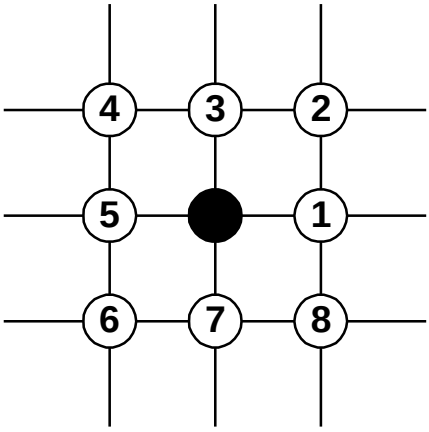
Các thuật toán vẽ đường

Dẫn nhập

- Giả sử tọa độ các điểm nguyên sau khi xấp xỉ đối tượng thực lần lượt là $(x_i, y_i), i = 0,\ldots$. Đây là các điểm nguyên sẽ được hiển thị trên màn hình.
- Bài toán đặt ra là nếu biết được (x_i, y_i) là tọa độ nguyên xác định ở bước thứ i, điểm nguyên tiếp theo (x_{i+1}, y_{i+1}) sẽ được xác định như thế nào.
- Đối tượng hiển thị trên lưới nguyên được liền nét, các điểm mà (x_{i+1}, y_{i+1}) có thể chọn chỉ là một trong tám điểm được đánh số từ 1 đến 8 trong hình sau (điểm đen chính là (x_i, y_i)).Hay nói cách khác : $(x_{i+1}, y_{i+1}) = (x_i \pm 1, y_i \pm 1)$.

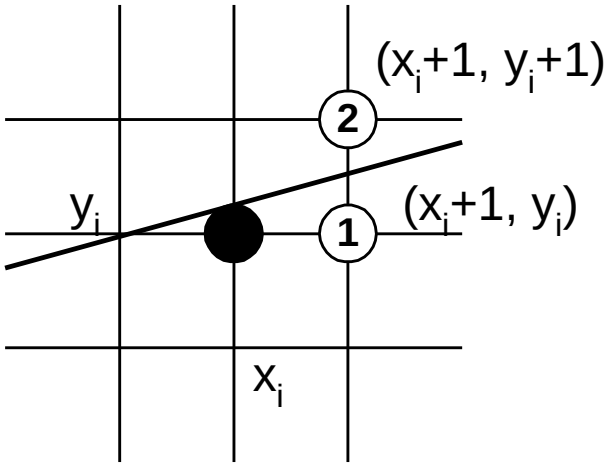


- Dáng điệu của đường sẽ cho ta gợi ý khi chọn một trong tám điểm trên. Cách chọn các điểm như thế nào sẽ tùy thuộc vào từng thuật toán trên cơ sở xem xét tới vấn đề tối ưu tốc độ.

Thuật toán vẽ đường thẳng

- Xét đoạn thẳng có hệ số góc $0 < m < 1$ và $Dx > 0$.
- Với các đoạn thẳng dạng này, nếu (x_i, y_i) là điểm đã xác định được ở bước thứ i (điểm màu đen) thì điểm cần chọn (x_{i+1}, y_{i+1}) ở bước thứ (i+1) sẽ là một trong hai trường hợp như hình vẽ sau :

$$\begin{cases} x_{i+1} = x_i + 1 \\ y_{i+1} \in \{y_i, y_i + 1\} \end{cases}$$



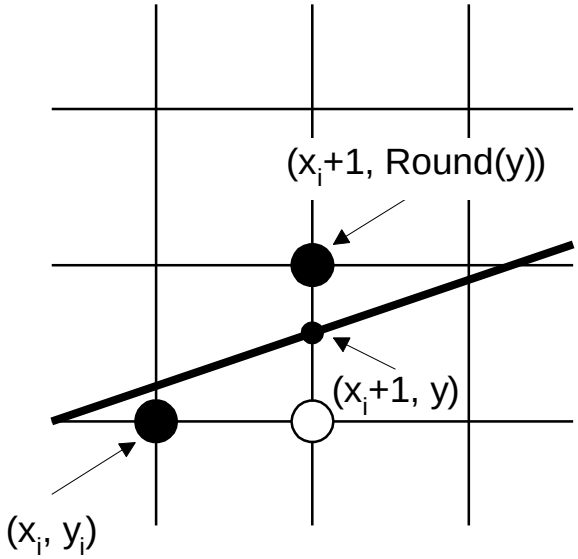
- Vấn đề còn lại, là cách chọn một trong hai điểm trên như thế nào để có thể tối ưu về mặt tốc độ.

Thuật toán DDA (Digital Differential Analyzer)

- Việc quyết định chọn y_{i+1} là y_i hay $y_i + 1$, dựa vào phương trình của đoạn thẳng $y = mx + b$. Nghĩa là, ta sẽ tính tọa độ của điểm $(x_i + 1, y)$ thuộc về đoạn thẳng thực. Tiếp đó, y_{i+1} sẽ là giá trị sau khi làm tròn giá trị tung độ y.

$$y = m(x_i + 1) + b$$

- Như vậy : $y_{i+1} = Round(y)$



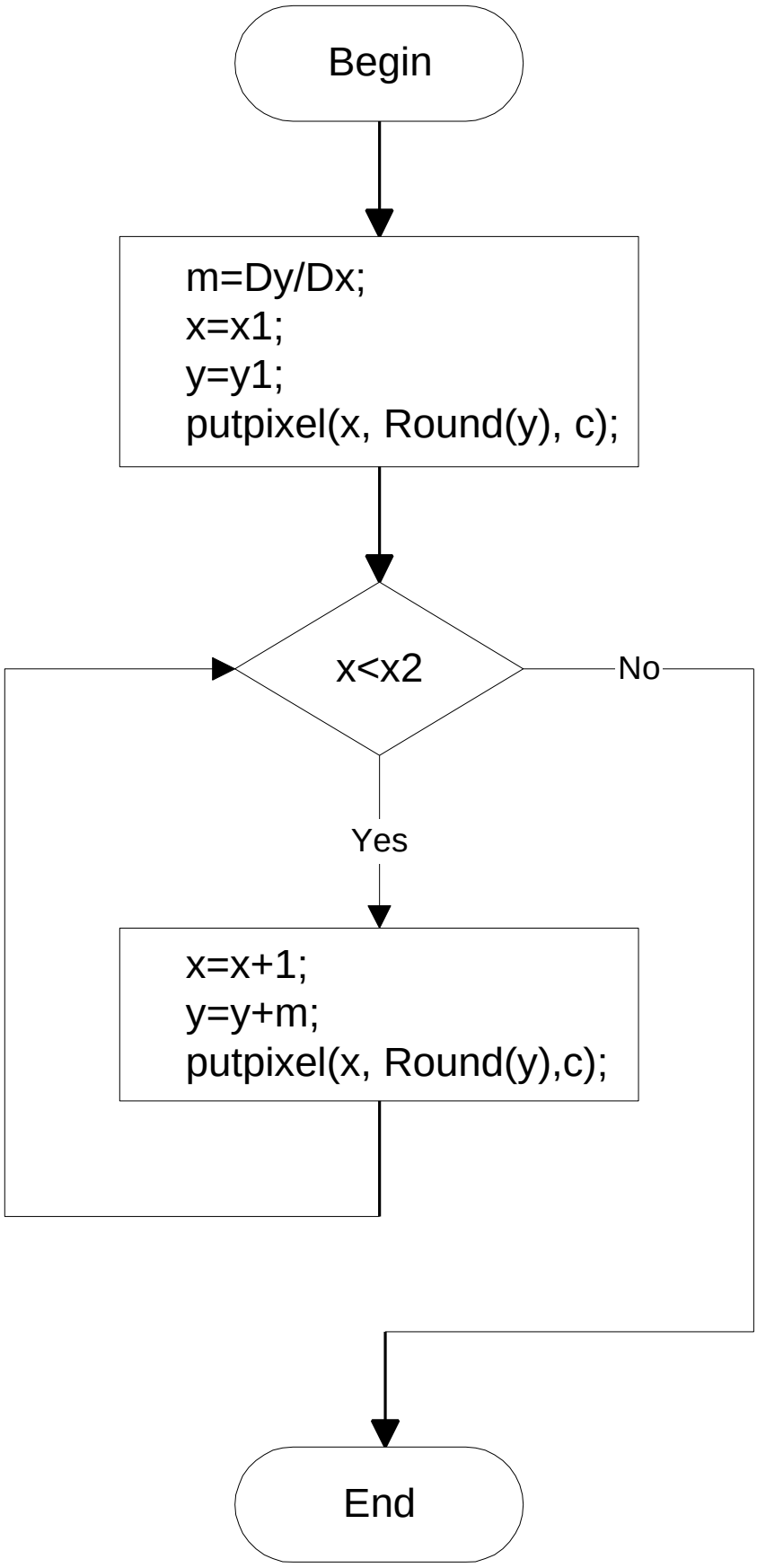
- Nếu tính trực tiếp giá trị thực y ở mỗi bước từ phương trình $y = mx + b$ thì phải cần một phép toán nhân và một phép toán cộng số thực. Để cải thiện tốc độ, người ta tính giá trị thực của y ở mỗi bước theo cách sau để khử phép tính nhân trên số thực :

- Nhận xét rằng : $y_{sau} = mx_{i+1} + b = m(x_i + 1) + b$

$$y_{trước} = mx_i + b$$

$$\Rightarrow y_{sau} = y_{trước} + m$$

Lưu đồ thuật toán DDA



- Ví dụ : Cho A(12, 20) và B(22, 27), ta có m= 0.7

i	x_i	y_i	y
0	12	20	20
1	13	21	20.7
2	14	21	21.4
3	15	22	22.1
4	16		
5	17		
6	18		
7	19		
8	20		
9	21		
10	22	27	

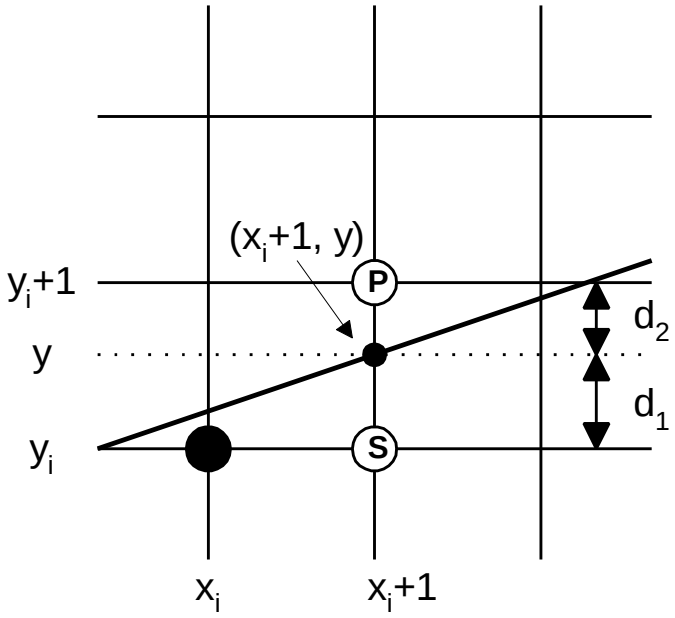
- Cài đặt minh họa thuật toán DDA

```
#define Round(a)  int(a+0.5)
int Color = GREEN;

void LineDDA (int x1, int y1, int x2, int y2)
{
    int x = x1;
    float y = y1;
    float m = float(y2-y1)/(x2-x1);

    putpixel(x, Round(y), Color);
    for(int i=x1; i<x2; i++)
    {
        x++;
        y +=m;
        putpixel(x, Round(y), Color);
    }
} // LineDDA
```

Thuật toán Bresenham



- Gọi $(x_i + 1, y)$ là điểm thuộc đoạn thẳng. Ta có: $y = m(x_i + 1) + b$.

  $d_1 = y - y_i$

- Đặt $d_2 = (y_i + 1) - y$

- Xét tất cả các vị trí tương đối của y so với y_i và $y_i + 1$, việc chọn điểm (x_{i+1}, y_{i+1}) là S hay P phụ thuộc vào việc so sánh d_1 và d_2 hay dấu của $d_1 - d_2$:

  - Nếu $d_1 - d_2 < 0$, ta sẽ chọn điểm S, tức là $y_{i+1} = y_i$.
  - Ngược lại, nếu $d_1 - d_2 \geq 0$, ta sẽ chọn điểm P, tức là $y_{i+1} = y_i + 1$

- Xét $p_i = Dx(d_1 - d_2) = Dx(2y - 2y_i - 1)$

  $\Rightarrow p_i = Dx[2(m(x_i + 1) + b) - 2y_i - 1]$

- Thay $m = \frac{Dy}{Dx}$ vào phương trình trên ta được : $p_i = 2Dyx_i - 2Dxy_i + c$, với $c = 2Dy + (2b-1)Dx$.
- Nhận xét rằng nếu tại bước thứ i ta xác định được dấu của p_i thì xem như ta xác định được điểm cần chọn ở bước (i+1).
- Ta có :

$$p_{i+1} - p_i = (2Dyx_{i+1} - 2Dxy_{i+1} + c) - (2Dyx_i - 2Dxy_i + c)$$

$$\Leftrightarrow p_{i+1} - p_i = 2Dy(x_{i+1} - x_i) - 2Dx(y_{i+1} - y_i)$$

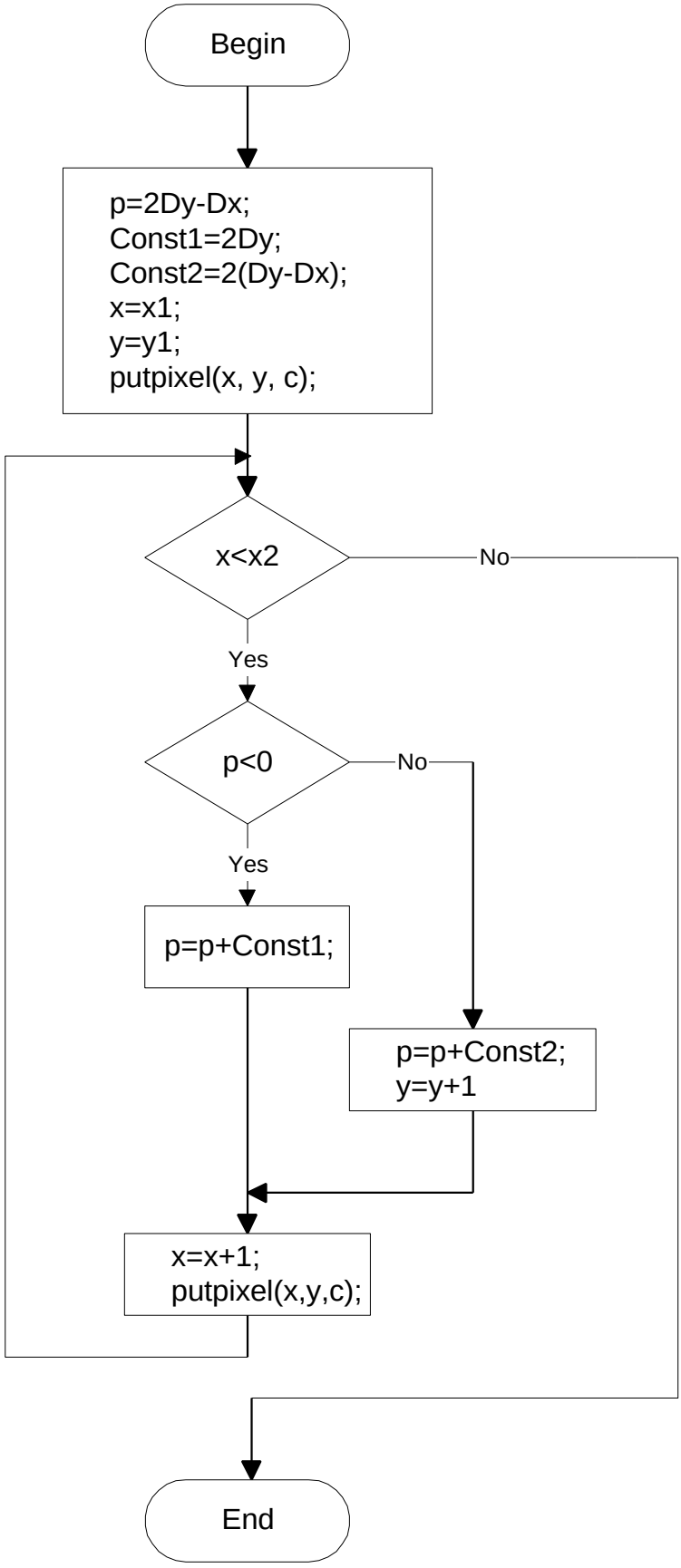
$$\Leftrightarrow p_{i+1} - p_i = 2Dy - 2Dx(y_{i+1} - y_i), \text{ do } x_{i+1} = x_i + 1$$

- Từ đây ta có thể suy ra cách tính p_{i+1} từ p_i như sau :
  - Nếu $p_i < 0$ thì $p_{i+1} = p_i + 2Dy$ do ta chọn $y_{i+1} = y_i$.
  - Ngược lại, nếu $p_i \geq 0$, thì $p_{i+1} = p_i + 2Dy - 2Dx$, do ta chọn $y_{i+1} = y_i + 1$.
- Giá trị p_0 được tính từ điểm vẽ đầu tiên (x_0, y_0) theo công thức :

$$p_0 = 2Dyx_0 - 2Dxy_0 + c = 2Dyx_0 - 2Dxy_0 + 2Dy - (2b-1)Dx$$

- Do (x_0, y_0) là điểm nguyên thuộc về đoạn thẳng nên ta có $y_0 = mx_0 + b = \frac{Dy}{Dx}x_0 + b$. Thế vào phương trình trên ta suy ra : $p_0 = 2Dy - Dx$.

Lưu đồ thuật toán Bresenham



- Ví dụ : Cho A(12, 20) và B(22, 27),
- Ta có
  - Dx = 22-12 = 10, Dy=27-20=7
  - Const1 = 2Dy = 14, Const2 = 2(Dy – Dx) = -6
  - p_0 = 2Dy – Dx = 14-10 = 4

i	x_i	y_i	p_i
0	12	20	4
1	13	21	-2
2	14	21	12
3	15	22	6
4	16	23	0
5	17	24	-6
6	18	24	8
7	19	25	2
8	20	26	-4
9	21	26	10
10	22	27	4

- Nhận xét
  - Thuật toán Bresenham chỉ làm việc trên số nguyên và các thao tác trên số nguyên chỉ là phép cộng và phép dịch bit (phép nhân 2) điều này là một cải tiến làm tăng tốc độ đáng kể so với thuật toán DDA. Ý tưởng chính của thuật toán nằm ở chỗ xét dấu p_i để quyết định điểm kế tiếp, và sử dụng công thức truy hồi $p_{i+1} - p_i$ để tính p_i bằng các phép toán đơn giản trên số nguyên.
  - Thuật toán này cho kết quả tương tự như thuật toán DDA.

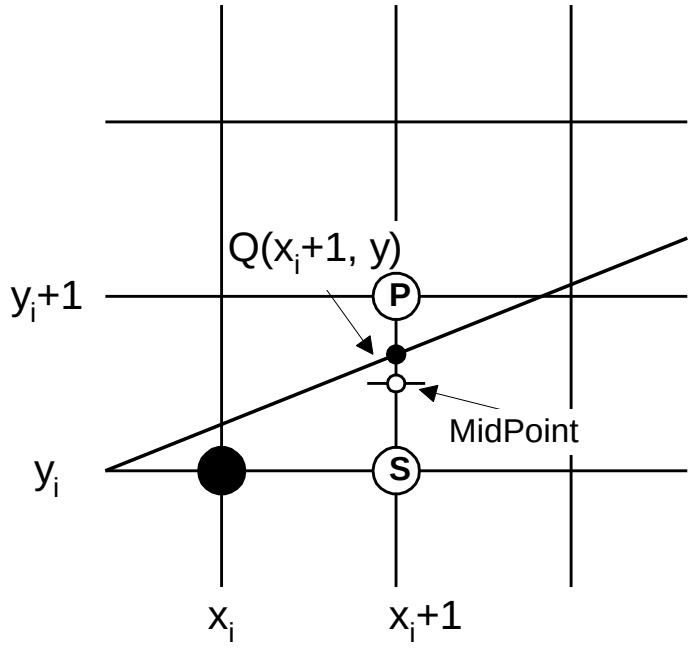
- Cài đặt minh họa thuật toán Bresenham

```
void LineBres (int x1, int y1, int x2, int y2)
{
    int Dx, Dy, p, Const1, Const2;
    int x, y;

    Dx     = x2 - x1;
    Dy     = y2 - y1;
    p   = 2*Dy - Dx; // Dy <<1 - Dx
    Const1 = 2*Dy; // Dy <<1
    Const2 = 2*(Dy-Dx); // (Dy-Dx) <<1
    x = x1;
    y = y1;
    putpixel(x, y, Color);
    for(i=x1; i<x2; i++)
    {
        if (p<0)
            p += Const1;
        else
        {
            p += Const2;
            y++;
        }
        x++;
        putpixel(x, y, Color);
    }
} // LineBres
```

Thuật toán MidPoint

- Thuật toán MidPoint đưa ra cách chọn y_{i+1} là y_i hay $y_i + 1$ bằng cách so sánh điểm thực $Q(x_i + 1, y)$ với điểm MidPoint là trung điểm của S và P. Ta có :
  - Nếu điểm Q nằm dưới điểm MidPoint, ta chọn S.
  - Nếu điểm Q nằm trên điểm MidPoint ta chọn P.



- Ta có dạng tổng quát của phương trình đường thẳng :

$Ax + By + C = 0$ với $A = y_2 - y_1, B = -(x_2 - x_1), C = x_2 y_1 - x_1 y_2$

- Đặt $F(x, y) = Ax + By + C$, ta có nhận xét :

$$F(x,y)\begin{cases} < 0, \text{nếu } (x,y) \text{ nằm phía trên đường thẳng} \\ = 0, \text{nếu } (x,y) \text{ thuộc về đường thẳng} \\ > 0, \text{nếu } (x,y) \text{ nằm phía dưới đường thẳng.} \end{cases}$$

- Lúc này việc chọn các điểm S, P ở trên được đưa về việc xét dấu của $p_i = 2F(\text{MidPoint}) = 2F\left(x_i + 1, y_i + \frac{1}{2}\right)$.
  - Nếu $p_i < 0$, điểm MidPoint nằm phía trên đoạn thẳng. Lúc này điểm thực Q nằm dưới điểm MidPoint nên ta chọn S, tức là $y_{i+1} = y_i$.
  - Ngược lại, nếu $p_i \geq 0$, điểm MidPoint nằm phía dưới đoạn thẳng. Lúc này điểm thực Q nằm trên điểm MidPoint nên ta chọn P, tức là $y_{i+1} = y_i + 1$.
- Mặt khác :

$$p_{i+1} - p_i = 2F\left(x_{i+1} + 1, y_{i+1} + \frac{1}{2}\right) - 2F\left(x_i + 1, y_i + \frac{1}{2}\right)$$

$$\Leftrightarrow p_{i+1} - p_i = 2\left[A(x_{i+1} + 1) + B\left(y_{i+1} + \frac{1}{2}\right) + C\right] - 2\left[A(x_i + 1) + B\left(y_i + \frac{1}{2}\right) + C\right]$$

$$\Leftrightarrow p_{i+1} - p_i = 2A + 2B(y_{i+1} - y_i) = 2Dy - 2Dx(y_{i+1} - y_i)$$

- Như vậy :
  - $p_{i+1} = p_i + 2Dy$, nếu $p_i < 0$ do ta chọn $y_{i+1} = y_i$.
  - $p_{i+1} = p_i + 2Dy - 2Dx$, nếu $p_i \geq 0$ do ta chọn $y_{i+1} = y_i + 1$.
- Ta tính giá trị p_0 ứng với điểm ban đầu (x_0, y_0), với nhận xét rằng (x_0, y_0) là điểm thuộc về đoạn thẳng, tức là có : $Ax_0 + By_0 + C = 0$

$$p_0 = 2F\left(x_0 + 1, y_0 + \frac{1}{2}\right) = 2\left[A(x_0 + 1) + B\left(y_0 + \frac{1}{2}\right) + C\right]$$

$$\Rightarrow p_0 = 2(Ax_0 + By_0 + C) + 2A + B = 2A + B = 2Dy - Dx$$

Câu hỏi kiểm tra

- Xét thuật toán Bresenham, với cách đặt d_1 và d_2 như trên, có khi nào d_1 hay d_2 âm hay không ? Cho ví dụ minh họa.

- Tại sao phải so sánh giá trị p_i với 0 trong các thuật toán MidPoint và Bresenham, bản chất của việc so sánh này là gì ?

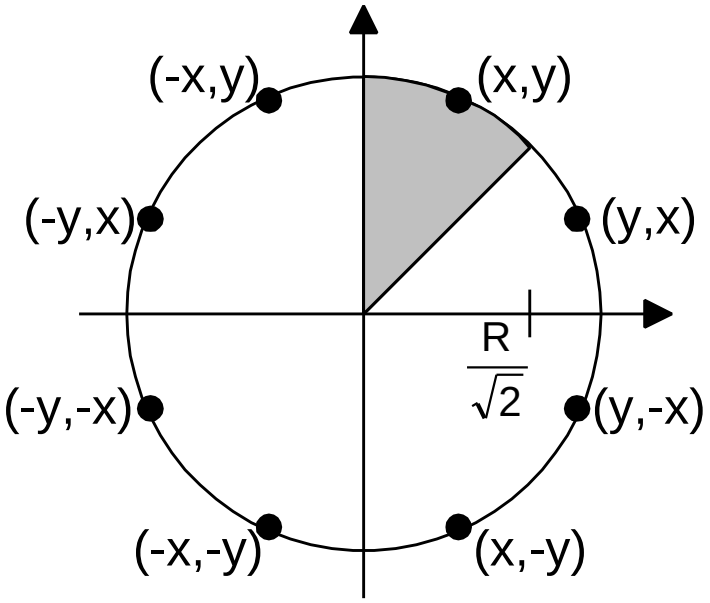
- Tại sao phải nhân F(MidPoint) với 2 khi gán cho p_i theo công thức p_i=2*F(MidPoint) ?

- Cài đặt thuật toán cho trường hợp $0 \le m \le 1$, Dx<0. Ta sử dụng thuật toán với trường hợp $0 \le m \le 1$, Dx>0 đã cài đặt cộng thêm một số thay đổi sau :
  - Thay biểu thức x=x+1 bằng x=x-1 và y=y+1 bằng y=y-1 vì trong trường hợp này x và y đều giảm dần.
  - Nhận xét rằng khi p<0 ta gán p=p+Const1, như vậy để hướng đến sự cân bằng Const1 phải là một giá trị dương. Tương tự như vậy, khi p≥0 ta gán p=p+Const2, Const2 phải là giá trị âm.
  - Từ nhận xét trên, trong các công thức ta sẽ thay Dx bằng abs(Dx), Dy bằng abs(Dy).
- Mở rộng thuật toán trên để vẽ đoạn thẳng trong trường hợp m bất kì.
  - Trường hợp đặc biệt $m=\infty$: Đoạn thẳng song song trục tung nên rất đơn giản khi vẽ.
  - Trường hợp $-1 \le m \le 1$: Sử dụng các công thức của thuật toán vẽ trong trường hợp $0 \le m \le 1$, Dx>0 và thay đổi một số điểm sau :
    - Nếu Dx<0 thì bước nhảy của x sẽ thay bằng −1. Tương tự nếu Dy<0, bước nhảy của y cũng sẽ là −1.
    - Thay Dx bằng abs(Dx), Dy=abs(Dy) trong tất cả các công thức có chứa Dx, Dy.
  - Trường hợp $m \le -1$ hay $m \ge 1$:
    - Thay đổi vai trò của x và y, nghĩa là thay x bằng y, y bằng x, Dx bằng Dy, Dy bằng Dx trong tất cả các công thức.
    - Thực hiện nguyên tắc về bước nhảy, thay đổi công thức Dx, Dy như trong trường hợp $-1 \le m \le 1$

Vẽ đường tròn bằng thuật toán MidPoint

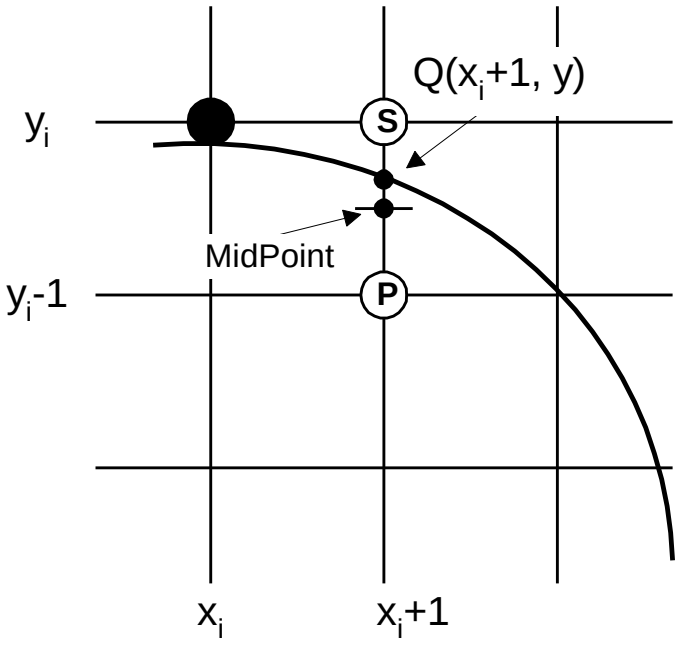
- Do tính đối xứng của đường tròn (C) nên ta chỉ cần vẽ cung ($C^{1/8}$) là cung 1/8 đường tròn, sau đó lấy đối xứng. Cung ($C^{1/8}$) được mô tả như sau (cung của phần tô xám trong hình vẽ) :

$$\begin{cases} 0 \le x \le R\dfrac{\sqrt{2}}{2} \\ R\dfrac{\sqrt{2}}{2} \le y \le R \end{cases}$$



- Như vậy nếu có (x, y) ∈ ($C^{1/8}$) thì các điểm : (y, x), (y,-x), (x,-y), (-x,-y), (-y,-x), (-y,x), (-x,y) sẽ thuộc (C).

- Chọn điểm bắt đầu để vẽ là điểm (0,R).
- Dựa vào hình vẽ, nếu (x_i, y_i) là điểm nguyên đã tìm được ở bước thứ i, thì điểm (x_{i+1}, y_{i+1}) ở bước thứ (i+1) là sự lựa chọn giữa S và P.
- Như vậy : $\begin{cases} x_{i+1} = x_i + 1 \\ y_{i+1} \in \{y_i, y_i - 1\} \end{cases}$



- Đặt $F(x,y) = x^2 + y^2 - R^2$, ta có :

$$F(x,y)\begin{cases} < 0, \text{nếu } (x,y) \text{ nằm trong đường tròn} \\ = 0, \text{nếu } (x,y) \text{ nằm trên đường tròn} \\ > 0, \text{nếu } (x,y) \text{ nằm ngoài đường tròn.} \end{cases}$$

- Xét $p_i = F(\text{MidPoint}) = F\left(x_i + 1, y_i - \frac{1}{2}\right)$. Ta có :
  - Nếu $p_i < 0$, điểm MidPoint nằm trong đường tròn. Lúc này điểm thực Q gần S hơn nên ta chọn S, tức là $y_{i+1} = y_i$.
  - Ngược lại, nếu $p_i \geq 0$, điểm MidPoint nằm ngoài đường tròn. Lúc này điểm thực Q gần P hơn nên ta chọn P, tức là $y_{i+1} = y_i - 1$.
- Mặt khác :

$$p_{i+1} - p_i = F\left(x_{i+1} + 1, y_{i+1} - \frac{1}{2}\right) - F\left(x_i + 1, y_i - \frac{1}{2}\right)$$

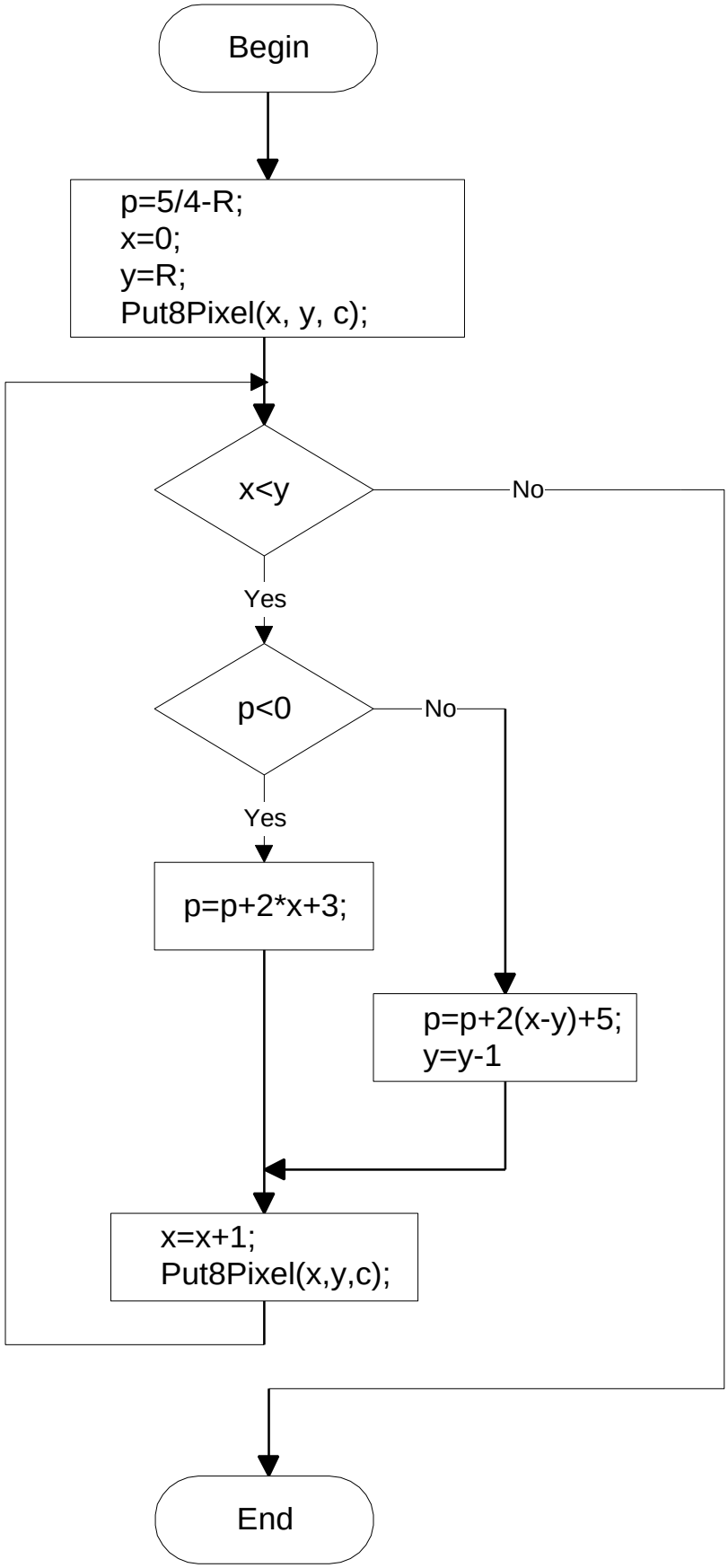
$$\Leftrightarrow p_{i+1} - p_i = \left[(x_{i+1} + 1)^2 + \left(y_{i+1} - \frac{1}{2}\right)^2 - R^2\right] - \left[(x_i + 1)^2 + \left(y_i - \frac{1}{2}\right)^2 - R^2\right]$$

$$\Leftrightarrow p_{i+1} - p_i = 2x_i + 3 + \left(y_{i+1}^2 - y_i^2\right) - \left(y_{i+1} - y_i\right)$$

- Vậy :
  - $p_{i+1} = p_i + 2x_i + 3$, nếu $p_i < 0$ do ta chọn $y_{i+1} = y_i$.
  - $p_{i+1} = p_i + 2x_i - 2y_i + 5$, nếu $p_i \geq 0$ do ta chọn $y_{i+1} = y_i - 1$.
- p_0 ứng với điểm ban đầu $(x_0, y_0) = (0, R)$.

$$p_0 = F\left(x_0 + 1, y_0 - \frac{1}{2}\right) = F\left(1, R - \frac{1}{2}\right) = \frac{5}{4} - R$$

Lưu đồ thuật toán MidPoint vẽ đường tròn



Cài đặt minh họa thuật toán MidPoint vẽ đường tròn

```
void CircleMidPoint (int R)
{
    int x, y;

    x = 0;
    y = R;
    Put8Pixel(x, y);
    p = 1 - R; // 5/4-R
    while (x < y)
    {
        if (p < 0)
            p += 2*x + 3;
        else
        {
            p += 2*(x -y) + 5;
            y--;
        }

        x++;
        Put8Pixel(x, y);
    }

} // CircleMidPoint
```

- Ví dụ : Vẽ đường tròn tâm I(0,0), bán kính R=15.

i	x_i	y_I	p_i		Delta1	Delta2
0	0	15	-14	1-15	3	-25
1	1	15	-11	-14+2*(0)+3	5	-23
2	2	15	-6	-11+2*(1)+3	7	-21
3	3	15	1	-6+2*(2)+3	9	-19
4	4	14	-18	1+2*(3-15)+5	11	-15
5	5	14	-7	-18+2*(4)+3	13	-13
6	6	14	6	-7+2*(5)+3	15	-11
7	7	13	-5	6+2(6-14)+5	17	-7
8	8	13	12	-5+2(7)+3	19	-5
9	9	12	7	12+2(8-13)+5	21	-1
10	10	11	6	7+2(9-12)+5	23	3
11	11	10	9	6+2(10-11)+5	25	7

Nhận xét :

- Nếu đặt Delta1 = 2*x+3, Delta2 = 2*(x-y)+5 thì
  - Do mỗi bước đều tăng x nên sau mỗi lần lặp giá trị Delta1 luôn tăng 2.
  - Do y bị giảm 1 khi gặp p≥0 và giữ nguyên giá trị trong trường hợp ngược lại nên nếu lần lặp trước giá trị p≥0 thì giá trị Delta2 sẽ được tăng 4 và nếu lần lặp trước giá trị p<0 thì giá trị Delta2 sẽ được tăng 2 mà thôi.
- Hãy tối ưu hóa cài đặt thuật toán MidPoint vẽ đường tròn từ nhận xét trên.

Vẽ đường conics và một số đường cong khác

Phương trình tổng quát của các đường conics có dạng : $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$. Giá trị của các hằng số A, B, C, D, E, F sẽ quyết định dạng của đường conics, cụ thể là nếu:

$$B^2 - 4AC \begin{cases} < 0, \text{ dạng đường tròn (nếu A = C và B = 0) hay ellipse} \\ = 0, \text{ dạng parabol} \\ > 0, \text{ dạng hyperbol.} \end{cases}$$

Ta sẽ áp dụng ý tưởng của thuật toán MidPoint để vẽ các đường conics và một số đường cong khác, theo các bước tuần tự sau:

- ***Bước 1 :*** Dựa vào dáng điệu và phương trình đường cong, để xem thử có thể rút gọn phần đường cong cần vẽ hay không.
- ***Bước 2 :*** Tính đạo hàm để từ đó phân thành các vùng vẽ :
  - Nếu $0 \le f'(x) \le 1$ thì $\begin{cases} x_{i+1} = x_i + 1 \\ y_{i+1} \in \{y_i, y_i + 1\} \end{cases}$ (*)
  - Nếu $-1 \le f'(x) \le 0$ thì $\begin{cases} x_{i+1} = x_i + 1 \\ y_{i+1} \in \{y_i, y_i - 1\} \end{cases}$ (*)
  - Nếu $f'(x) > 1$ thì $\begin{cases} y_{i+1} = y_i + 1 \\ x_{i+1} \in \{x_i, x_i + 1\} \end{cases}$ (*)
  - Nếu $f'(x) < -1$ thì $\begin{cases} y_{i+1} = y_i + 1 \\ x_{i+1} \in \{x_i, x_i - 1\} \end{cases}$ (*)

- ***Bước 3 :*** Xác định công thức của p_i cho từng trường hợp để quyết định (*) dựa trên dấu của p_i. p_i thường là hàm được xây dựng từ phương trình đường cong để cho $p_i = 0$ nếu (x_i, y_i) thuộc về đường cong. Việc chọn p_i cần phải chú ý sao cho thao tác tính p_i sau này hạn chế phép toán trên số thực.

- ***Bước 4 :*** Tìm mối liên quan của p_{i+1} và p_i bằng cách xét hiệu $p_{i+1} - p_i$.

- ***Bước 5 :*** Tính p_0 và hoàn chỉnh thuật toán.

Bài tập

- Giải thích tại sao chỉ chọn cung 1/8 đường tròn để vẽ rồi lấy đối xứng mà không mở rộng cho cung 1/16 hay 1/32.

- Giải thích tại sao có thể thay công thức p_0=5/4-R bằng công thức p_0=1-R khi cài đặt.

- Hãy trình bày thuật toán MidPoint vẽ cung 1/8 đường tròn sau :

$$\begin{cases} R\dfrac{\sqrt{2}}{2} \le x \le R \\ 0 \le y \le R\dfrac{\sqrt{2}}{2} \end{cases}$$

- Áp dụng các bước trên để vẽ đoạn thẳng trong trường hợp tổng quát.

- Hãy trình bày khung chính của thuật toán vẽ ellipse, parabol, hyperbol dựa vào các bước trên.