Fundamentals of Computer Graphics and Image Processing

2.LECTURE – STRAIGHT LINE ALGORITHM

Lecture plan

What is an algorithm? How to describe it?

Mathematical description of the straight line.

Straight line drawing algorithms:

- Straight line direct scanning conversion (using the mathematical formula)
- Bresenham's algorithm

Programming of Bresenham's algorithm

Mathematical calculation of the straight line points.

Algorithms

DESCRIPTIONS, FLOW CHARTS, DIAGRAMS

Algorithms: instructions and pseudo code

Step 1. Assign values to variables M and N.

Step 2. Divide M by N and assign the remainder to the variable P.

Step 3. If P value is not equal to 0

3.1. then assign the value of N to variable M and the value of P to N and go back to step 2

3.2. otherwise, go to step 4.

Step 4. Algorithm stops. The greatest common divisor is the value stored in variable N.

P ← M MOD N

WHILE P ≠ 0 DO

M ← N

N ← P

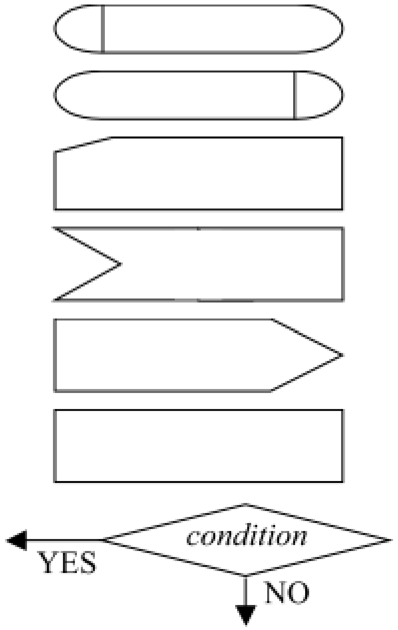
P ← M MOD N

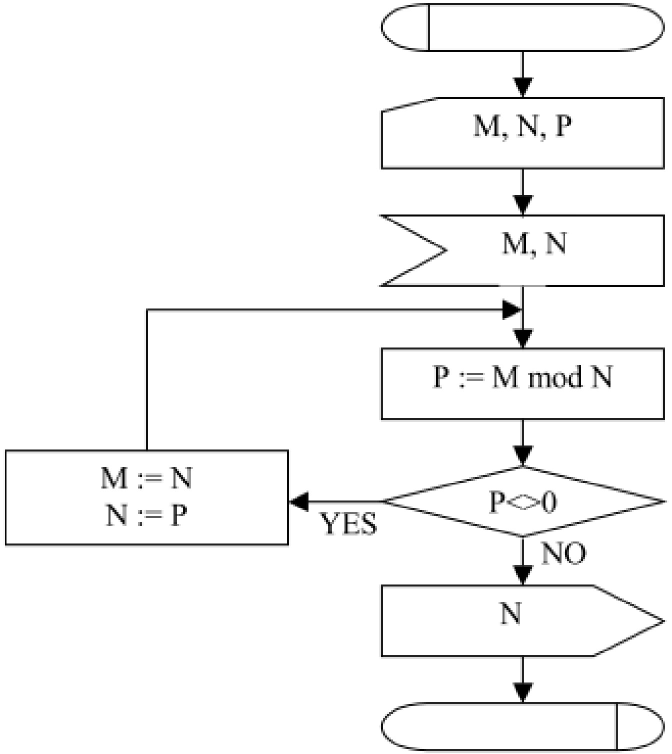
END WHILE

RETURN N

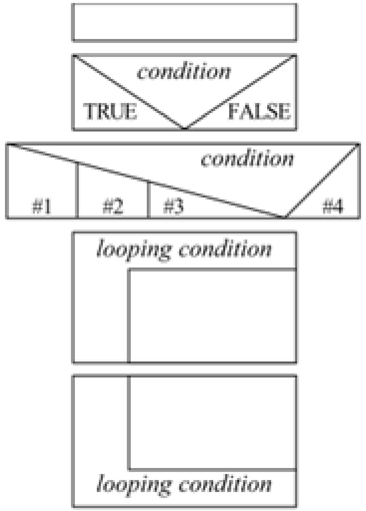
END FUNCTION

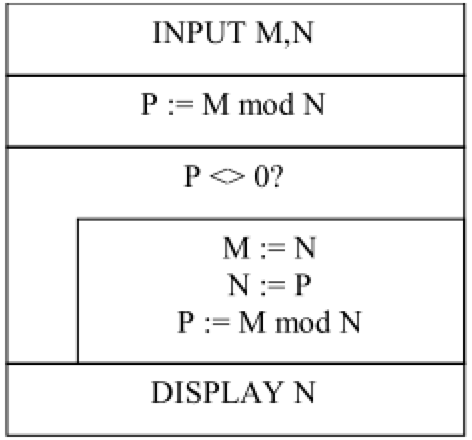
Algorithms: Flow charts.



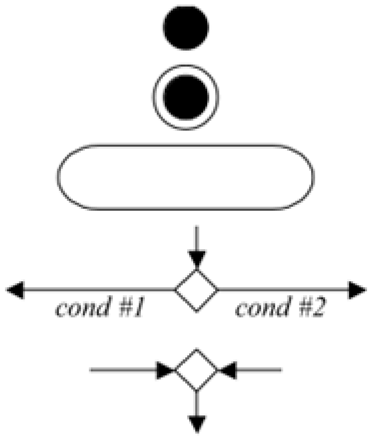


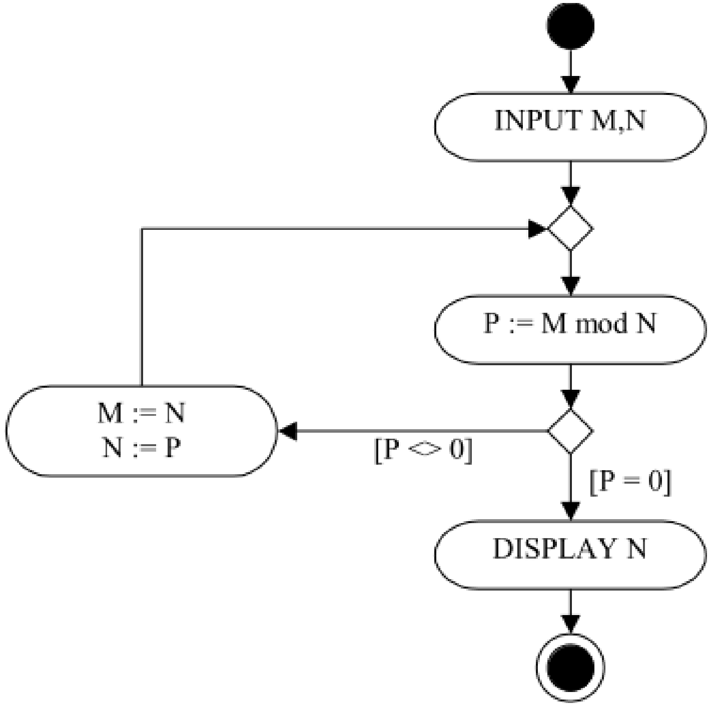
Algorithms: Nassi-Schneiderman structural diagramm





Algorithms: Unified modeling language action diagram

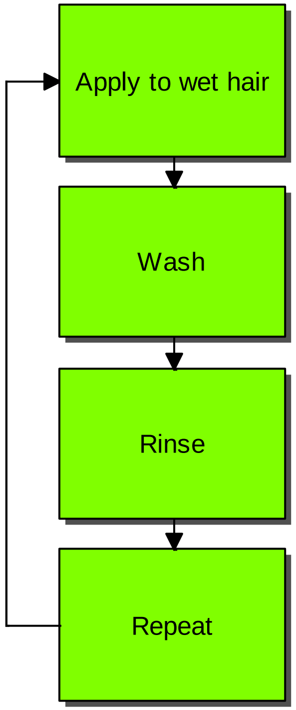




Algorithms: Example

Instructions:

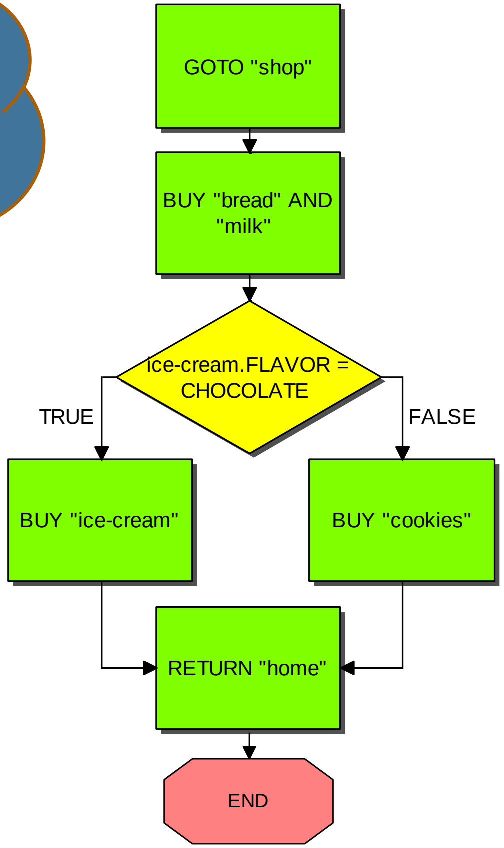
Apply to wet hair,
wash and rinse,
repeat



Algorithms: Example

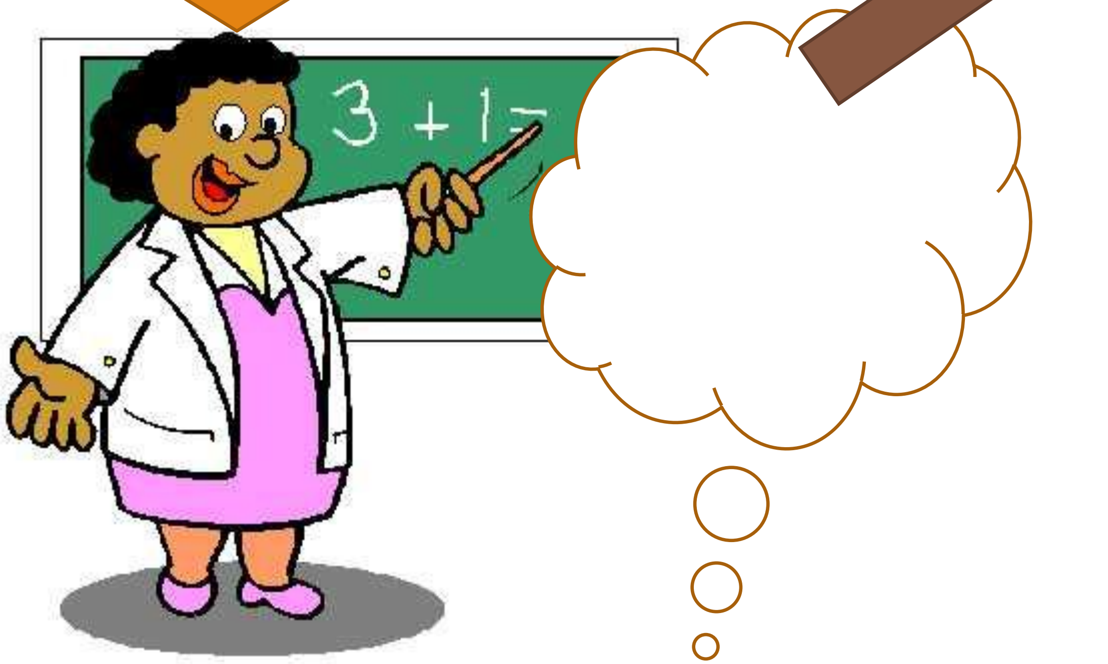
Go to the shop and buy some bread, milk and ice-cream! Oh, please buy chocolate ice-cream! If it’s not available, then buy some cookies.

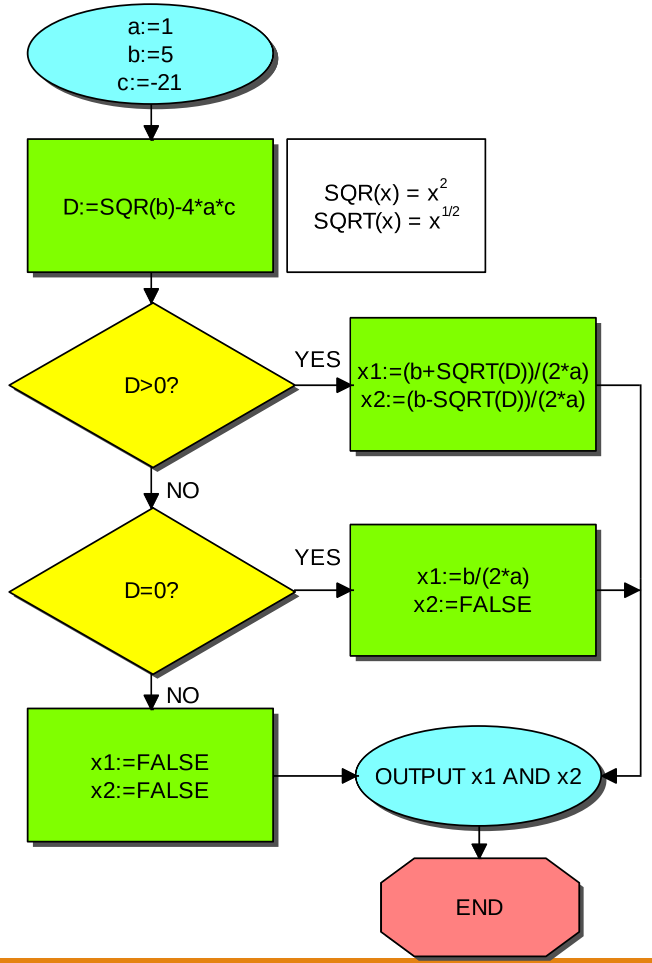
1. Go to shop
2. Buy bread and milk
3. If there is chocolate ice-cream, buy it… if not, buy cookies
4. Return home



Algorithms: Example

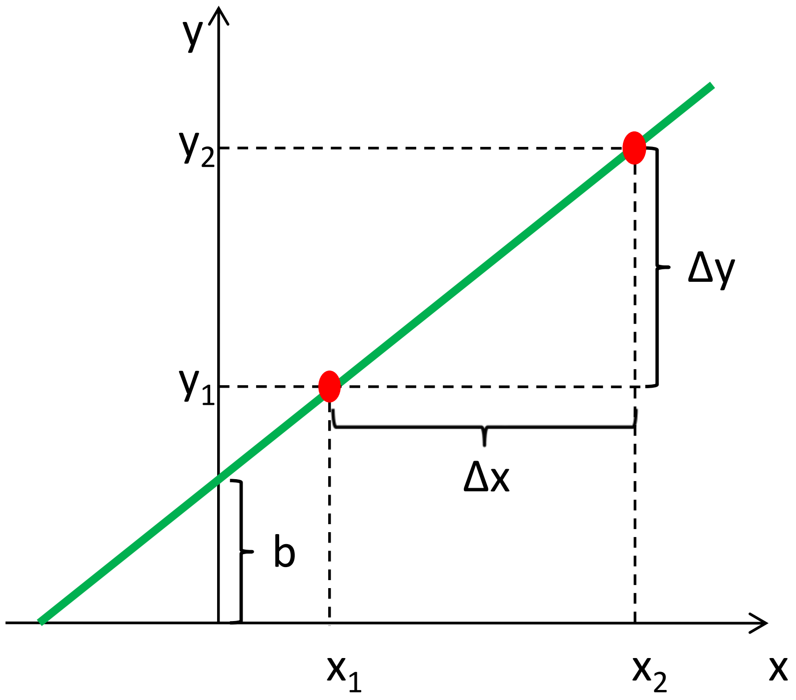
Find x for equation
$x^2+5x-21=0$!





Straight line drawing algorithms

Straight line mathematical decription

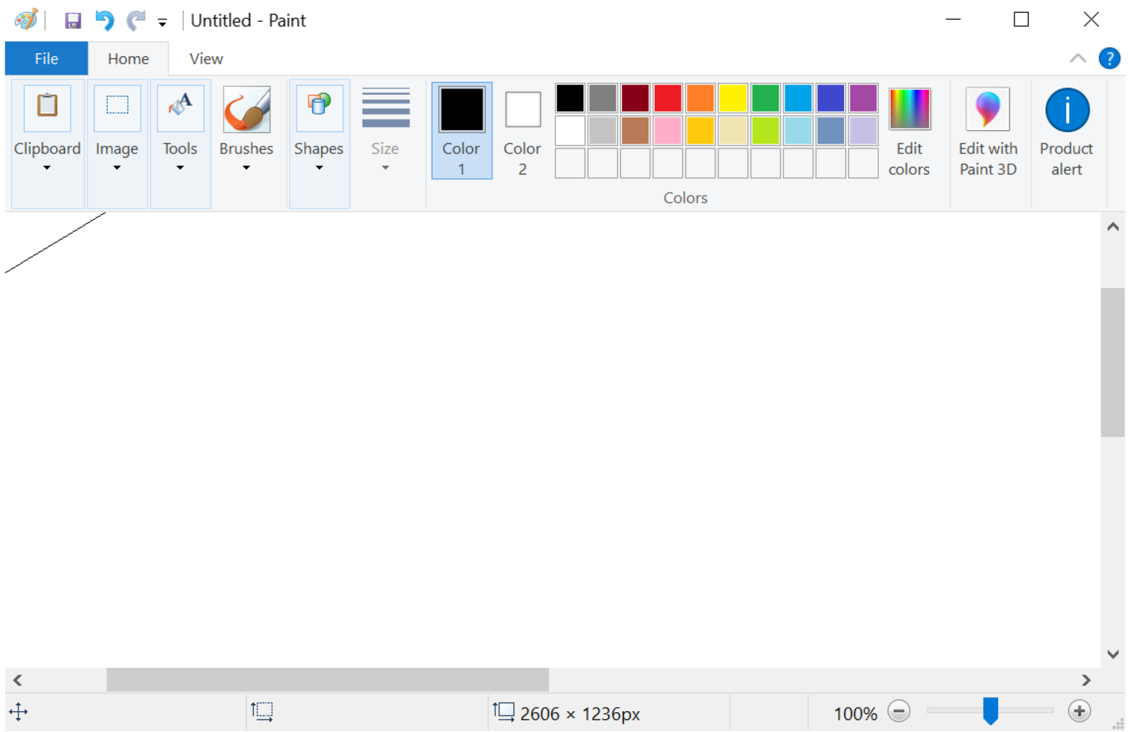


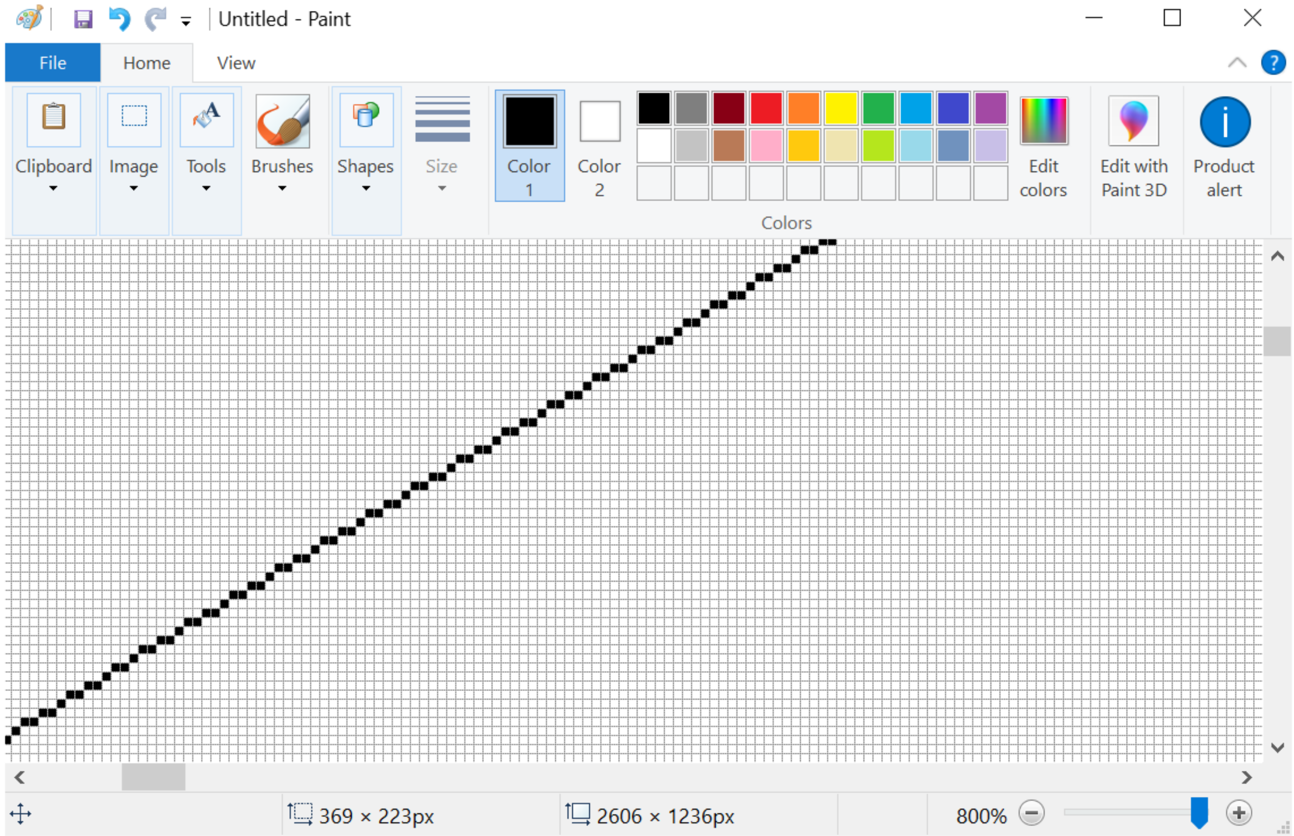
$$y = kx + b$$

$$k = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

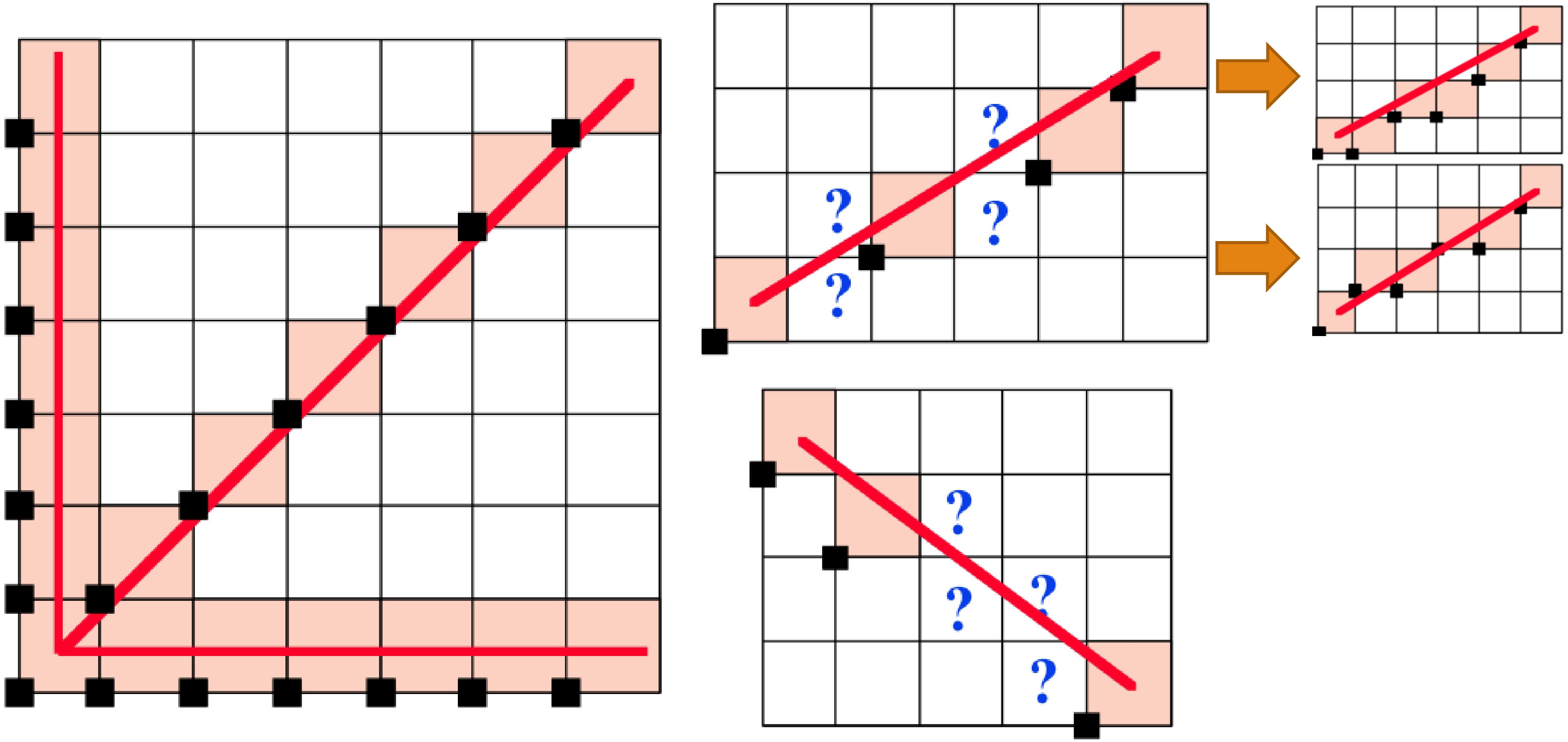
$$b = y_1 - kx_1$$

How to draw a line on a computer screen?





Line rasterization: the idea



Line rasterization: direct approach

$y = kx + b$

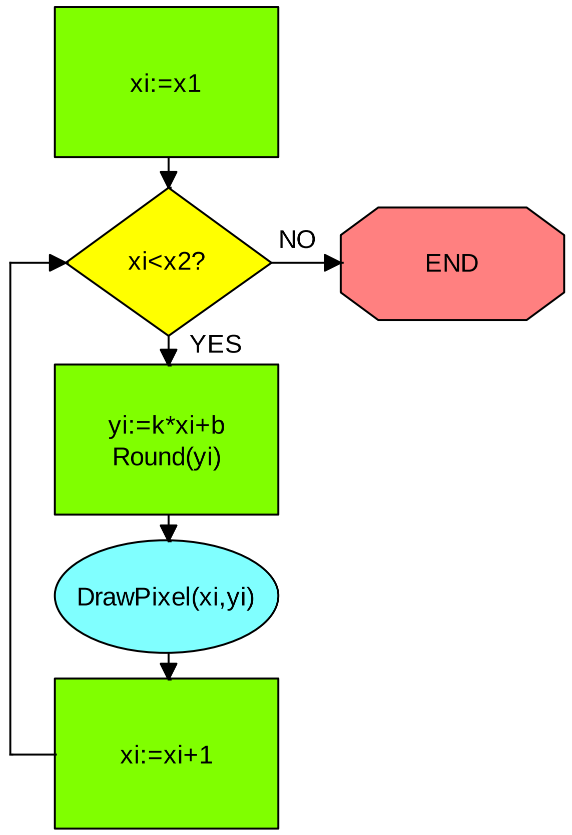
$x_1 = 5, y_1 = ?$
$x_2 = 6, y_2 = ?$
$x_3 = 7, y_3 = ?$
$x_4 = 8, y_4 = ?$

We can use the line formula for algorithm!

15

Line rasterization: direct approach

1. Begin with leftmost pixel x_1
2. Go through all pixel until algorithm reaches rightmost pixel x_2. With each pixel do the following:
   - For each x_i calculate the according y_i
   - Round the acquired y_i value to integer and draw the pixel (x_i, y_i)



Line rasterization: direct approach faults

$$y = kx + b$$

k: real!

$$k = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

b: real!

$$b = y_1 - kx_1$$

The direct approaches uses operations with floating point!

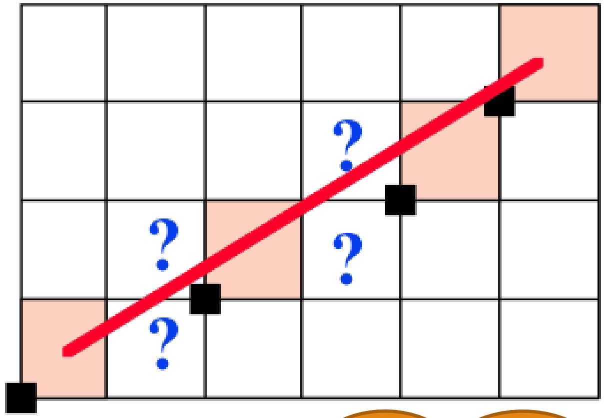
Bresenham algorithm

MATHEMATICAL DESCRIPTION, ALGORITHMIZATION, PROGRAMMING

Bresenham algorithm: History

We won't use math formula to calculate y!

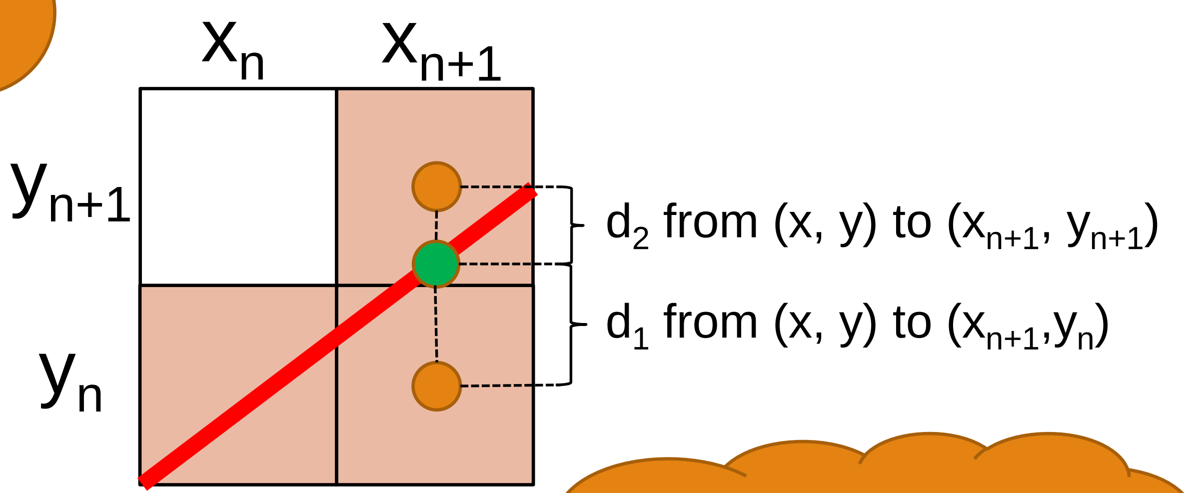
Jack Elton Bresenham, 1962



We will decide where to draw the next pixel using the decision parameter and previous pixel values!

Bresenham algorithm: Mathematical description

How to find the decisional parameter?



We first find out wich pixel the line is closer to!

Bresenham algorithm: Mathematical description

The straight line formula is $y = kx + b$,

It is known that, ka $x_{n+1} = x_n + 1$ un $y_{n+1} = y_n + 1$ then

$d_1 = y - y_n = k(x_n + 1) + b - y_n$

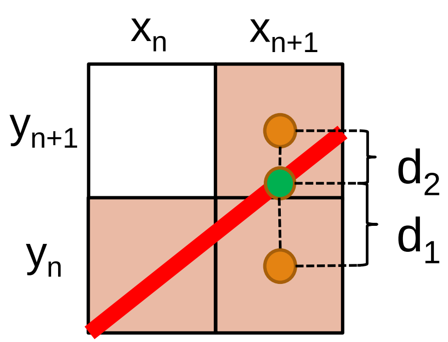
$d_2 = (y_n + 1) - y = y_n + 1 - k(x_n + 1) - b$

Ja $d_1 > d_2$ then the next pixel will be $(x_n + 1, y_n + 1)$

Ja $d_1 < d_2$ then the next pixel will be $(x_n + 1, y_n)$

Ja $d_1 = d_2$ then any pixel may be used

Example:
if $x_1 = 5$ then
$x_2 = 5 + 1 = 6$



Bresenham algorithm: Mathematical description

To decide the choice between d_1 un d_2, Bresenham proposed to use the difference between d_1 and d_2 and called it the decision parameter (p_n):

$$p_n = \Delta x(d_1 - d_2)$$

If $p_n > 0$ then $d_1 > d_2$, if $p_n < 0$, then $d_1 < d_2$

Why to multiply the difference by Δx? Because we don't want to use the floating point numbers $k = \frac{\Delta y}{\Delta x}$. In such a way we will only have to calculate integer values. Now, we should find $d_1 - d_2$, and describe it for programming:

$$d_1 - d_2 = k(x_n + 1) + b - y_n - (y_n + 1 - k(x_n + 1) - b) = 2k(x_n + 1) + 2b - 2y_n - 1$$

$$p_n = 2\Delta y \cdot x_n - 2\Delta x \cdot y_n + 2\Delta y + \Delta x(2b - 1)$$

Bresenham algorithm: Mathematical description

Now we have the decision parameter. We should calculate it for every pixel! Almost identical to the direct approach, where we calculate x first then y, we will need to calculate p first then (x, y). So, how to calculate the p_{n+1} for the next pixel?

$$p_{n+1} = 2\Delta y \cdot x_{n+1} - 2\Delta x \cdot y_{n+1} + 2\Delta y + \Delta x(2b - 1)$$

Let's subtract the p_n value from p_{n+1}. This will allow us to lose the variable b (that is a floating point value), and will also give us the possibility to calculate the next parameter iteratively.

$$p_{n+1} - p_n = 2\Delta y(x_{n+1} - x_n) - 2\Delta x(y_{n+1} - y_n)$$

It is known that $x_{n+1} = x_n + 1$, then:

$$p_{n+1} = p_n + 2\Delta y - 2\Delta x(y_{n+1} - y_n),$$

where $(y_{n+1} - y_n)$ takes value of 0 or 1, depending on p_n value.

Bresenham algorithm: Algorithmization

The algorithm works as follows:

Two points ar given for the line – the starting point (x_0, y_0) and the ending point (x_g, y_g).

Algorithm first calculates the initial value of p_0:

$p_0 = 2\Delta y - \Delta x$

Ja $p_0 < 0$ tad $d_1 < d_2$, meaning that the next pixel will be

$(x_n + 1, y_n)$ and $p_1 = p_0 + 2\Delta y - 2\Delta x(y_{n+1} - y_n)$, but since $y_{n+1} = y_n$, then

$p_1 = p_0 + 2\Delta y$

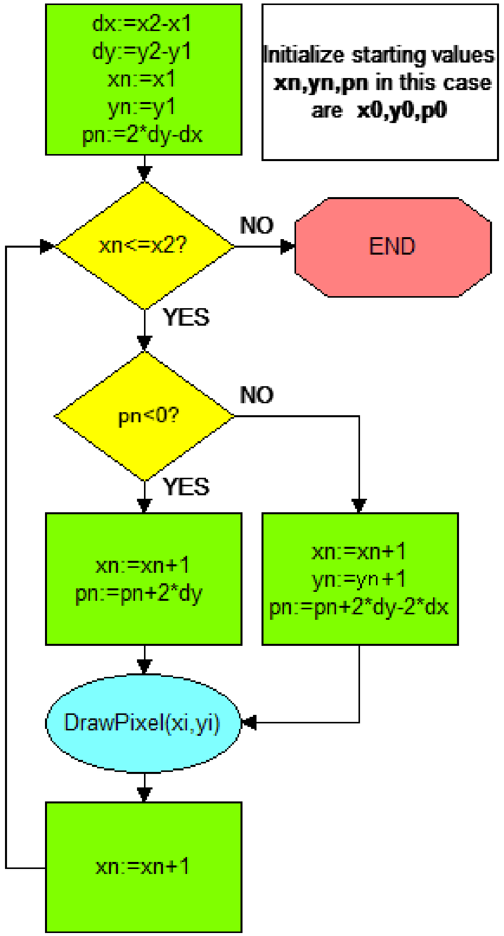
Ja $p_0 > 0$ tad $d_1 > d_2$, meaning that the next pixel will be

$(x_n + 1, y_n + 1)$ and $p_1 = p_0 + 2\Delta y - 2\Delta x(y_{n+1} - y_n)$, but since $y_{n+1} = y_n + 1$, then

$p_1 = p_0 + 2\Delta y - 2\Delta x$

Bresenham algorithm: Algorithmization

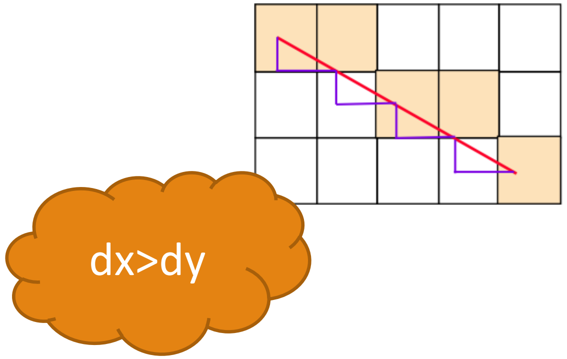
1. First calculate value p_0 sfor starting point (x_0, y_0) and $p_0 = 2\Delta y - \Delta x$
2. For each x_n, starting with $n = 0$, while $x_n < x_2$, calculate p_n and if:
   1. $p_n < 0$, tad $d_1 < d_2$, the next pixel will be $(x_n + 1, y_n)$ and $p_{n+1} = p_n + 2\Delta y$
   2. $p_n \geq 0$, tad $d_1 \geq d_2$, the next pixel will be $(x_n + 1, y_n + 1)$ and $p_{n+1} = p_n + 2\Delta y - 2\Delta x$

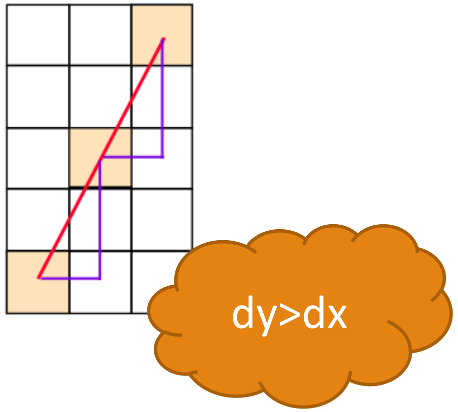


Bresenham algorithm: Algorithmization

But the algorithm should be updated!

1. It works only in cases when x and y are increasing $(x_n + 1, y_n + 1)$, but what is to be done than one of the coordinates decreases it's value? For example, $(x_1, y_1) = (0, 0)$, but $(x_2, y_2) = (5, -5)$. Then x coordinate increases, but y coordinate decreases $(x_n + 1, y_n - 1)$!
2. It works only in cases when $dx > dy$, because for each x only one y value is found. If coordinate x has several y values, then the resulting line will have holes:





Bresenham algorithm: Algorithmization

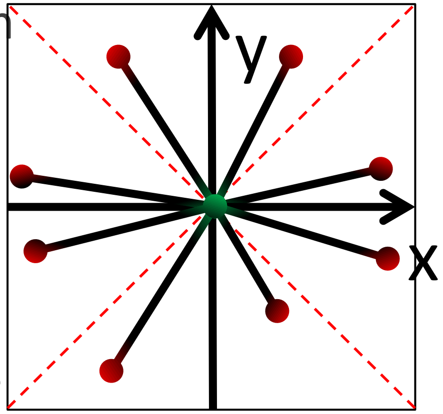
First solution:

Instead of solid increase (+1) we introduce a special step variable – x_s and y_s. They will equal -1 or +1 depending on points (x_1, y_1) and (x_2, y_2):

- If $x_2 > x_1$, then x increases, $(x_s = 1)$
- If $x_2 < x_1$, then x decreases, $(x_s = -1)$
- If $y_2 > y_1$, then y increases, $(y_s = 1)$
- If $y_2 < y_1$, then y decreases, $(y_s = -1)$

Since this step will differ depending on the line, we can use the absolute values of |Δx| and |Δy|, because it doesn't matter if they are positive or negative.

- $dx \ = \ |x_2 \ - \ x_1|$
- $dy \ = \ |y_2 \ - \ y_1|$

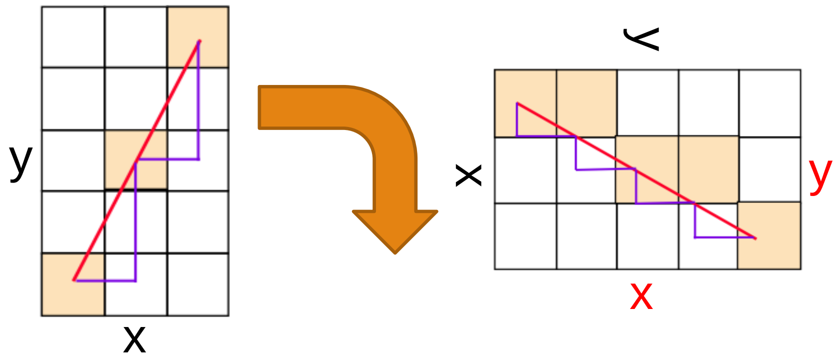


x_n+1? or x_n-1?
y_n+1? or y_n-1?

Bresenham algorithm: Algorithmization

Second solution:

To avoid holes in the lines where $dy > dx$, we must simply switch x and y places in every formula and condition. So for each y we will find one corresponding x value.



Bresenham algorithm: Algorithmization

Input data: starting point (x_{sp}, y_{sp}), and endpoint (x_{gp}, y_{gp})

1. Calculate dx un dy: $dx = |x_{gp} - x_{sp}|$ $dy = |y_{gp} = y_{sp}|$
2. Define x_s un y_s,
   - $x_s = 1$, if $x_{sp} < x_{gp}$, else $x_s = -1$
   - $y_s = 1$, if $y_{sp} < y_{gp}$, else $y_s = -1$
3. Define initial values for variables x_n un y_n, $n \in [0,1,2, \dots]$, initially $(x_0, y_0) = (x_{sp}, y_{sp})$

If $dx > dy$, then:

4. $P_0 = 2dy - dx;$

5. Until x_n reaches the endpoint, repeat:

- If $Pn > 0$, then $(x_{n+1}, y_{n+1}) = (x_n + x_s, y_n + y_s)$ and $P_{n+1} = P_n + 2dy - 2dx$
- If $Pn \leq 0$, then $(x_{n+1}, y_{n+1}) = (x_n + x_s , y_n)$ and $Pn = Pn + 2 * dy$

If $dx \leq dy$, then:

4. $P_0 = 2dx - dy;$

5. Until y_n reaches endpoint, repeat:

- If $P_n > 0$, then $(x_{n+1}, y_{n+1}) = (x_n + x_s, y_n + y_s)$ and $P_{n+1} = P_n + 2dx - 2dy$
- If $Pn \leq 0$, then $(x_{n+1}, y_{n+1}) = (x_n, y_n + y_s)$ and $P_{n+1} = P_n + 2 * dx$

Mathematical Calculations

PIEMĒRS

Example of Bresenham Algorithm Calculations

Let the starting point of the line be (x_{sp}=10, y_{sp}=10) and the and point be (x_{gp}=19, y_{gp}=15).

Let's calculate:

$|\Delta x| = |19 – 10| = 9$ $|\Delta y| = |15-10| = 5$

$x_s = 1$ $y_s = 1$ $(x_0, y_0) = (10, 10)$

And the initial decisional parameter value p_0 will be

$p0 = 2\Delta y - \Delta x = 10 - 9 = 1$

Example of Bresenham Algorithm Calculations

Now we calculate for each x:

1. $n = 0$, $p_0 = 1$

   since $p_0>0$, then the next pixel will be $(x_1,y_1)=(x_0+1,y_0+1)$, atceramies, ka $x_0=10$, $y_0=10$, tad $(x_1,y_1)=(11, 11)$, un rēķinam p_1 pēc formulas

   $p_1 = p_0 + 2\Delta y - 2\Delta x = 1 + 10 - 18 = -7$

2. $n = 1$, $p_1 = -7$

   since $p_1<0$, then the next pixel will be $(x_2,y_2)= (x_1+1,y_1)$, t.i (12, 11) un

   $p_2 = p_1 + 2\Delta y = -7 + 10 = 3$

Example of Bresenham Algorithm Calculations

3. $n = 2$, $p_2 = 3$

   since $p_2>0$, then the next pixel will be $(x_3,y_3)= (x_2+1,y_2+1)$, t.i (13, 12) un

   $$p_3 = p_2 + 2\Delta y - 2\Delta x = 3 + 10 - 18 = -5$$

4. $n = 3$, $p_3 = -5$

   since $p_3<0$, then the next pixel will be $(x_4,y_4)= (x_3+1,y_3)$, t.i (14, 12) un

   $$p_4 = p_3 + 2\Delta y = -5 + 10 = 5$$

Example of Bresenham Algorithm Calculations

5. $n = 4$, $p_4 = 5$

   since $p_4>0$, then the next pixel will be $(x_5,y_5)=(x_4+1,y_4+1)$, t.i (15, 13) un

   $$p_5 = p_4 + 2\Delta y - 2\Delta x = 5 + 10 - 18 = -3$$

6. $n = 5$, $p_5 = -3$

   since $p_5<0$, then the next pixel will be $(x_6,y_6)=(x_5+1,y_5)$, t.i (16, 13) un

   $$p_6 = p_5 + 2\Delta y = -3 + 10 = 7$$

Example of Bresenham Algorithm Calculations

7. $n = 6, p_6 = 7$

   since $p_6 > 0$, then the next pixel will be $(x_7, y_7) = (x_6 + 1, y_6 + 1)$, t.i (17, 14) un

   $$p_7 = p_6 + 2\Delta y - 2\Delta x = 7 + 10 - 18 = -1$$

8. $n = 7, p_7 = -1$

   since $p_8 < 0$, then the next pixel will be $(x_7, y_7) = (x_n + 1, y_n)$, t.i (18, 14) un

   $$p_8 = p_7 + 2\Delta y = -1 + 10 = 9$$

Example of Bresenham Algorithm Calculations

9. $n = 8$, $p_8 = 9$

   since $p_8>0$, then the next pixel will be (x_9+1,y_9+1), and that is the end point (19, 15)