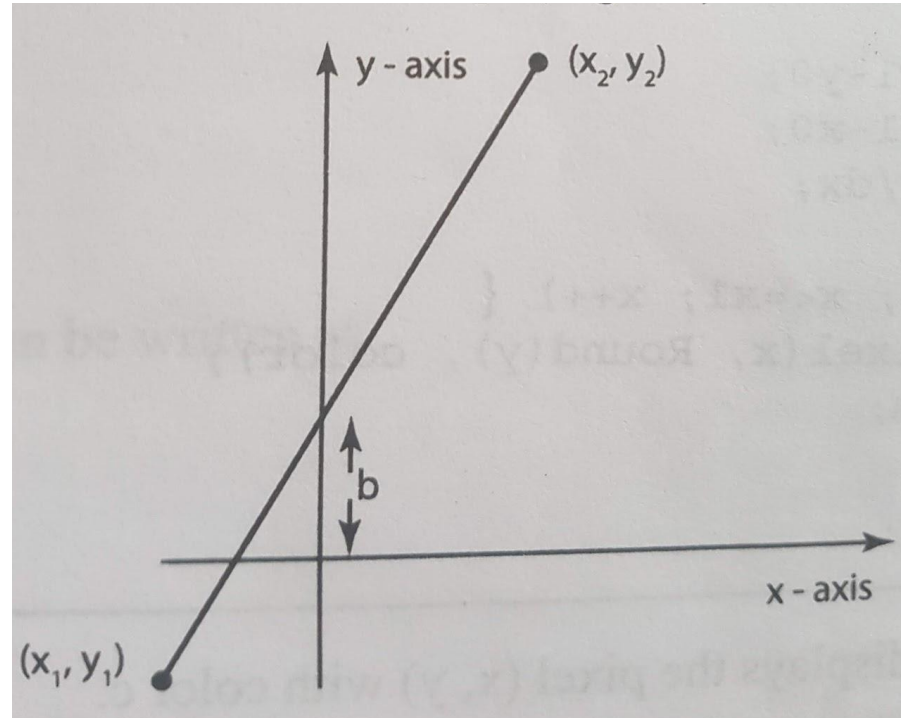


Line Basics

Line Basics

Line equation: $y=mx+b$
or
 $f(x,y)=ax+by+c=0$

By comparison
 $a=dy, b=-dx$ and $c=B.dx$



$m=dy/dx$
and
 $B= y\text{-intercept}$

Line Basics

- For a point (x_a, y_a) above the line $f(x, y)$
 $f(x_a, y_a)$ is -ve
- For a point (x_b, y_b) below the line $f(x, y)$
 $f(x_b, y_b)$ is +ve

Given +ve a

Bresenham Line Algo



Dr. Jack E. Bresenham
Researcher at IBM

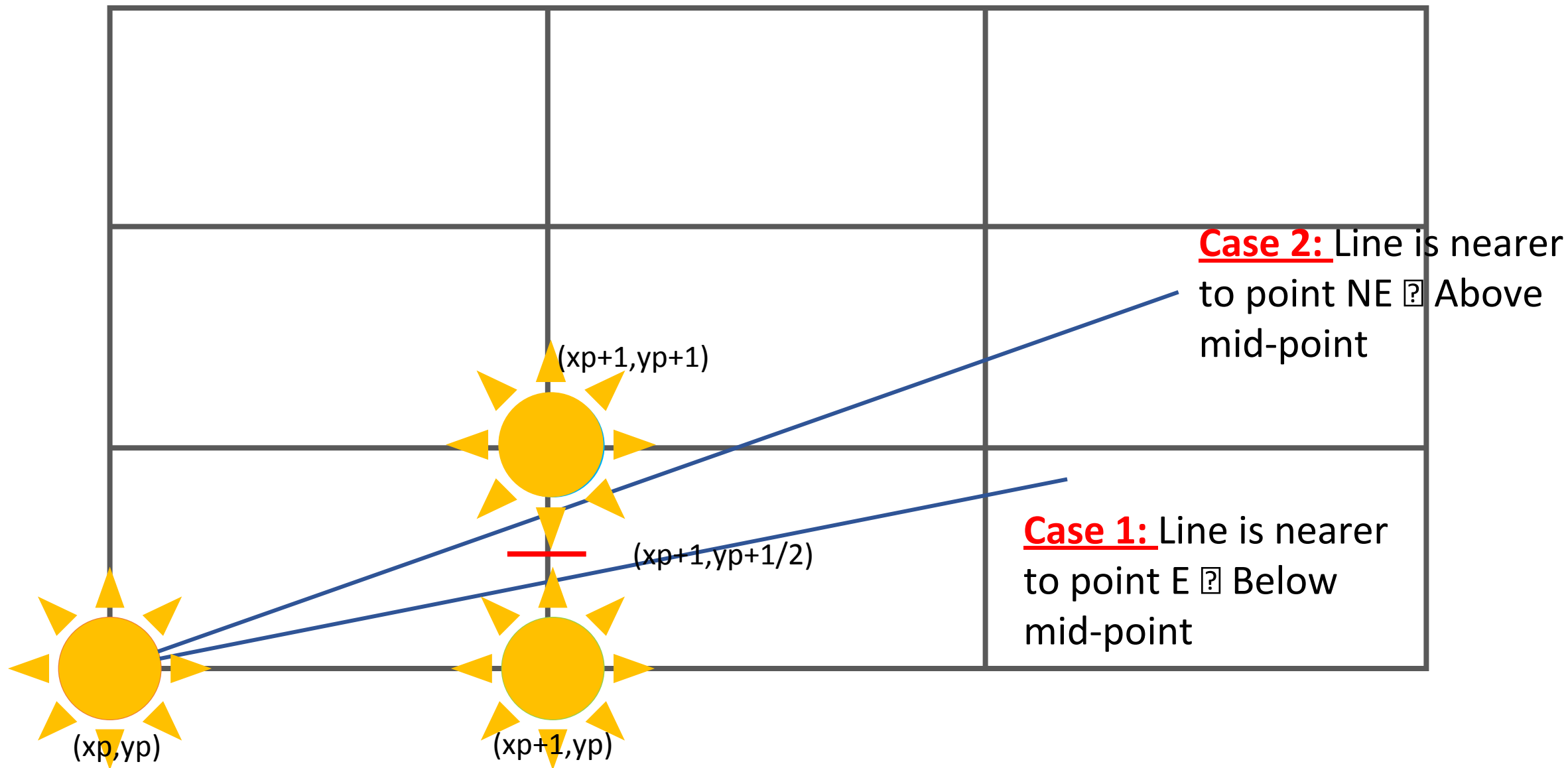
“

• • •

Thou shalt not
but use my algorithm
to draw lines
Henceforth

”

• • •

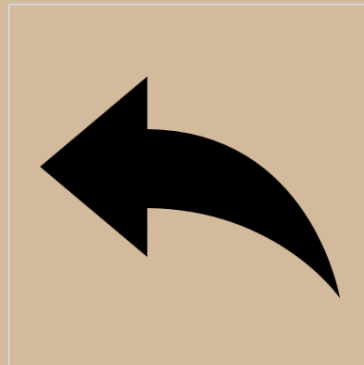




How can we determine if the line is above or below the mid-point without plotting



Put Mid-point (x_{p+1} , $y_{p+1/2}$) in line equation





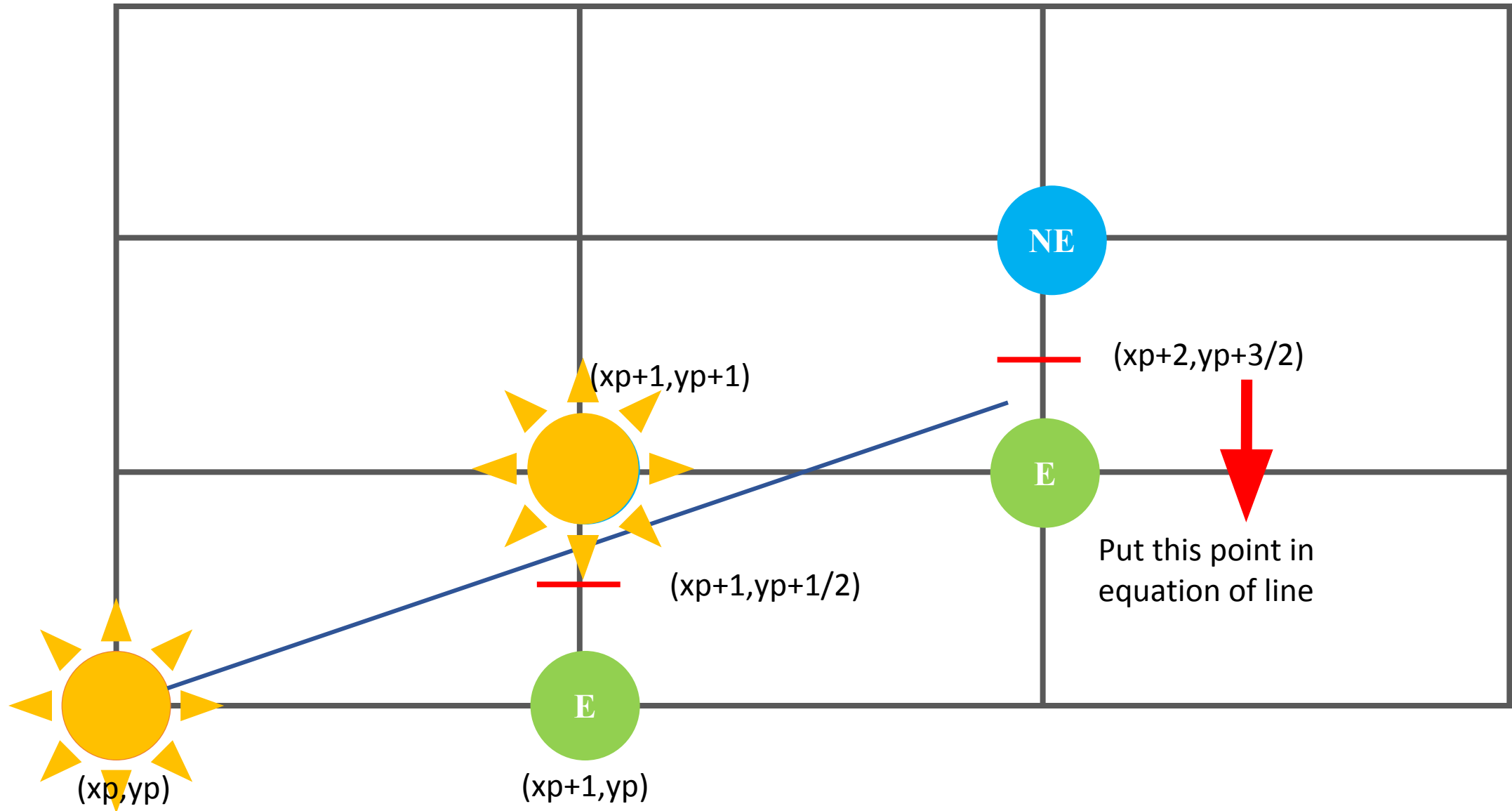
Task

Evaluate

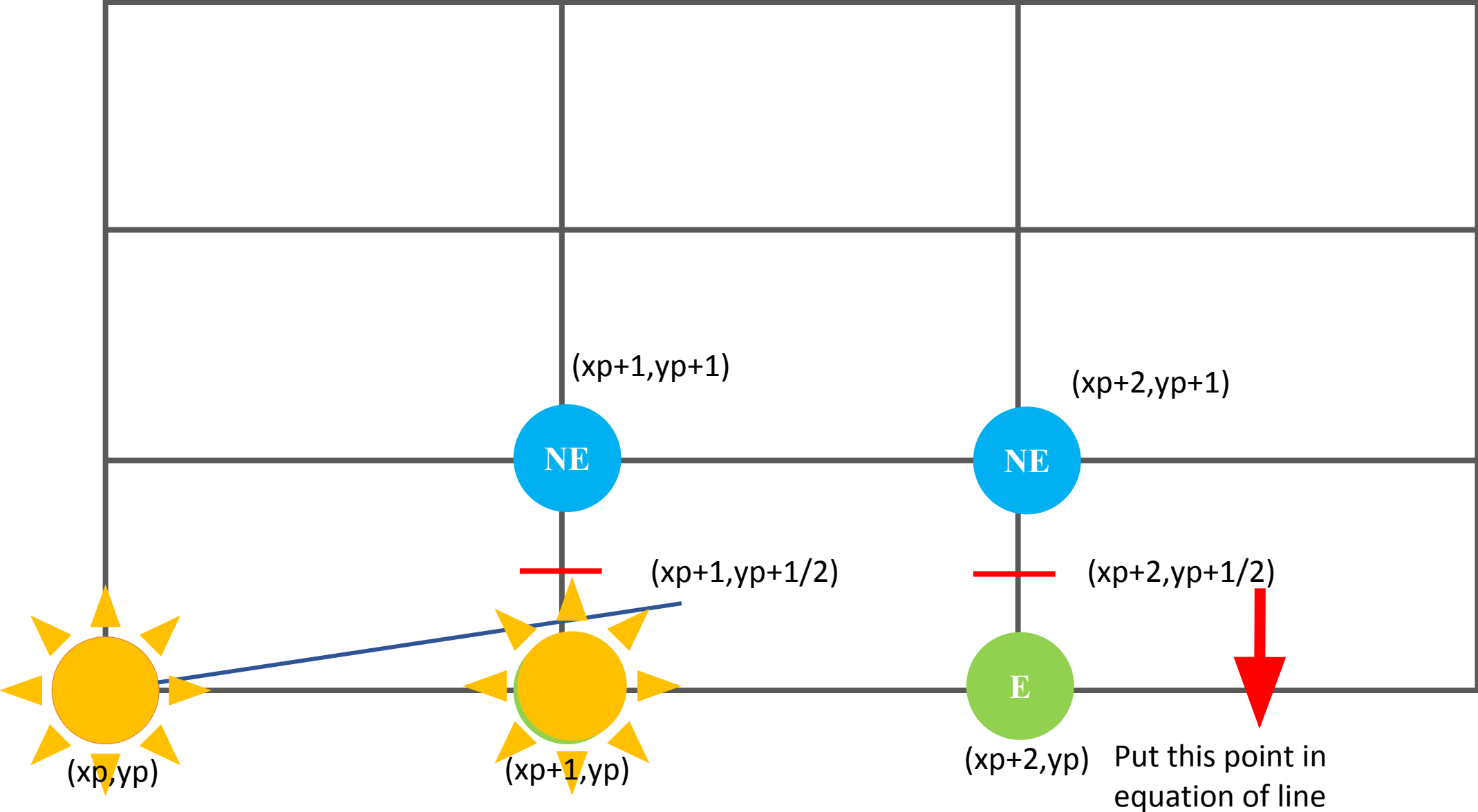


$$f(x_{p+1}, y_{p+1/2}) = ? = d$$

If d is +ve



If d is -ve



*Complete
Bresenham
Line
Algo
for $m < 1$*

1. Input the twoline endpoints and store the left endpoint in (x_0, y_0)
2. Plot (x_0, y_0)
3. Calculate constants dx , dy , $2dy$, and $2dy - dx$, and obtain the starting value for the decision parameter as d_{start} or $d_0 = 2dy - dx$
4. At each x_k along the line, starting at $k = 0$, perform the following test:
 - a. If $d_k < 0$, the next point to plot is $(x_k + 1, y_k)$ and $d_{k+1} = d_k + 2dy$ (or $d_{k+1} = d_k + dE$)
 - b. Otherwise, the next point to plot is $(x_k + 1, y_k + 1)$ and $d_{k+1} = d_k + 2dy - 2dx$ (or $d_{k+1} = d_k + dNE$)
5. Repeat step 4 dx times.

Pseudo code

```
void MidpointLine(int x0, int y0, int x1, int y1, int color)
//Assuming the lines slope is between 0 and 1
// (x0, y0) is lower left-end point
// (x1, y1) is upper right-end point
{
    int dx, dy, deltaE, deltaNE, d, x, y;
    dx = x1 - x0;
    dy = y1 - y0;
    d = 2 * dy - dx;          // Initial value of d
    deltaE = 2 * dy;          // Increment used to move to East pixel
    deltaNE = 2 * (dy - dx);
    x = x0;
    y = y0;
    WritePixel(x, y, color) // The start pixel
    while (x < x1) {
        if (d <= 0) {
            d += deltaE;
            x++;
        }
        else {
            d += deltaNE;
            x++;
            y++;
        }
        WritePixel(x, y, color);
    }
}
```

Example on board

- Plot a line between (2,1) and (7,4)
- how to make lines whose gradient are >1 or -ve

By Matlab Code

```
% testing Bresenham line algo  
% Line between (2,1) and (7,4)
```

```
x0=2;y0=1;x1=7;y1=4;
```

```
x=x0;
```

```
y=y0;
```

```
dy = 4-1;
```

```
dx = 7-2;
```

```
d = 2 * dy - dx;
```

```
deltaE = 2 * dy;
```

```
deltaNE = 2* (dy - dx);
```

```
scatter (x,y,60,"filled")
```

```
axis([0 8 0 5])
```

```
hold on
```

```
grid on
```

```
while (x < x1)
```

```
    if(d<=0)
```

```
        d=d+deltaE;
```

```
        x++;
```

```
    else
```

```
        d=d+deltaNE;
```

```
        x++;
```

```
        y++;
```

```
    end
```

```
    scatter(x,y,60,"filled");
```

```
end
```

