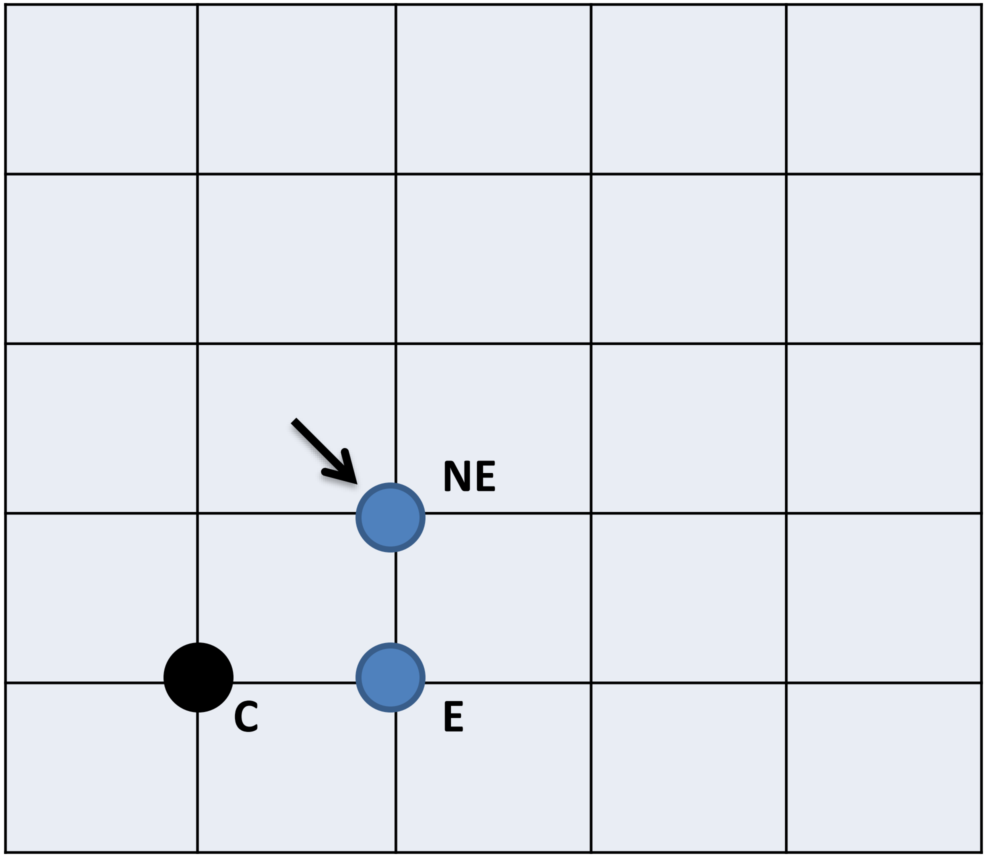


Bresenham's Mid Point Criteria: How it works



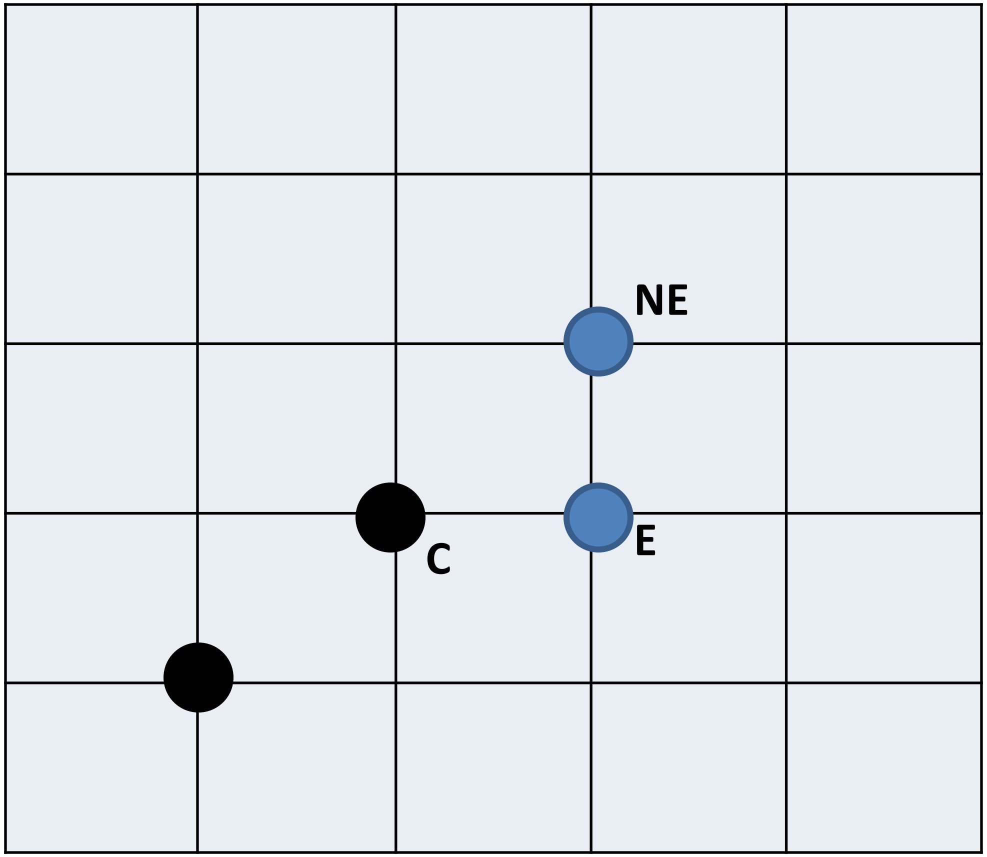
Next pixel is chosen (from E or NE) to build the line successively

Bresenham's Mid Point Criteria: How it works



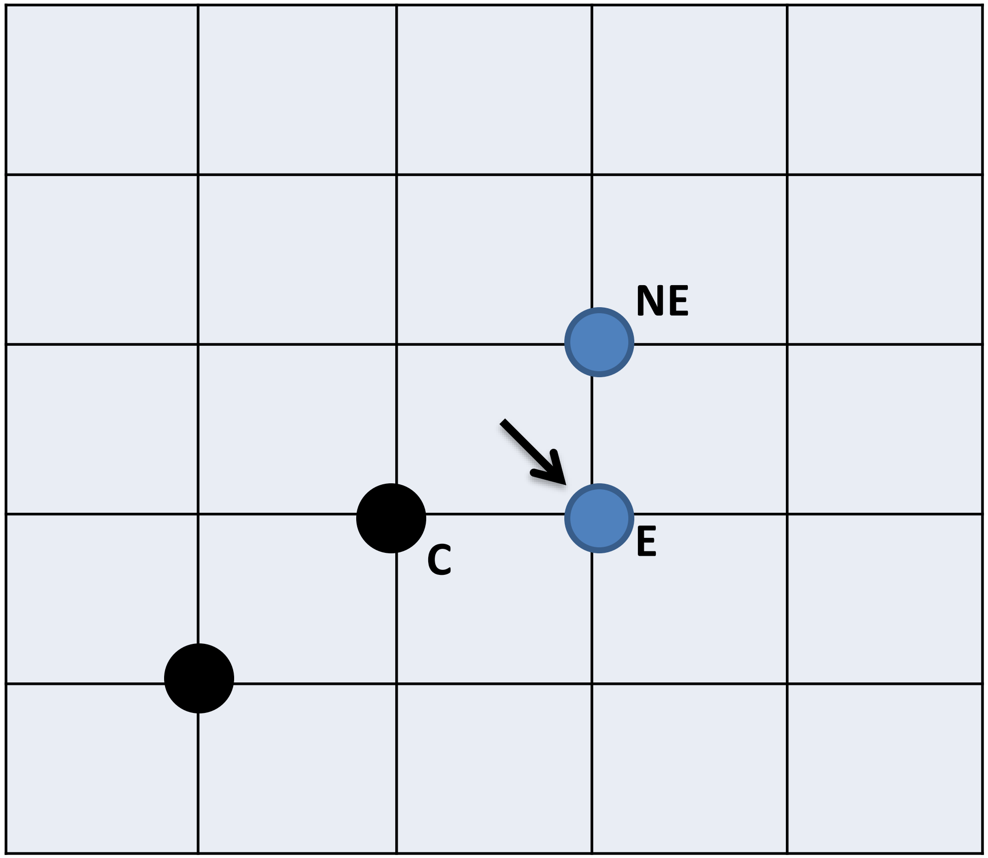
Next pixel is chosen (from E or NE) to build the line successively

Bresenham's Mid Point Criteria: How it works



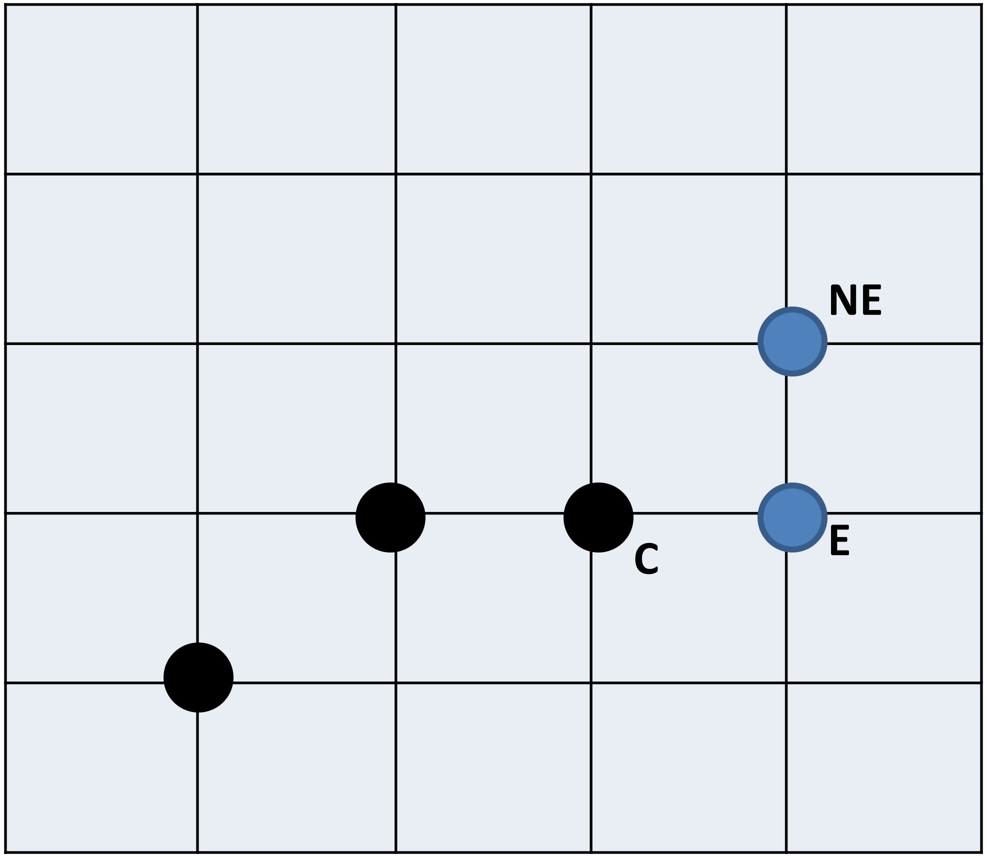
Next pixel is chosen (from E or NE) to build the line successively

Bresenham's Mid Point Criteria: How it works



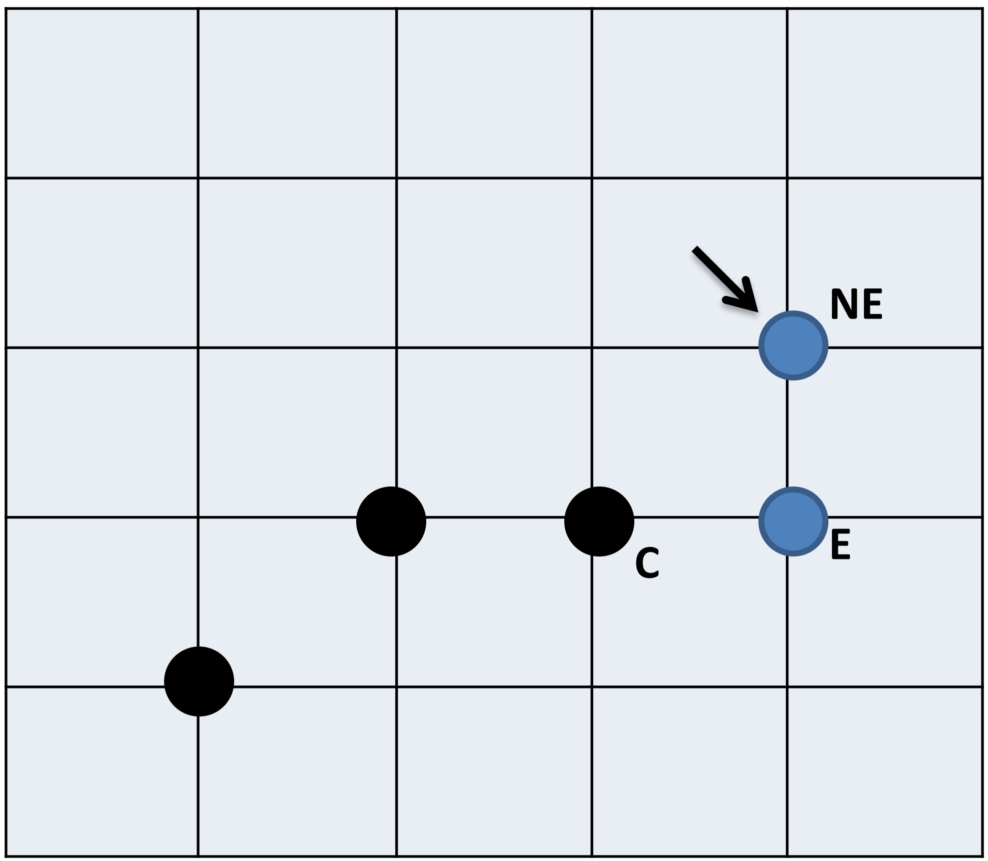
Next pixel is chosen (from E or NE) to build the line successively

Bresenham's Mid Point Criteria: How it works



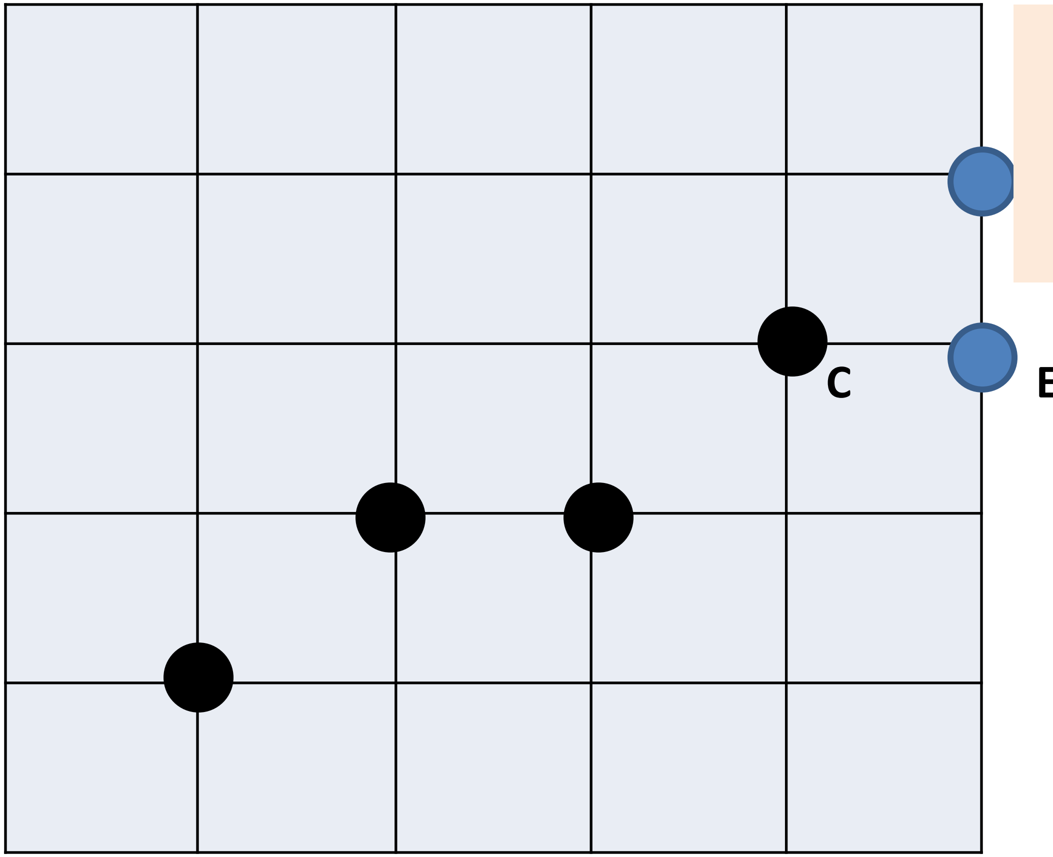
Next pixel is chosen (from E or NE) to build the line successively

Bresenham's Mid Point Criteria: How it works



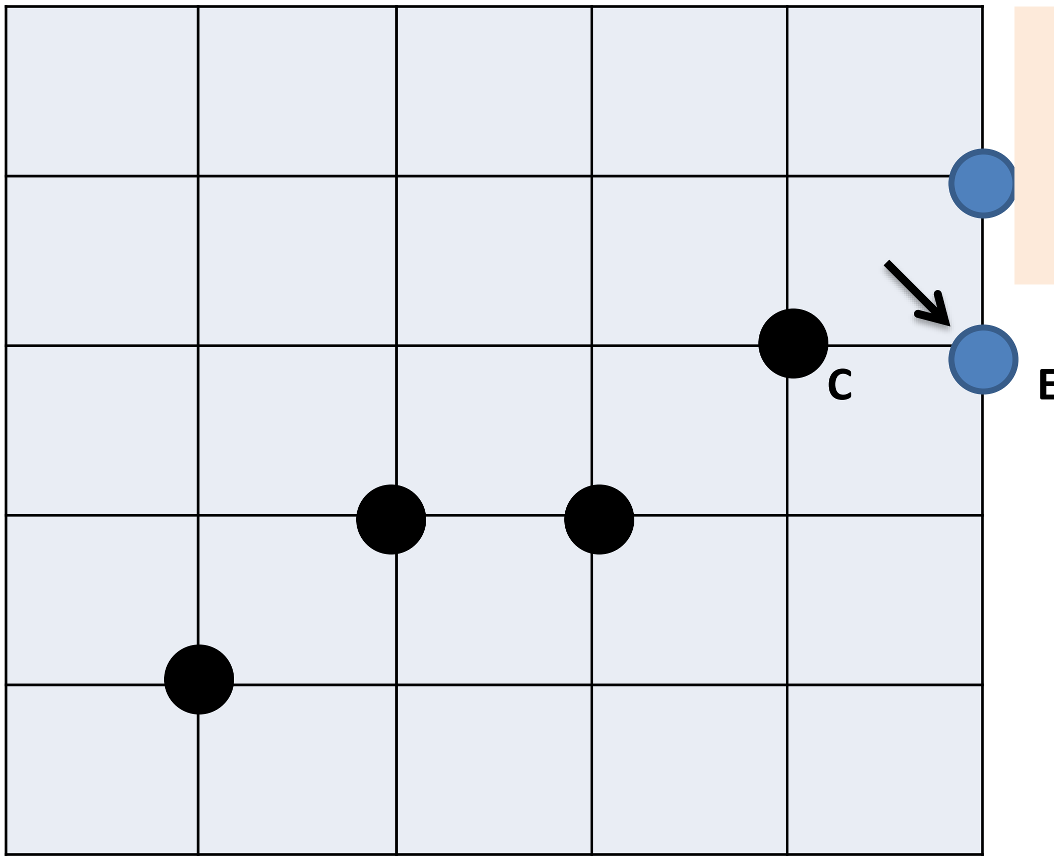
Next pixel is chosen (from E or NE) to build the line successively

Bresenham's Mid Point Criteria: How it works



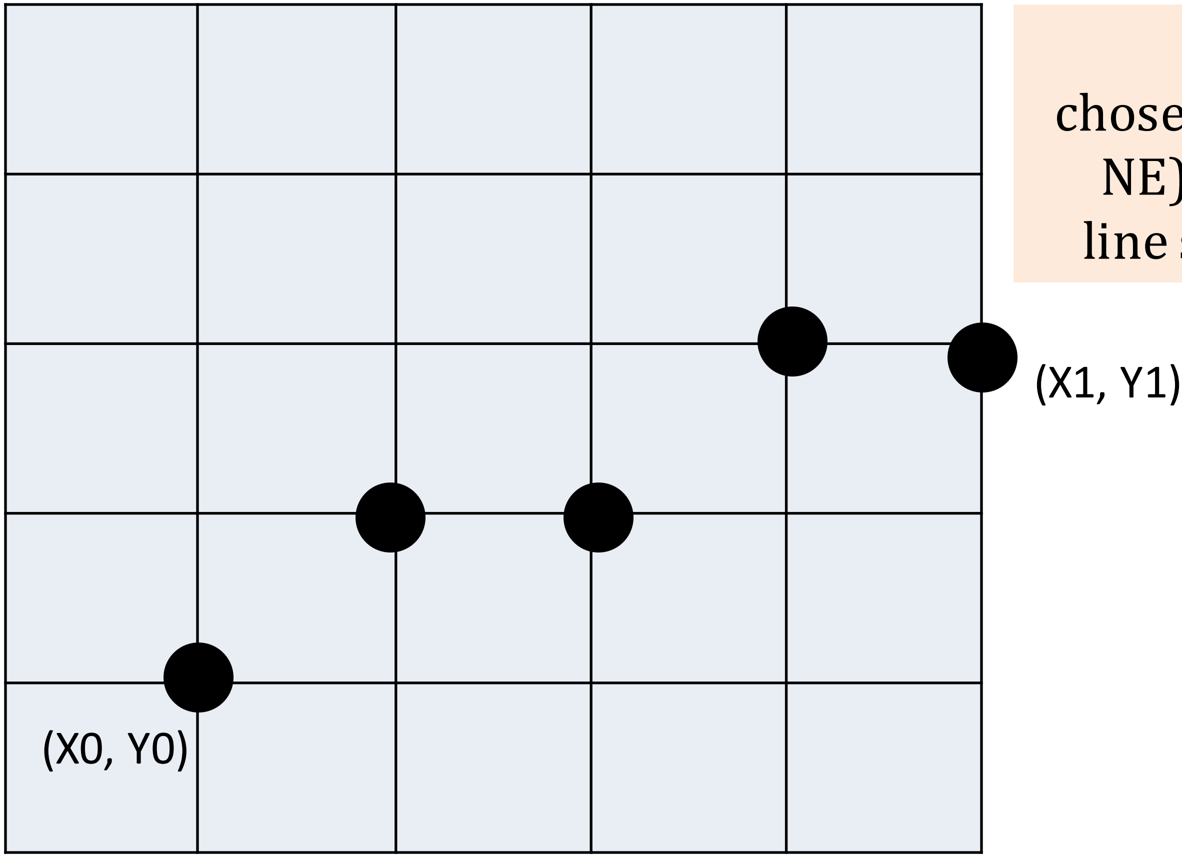
Next pixel is chosen (from E or NE) to build the line successively

Bresenham's Mid Point Criteria: How it works



Next pixel is chosen (from E or NE) to build the line successively

Bresenham's Mid Point Criteria: How it works



Next pixel is chosen (from E or NE) to build the line successively

A Line is defined by a function F (X,Y)

$$Y = mX + B$$

$$or, Y = \frac{dy}{dx} * X + B$$

$$or, Ydx = Xdy + Bdx$$

$$or, Xdy - Ydx + Bdx = 0$$

$$or, aX + bY + c = 0 \ [here, a = dy, b = -dx, c = Bdx]$$

$$F(X,Y) = aX + bY + c = 0$$

A Line is defined by a function F (X,Y)

$$Y = mX + B$$

$$or, Y = \frac{dy}{dx} * X + B$$

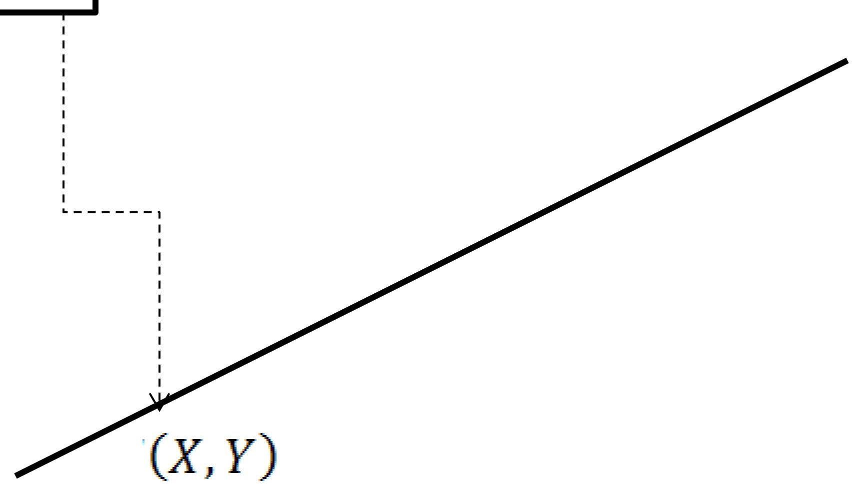
$$or, Ydx = Xdy + Bdx$$

$$or, Xdy - Ydx + Bdx = 0$$

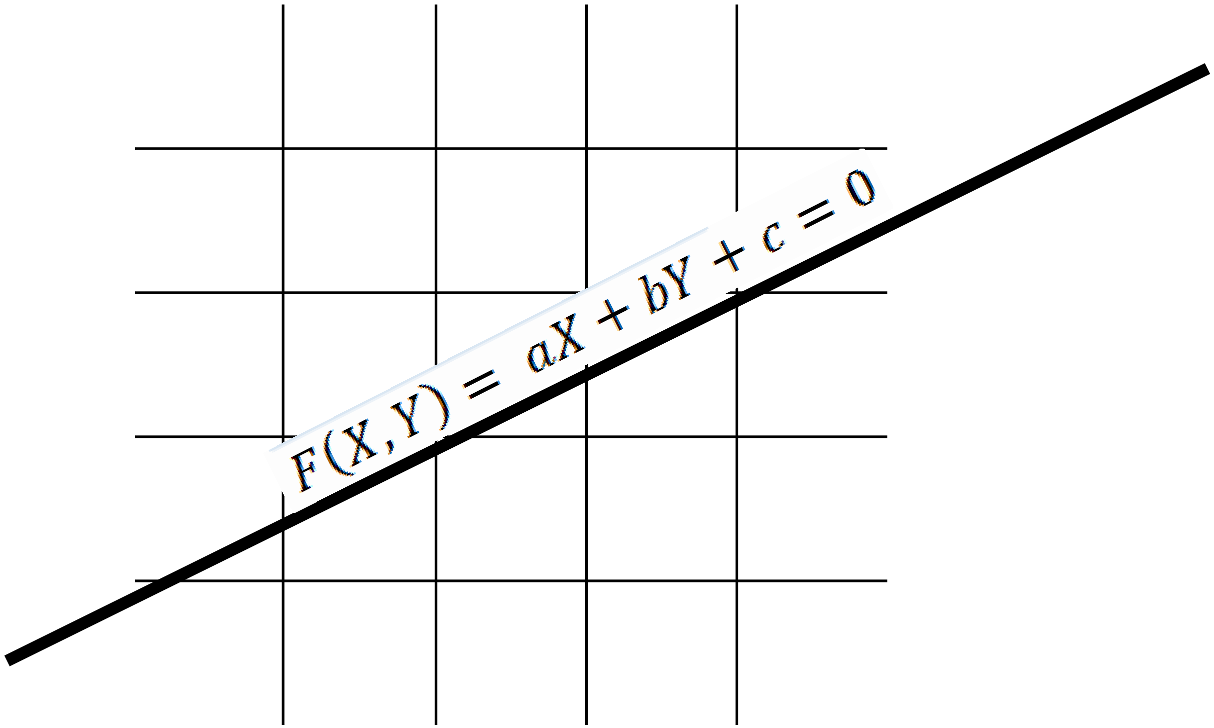
$$or, aX + bY + c = 0 \;\; [here, a = dy, b = -dx, c = Bdx]$$

$$\boxed{F(X,Y) = aX + bY + c = 0}$$

For every point (X,Y) on the line the function is Zero

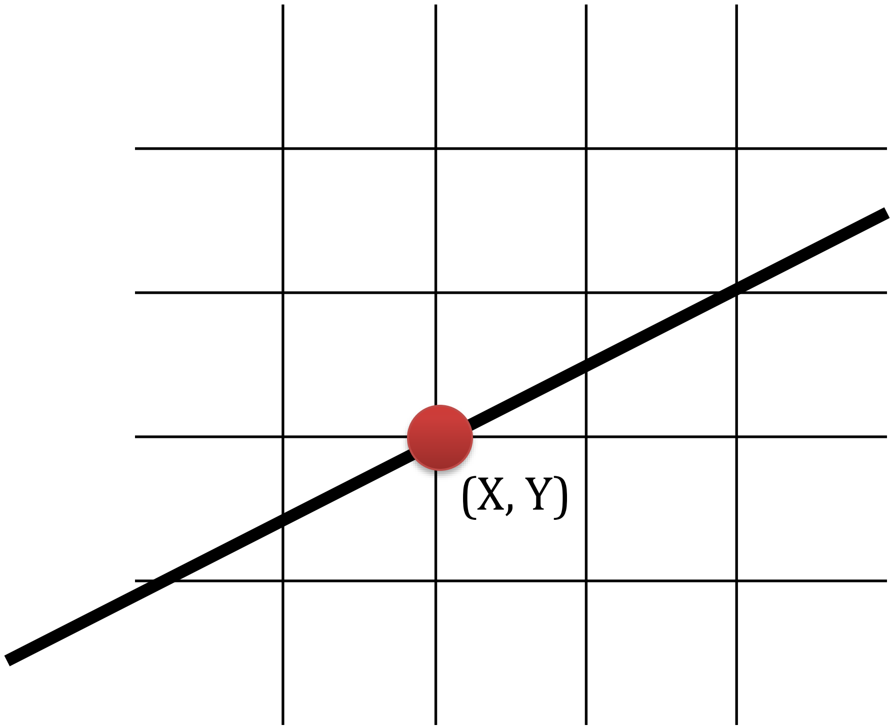


A Line is defined by a function F (X,Y)



A Line is defined by a function F (X,Y)

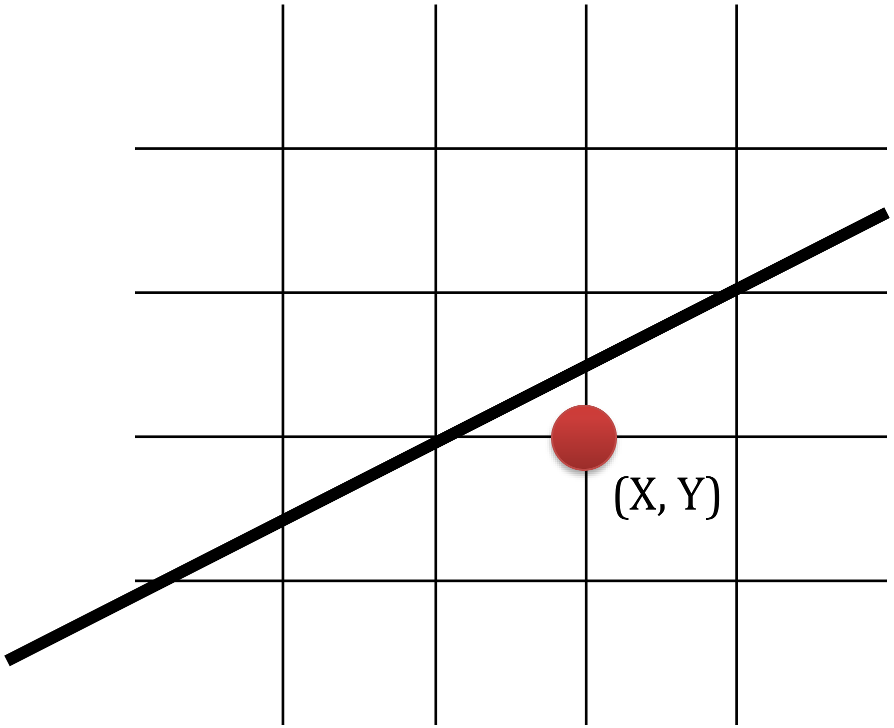
$$F(X,Y) = aX + bY + c = 0$$



If **F(X,Y) = 0,** the point (X,Y) on lying on the line

A Line is defined by a function F (X,Y)

$$F(X,Y) = aX + bY + c = 0$$

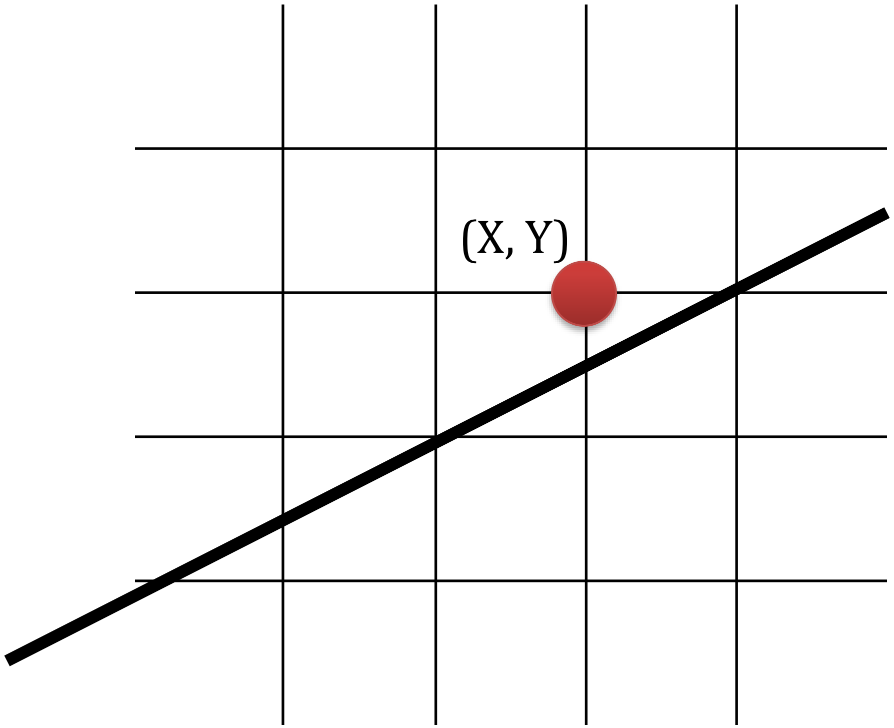


If **F(X,Y) = 0,** the point (X,Y) on lying on the line

If **F (X, Y) > 0,** the point (X,Y) is below the line

A Line is defined by a function F (X,Y)

$$F(X,Y) = aX + bY + c = 0$$

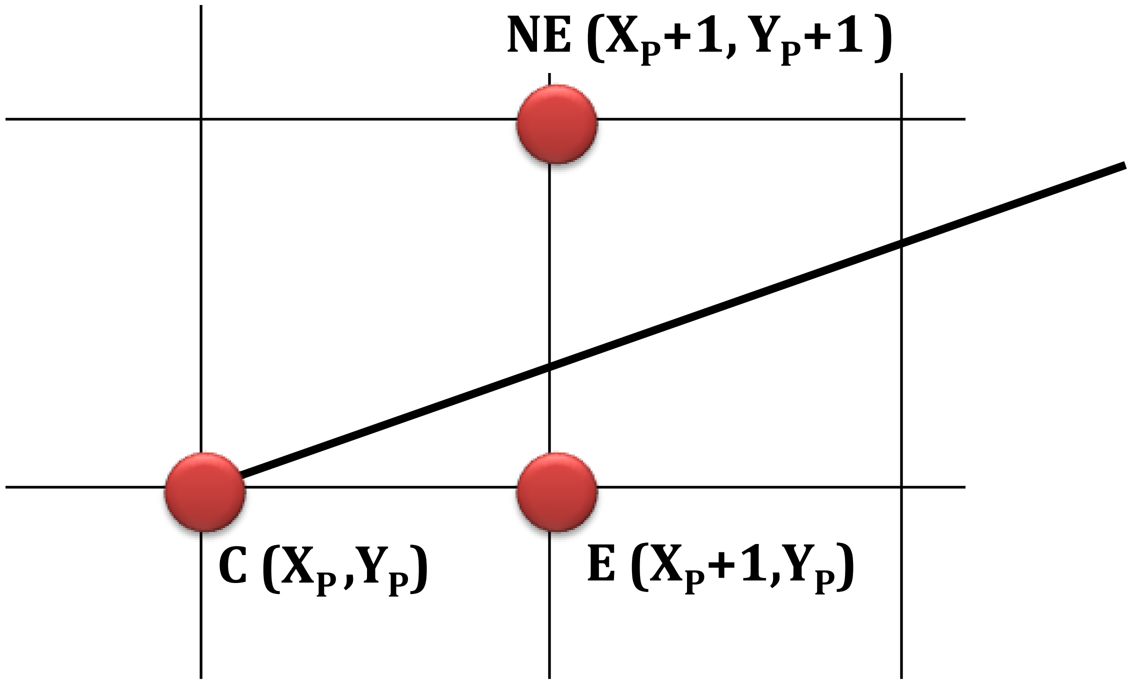


If **F(X,Y) = 0,** the point (X,Y) on lying on the line

If **F (X, Y) > 0,** the point (X,Y) is below the line

If **F (X, Y) < 0,** the point (X,Y) is above the line

A Line is defined by a function F (X,Y)

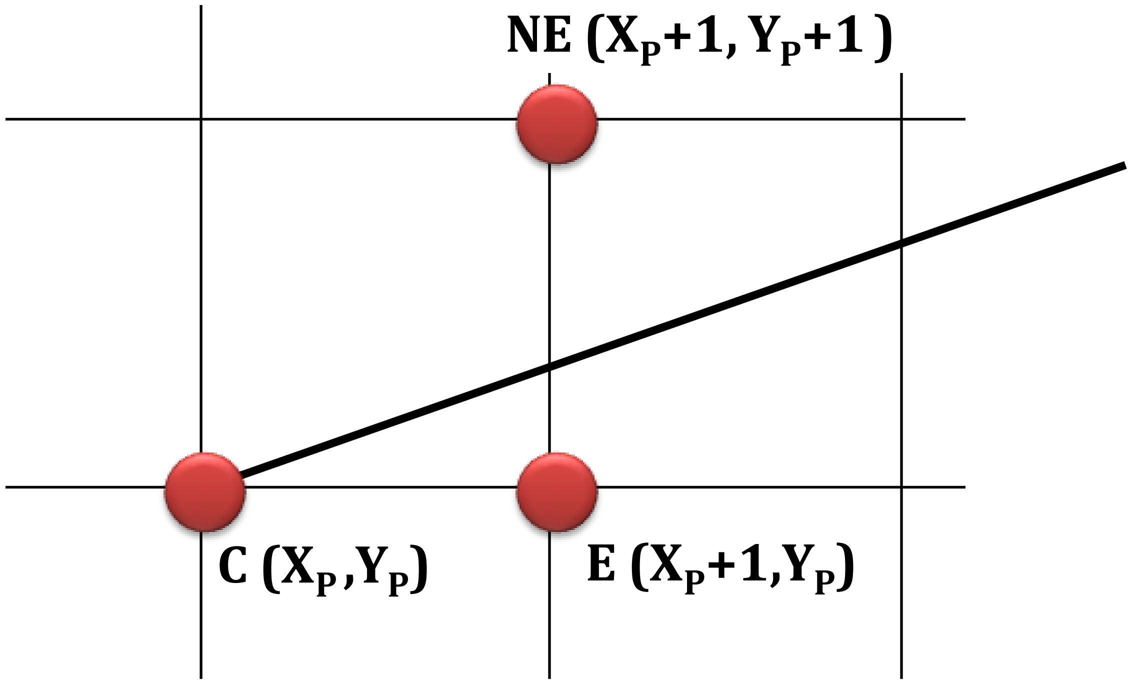


Bresenham's Mid Point Criteria: The Logic

Selecting E or NE depends on closeness to the line.

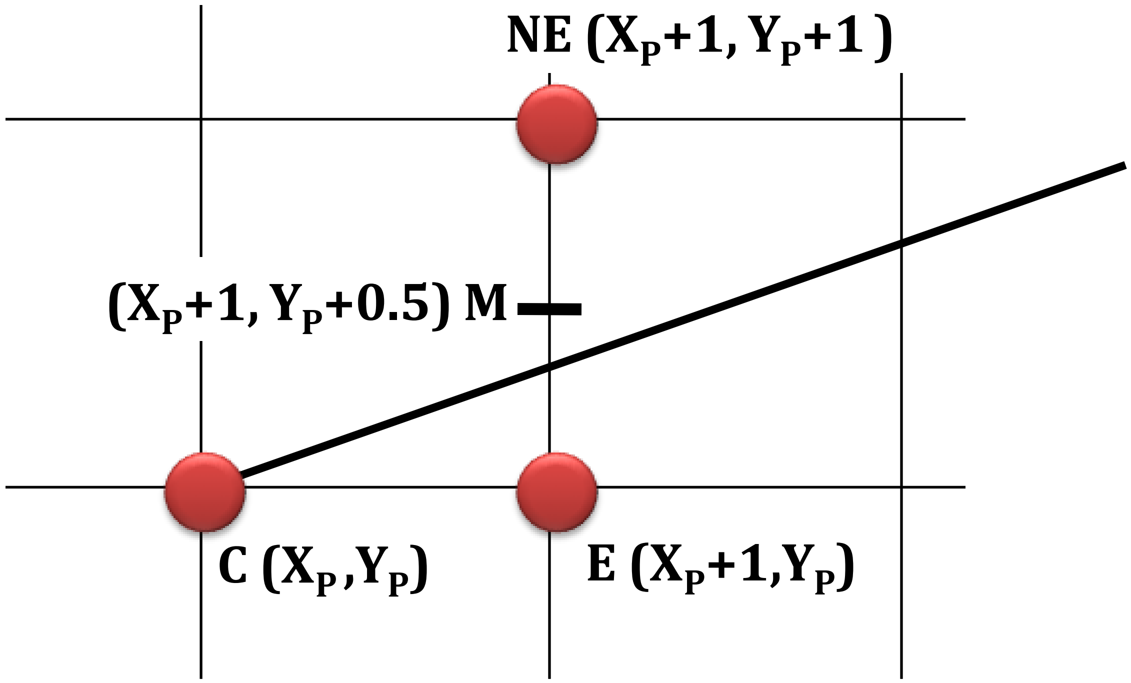
If E is closer to line, then E is selected

If NE is closer, then NE is selected

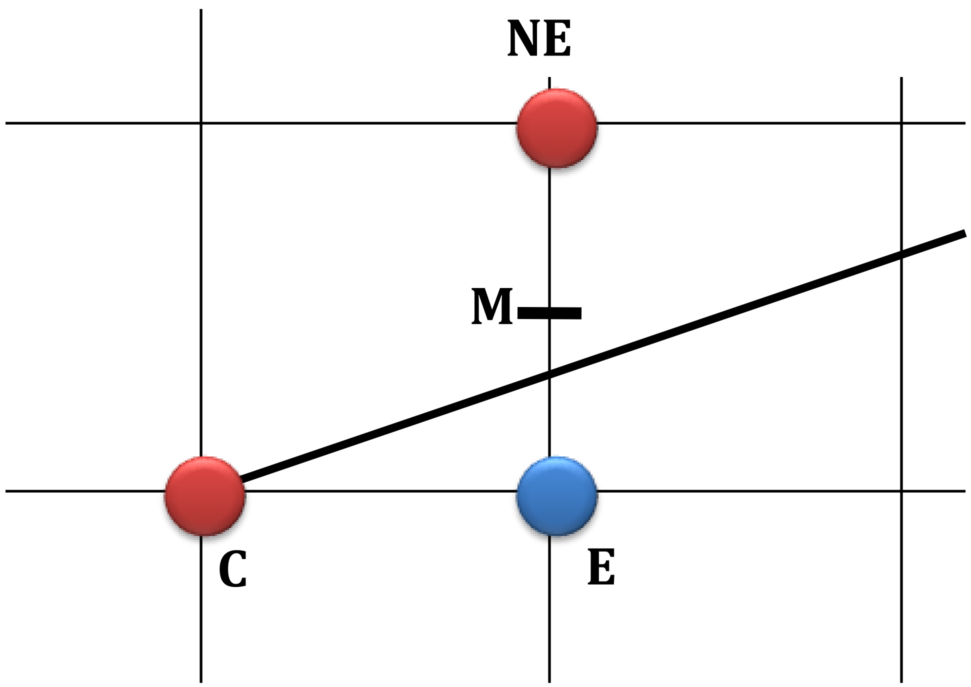


Bresenham's Mid Point Criteria: The Logic

To determine the nearness, mid point between E and NE is used

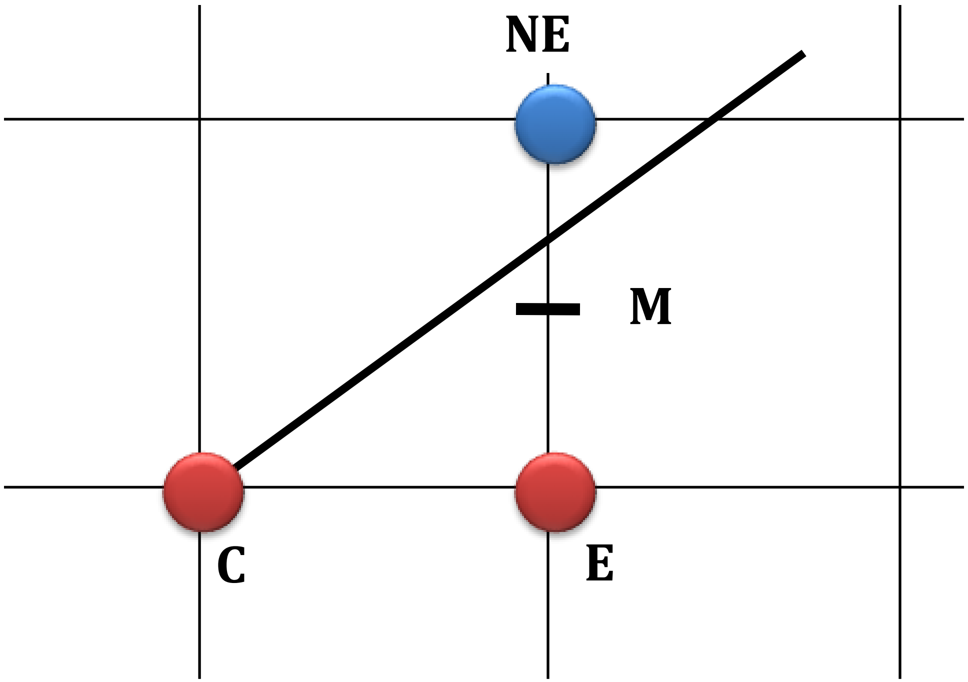


Bresenham's Mid Point Criteria: The Logic



If M is above the line, then E is closer to the line
→**E** is selected

Bresenham's Mid Point Criteria: The Logic

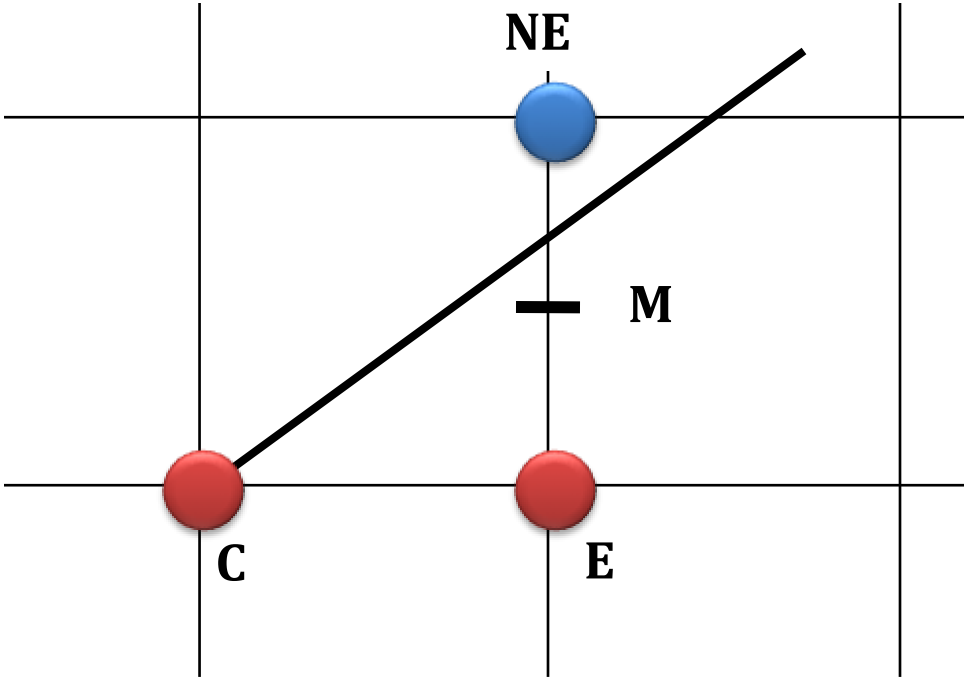


If M is above the line, then E is closer to the line
→**E** is selected

If M is below the line, then NE is closer to the line
→ **NE** is selected

Bresenham’s Mid Point Criteria: The Logic

Now, we have to evaluate whether the mid point is below or above the line



If M is above the line, then E is closer to the line
→ **E** is selected

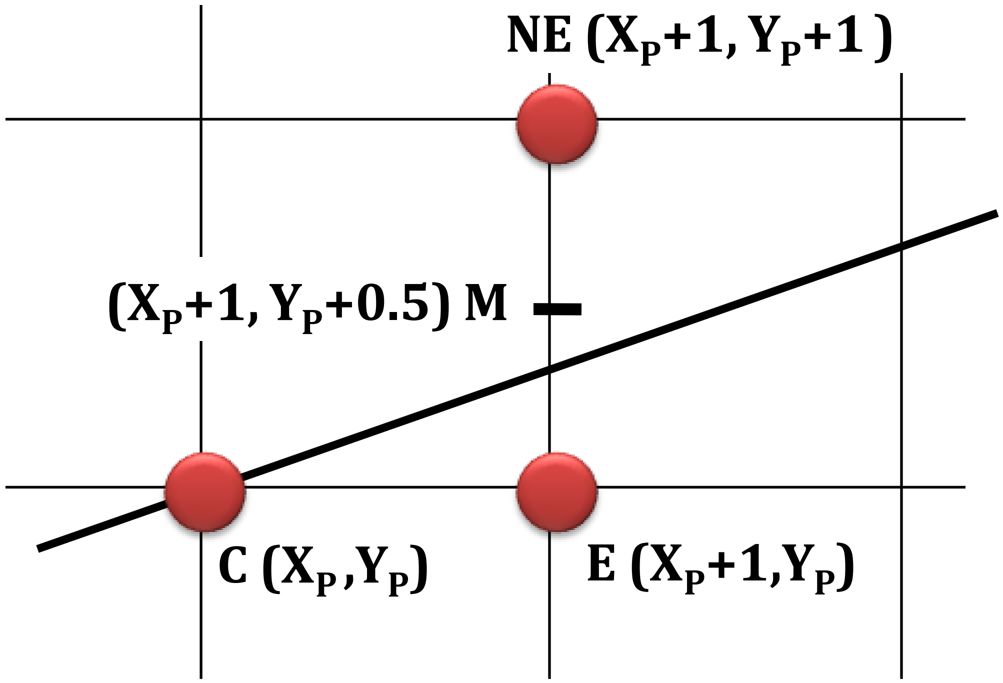
If M is below the line, then NE is closer to the line
→ **NE** is selected

Bresenham's Mid Point Criteria

We know, $F(X,Y) = aX + bY + c$

Lets put the mid point **M**'s coordinate in function F(X,Y)

$F(M) = F(\mathbf{X_P+1, Y_P+0.5}) = a(X_P+1) + b(Y_P+0.5) + c$

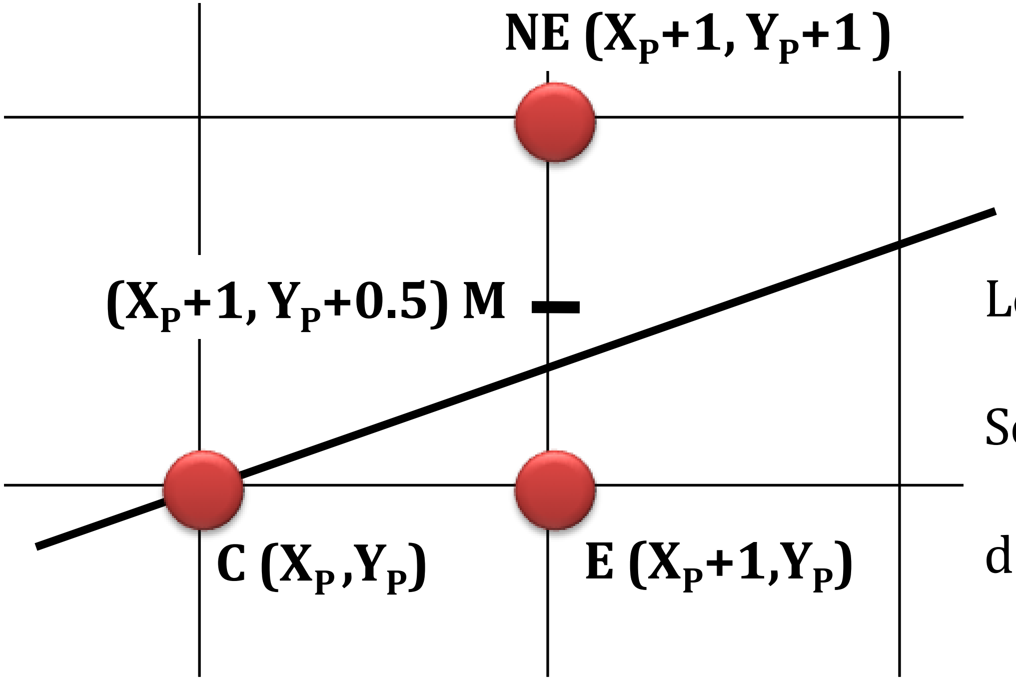


Bresenham's Mid Point Criteria

We know, $F(X,Y) = aX + bY + c$

Lets put the mid point **M**'s coordinate in function F(X,Y)

$F(M) = F(\mathbf{X_P+1, Y_P+0.5}) = a(X_P+1) + b(Y_P+0.5) + c$



Lets store **F(M)** in a variable **d**

So, **d = F(M)**

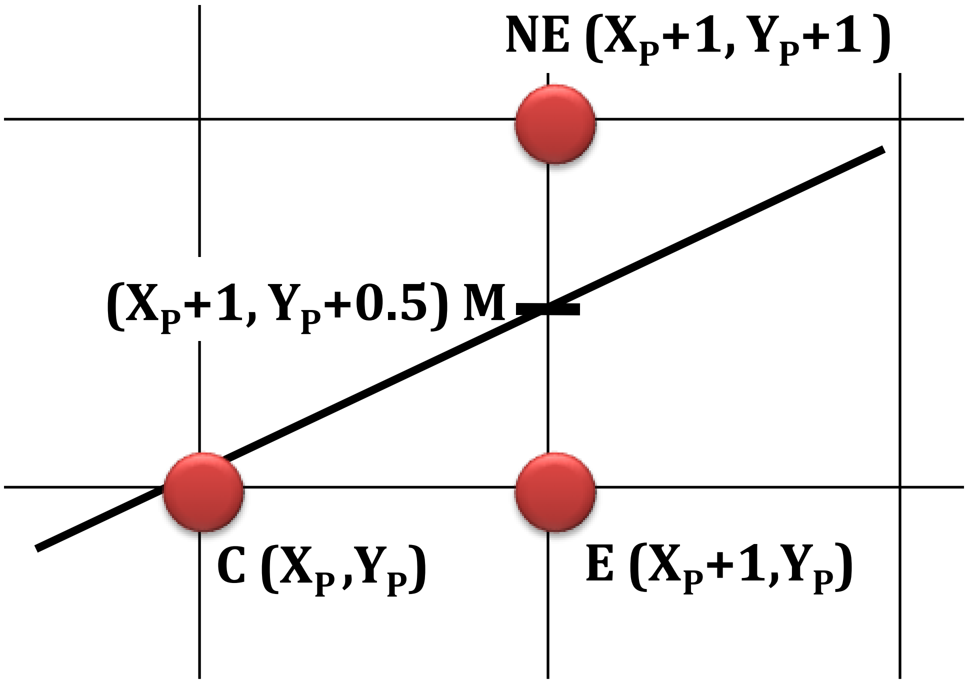
d is called 'decision variable'

Bresenham's Mid Point Criteria

So, d = F(M)

$$= F(X_P+1, Y_P+0.5)$$

$$= a(X_P+1) + b(Y_P+0.5) + c$$



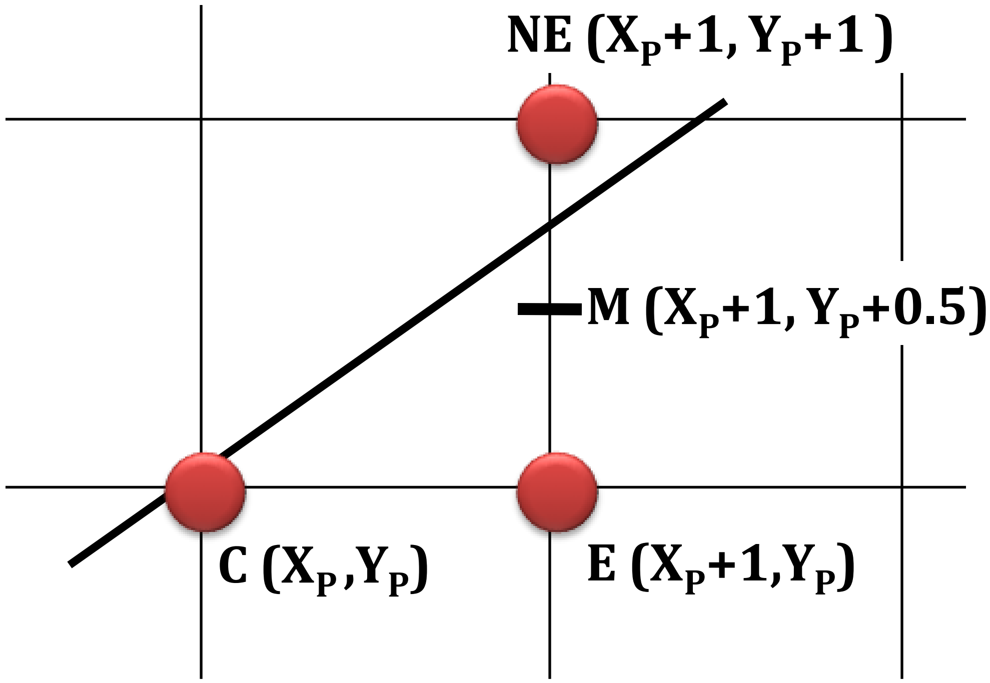
if **d = 0**, then midpoint is on the line

Bresenham's Mid Point Criteria

So, d = F(M)

$$= F(X_P+1, Y_P+0.5)$$

$$= a(X_P+1) + b(Y_P+0.5) + c$$

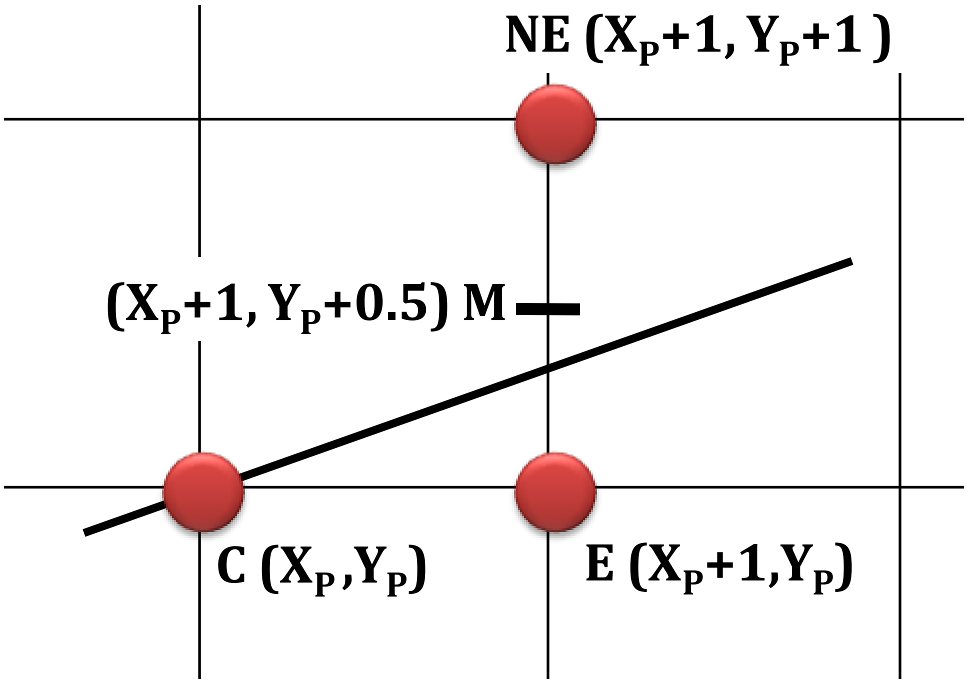


if **d = 0**, then midpoint M is on the line

If **d > 0**, then midpoint M is below the line

Bresenham's Mid Point Criteria

$$\begin{aligned}\text{So, } d &= F(M)\\ &= F(X_P+1, Y_P+0.5)\\ &= a(X_P+1) + b(Y_P+0.5) + c\end{aligned}$$



if **d = 0**, then midpoint M is on the line

If **d > 0**, then midpoint M is below the line

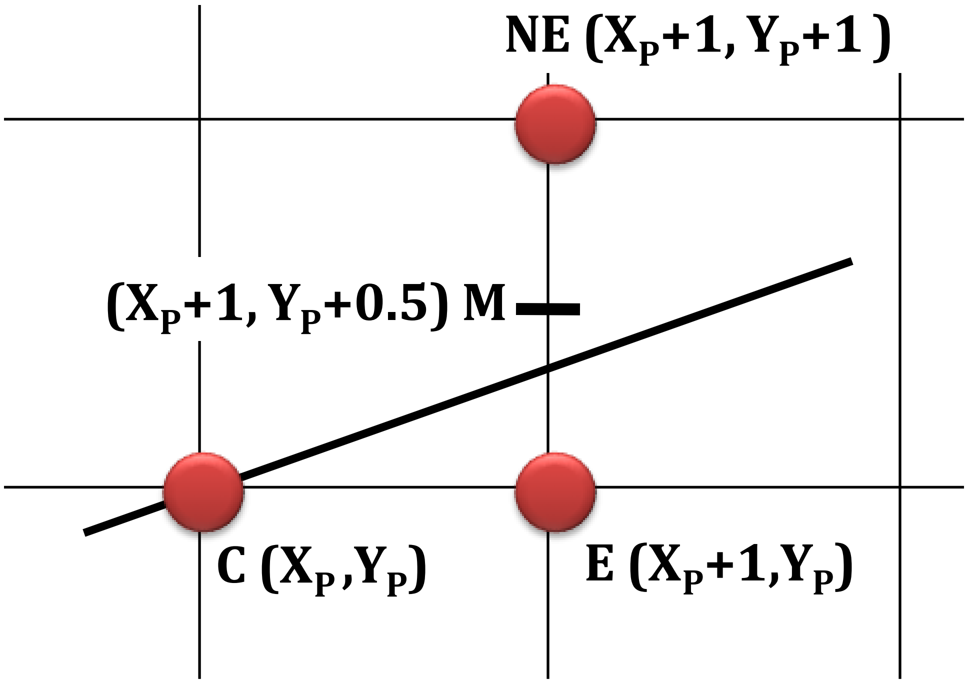
If **d < 0**, then midpoint M is above the line

Bresenham's Mid Point Criteria

So, $d = F(M)$

$= F(X_P+1, Y_P+0.5)$

$= a(X_P+1) + b(Y_P+0.5) + c$

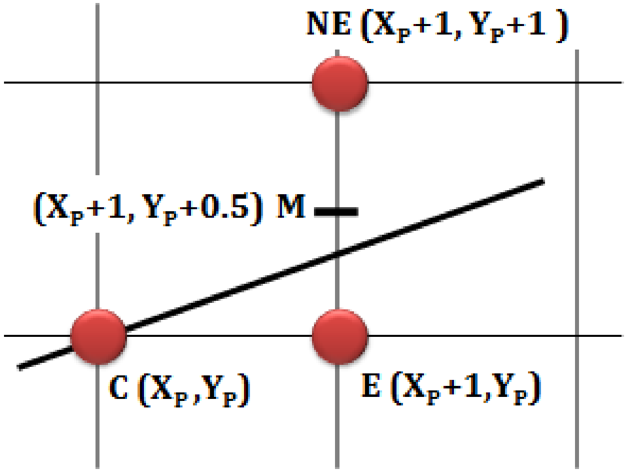


As we must select E or NE:

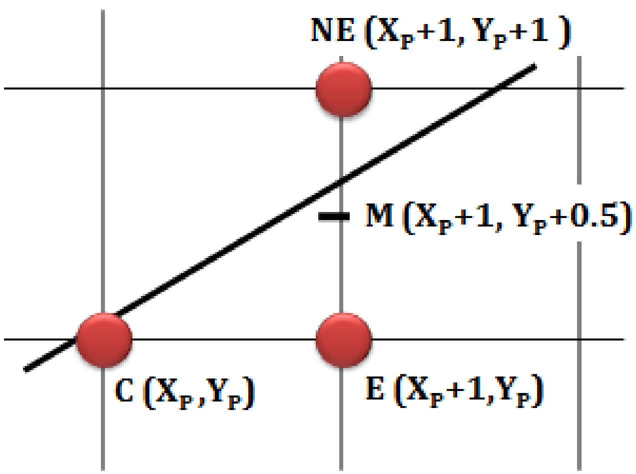
If **$d > 0$**, then midpoint M is below the line

If **$d \leq 0$**, then midpoint M is above the line

Bresenham's Mid Point Criteria : Summary

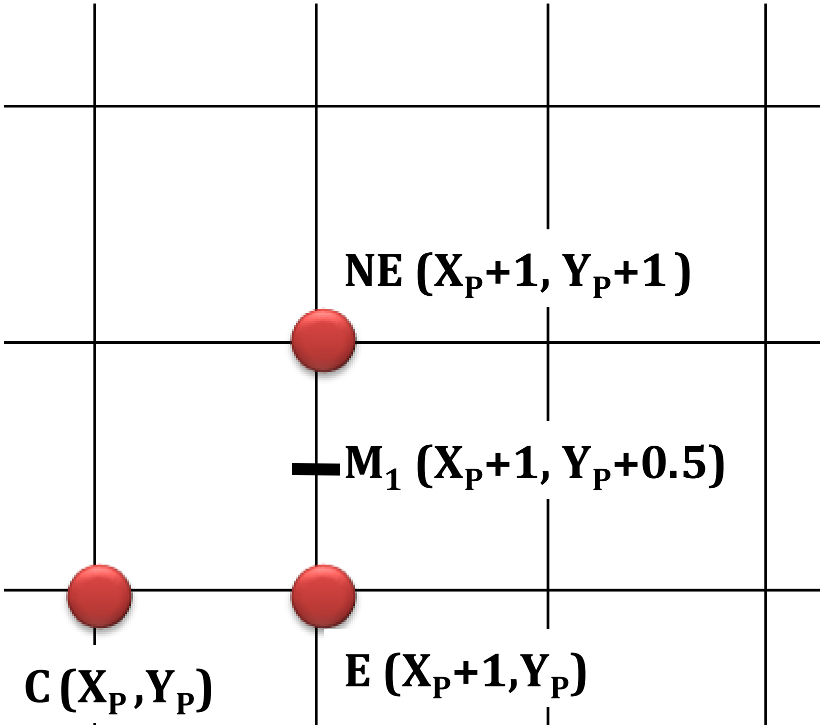


If **d ≤ 0**, then midpoint M is above the line, and E is closer to line,
So, **E** is selected



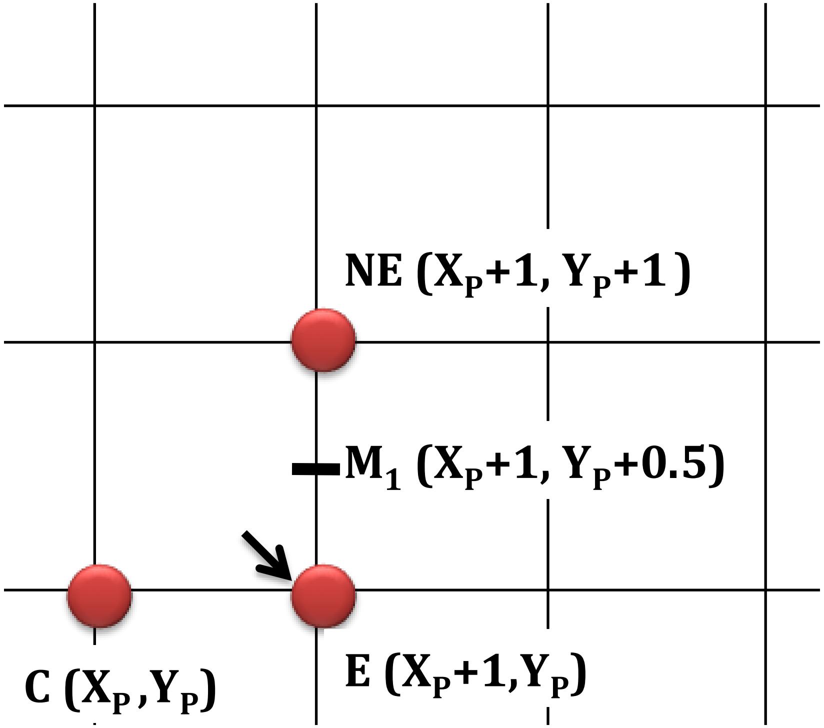
If **d > 0**, then midpoint M is below the line, and NE is closer to line,
So, **NE** is selected

Bresenham's Mid Point Criteria : Successive Updating (for selecting E)



$$
\begin{aligned}
d_1 &= F(M_1) \\
&= F(X_P+1, Y_P+0.5) \\
&= a(X_P+1) + b(Y_P+0.5) + c
\end{aligned}
$$

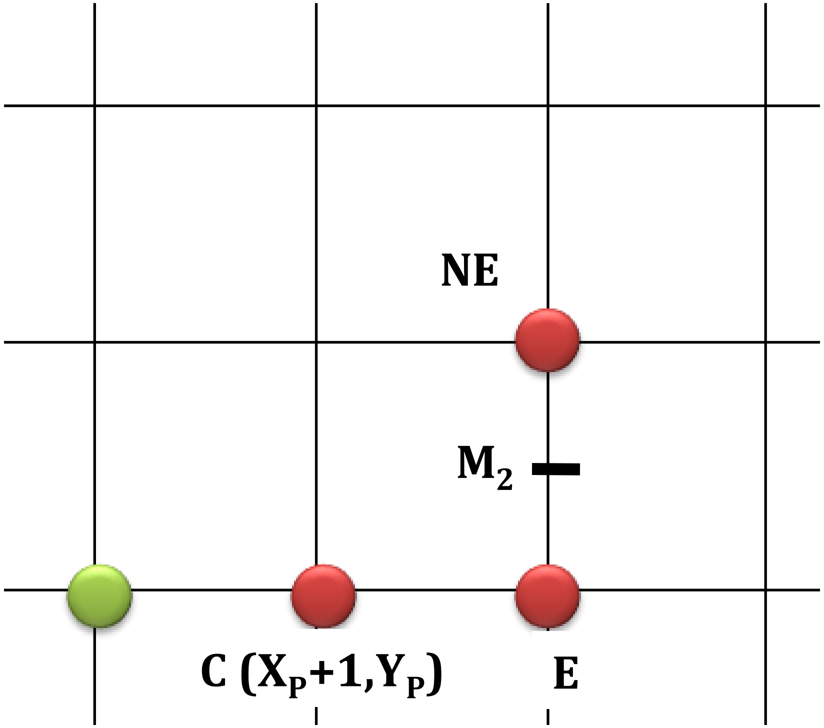
Bresenham's Mid Point Criteria : Successive Updating (for selecting E)



$$d_1 = F(M_1)$$
$$= F(X_P+1, Y_P+0.5)$$
$$= a(X_P+1) + b(Y_P+0.5) + c$$

IF $d_1 \leq 0$, *select E* $(X_P = X_P+1, Y_P)$

Bresenham's Mid Point Criteria : Successive Updating (for selecting E)



$d_1 = F(M_1)$

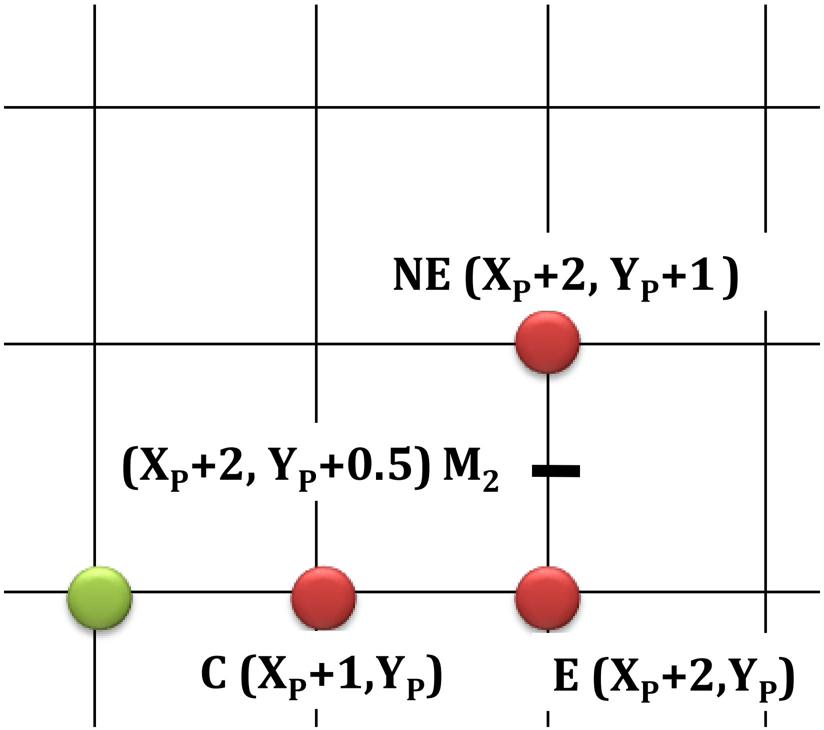
$= F(X_P+1, Y_P+0.5)$

$= a(X_P+1) + b(Y_P+0.5) + c$

IF $d_1 \leq 0$, *select E* $(X_P = X_P+1, Y_P)$

$d_2 = F(M_2)$

Bresenham's Mid Point Criteria : Successive Updating (for selecting E)



$d_1 = F(M_1)$

$= F(X_P+1, Y_P+0.5)$

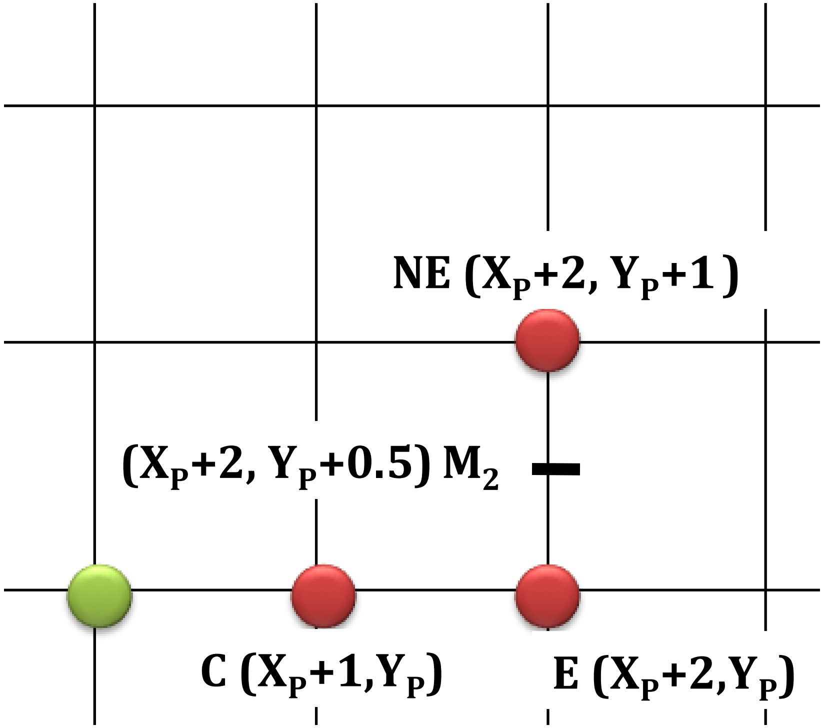
$= a(X_P+1) + b(Y_P+0.5) + c$

IF $d_1 \leq 0$, *select E* $(X_P = X_P+1, Y_P)$

$d_2 = F(M_2)$

$= F(X_P+2, Y_P+0.5)$

Bresenham's Mid Point Criteria : Successive Updating (for selecting E)

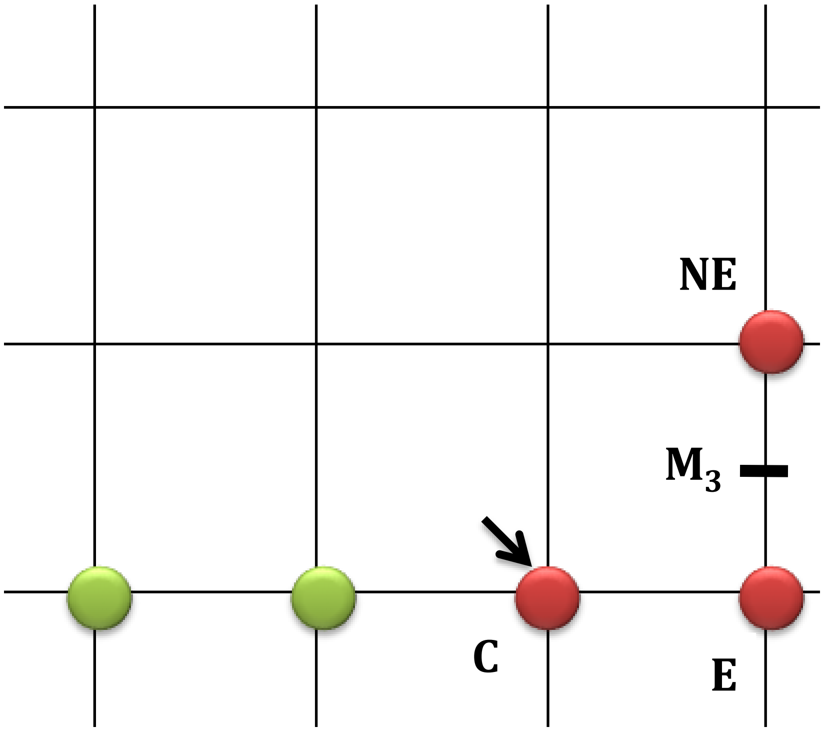


$$
\begin{aligned}
d_1 &= F(M_1) \\
&= F(X_P+1, Y_P+0.5) \\
&= a(X_P+1) + b(Y_P+0.5) + c
\end{aligned}
$$

IF $d_1 \le 0$, *select E* ($X_P = X_P+1, Y_P$)

$$
\begin{aligned}
d_2 &= F(M_2) \\
&= F(X_P+2, Y_P+0.5) \\
&= a(X_P+2) + b(Y_P+0.5) + c \\
&= aX_P + 2a + bY_P + 0.5b + c \\
&= aX_P + a + bY_P + 0.5b + c + a \\
&= [\, a(X_P+1) + b(Y_P+0.5) + c \,] + a \\
&= d_1 + a
\end{aligned}
$$

Bresenham's Mid Point Criteria : Successive Updating (for selecting E)



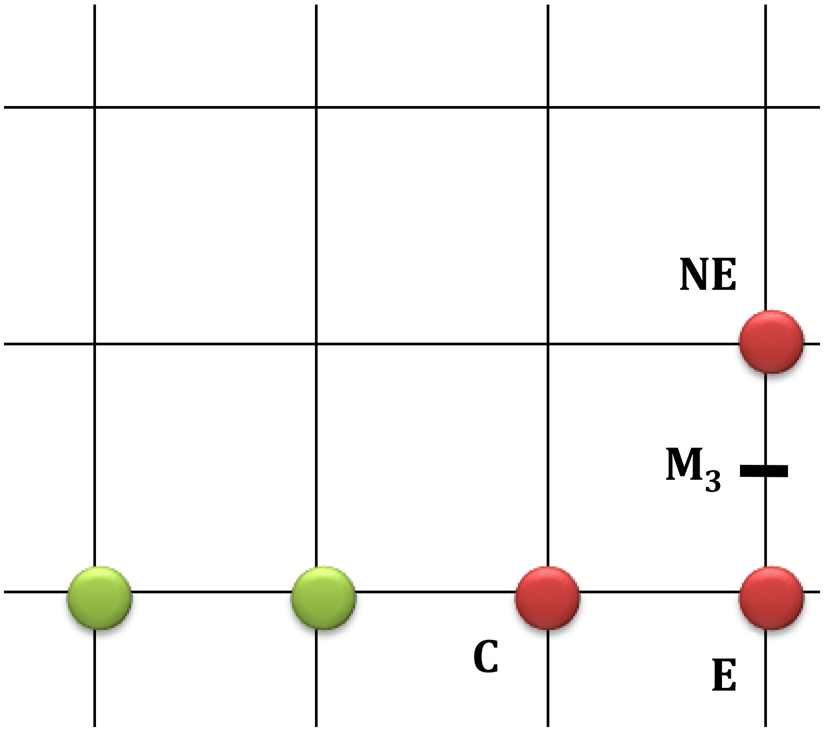
$$
\begin{aligned}
d_1 &= F(M_1) \\
&= F(X_P+1, Y_P+0.5) \\
&= a(X_P+1) + b(Y_P+0.5) + c
\end{aligned}
$$

IF $d_1 \leq 0$, *select E* $(X_P = X_P+1, Y_P)$

$$
\begin{aligned}
d_2 &= F(M_2) \\
&= F(X_P+2, Y_P+0.5) \\
&= a(X_P+2) + b(Y_P+0.5) + c \\
&= aX_P + 2a + bY_P + 0.5b + c \\
&= aX_P + a + bY_P + 0.5b + c + a \\
&= [\ a(X_P+1) + b(Y_P+0.5) + c\ ] + a \\
&= d_1 + a
\end{aligned}
$$

IF $d_2 \leq 0$, *select E* $(X_P = X_P+1, Y_P)$

Bresenham's Mid Point Criteria : Successive Updating (for selecting E)



$$d_1 = F(M_1)$$
$$= F(X_P+1, Y_P+0.5)$$
$$= a(X_P+1) + b(Y_P+0.5) + c$$

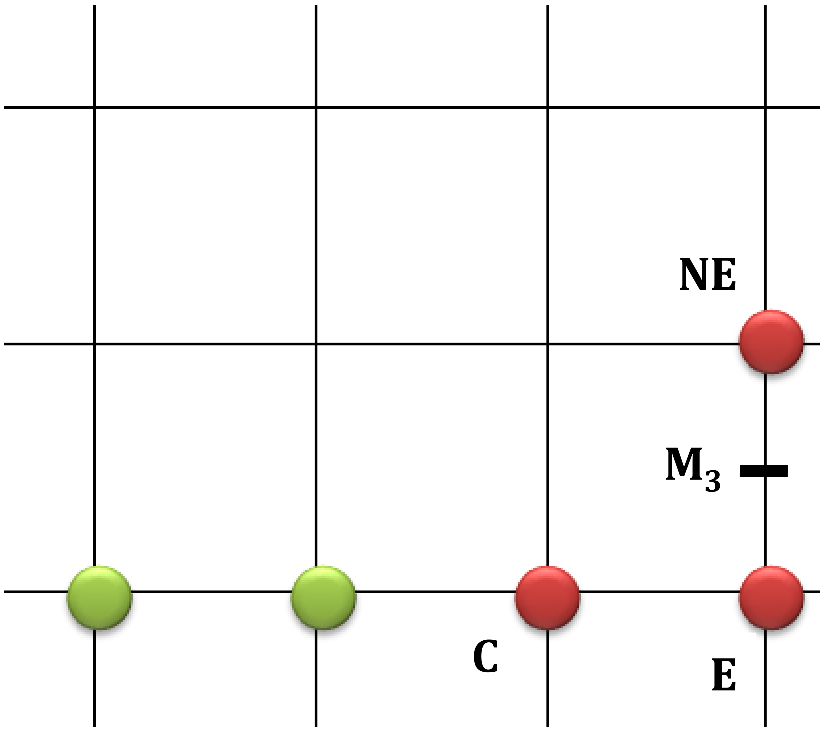
IF $d_1 \leq 0$ [*select E*]

$$d_2 = F(M_2)$$
$$= F(X_P+2, Y_P+0.5)$$
$$= a(X_P+2) + b(Y_P+0.5) + c$$
$$= aX_P + 2a + bY_P + 0.5b + c$$
$$= aX_P + a + bY_P + 0.5b + c + a$$
$$= [\ a(X_P+1) + b(Y_P+0.5) + c\ ] + a$$
$$= d_1 + a$$

IF $d_2 \leq 0$, *select E* $(X_P = X_P+1, Y_P)$

Similarly, $d_3 = F(M_3) = d_2 + a$

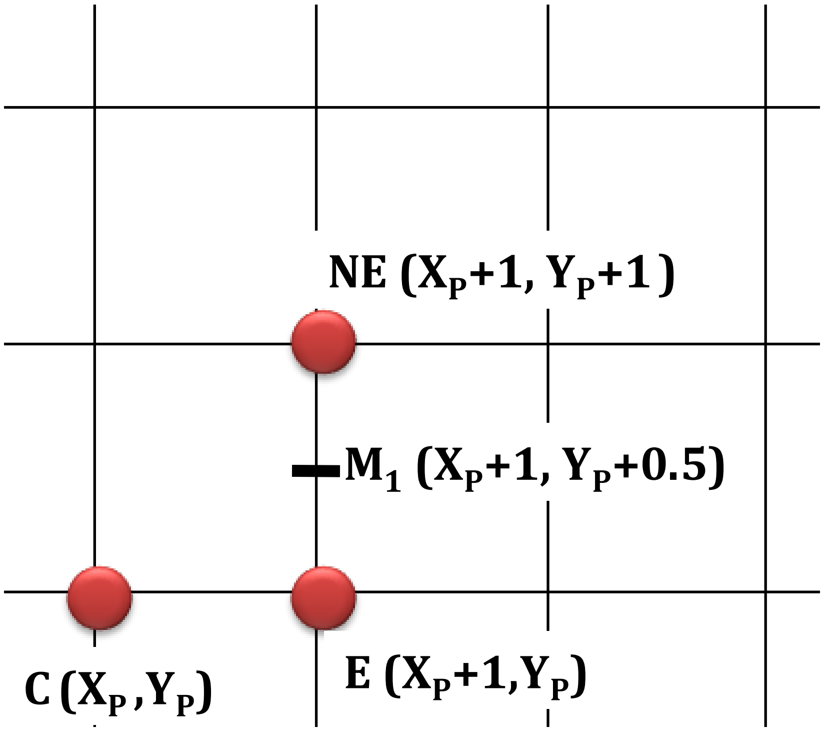
Bresenham's Mid Point Criteria : Successive Updating (for selecting E)



Every iteration after **selecting E**, we can successively update our decision variable with-

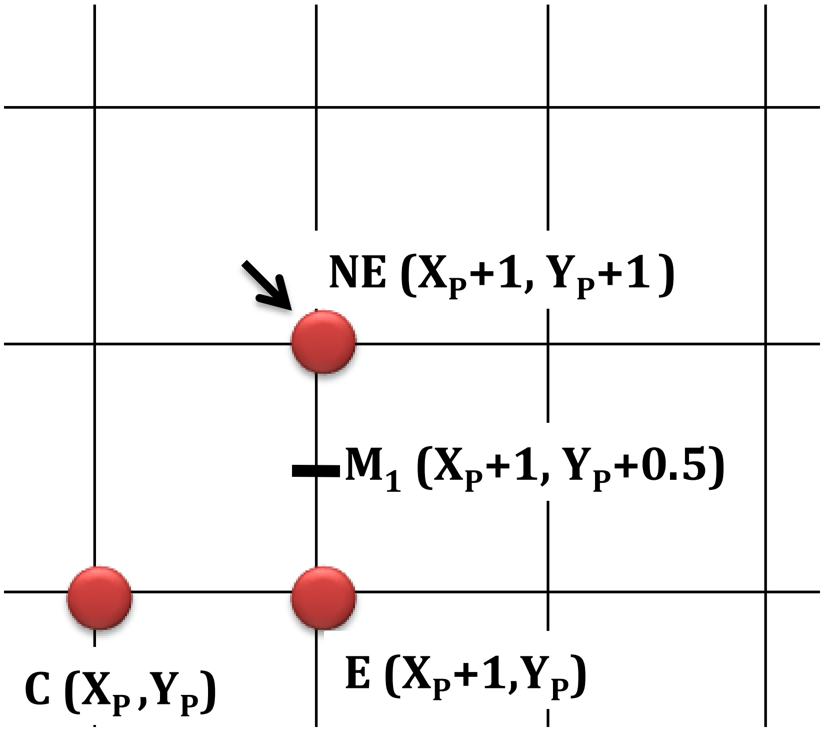
$$\mathbf{d_{NEW} = d_{OLD} + a}$$
$$\mathbf{= d_{OLD} + dy}$$

Bresenham's Mid Point Criteria : Successive Updating (for selecting NE)



$$d_1 = F(M_1)$$
$$= F(X_P+1, Y_P+0.5)$$
$$= a(X_P+1) + b(Y_P+0.5) + c$$

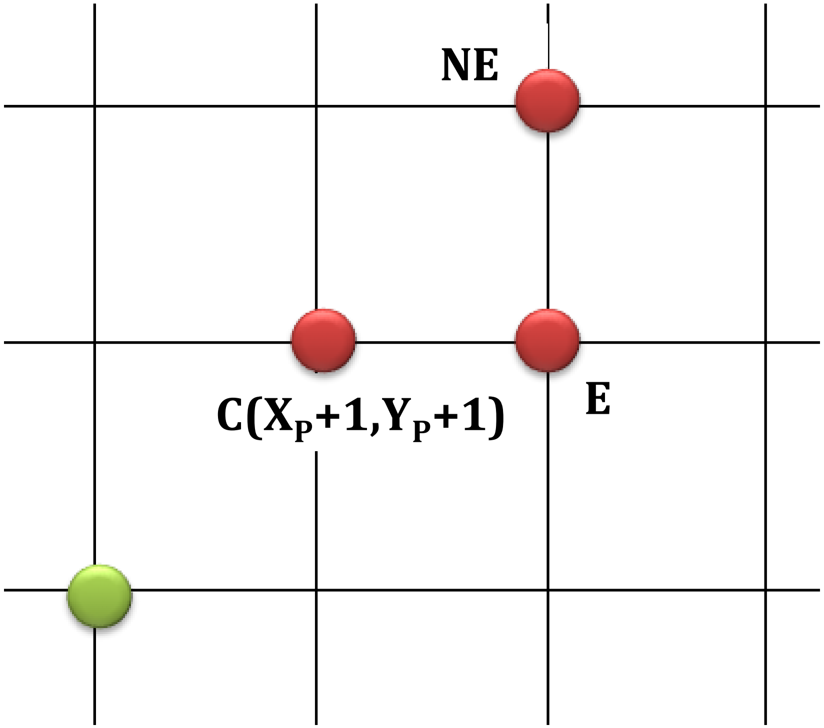
Bresenham's Mid Point Criteria : Successive Updating (for selecting NE)



$$d_1 = F(M_1)$$
$$= F(X_P+1, Y_P+0.5)$$
$$= a(X_P+1) + b(Y_P+0.5) + c$$

IF $d_1 > 0$, *select NE* ($X_P=X_P+1$, $Y_P=Y_P+1$)

Bresenham's Mid Point Criteria : Successive Updating (for selecting NE)



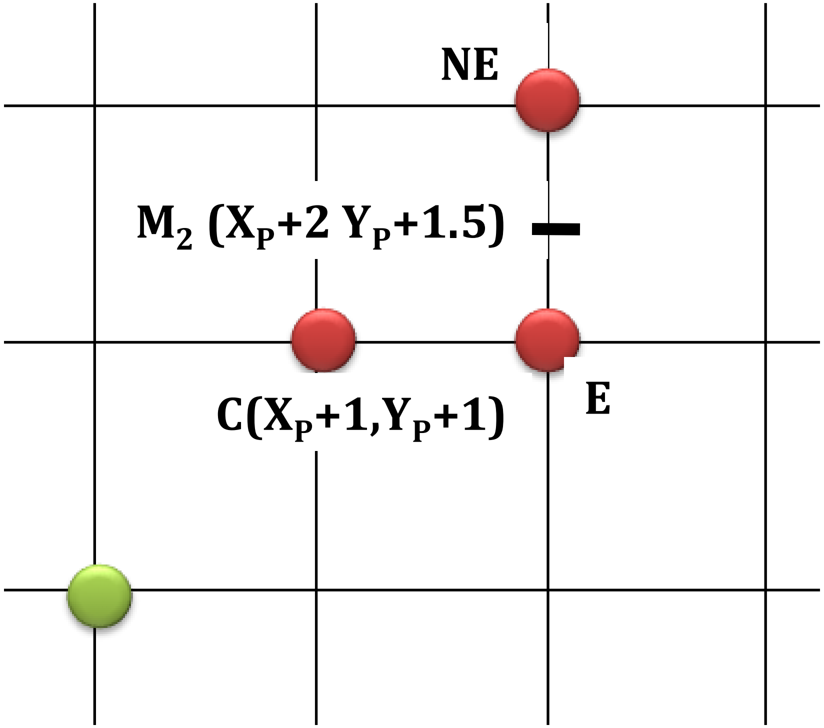
$d_1 = F(M_1)$

$= F(X_P+1, Y_P+0.5)$

$= a(X_P+1) + b(Y_P+0.5) + c$

IF $d_1 > 0$, *select NE* $(X_P=X_P+1,\ Y_P=Y_P+1)$

Bresenham's Mid Point Criteria : Successive Updating (for selecting NE)

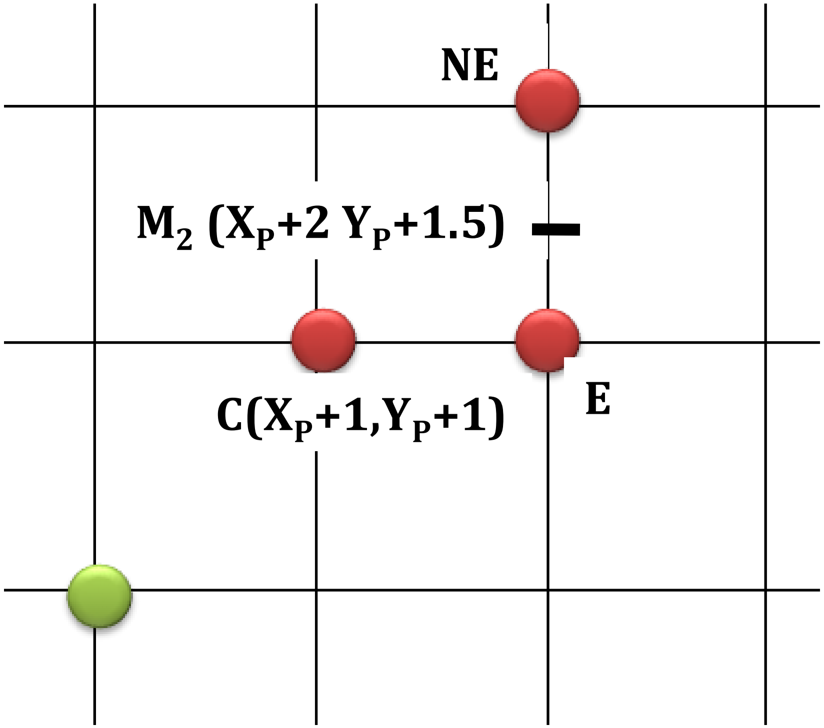


$$
\begin{aligned}
d_1 &= F(M_1) \\
&= F(X_P+1, Y_P+0.5) \\
&= a(X_P+1) + b(Y_P+0.5) + c
\end{aligned}
$$

IF $d_1 > 0$, *select NE* (X_P=X_P+1, Y_P=Y_P+1)

$$
\begin{aligned}
d_2 &= F(M_2) \\
&= F(X_P+2, Y_P+1.5)
\end{aligned}
$$

Bresenham's Mid Point Criteria : Successive Updating (for selecting NE)

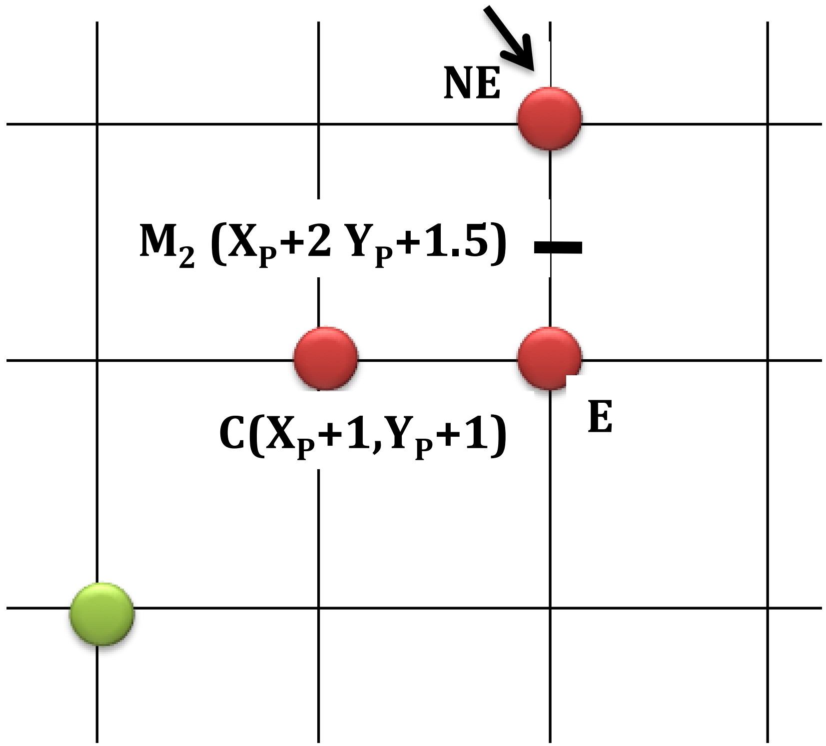


$$d_1 = F(M_1)$$
$$= F(X_P+1, Y_P+0.5)$$
$$= a(X_P+1) + b(Y_P+0.5) + c$$

IF $d_1 > 0$, *select NE* $(X_P = X_P+1, Y_P = Y_P+1)$

$$d_2 = F(M_2)$$
$$= F(X_P+2, Y_P+1.5)$$
$$= a(X_P+2) + b(Y_P+1.5) + c$$
$$= aX_P + 2a + bY_P + 1.5b + c$$
$$= aX_P + a + bY_P + 0.5b + c + a + b$$
$$= [\ a(X_P+1) + b(Y_P+0.5) + c\ ] + a + b$$
$$= d_1 + (a + b)$$

Bresenham's Mid Point Criteria : Successive Updating (for selecting NE)



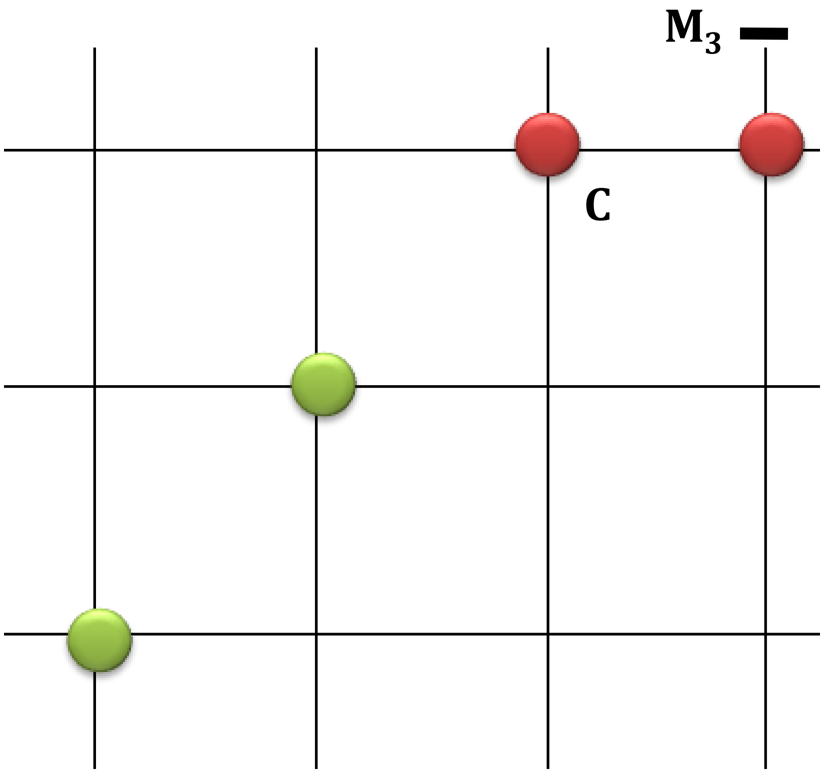
$$
\begin{aligned}
d_1 &= F(M_1) \\
&= F(X_P+1, Y_P+0.5) \\
&= a(X_P+1) + b(Y_P+0.5) + c
\end{aligned}
$$

IF $d_1 > 0$, *select NE* $(X_P=X_P+1,\ Y_P=Y_P+1)$

$$
\begin{aligned}
d_2 &= F(M_2) \\
&= F(X_P+2, Y_P+1.5) \\
&= a(X_P+2) + b(Y_P+1.5) + c \\
&= aX_P + 2a + bY_P + 1.5b + c \\
&= aX_P + a + bY_P + 0.5b + c + a + b \\
&= [\ a(X_P+1) + b(Y_P+0.5) + c\ ] + a + b \\
&= d_1 + (a + b)
\end{aligned}
$$

IF $d_2 > 0$, *select NE* $(X_P=X_P+1,\ Y_P=Y_P+1)$

Bresenham's Mid Point Criteria : Successive Updating (for selecting NE)



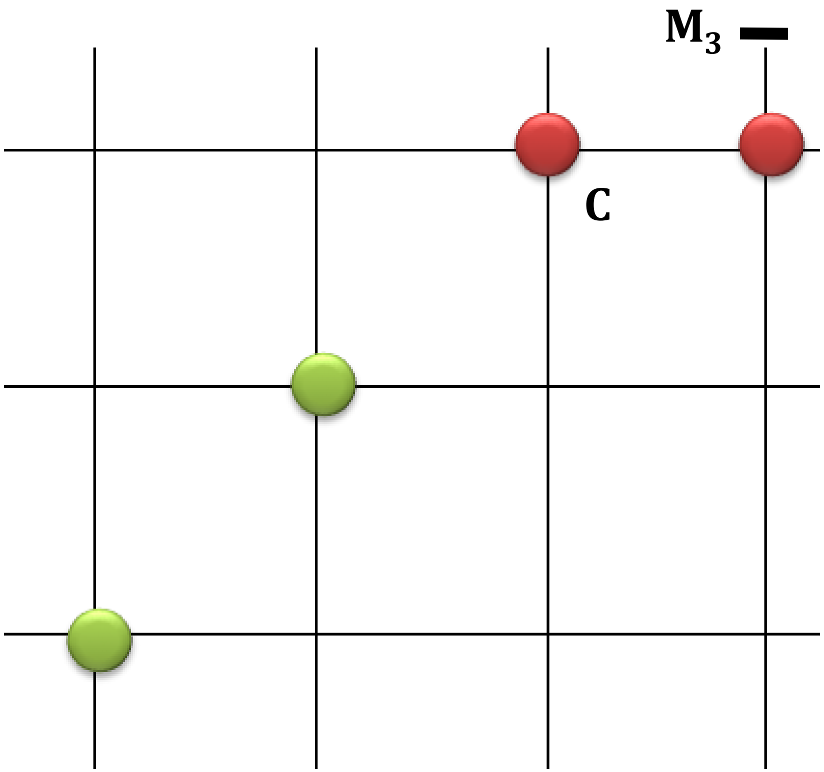
$d_1 = F(M_1)$
$= F(X_P+1, Y_P+0.5)$
$= a(X_P+1) + b(Y_P+0.5) + c$

IF $d_1 > 0$, *select NE* $(X_P=X_P+1, Y_P=Y_P+1)$

$d_2 = F(M_2)$
$= F(X_P+2, Y_P+1.5)$
$= a(X_P+2) + b(Y_P+1.5) + c$
$= aX_P + 2a + bY_P + 1.5b + c$
$= aX_P + a + bY_P + 0.5b + c + a + b$
$= [\ a(X_P+1) + b(Y_P+0.5) + c\ ] + a + b$
$= d_1 + (a + b)$

IF $d_2 > 0$, *select NE* $(X_P=X_P+1, Y_P=Y_P+1)$
Similarly, $d_3 = F(M_3) = d_2 + (a + b)$

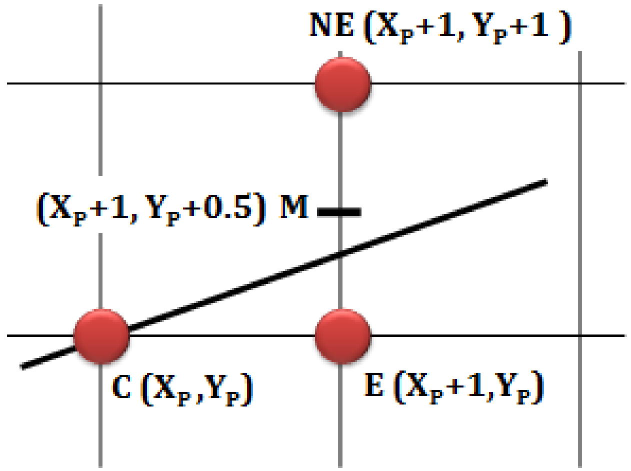
Bresenham's Mid Point Criteria : Successive Updating (for selecting NE)



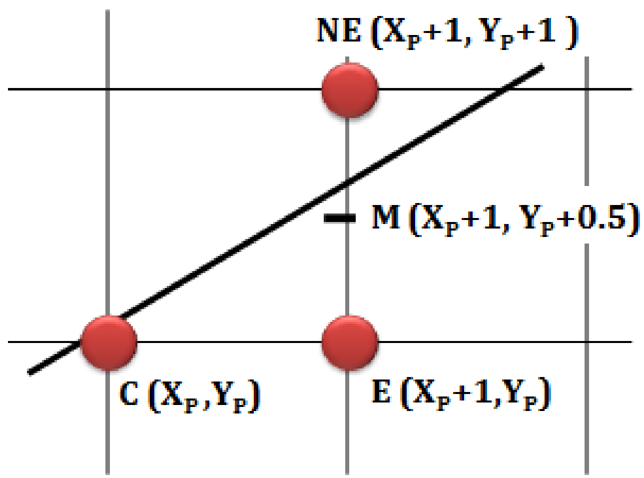
Every iteration after **selecting NE**, we can successively update our decision variable with-

$$\mathbf{d_{NEW} = d_{OLD} + (a + b)}$$
$$\mathbf{= d_{OLD} + (dy - dx)}$$

Bresenham's Mid Point Criteria : Successive Updating (Summary)



If **d ≤ 0**, then midpoint M is above the line, and E is closer to line,
So, **E** is selected
And, do-
d = d + ΔE
Where, ΔE = dy



If **d > 0**, then midpoint M is below the line, and NE is closer to line,
So, **NE** is selected
And, do-
d = d + ΔNE
Where, ΔE = dy - dx

Bresenham's Mid Point Algorithm

```
while (x <= x1)
    if d <=0 /* Choose E */
        x = x+1
        d = d + ΔE;

    else /* Choose NE */
        x = x+1
        y = y+1
        d = d + ΔNE
    Endif
    PlotPoint(x, y)
end while
```

Bresenham's Mid Point Algorithm

```
while (x <= x1)
    if d <=0 /* Choose E */
        x = x+1
        d = d + ΔE;

    else /* Choose NE */
        x = x+1
        y = y+1
        d = d + ΔNE
    Endif
    PlotPoint(x, y)
end while
```

Modified

```
while (x <= x1)
    if d <=0 /* Choose E */
        d = d + ΔE;

    else /* Choose NE */
        y = y+1
        d = d + ΔNE
    Endif
    x = x+1
    PlotPoint(x, y)
end while
```

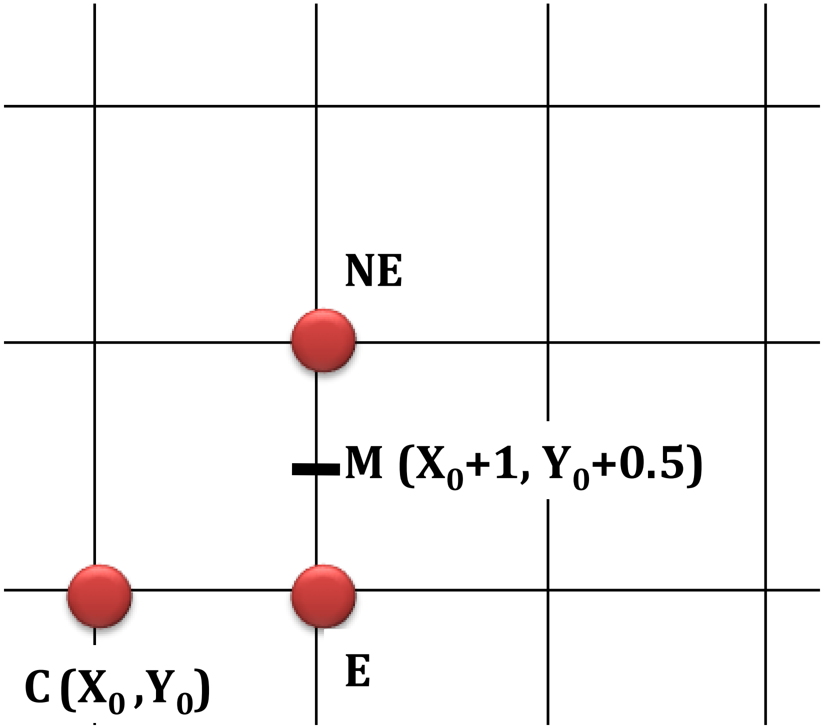
Bresenham's Mid Point Algorithm

*This **d** must be initialized to start the successive operation*

```
while (x <= x1)
    if d <=0 /* Choose E */
        x = x+1
        d = d + ΔE;

    else /* Choose NE */
        x = x+1
        y = y+1
        d = d + ΔNE
    Endif
    PlotPoint(x, y)
end while
```

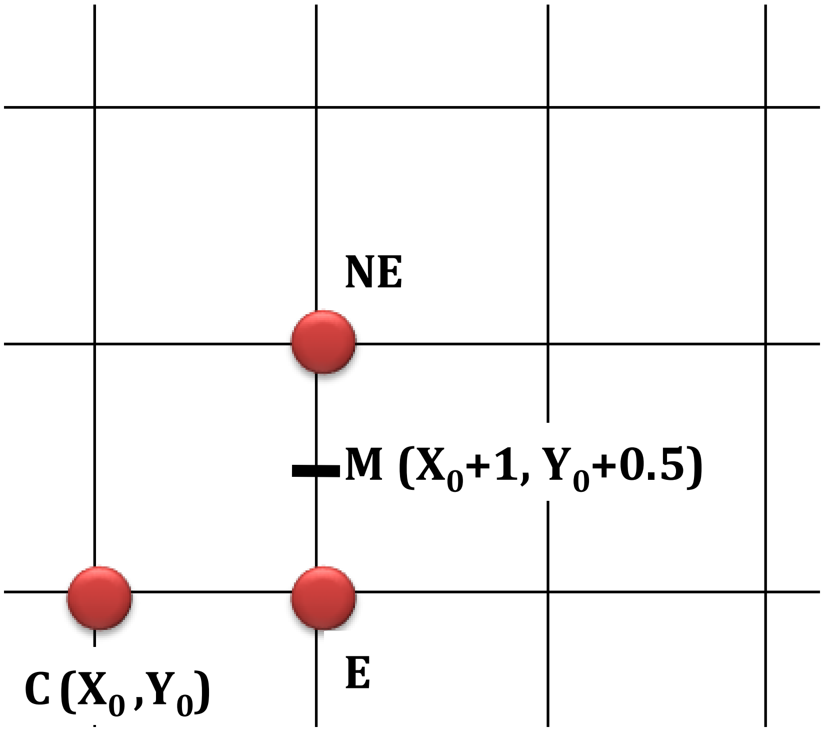
```
while (x <= x1)
    if d <=0 /* Choose E */
        d = d + ΔE;

    else /* Choose NE */
        y = y+1
        d = d + ΔNE
    Endif
    x = x+1
    PlotPoint(x, y)
end while
```

Bresenham's Mid Point Algorithm: Initializing Decision Variable



$$
\begin{aligned}
d_{INIT} &= F(M) \\
&= F(X_0+1, Y_0+0.5) \\
&= a(X_0+1) + b(Y_0+0.5) + c \\
&= aX_0 + a + bY_0 + 0.5b + c \\
&= aX_0 + bY_0 + c + a + 0.5b \\
&= (aX_0 + bY_0 + c) + a + 0.5b \\
&= F(X_0, Y_0) + a + 0.5b \\
&= a + 0.5b \\
&= dy - 0.5dx
\end{aligned}
$$

Bresenham's Mid Point Algorithm: Initializing Decision Variable



$$
\begin{aligned}
d_{INIT} &= F(M) \\
&= F(X_0+1, Y_0+0.5) \\
&= a(X_0+1) + b(Y_0+0.5) + c \\
&= aX_0 + a + bY_0 + 0.5b + c \\
&= aX_0 + bY_0 + c + a + 0.5b \\
&= (aX_0 + bY_0 + c) + a + 0.5b \\
&= F(X_0, Y_0) + a + 0.5b \\
&= a + 0.5b \\
&= dy - 0.5dx
\end{aligned}
$$

Still there is floating point. floating point operation is slower than integer operation

Bresenham's Mid Point Algorithm: Initialization

→ $d_{INIT} = dy - 0.5dx$

$= 2dy - dx$

→ $\Delta E = 2dy$

→ $\Delta NE = 2(dy - dx)$

2 is multiplied with $\mathbf{d_{INIT}}$ to remove the floating point. Observe that, **ΔE** and **ΔNE** also multiplied by 2 as those two will be added with $\mathbf{d_{INIT}}$ depending on condition. The **sign** of the decision variable d is needed to select E or NE pixel. (+ve / -ve) **Value** is influencing the decision here.

Bresenham's Mid Point Algorithm

Given:

Start point (x0,y0)
End point (x1, y1)

Initialization:

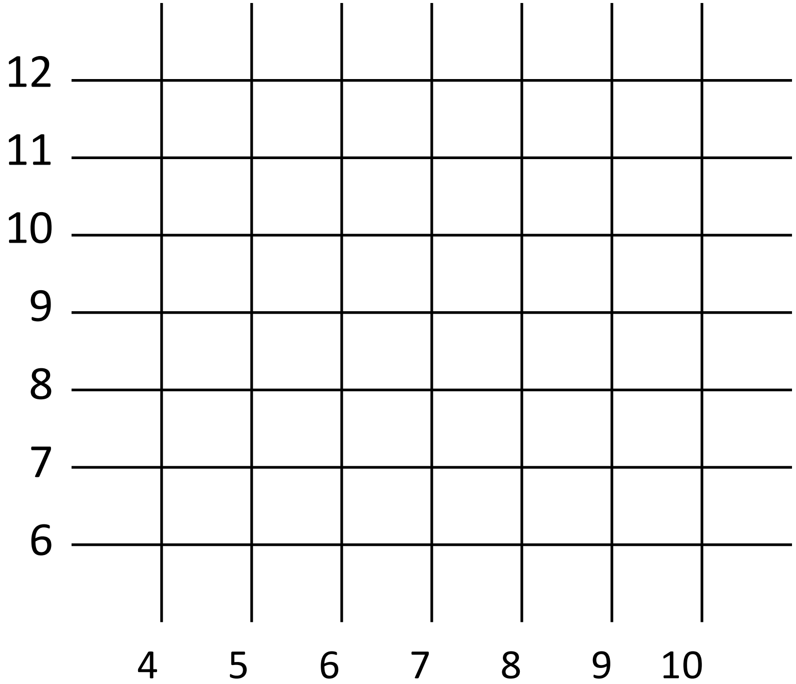
x = x0, y = y0
dx = x1-x0, dy = y1-y0
d = 2dy – dx
ΔE = 2dy
ΔNE = 2(dy – dx)

PlotPoint(x, y)

Loop:

```
while (x <= x1)
      if d <=0 /* Choose E */
            d = d + ΔE;

      else /* Choose NE */
            y = y+1
            d = d + ΔNE
      Endif
      x = x+1
      PlotPoint(x, y)
end while
```

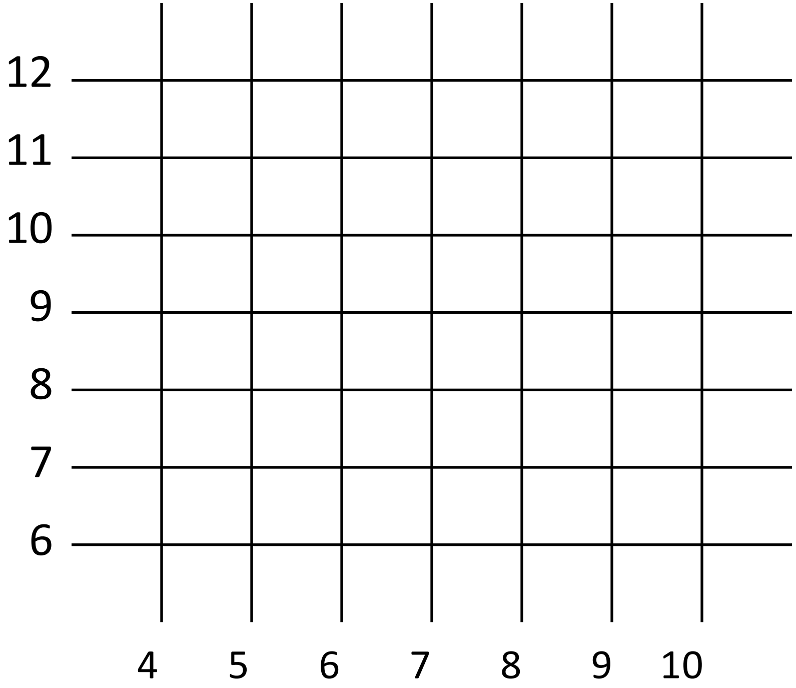
Bresenham's Mid Point Algorithm : Example



Given:

Start point (5, 8)
End point (9, 11)

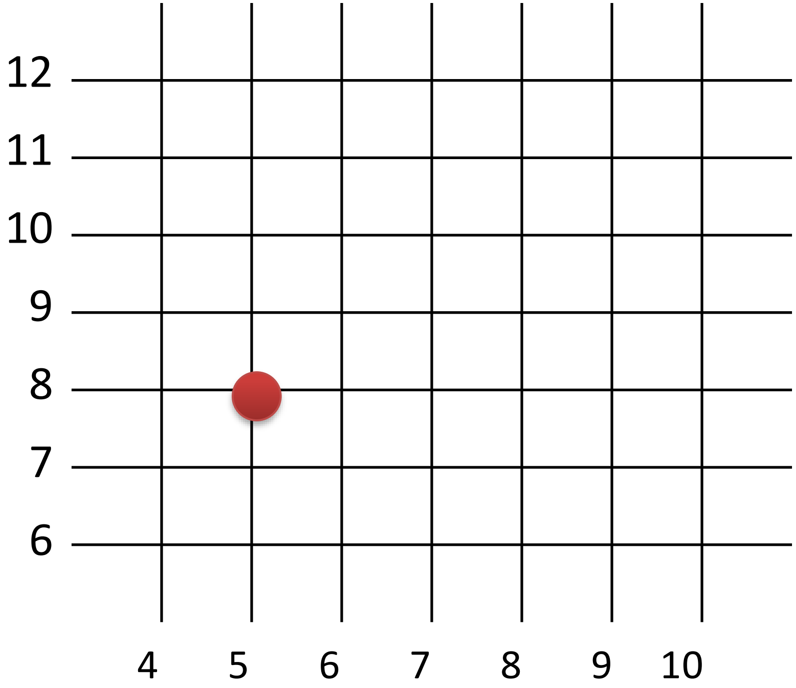
Bresenham's Mid Point Algorithm : Example



Start point (5, 8)
End point (9, 11)

dy = 3, dx = 4
d = 2dy – dx = 2
ΔE = 2dy = 6
ΔNE = 2(dy – dx) = -2

Bresenham's Mid Point Algorithm : Example



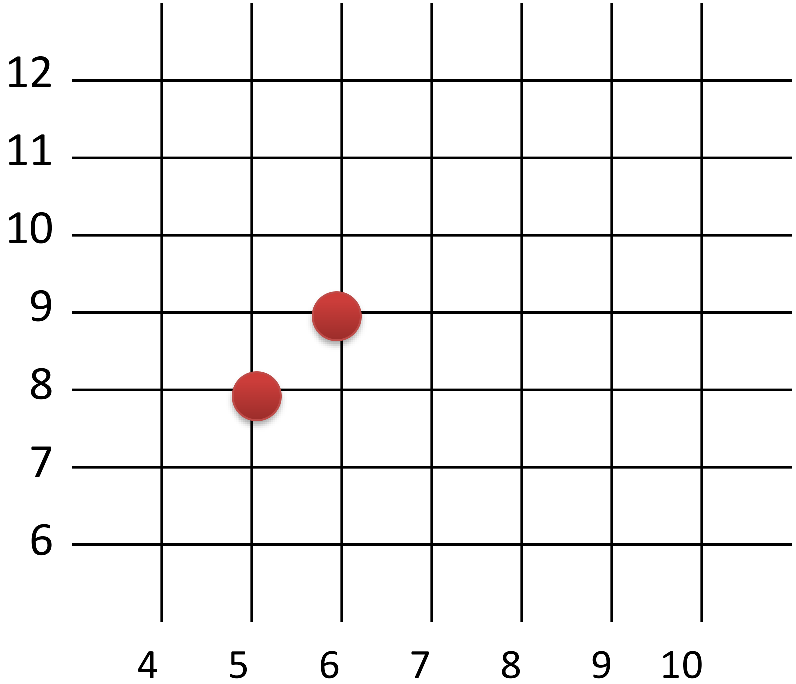
Start point (5, 8)
End point (9, 11)

dy = 3, dx = 4
d = 2dy – dx = 2
ΔE = 2dy = 6
ΔNE = 2(dy – dx) = -2

d = 2

d	2			
(X, Y)				

Bresenham's Mid Point Algorithm : Example



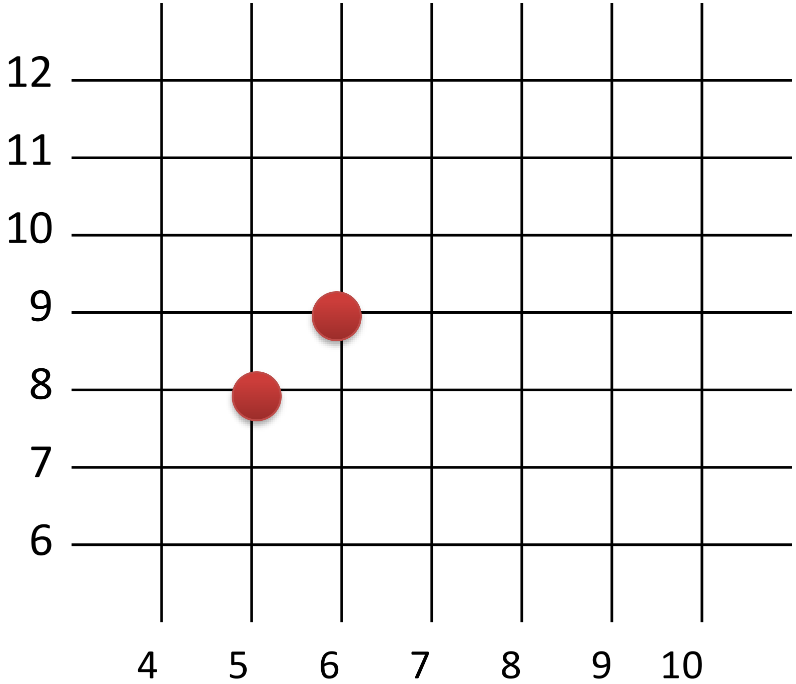
Start point (5, 8)
End point (9, 11)

dy = 3, dx = 4
d = 2dy – dx = 2
ΔE = 2dy = 6
ΔNE = 2(dy – dx) = -2

d	2			
(X, Y)	NE (6, 9)			

d > 0, NE

Bresenham's Mid Point Algorithm : Example



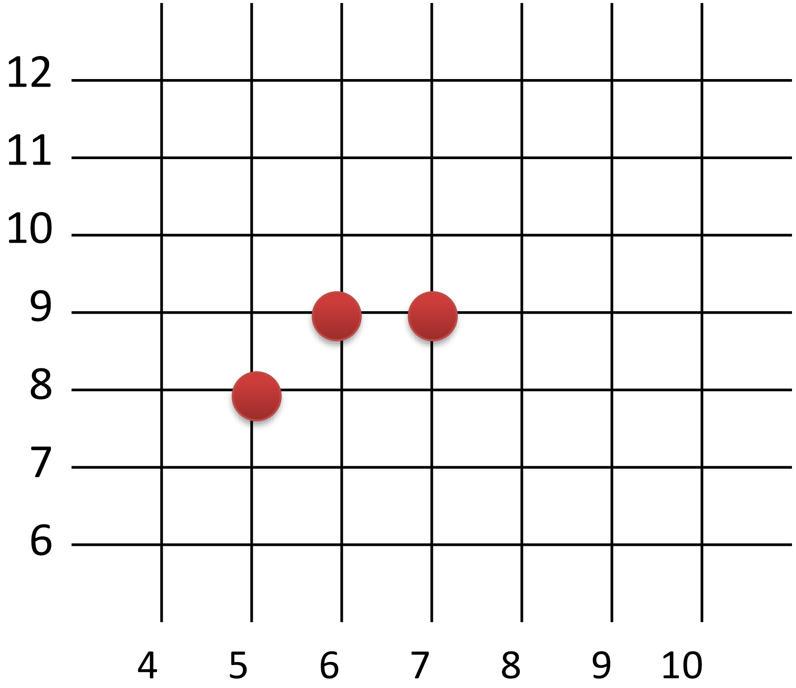
Start point (5, 8)
End point (9, 11)

dy = 3, dx = 4
d = 2dy – dx = 2
ΔE = 2dy = 6
ΔNE = 2(dy – dx) = -2

d = 2 + ΔNE

d	2	0		
(X, Y)	NE (6, 9)			

Bresenham's Mid Point Algorithm : Example



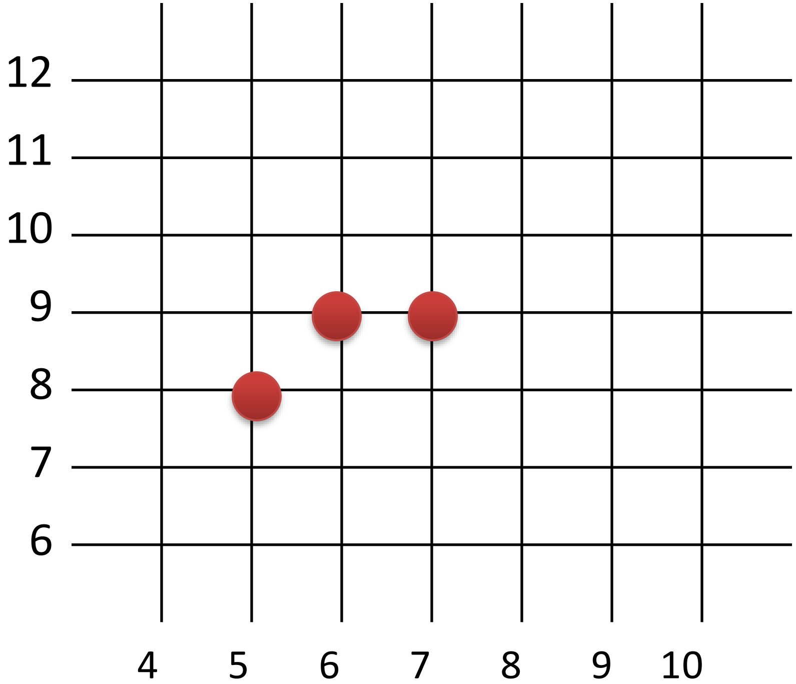
Start point (5, 8)
End point (9, 11)

dy = 3, dx = 4
d = 2dy – dx = 2
ΔE = 2dy = 6
ΔNE = 2(dy – dx) = -2

d	2	0		
(X, Y)	NE (6, 9)	E (7,9)		

d <= 0, E

Bresenham's Mid Point Algorithm : Example



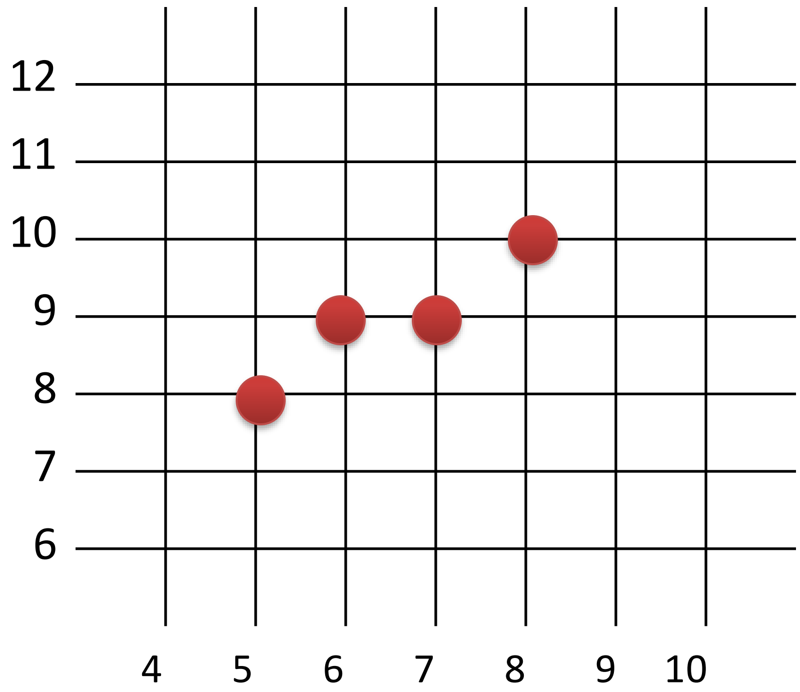
Start point (5, 8)
End point (9, 11)

dy = 3, dx = 4
d = 2dy – dx = 2
ΔE = 2dy = 6
ΔNE = 2(dy – dx) = -2

d = 0 + ΔE

d	2	0	6	
(X, Y)	NE (6, 9)	E (7,9)		

Bresenham's Mid Point Algorithm : Example



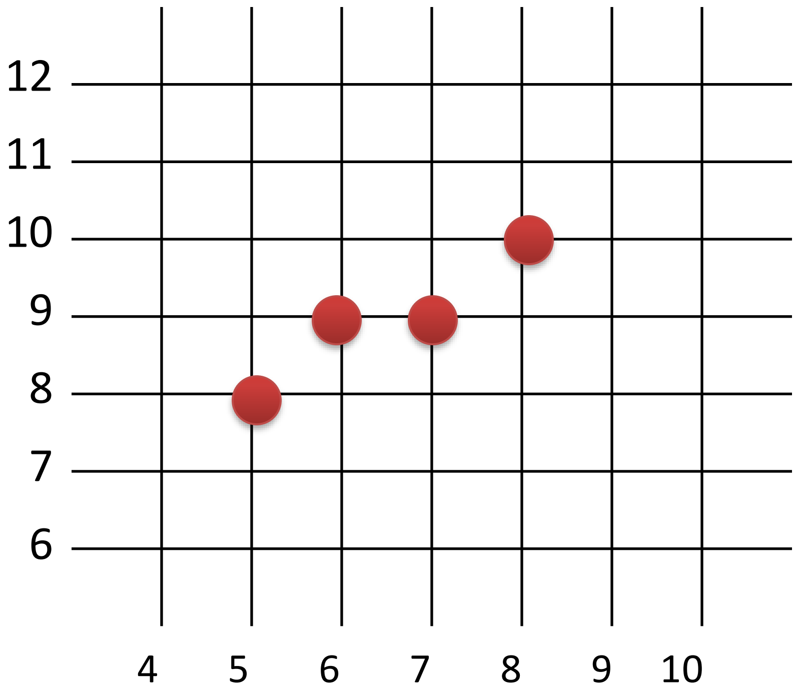
Start point (5, 8)
End point (9, 11)

dy = 3, dx = 4
d = 2dy – dx = 2
ΔE = 2dy = 6
ΔNE = 2(dy – dx) = -2

d	2	0	6	
(X, Y)	NE (6, 9)	E (7,9)	NE (8,10)	

d > 0, NE

Bresenham's Mid Point Algorithm : Example



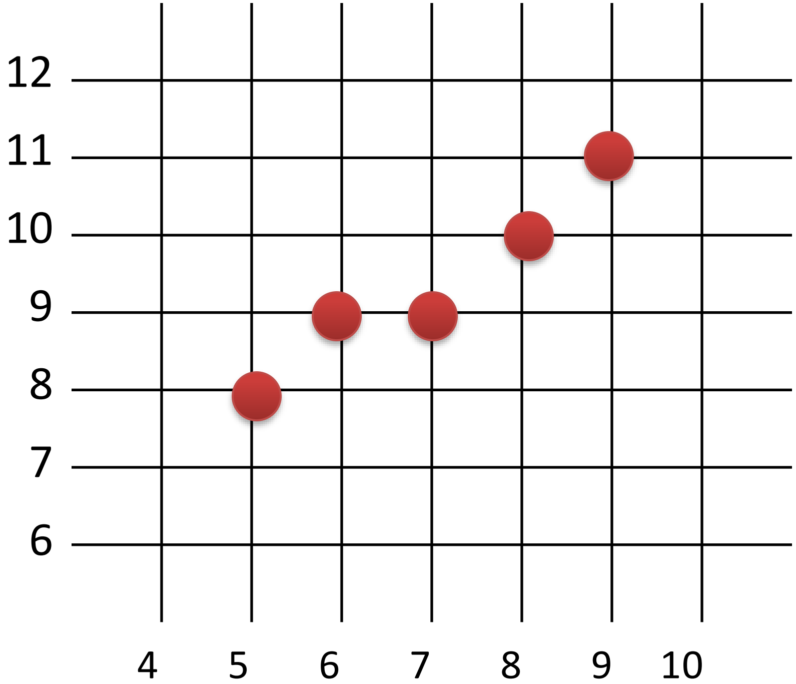
Start point (5, 8)
End point (9, 11)

dy = 3, dx = 4
d = 2dy – dx = 2
ΔE = 2dy = 6
ΔNE = 2(dy – dx) = -2

d = 6 + ΔNE

d	2	0	6	4
(X, Y)	NE (6, 9)	E (7,9)	NE (8,10)	

Bresenham's Mid Point Algorithm : Example



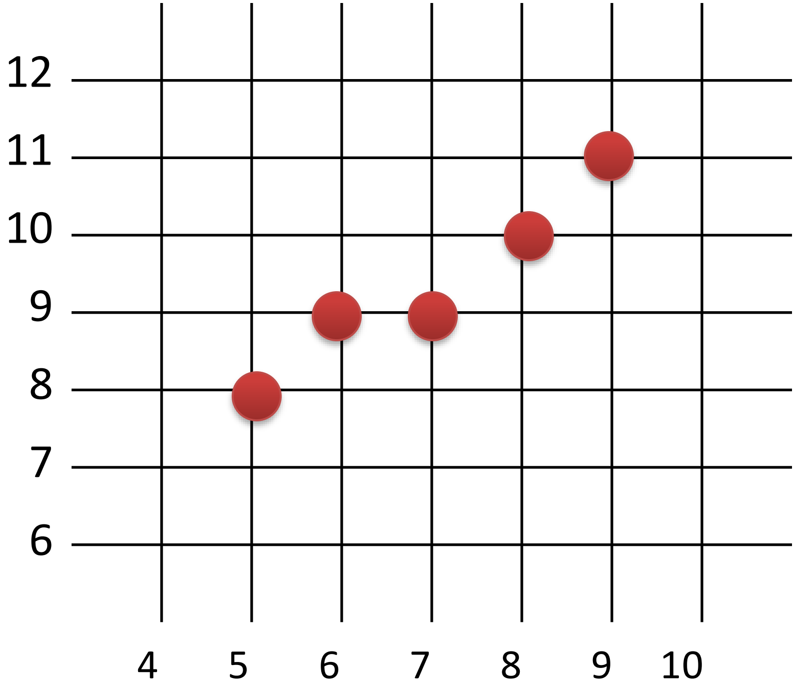
Start point (5, 8)
End point (9, 11)

dy = 3, dx = 4
d = 2dy – dx = 2
ΔE = 2dy = 6
ΔNE = 2(dy – dx) = -2

d	2	0	6	4
(X, Y)	NE (6, 9)	E (7,9)	NE (8,10)	NE (9,11)

d > 0, NE

Bresenham's Mid Point Algorithm : Example



Start point (5, 8)
End point (9, 11)

dy = 3, dx = 4
d = 2dy – dx = 2
ΔE = 2dy = 6
ΔNE = 2(dy – dx) = -2

d	2	0	6	4
(X, Y)	NE (6, 9)	E (7,9)	NE (8,10)	NE (9,11)

d > 0, NE