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?

   *It won’t necessarily work if the space to be filled has concavities.*
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:

```plaintext
recFill(x,y,c0):
  if color(x,y) ≠ c0
    (just return)
  else
    ink(x,y)
    recFill(x+1,y,c0);  // east
    recFill(x-1,y,c0);  // west
    recFill(x,y+1,c0);  // south
    recFill(x,y-1,c0);  // 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. Now assume that a 4-connected sweep fill algorithm of the sort described in the video lecture is used. Number the pixels by their fill order, starting from the pixel labeled 1.

Recall the pseudocode for the sweep fill:

- Fill pixels to the left of current
- Fill pixels to the right of current
- Scan leftmost to rightmost, making recursive calls above & below