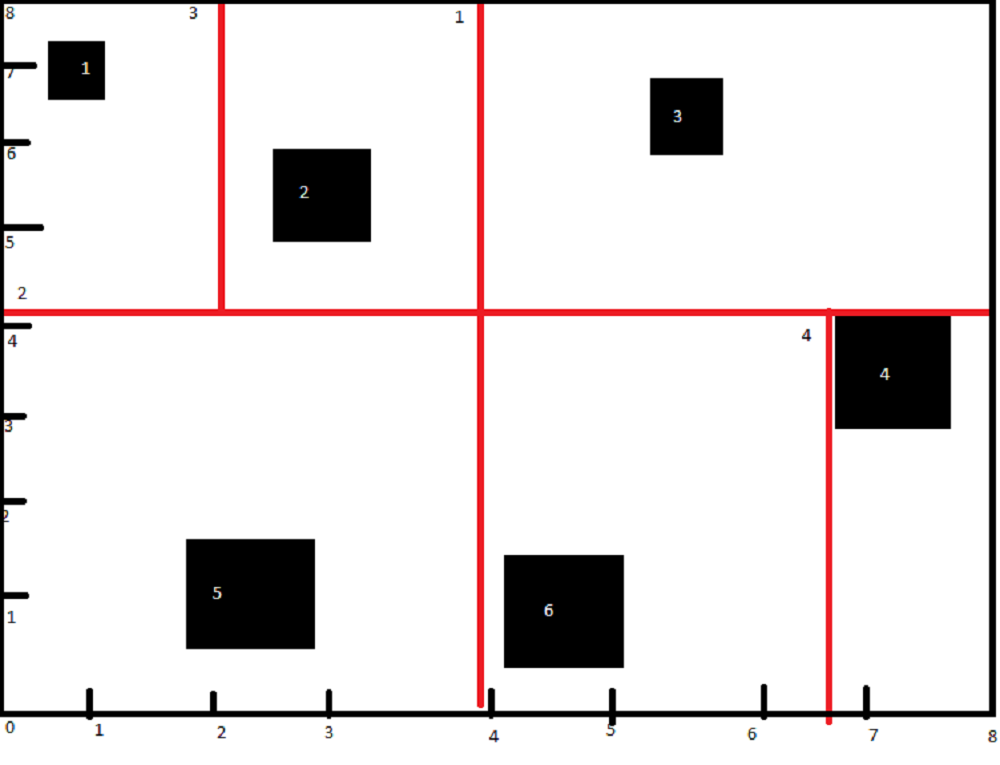
**Part 3:** Write a program to compute a quadtree on a grid (up to 64x64), randomly generate 400 x,y pairs and compute the quadree for that setup. Test this by hand on an 8x8 grid with 6 objects to show correctness.

* [1] Divide in half vertically
* [2] Divide in half horizontally
* Bottom left quadrant
  + Divide in half vertically
  + Can’t: break
* Top Left
  + [3] Divide in half vertically
  + Divide in half horizontally
  + Can’t: break
* Top right
  + 1 member: break
* Bottom right
  + [4] Divide in half vertically
  + Divide in half horizontally
  + Can’t: break
* Every object is isolated: done



**Part 4:** Maths stuff (Do this by hand).

Show the Cohen Sutherland and Liang Barksy (line clipping algorithms) for the following line segments with a:

Window from (0,0) to (10,10)

1. (-4, -2), (8,12)

Cohen Sutherland: P

P1 – 1010

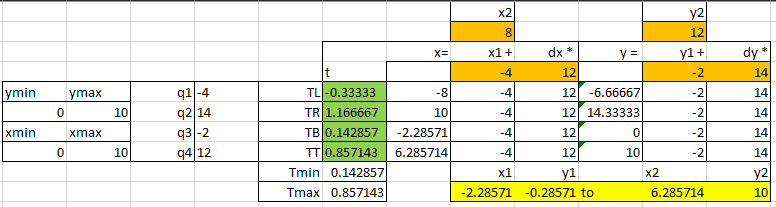
P2 – 0100

AND – 0000

* Onscreen

OR – 1110

* Partially onscreen

Liang Barksy:

Window (-10,-10) to (10,10).

1. (9, -3), (-9, -9)

Cohen Sutherland:

P1 – 0000

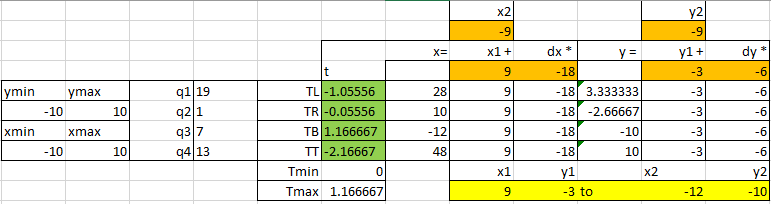
P2 – 0000

AND – 0000

* Onscreen

OR – 0000

* Completely onscreen

Liang Barksy: 

Window (-10,-10) to (10,10).

1. (19,8) (-16,17)

Cohen Sutherland:

P1 – 0100

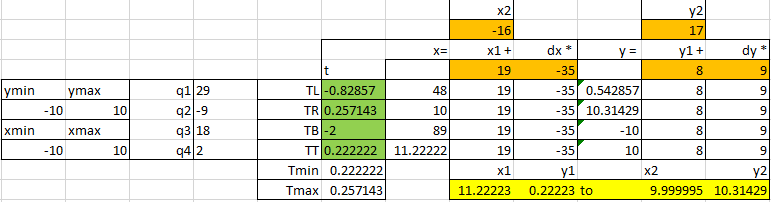
P2 – 1001

AND – 0000

* Onscreen

OR – 1101

* Partially on screen

Liang Barksy: 

Window 0,0 to 1920,1080

1. (2496, 864), (768, 324)

Cohen Sutherland:

P1 – 0100

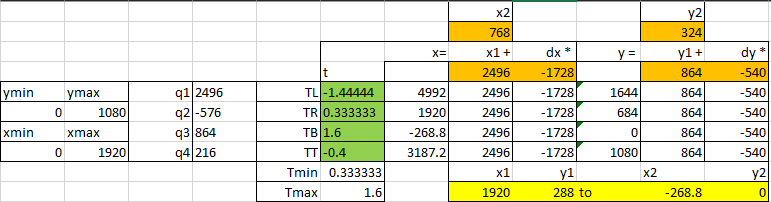
P2 – 0000

AND – 0000

* Onscreen

OR – 0100

* Partially onscreen

Liang Barksy: 

Window from (-960, -540 to +960, 540)

1. (-480, 432), (-864, -1134)

Cohen Sutherland:

P1 – 0000

P2 – 1000

AND – 0000

* Onscreen

OR – 1000

* Partially onscreen

Liang Barksy: 