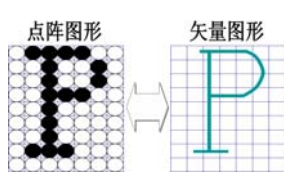
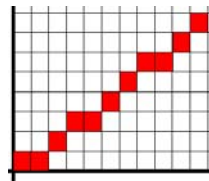


《计算机图形学》作业三

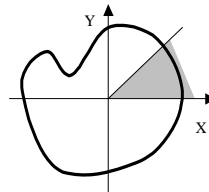
- 1、下图示出了图形的两种表示方法：矢量表示和点阵表示，试从两种图形表示方式相互转换的角度描述线画图形生成扫描转换算法的基本思想。
- 2、简要阐明 Bresenham 画线算法的基本原理，计算端点分别为(20,10)和(30,18)的直线从点(20,10)开始的像素生成位置。分析生成如下图所示线条“锯齿”现象的成因，并比较解决这一问题的主要策略。
- 3、如图所示的封闭曲线的隐式方程为 $f(x, y) = 0$ ，图中阴影部分曲线段的切线斜率绝对值大于 1，可采用中点算法绘制生成阴影区域内曲线上点。试推导生成该曲线段的中点算法决策参数函数表达式及其增量计算表达式。若曲线段的切线斜率绝对值小于 1，给出其生成过程与上述过程的差异，并分析中点算法与 Bresenham 画线算法的异同点。
- 4、，如下图，假设一条边的斜率为 $m=7/3$ ，且已知其与扫描线的某交点为(3,4)。试采用扫描线与边界求交整数增量运算方法计算自该交点以上 4 个与扫描线的连续交点坐标位置。



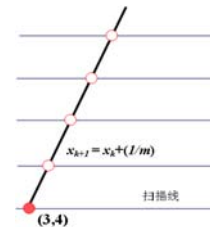
第 1 题图



第 2 题图

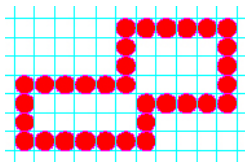


第 3 题图



第 4 题图

- 5、下图给出了区域的边界形状，如何定义该区域的边界和内部像素才能正确填充其内部区域?并说明理由。
- 6、试比较分别采用多边形扫描转换填充和区域填充方法对下图所示的区域形状进行填充生成的形状表示要求及填充过程。



第 5 题图



第 6 题图

- 4、下图给出了两个图元填充算法子功能函数 A 和 B，试给出两个函数的填充算法名称并分别按给定的函数描述其算法处理过程。

功能 A:

```
void FunctionFill4 (int x, int y, int fillColor, int oldColor)
{
    if (getPixel (x, y) == oldColor)
    {
        setColor (fillColor);
        setPixel (x, y);
        FunctionFill4(x+1, y, fillColor, oldColor);
        FunctionFill4(x-1, y, fillColor, oldColor);
        FunctionFill4(x, y+1, fillColor, oldColor);
        FunctionFill4(x, y-1, fillColor, oldColor);
    }
}
```

功能 B:

```
void FunctionFill4 (int x, int y, int fill, int boundary)
{
    int current; current=getPixel (x, y);
    if (((current != boundary) && (current != fill))
    {
        setColor (fill);
        setPixel (x, y);
        FunctionFill4 (x+1, y, fill, boundary);
        FunctionFill4 (x-1, y, fill, boundary);
        FunctionFill4 (x, y+1, fill, boundary);
        FunctionFill4 (x, y-1, fill, boundary);
    }
}
```