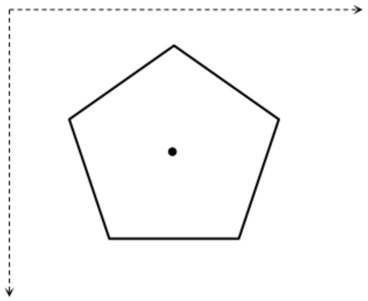
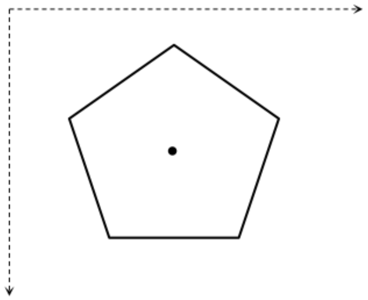
CS 240: Computer Graphics
Handout 2

Flood Fill

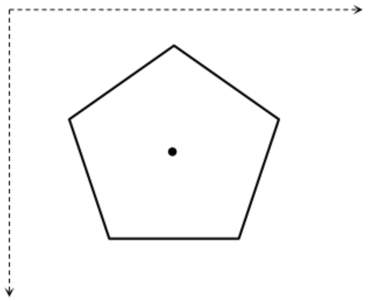
1) In the flood fill algorithm, if there is just one recursive call (to the **north**), what area is filled in the pentagon below? For all questions, assume the first flood fill call is to the marked center pixel.



2) Answer the same question as above, but for two recursive calls (to the **north** and **east**). Shade the filled area in the pentagon below.

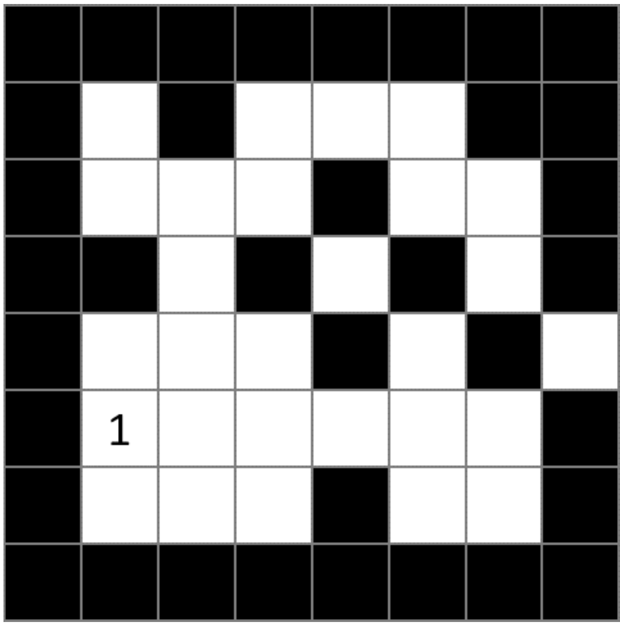


3) Now assume we have three recursive calls (to the **north**, **east**, and **south**). Shade the filled area in the pentagon below.



4) Could the stack overflow problem be addressed by running four separate recursion operations, one for each quadrant? What problems, if any, do you see with this approach?

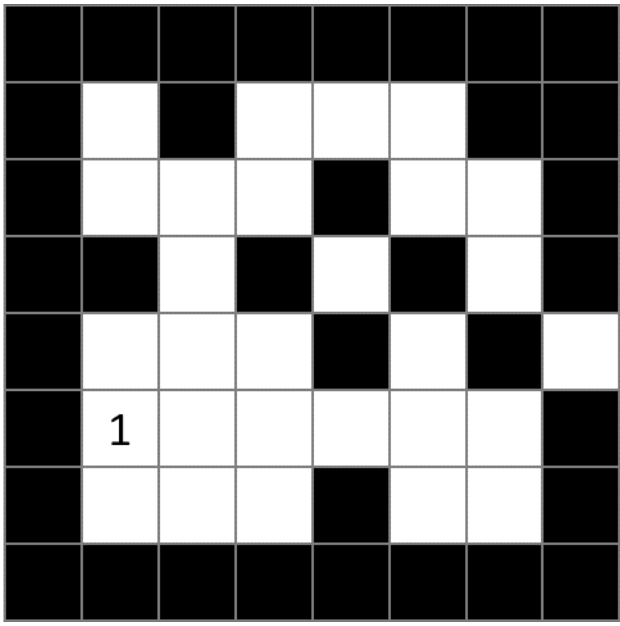
5) Consider the figure at right. Assuming that a recursive flood fill uses E-W-S-N for the sequence of recursive calls, number the pixels by their fill order, starting from the pixel labeled 1.



Recall the pseudocode for the recursive fill:

```
recFill(x,y,$c_0$):
   if color(x,y) $\neq c_0$
      (just return)
   else
      ink(x,y)
      recFill(x+1,y,$c_0$);  // east
      recFill(x-1,y,$c_0$);  // west
      recFill(x,y+1,$c_0$);  // south
      recFill(x,y-1,$c_0$);  // north
   endif
```

6) Now assume that an 8-connected fill is used, with the fill order N-NE-E-SE-S-SW-W-NW. Again number the pixels by their fill order, starting from the pixel labeled 1.



7) Suppose we are doing a scanline fill of a polygon with four vertices, at (4,4), (14,12), (7,15), and (16,20) in order around the boundary.
   a) What y values will the scanline algorithm need to loop over?
   b) We are working on the computation for the scanline at $y = 10$. Which edges will intersect this line?
   c) What are the points of intersection between the edges and scanline $y = 10$?

8) For a different polygon, we compute edge intersection points at (10,10), (4,10), (15,10) and (8,10). What sections of this scanline should be filled in?