1. Towers of *n*-Towers (*n*)
   1. Towers (*n* – 1)
   2. Move disk in
   3. Towers (*n* – 1)
2. Towers (*n*, start, end)
   1. Towers (*n* – 1, start, mid)
   2. Move disk *n* from start to end
   3. Towers (*n* – 1, mid, end)
3. Look at the number of moves
   1. Let T(*n*) = number of moves with *n* disks.
   2. T(*n*) = T(*n* – 1) + 1 + T(*n* – 1) = 2T(*n* – 1) + 1
      1. //T(*n* – 1): move *n* – 1 from small to mid
      2. //T(*n* – 1): move *n* – 1 from mid to end
   3. Example: T(*n*) = 4T(*n*/2) + *n*
      1. Identities
         1. T(1) = 1
         2. T(*n*/2) = 4T(*n*/4) + *n*/2
         3. T(*n*/4) = 4T(*n*/8) + *n*/4
      2. *T*(*n*) = 4(4T(*n*/4) + n/2) + *n*
         1. = 16T(*n*/4) + 3*n*
         2. = 16T(4T(*n*/8) + *n*/4) +3*n*
         3. = 64T(*n*/8) + 7*n*
4. Iteration Technique
   1. T(*n*) = 2T(*n* – 1) + 1, T(*n* – 1) = 2T(*n* – 2) + 1
   2. T(*n*) = 2T(*n* – 1) + 1
      1. = 2(2T(*n* – 2) +1) +1
      2. = 4T(*n* – 2) + 3
      3. = 4(2T(*n* – 3) + 1) + 3
      4. = 8T(*n* – 3) + 7
5. After *k* steps, 2*k*T(*n* – *k*) + (2*k* – 1)
   1. //Only know T(1)
   2. //Let *k* = *n* – 1, T(*n*) = 2*n* – 1
   3. T(*n* – (*n* – 1)) + 2 – 1 = 2*n* – 1T(1) + 2*n* – 1 – 1
      1. = 2*n* – 1 + 2*n* – 1 – 1
      2. = 2*n* – 1 (1+1) – 1
      3. = 2*n* – 1
6. For 5 disks, 31 moves, 64 disks
   1. 264 – 1 = disk #
   2. 108 or 109
   3. O(2*n*)
7. Flood fill
   1. Fill location (x, y) marking that I’ve been there.
   2. For each unvisited neighbor (u, v) fill (u, v)
   3. Code
      * 1. for (i = 0; i < NUMBER; i++) { //inbonnais (x + DX[i], y + DY[i])
        2. if c: visited (x + DX[i], y + DY[i])
        3. fill (x + DX[i], y + DY[i]);
      1. }
8. Minesweeper