

计算几何 第二次作业

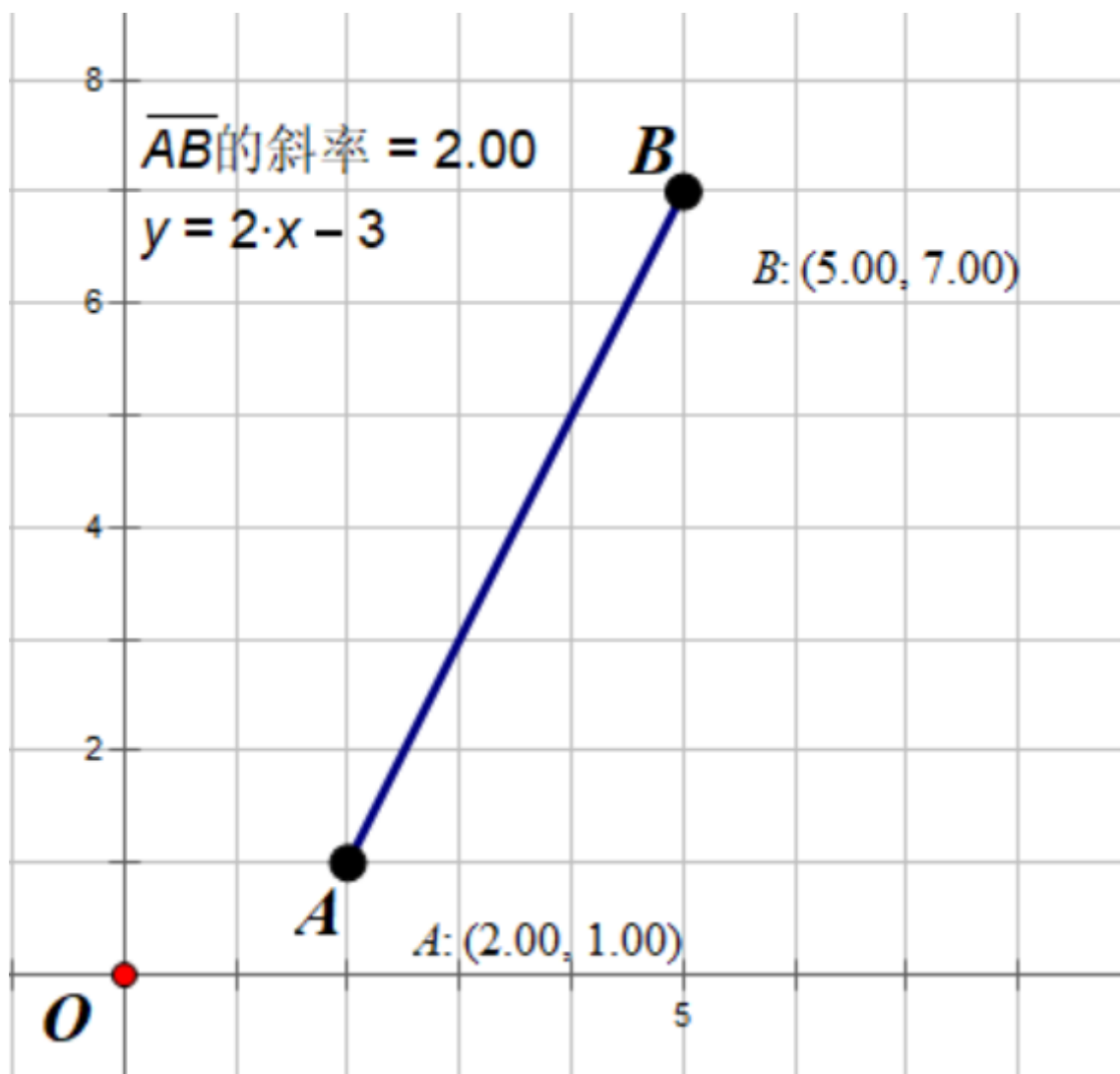
数学三班 李岳锴 200810301

第三章 基本图形生成算法

1. 用中点画线法和Bresenham画线法画线段：起点(2, 1)，终点(5, 7)

(1) 中点画线法：

设 $A(2, 1), B(5, 7)$ ，首先我们求得 AB 两点所在直线的方程： $y = 2x - 3$



注意到线段 AB 的斜率大于1，因此我们首先给出线段斜率大于1的中点画线算法流程：

1. 初始化。令 $a = y_1 - y_2, b = x_2 - x_1, d = a + 2b, x = x_1, y = y_1$
2. 使用指定颜色color绘制像素 (x, y)
3. 判断 y 是否小于 y_2 。如果 $y < y_2$ ，则执行步骤4，否则算法结束

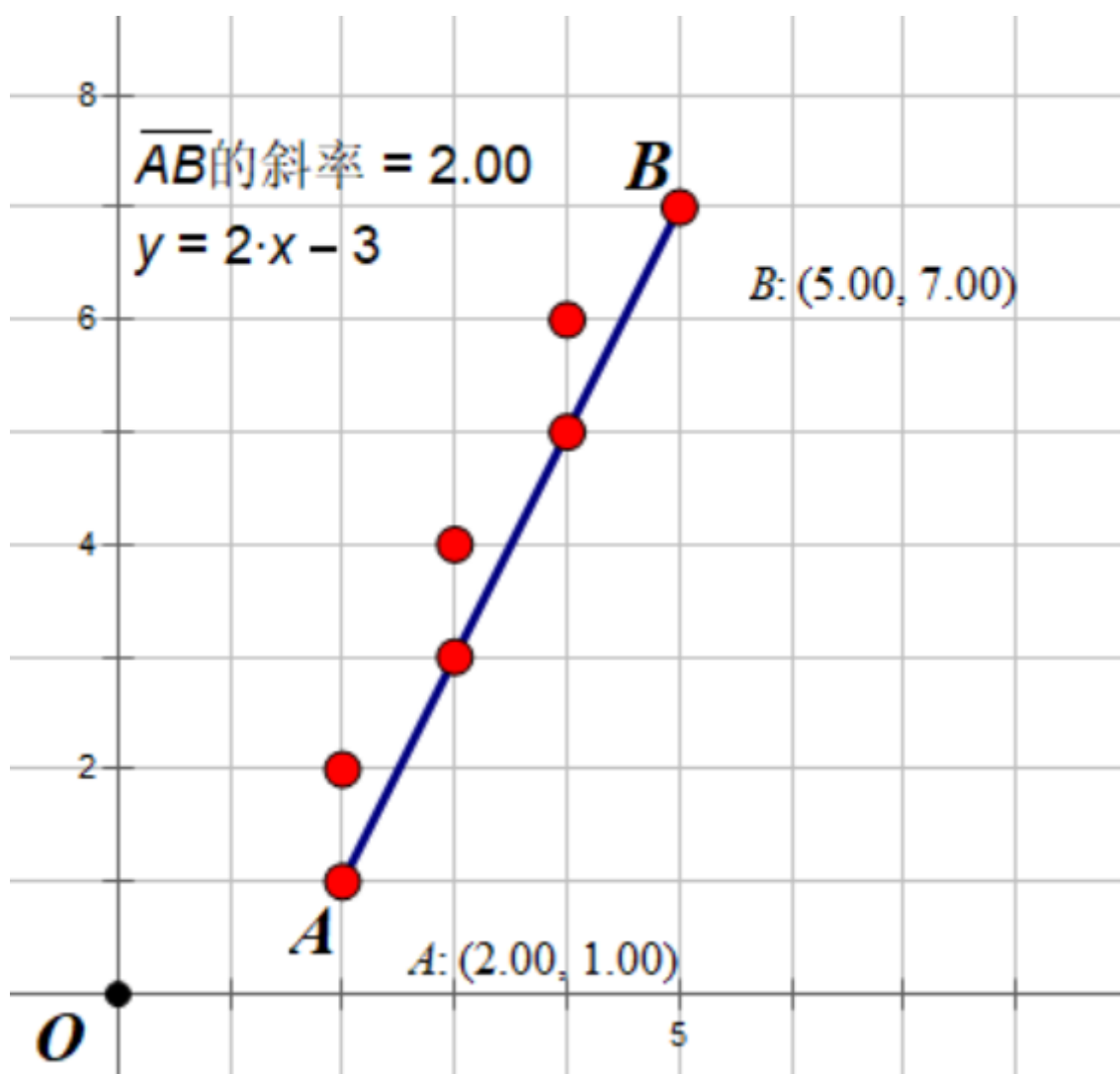
4. 如果 $d > 0$, 则执行 $x = x + 1, y = y + 1, d = d + 2(a + b)$,
否则执行 $y = y + 1, d = d + 2b$
5. 使用指定颜色color绘制像素 (x, y) , 并转至步骤3

迭代过程如下:

初始化: $a = -6, b = 3, d = 0, x = 2, y = 1$, 绘制像素点 $(2, 1)$

- ① $d = 0$, 令 $y = 2, d = 6$, 绘制像素点 $(2, 2)$
- ② $d = 6$, 令 $x = 3, y = 3, d = 0$, 绘制像素点 $(3, 3)$
- ③ $d = 0$, 令 $y = 4, d = 6$, 绘制像素点 $(3, 4)$
- ④ $d = 6$, 令 $x = 4, y = 5, d = 0$, 绘制像素点 $(4, 5)$
- ⑤ $d = 0$, 令 $y = 6, d = 6$, 绘制像素点 $(4, 6)$
- ⑥ $d = 6$, 令 $x = 5, y = 7, d = 0$, 绘制像素点 $(5, 7)$

此时 $y = y_2$, 到达终点。绘制的示意图如下:



(2) Bresenham画线法:

由(1)可知, 该直线位于第二八分圆域内。因此, 我们给出第二八分圆域内的Bresenham画线算法流程:

1. 初始化。令 $e = 2|\Delta x| - 2|\Delta y|$
2. 使用指定颜色color绘制像素 (x, y)

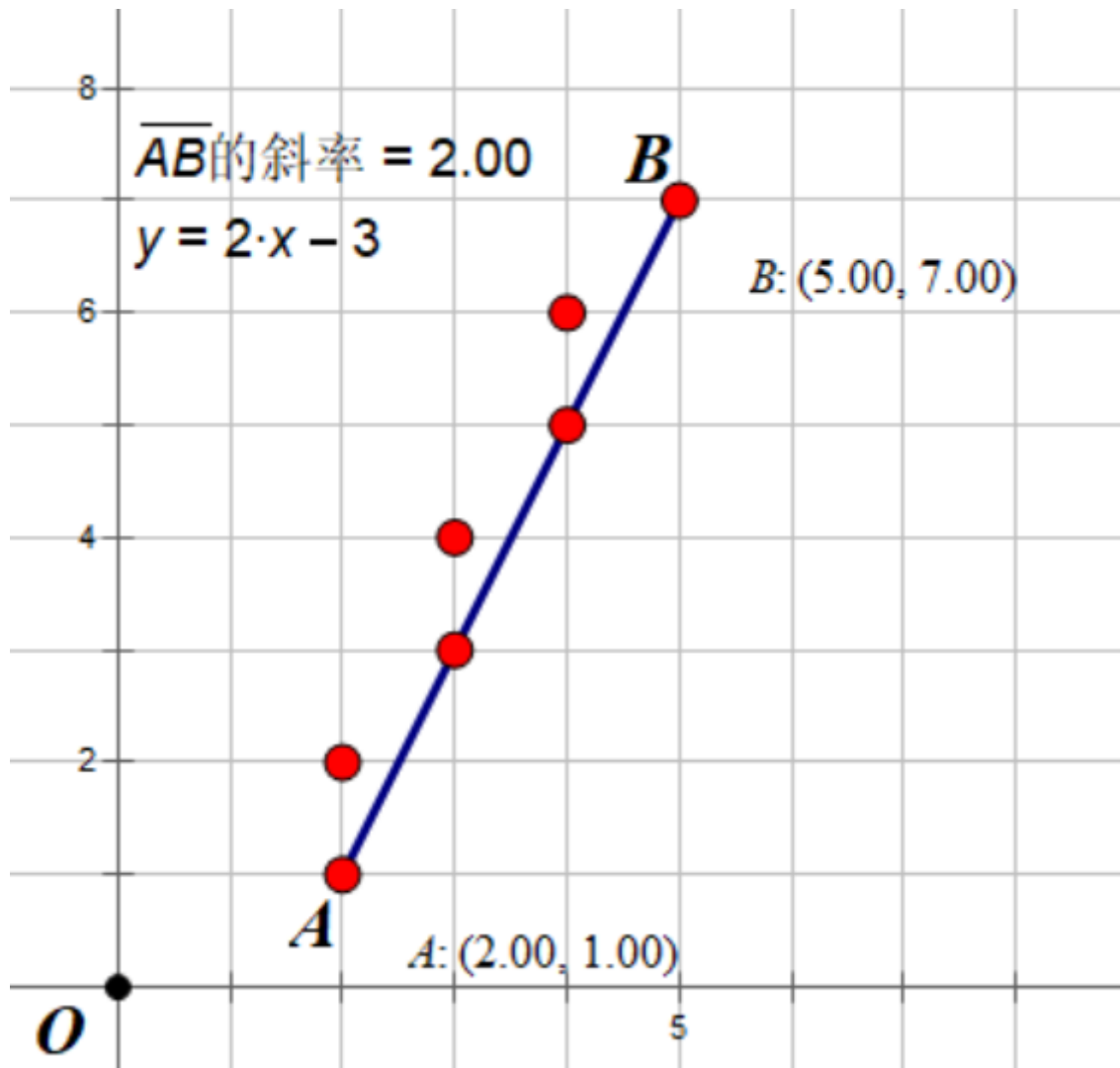
3. 判断 y 是否小于 y_2 。如果 $y < y_2$ ，则执行步骤4，否则算法结束
4. 如果 $e < 0$ ，则执行 $y = y + 1, e = e + 2|\Delta x|$ ，
否则执行 $x = x + 1, y = y + 1, e = e + 2|\Delta x| - 2|\Delta y|$
5. 使用指定颜色color绘制像素 (x, y) ，并转至步骤3

迭代过程如下：

初始化： $e = -6, \Delta x = 3, \Delta y = 6$ ，绘制像素点 $(2, 1)$

- ① $e = -6$ ，令 $y = 2, e = 0$ ，绘制像素点 $(2, 2)$
- ② $e = 0$ ，令 $x = 3, y = 3, d = 0$ ，绘制像素点 $(3, 3)$
- ③ $e = -6$ ，令 $y = 4, d = 6$ ，绘制像素点 $(3, 4)$
- ④ $e = 0$ ，令 $x = 4, y = 5, d = 0$ ，绘制像素点 $(4, 5)$
- ⑤ $e = -6$ ，令 $y = 6, d = 6$ ，绘制像素点 $(4, 6)$
- ⑥ $e = 0$ ，令 $x = 5, y = 7, d = 0$ ，绘制像素点 $(5, 7)$

此时 $y = y_2$ ，到达终点。绘制的示意图如下：



2. 用中点画圆法和Bresenham画圆法画下述圆在第一象限（屏幕，含坐标为0在坐标轴上的点）上的部分：圆心 $(2, 1)$ ，半径为6

(1) 中点画圆法：

算法流程如下：

1. 初始化。令初始点 $(x_0, y_0) = (0, r)$ ，初始判别式 $d = 1 - r$
2. 如果 $d < 0$ ，则取下一个像素为 $(x_n + 1, y_n)$ ，且 $d_{n+1} = d_n + 2x_n + 3$
否则令下一个像素为 $(x_n + 1, y_n - 1)$ ，且 $d_{n+1} = d_n + 2(x_n - y_n) + 5$
3. 确定 (x_{n+1}, y_{n+1}) 在其余7个八分圆中的对称点位置
4. 将计算出的每个像素位置 (x, y) 平移到圆心位于 (x_c, y_c) 的圆的轨迹上，即令 $x' = x + x_c, y' = y + y_c$
5. 重复步骤2-4，直到 $x \geq y$ 为止
6. 使用指定颜色color绘制像素点

迭代过程如下：

初始化： $(x_0, y_0) = (0, 6), d = -5$

① 由对称性，同时记录像素点 $(6, 0), (0, -6), (-6, 0)$

由平移，得到像素点 $(2, 7), (8, 1), (2, -5), (-4, 1)$

② 下一个像素点为 $(1, 6)$ ，令 $d = -2$ ，

由对称性，同时记录像素点： $(6, 1), (6, -1), (1, -6), (-1, -6), (-6, -1), (-6, 1), (-1, 6)$

由平移，得到像素点 $(3, 7), (8, 2), (8, 0), (3, -5), (1, -5), (-4, 0), (-4, 2), (1, 7)$ ：

③ 下一个像素点为 $(2, 6)$ ，令 $d = 3$ ，

由对称性，同时记录像素点： $(6, 2), (6, -2), (2, -6), (-2, -6), (-6, -2), (-6, 2), (-2, 6)$

由平移，得到像素点 $(4, 7), (8, 3), (8, -1), (4, -5), (0, -5), (-4, -1), (-4, 3), (0, 7)$ ：

④ 下一个像素点为 $(3, 5)$ ，令 $d = 0$ ，

由对称性，同时记录像素点： $(5, 3), (5, -3), (3, -5), (-3, -5), (-5, -3), (-5, 3), (-3, 5)$

由平移，得到像素点 $(5, 6), (7, 4), (7, -2), (5, -4), (-1, -4), (-3, -2), (-3, 4), (-1, 6)$ ：

⑤ 下一个像素点为 $(4, 4)$ ，此时 $x \geq y$ ，到达终点，故不再计算判别式

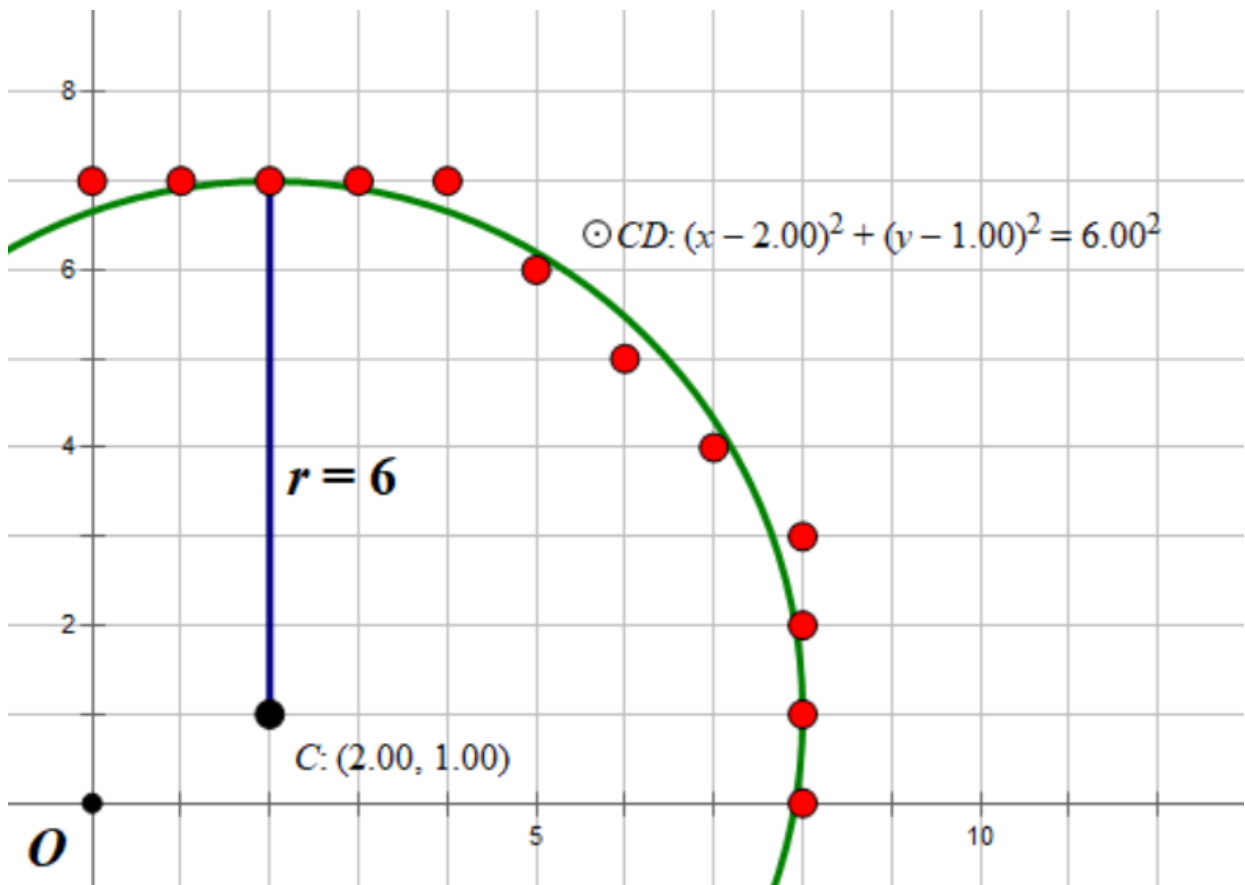
由对称性，同时记录像素点： $(4, -4), (-4, -4), (-4, 4)$

由平移，得到像素点 $(6, 5), (6, -3), (-2, -3), (-2, 5)$ ：

最终绘制出的第一象限像素点为：

$(2, 7), (8, 1), (3, 7), (8, 2), (8, 0), (1, 7), (4, 7), (8, 3), (0, 7), (5, 6), (7, 4), (6, 5)$

绘制的示意图如下：



(2) Bresenham画圆法:

首先定义一些判别式如下:

$$\Delta_n = (x_n + 1)^2 + (y_n - 1)^2 - r^2$$

$$\delta_{HD} = \left| (x_n + 1)^2 + y_n^2 - r^2 \right| - \left| (x_n + 1)^2 + (y_n - 1)^2 - r^2 \right| = 2(\Delta_n + y_n) - 1$$

$$\delta_{DV} = \left| (x_n + 1)^2 + (y_n - 1)^2 - r^2 \right| - \left| x_n^2 + (y_n - 1)^2 - r^2 \right| = 2(\Delta_n - x_n) - 1$$

算法流程如下:

1. 初始化。令初始点 $(x_0, y_0) = (0, r)$, 初始判别式 $\Delta = 2(1 - r)$
2. 当 $\Delta < 0$, 若 $\delta_{HD} < 0$, 则取下一个像素为 $H(x_n + 1, y_n)$, 否则取 $D(x_n + 1, y_n - 1)$;

当 $\Delta > 0$, 若 $\delta_{DV} < 0$, 则取下一个像素为 $D(x_n + 1, y_n - 1)$, 否则取 $V(x_n, y_n - 1)$;

当 $\Delta = 0$, 则取下一个像素为 $D(x_n + 1, y_n - 1)$
3. 当下一个像素为 H 点时, 令 $\Delta_{n+1} = \Delta_n + 2x_{n+1} + 1$;

当下一个像素为 D 点时, 令 $\Delta_{n+1} = \Delta_n + 2(x_{n+1} - y_{n+1} + 1)$;

当下一个像素为 V 点时, 令 $\Delta_{n+1} = \Delta_n - 2y_{n+1} + 1$
4. 确定 (x_{n+1}, y_{n+1}) 在其余7个八分圆中的对称点位置
5. 将计算出的每个像素位置 (x, y) 平移到圆心位于 (x_c, y_c) 的圆的轨迹上, 即令

$$x' = x + x_c, y' = y + y_c$$
6. 重复步骤2-5, 直到 $y \leq 0$ 为止
7. 使用指定颜色color绘制像素点

迭代过程如下:

初始化: $(x_0, y_0) = (0, 6), \Delta = -10$

① 由对称性, 同时记录像素点 $(0, -6)$

由平移, 得到像素点 $(2, 7), (2, -5)$

② 由于 $\delta_{HD} = -9 < 0$, 故下一个像素点为 $H(1, 6)$, 令 $\Delta = -7$,

由对称性, 同时记录像素点: $(1, -6), (-1, -6), (-1, 6)$

由平移, 得到像素点 $(3, 7), (3, -5), (1, -5), (1, 7)$:

③ 由于 $\delta_{HD} = -3 < 0$, 故下一个像素点为 $H(2, 6)$, 令 $\Delta = -2$,

由对称性, 同时记录像素点: $(2, -6), (-2, -6), (-2, 6)$

由平移, 得到像素点 $(4, 7), (4, -5), (0, -5), (0, 7)$:

④ 由于 $\delta_{HD} = 7 > 0$, 故下一个像素点为 $D(3, 5)$, 令 $\Delta = -4$,

由对称性, 同时记录像素点: $(3, -5), (-3, -5), (-3, 5)$

由平移, 得到像素点 $(5, 6), (5, -4), (-1, -4), (-1, 6)$:

⑤ 由于 $\delta_{HD} = 1 > 0$, 故下一个像素点为 $D(4, 4)$, 令 $\Delta = -2$,

由对称性, 同时记录像素点: $(4, -4), (-4, -4), (-4, 4)$

由平移, 得到像素点 $(6, 5), (6, -3), (-2, -3), (-2, 5)$:

⑥ 由于 $\delta_{HD} = 3 > 0$, 故下一个像素点为 $D(5, 3)$, 令 $\Delta = 4$,

由对称性, 同时记录像素点: $(5, -3), (-5, -3), (-5, 3)$

由平移, 得到像素点 $(7, 4), (7, -2), (-3, -2), (-3, 4)$:

⑦ 由于 $\delta_{DV} = -3 < 0$, 故下一个像素点为 $D(6, 2)$, 令 $\Delta = 14$,

由对称性, 同时记录像素点: $(6, -2), (-6, -2), (-6, 2)$

由平移, 得到像素点 $(8, -1), (-4, -1), (-4, 3), (-4, 3)$:

⑧ 由于 $\delta_{DV} = 15 > 0$, 故下一个像素点为 $V(6, 1)$, 令 $\Delta = 13$,

由对称性, 同时记录像素点: $(6, -1), (-6, -1), (-6, 1)$

由平移, 得到像素点 $(8, 2), (8, 0), (-4, 0), (-4, 2)$:

⑨ 由于 $\delta_{DV} = 13 > 0$, 故下一个像素点为 $V(6, 0)$, 此时 $y = 0$, 到达终点, 故不再计算判别式。

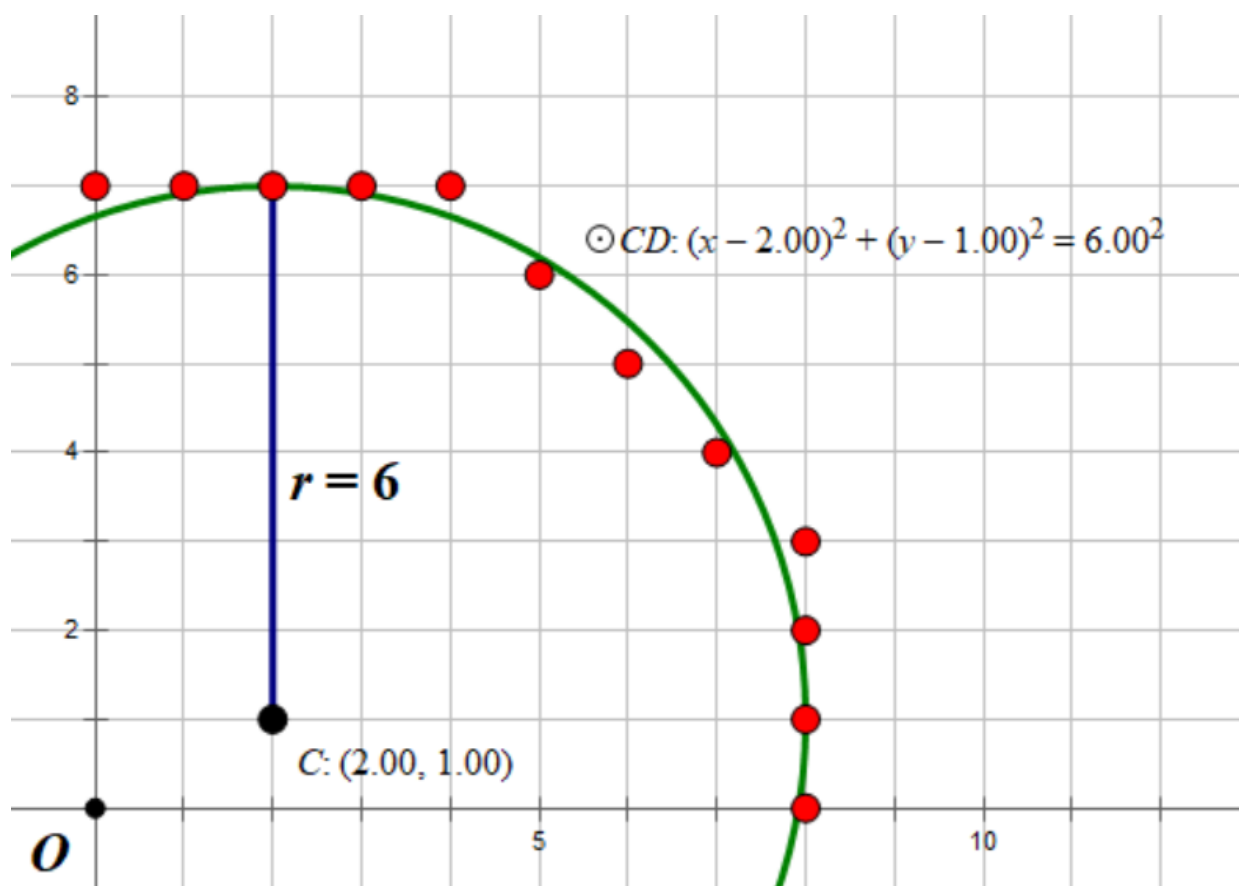
由对称性, 同时记录像素点: $(-6, 0)$

由平移, 得到像素点 $(-4, 1)$:

最终绘制出的第一象限像素点为:

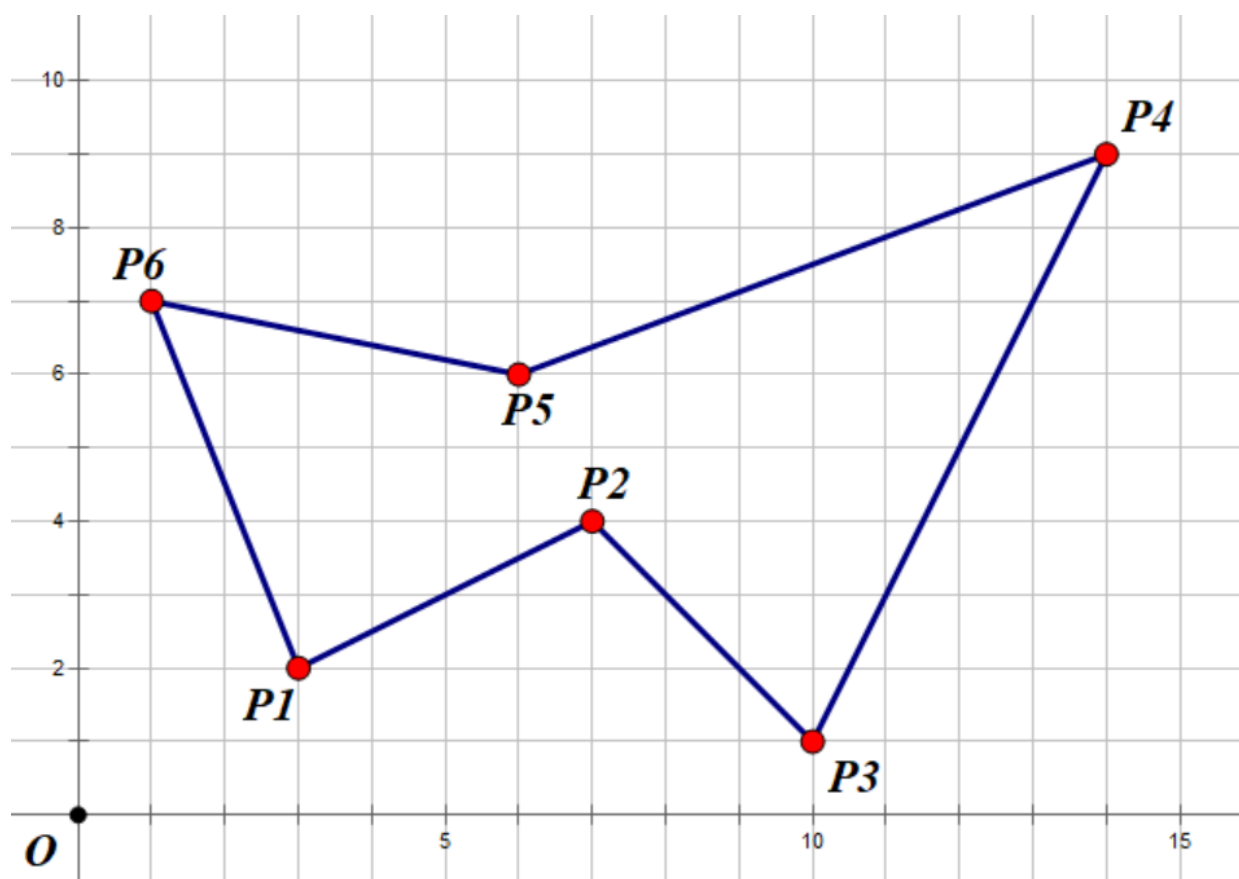
$(2, 7), (8, 1), (3, 7), (8, 2), (8, 0), (1, 7), (4, 7), (8, 3), (0, 7), (5, 6), (7, 4), (6, 5)$

绘制的示意图如下:

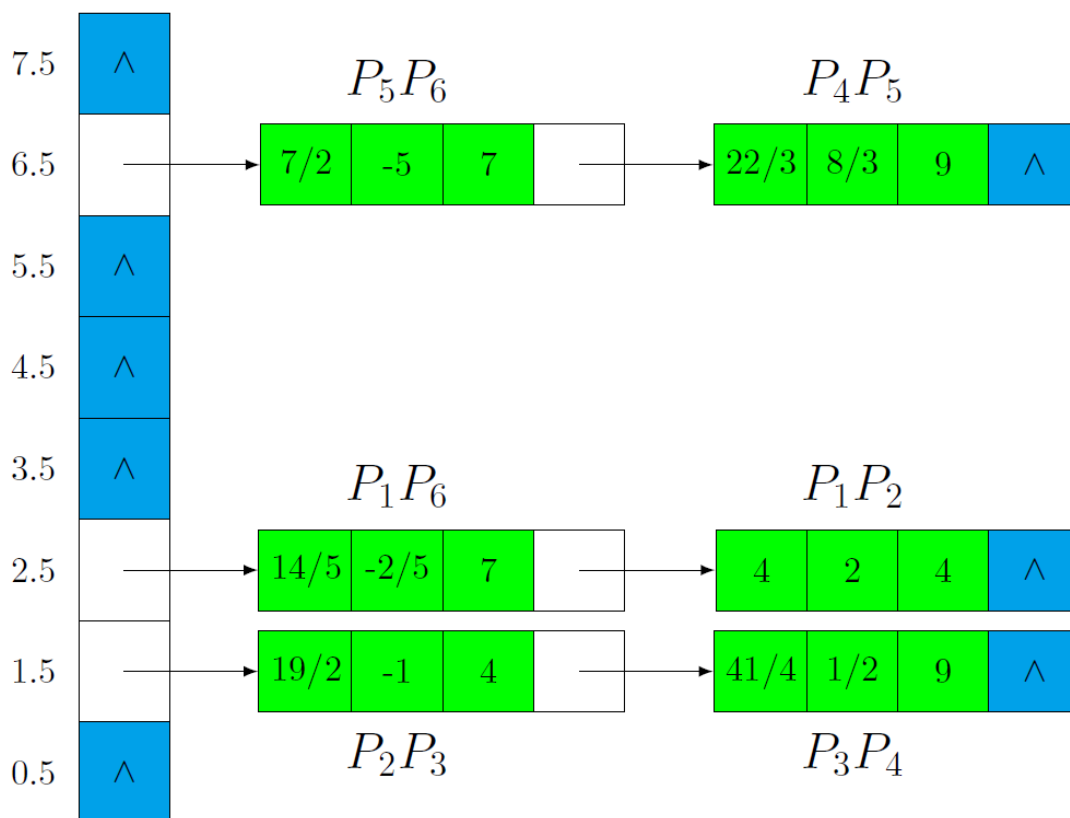


3. 多边形由下述顶点定义

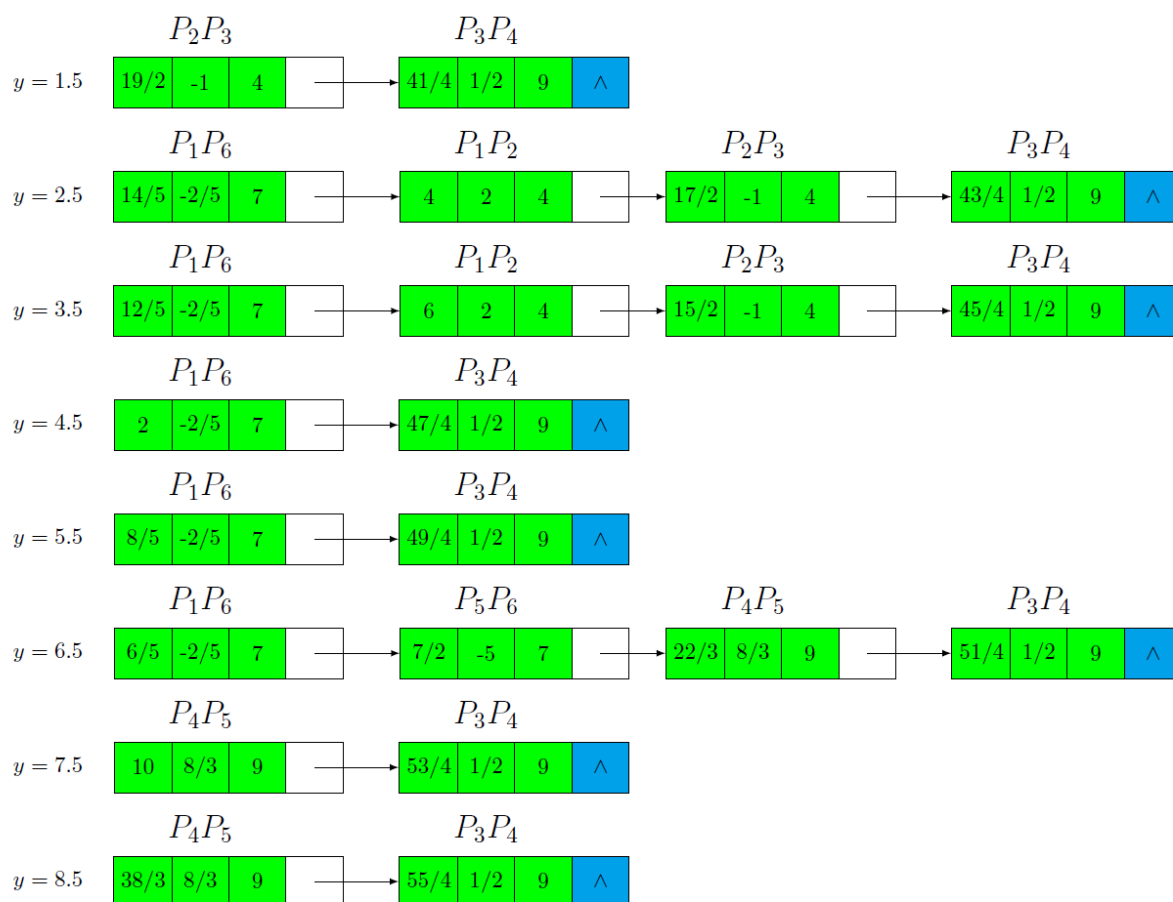
$\{(3, 2), (7, 4), (10, 1), (14, 9), (6, 6), (1, 7)\}$, 用活性边的方法 (扫描线填充) 填充, 给出新边表和活性边表。



(1) 新边表:

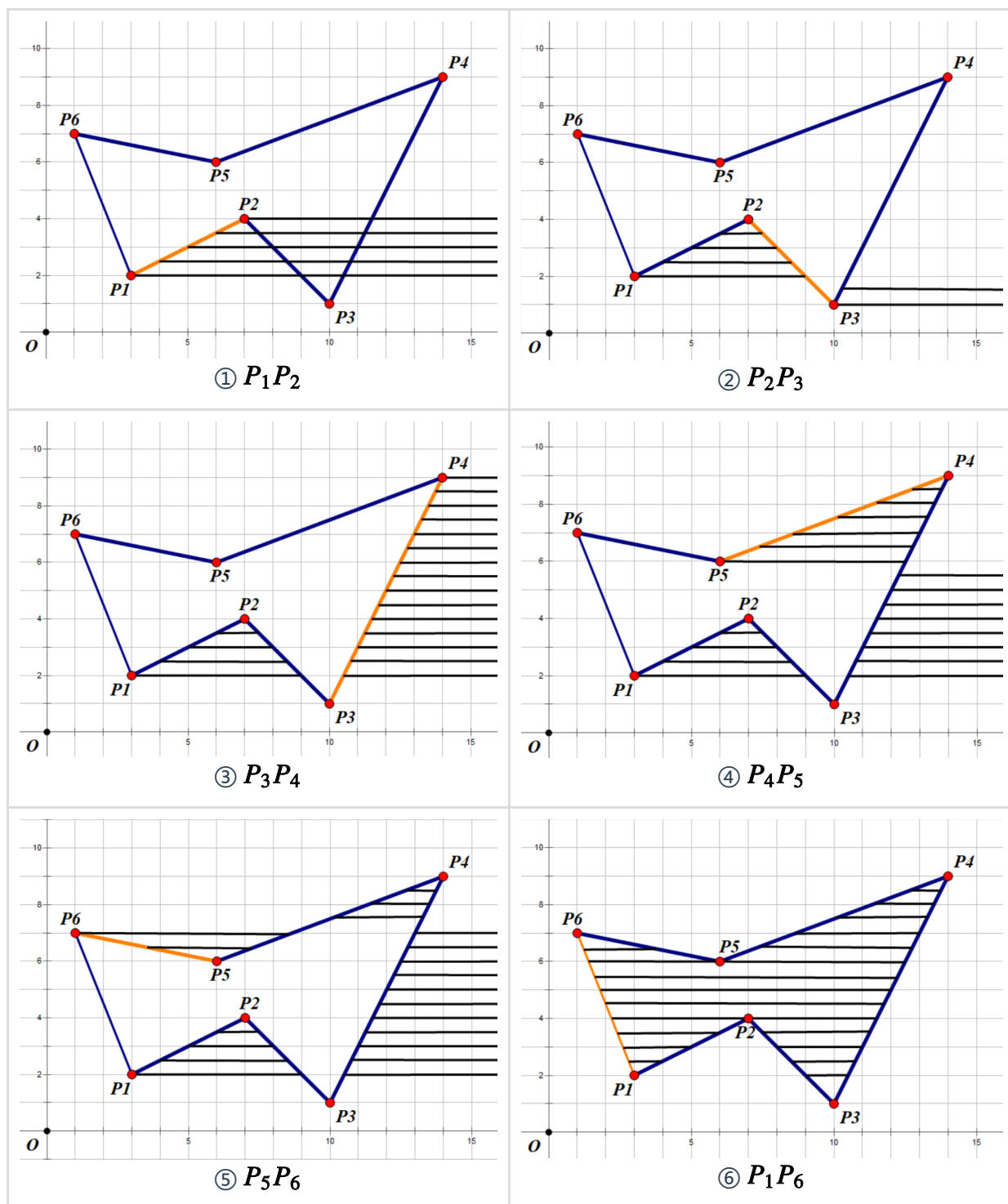


(2) 活性边表:



4. 对上述多边形，用边填充算法和栅栏填充算法填充，给出每步示意图。

(1) 边填充算法



(2) 栅栏填充算法

我们选取经过 P_2 的垂直于扫描线的直线作为栅栏线：

