Algorithm 1: 数值微分法 (DDA)

Input: 直线 L 两端点坐标 $P_0(x_0, y_0), P_1(x_1, y_1)$, 画线颜色 $color$

$\Delta x \leftarrow x_1 - x_0$, $\Delta y \leftarrow y_1 - x_0$;
$steps \leftarrow \max(|\Delta x|, |\Delta y|)$;
$\mathrm{d}x \leftarrow \frac{\Delta x}{steps}$, $\mathrm{d}y \leftarrow \frac{\Delta y}{steps}$;
$P \leftarrow P_0$;
for $i \leftarrow 1$ **to** $steps + 1$ **do**
    `putpixel` (`round`$(P), color$);
    $P \leftarrow P + (\mathrm{d}x, \mathrm{d}y)$;
end

Algorithm 2: 中点画线法

Input: 直线 L 两端点坐标 $P_0(x_0, y_0), P_1(x_1, y_1)$, 画线颜色 *color*

if $x_0 > x_1$ **then**
  `swap` (P_0, P_1);
end
$a \leftarrow y_0 - y_1, b \leftarrow x_1 - x_0, c \leftarrow x_0 y_1 - x_1 y_0$;
$P_i \leftarrow P_0$;
if $k \in [0, 1]$ **then**
  $d \leftarrow 2a + b$;
  $\Delta P_L \leftarrow (1, 1), \Delta d_L \leftarrow 2(a + b)$;
  $\Delta P_G \leftarrow (1, 0), \Delta d_G \leftarrow 2a$;
else if $k \in (1, +\infty)$ **then**
  $d \leftarrow a + 2b$;
  $\Delta P_L \leftarrow (0, 1), \Delta d_L \leftarrow 2b$;
  $\Delta P_G \leftarrow (1, 1), \Delta d_G \leftarrow 2(a + b)$;
else if $k \in [-1, 0)$ **then**
  $d \leftarrow 2a - b$;
  $\Delta P_L \leftarrow (1, 0), \Delta d_L \leftarrow 2a$;
  $\Delta P_G \leftarrow (1, -1), \Delta d_G \leftarrow 2(a - b)$;
else
  $d \leftarrow a - 2b$;
  $\Delta P_L \leftarrow (1, -1), \Delta d_L \leftarrow 2(a - b)$;
  $\Delta P_G \leftarrow (0, -1), \Delta d_G \leftarrow -2b$;
end
while $P \neq P_1$ **do**
  `putpixel` $(P_i$, color$)$;
  if $d < 0$ **then**
    $P \leftarrow P + \Delta P_L, d \leftarrow d + \Delta d_L$;
  else
    $P \leftarrow P + \Delta P_G, d \leftarrow d + \Delta d_G$;
  end
end
`putpixel` $(P_1$, color$)$;

Algorithm 3: Bresenham 画线法

Input: 直线 L 两端点坐标 $P_0(x_0, y_0), P_1(x_1, y_1)$, 画线颜色 *color*

if $x_0 > x_1$ **then**
  `swap` (P_0, P_1);
end
$\Delta x = x_1 - x_0, \Delta y = y_1 - y_0$;
$P_i \leftarrow P_0$;
if $k \in [0, 1]$ **then**
  $d \leftarrow 2\Delta y - \Delta x$;
  $\Delta P_L \leftarrow (1, 0), \Delta d_L \leftarrow 2\Delta y$;
  $\Delta P_G \leftarrow (1, 1), \Delta d_G \leftarrow 2(\Delta y - \Delta x)$;
else if $k \in (1, +\infty)$ **then**
  $d \leftarrow 2\Delta x - \Delta y$;
  $\Delta P_L \leftarrow (0, 1), \Delta d_L \leftarrow 2\Delta x$;
  $\Delta P_G \leftarrow (1, 1), \Delta d_G \leftarrow 2(\Delta x - \Delta y)$;
else if $k \in [-1, 0)$ **then**
  $d \leftarrow -2\Delta y - \Delta x$;
  $\Delta P_L \leftarrow (1, 0), \Delta d_L \leftarrow -2\Delta y$;
  $\Delta P_G \leftarrow (1, -1), \Delta d_G \leftarrow -2(\Delta y + \Delta x)$;
else
  $d \leftarrow 2\Delta x + \Delta y$;
  $\Delta P_L \leftarrow (0, -1), \Delta d_L \leftarrow 2\Delta x$;
  $\Delta P_G \leftarrow (1, -1), \Delta d_G \leftarrow 2(\Delta x + \Delta y)$;
end
while $P \neq P_1$ **do**
  `putpixel` (P_i, color);
  if $d < 0$ **then**
    $P \leftarrow P + \Delta P_L, d \leftarrow d + \Delta d_L$;
  else
    $P \leftarrow P + \Delta P_G, d \leftarrow d + \Delta d_G$;
  end
end
`putpixel` (P_1, color);