Algorithm for Line Drawing (all cases)

Step 1:

```
Input the starting and ending points:
x0, y0 (starting point)
x1, y1 (ending point)
```

Step 2:

```
Calculate the absolute differences:
$\Delta x = |x1 - x0|$
$\Delta y = |y1 - y0|$
```

Step 3:

```
Determine the step directions:
sx = 1 if x1 > x0 else -1
sy = 1 if y1 > y0 else -1
```

Step 4:

```
Initialize the starting point and lists to store the points:
xes = [x0], yes = [y0]
```

Step 5:

```
Check if $\Delta x > \Delta y$ (line is more horizontal):
  Calculate the initial decision parameter:
  $p = 2\Delta y - \Delta x$
  Iterate through the x-coordinates until x0 = x1:
    Increment x0 by sx: x0 = x0 + sx
    If p >= 0:
      Increment y0 by sy: y0 = y0 + sy
      Update p: $p = p + 2(\Delta y - \Delta x)$
    Else:
      Update p: $p = p + 2\Delta y$
    Append the new x0, y0 to xes, yes
```

Step 6:

```
Else (line is more vertical):
  Calculate the initial decision parameter:
  $p = 2\Delta x - \Delta y$
  Iterate through the y-coordinates until y0 = y1:
    Increment y0 by sy: y0 = y0 + sy
    If p >= 0:
      Increment x0 by sx: x0 = x0 + sx
```

```
        Update p: $p = p + 2(\Delta x - \Delta y)$
    Else:
        Update p: $p = p + 2\Delta x$
    Append the new x0, y0 to xes, yes
```

Step 7:

Plot the points:
Use a graphing library (e.g., Matplotlib) to plot the line with the coordinates stored in xes and yes.

Step 8:

Stop when x0 = x1 or y0 = y1.