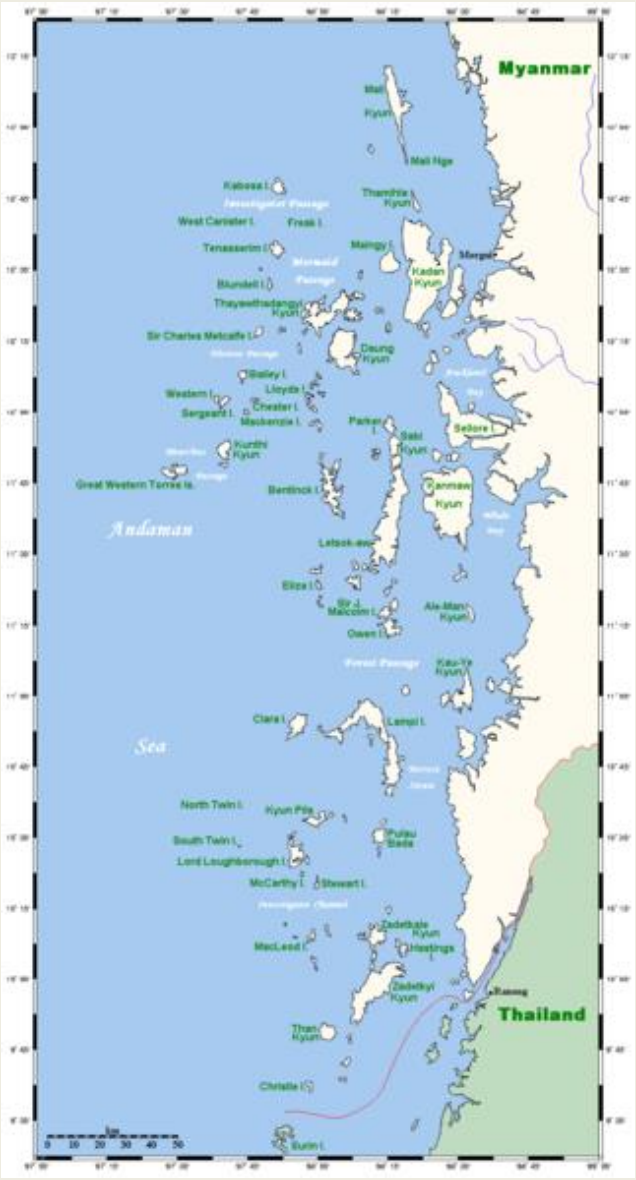
Island Problem

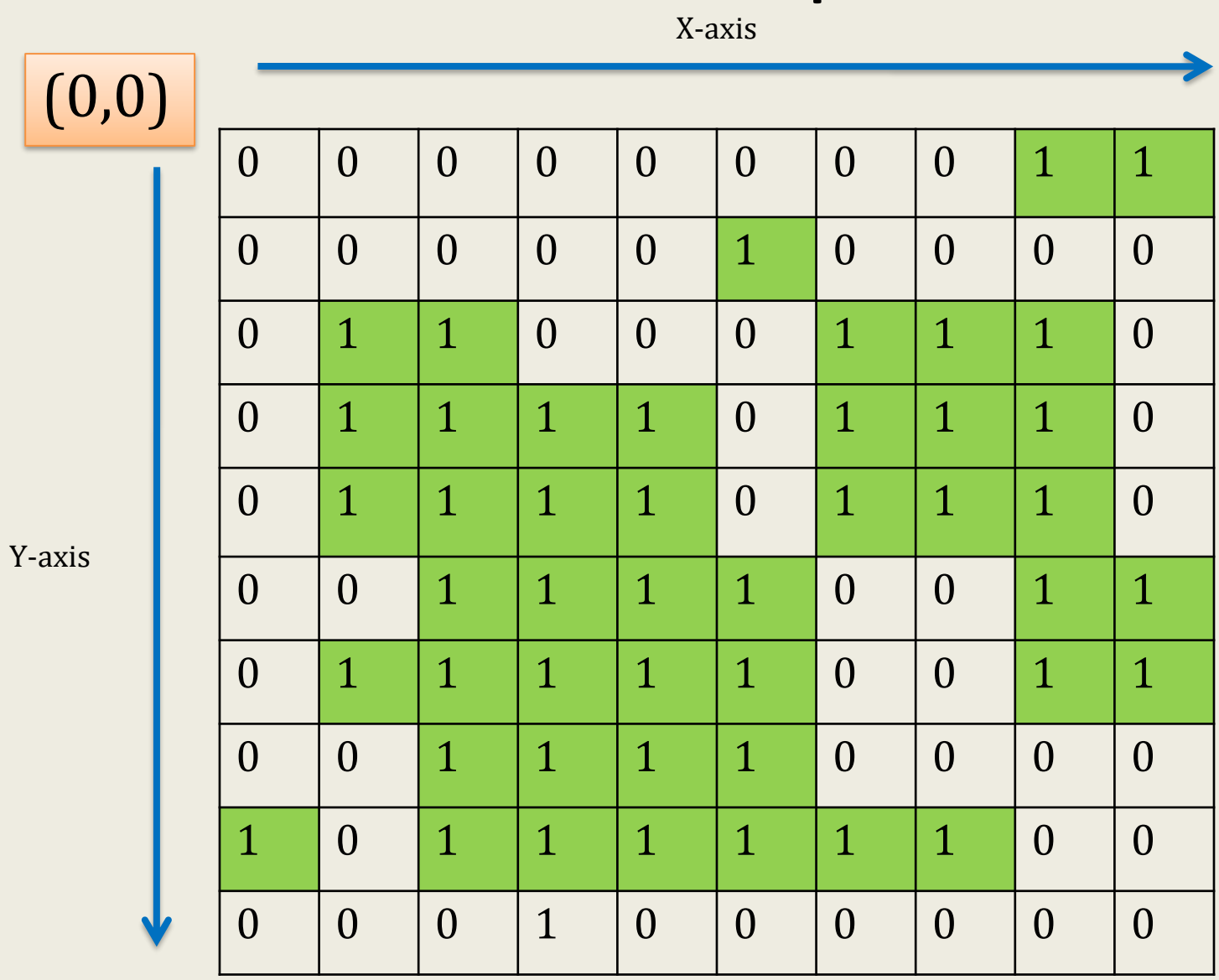
- Given a bitmap of aerial photographer of archipelago (chain of islands)
  - Bitmap is black & white
    - White means land
    - Black means sea
- Find the largest possible square land



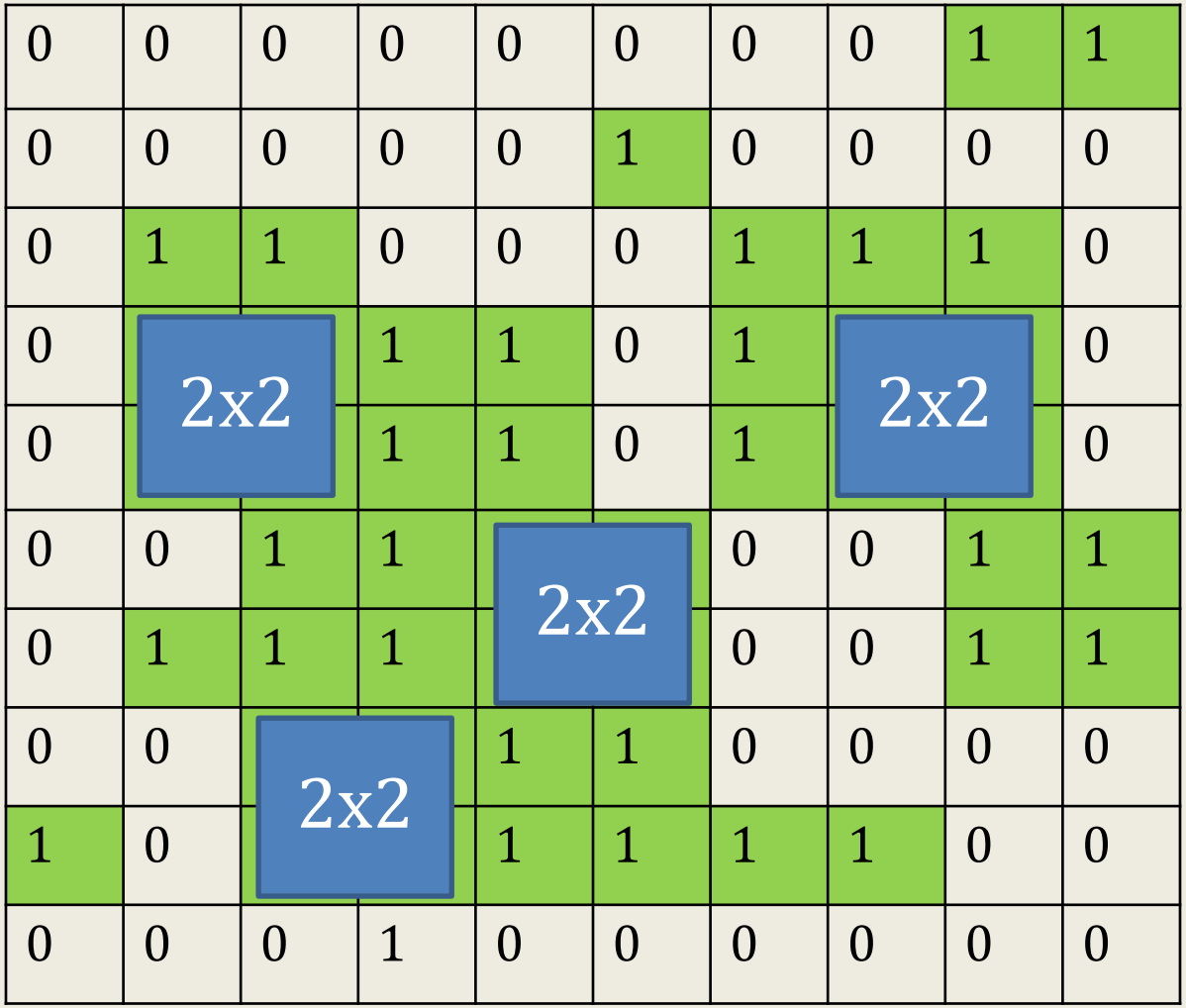
Island Problem

- Input
  - 2D array of Boolean called “Land”
    - Size m x n
    - Land[x][y] is true when cell x,y is a land
- Output
  - x,y the top-left coordinate of the largest square land and s, the size of the largest square land

Example


X-axis
(0,0)
Y-axis
0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 1 0 0 0 0
0 1 1 0 0 0 1 1 1 0
0 1 1 1 1 0 1 1 1 0
0 1 1 1 1 0 1 1 1 0
0 0 1 1 1 1 0 0 1 1
0 1 1 1 1 1 0 0 1 1
0 0 1 1 1 1 0 0 0 0
1 0 1 1 1 1 1 1 0 0
0 0 0 1 0 0 0 0 0 0

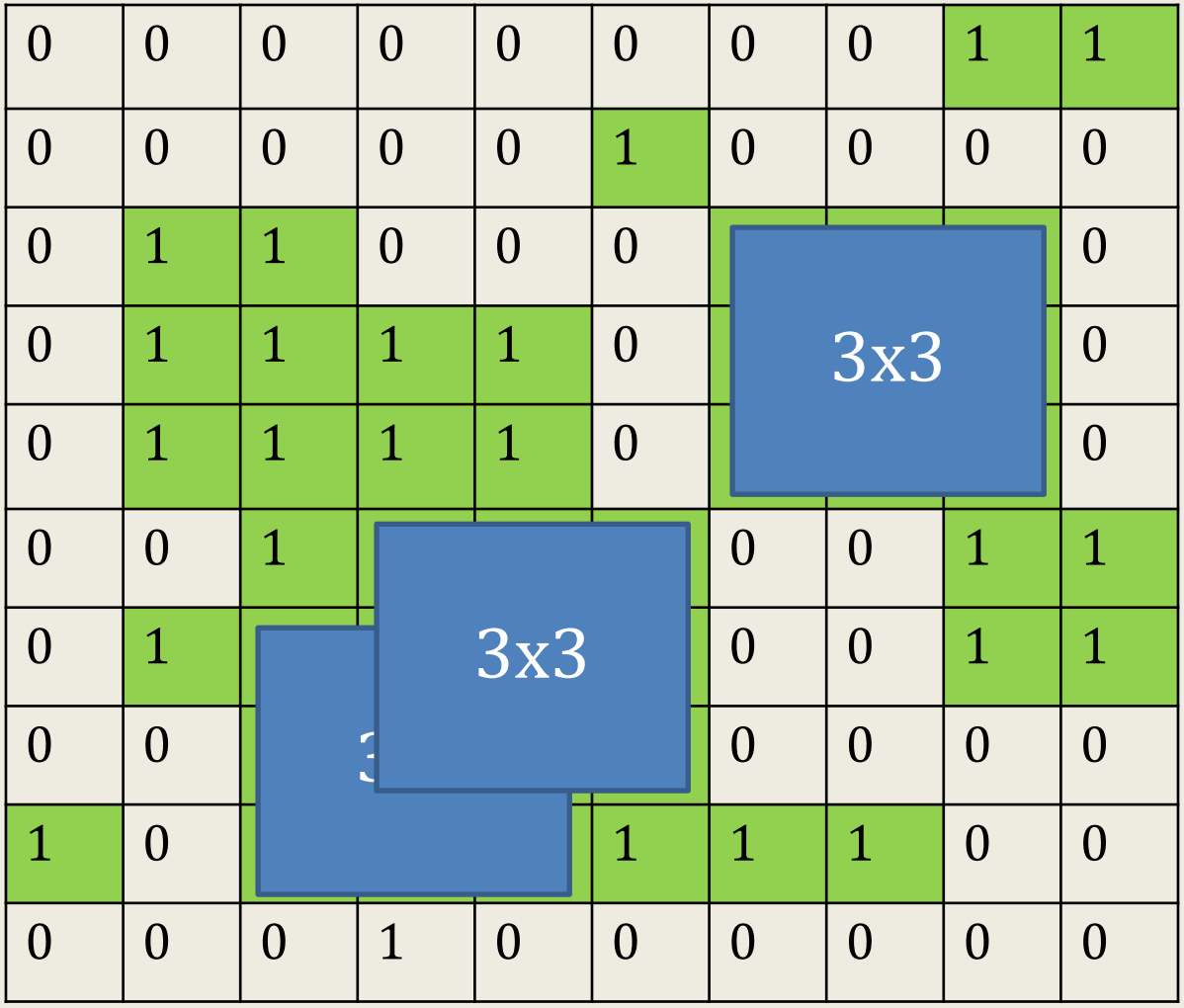
Example : Some of 2x2 square



Solution : The largest square

0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	1	1	1	0
0	1	1	1	1	0	1	1	1	0
0	1	1	1	1	0	1	1	1	0
0	0	1	1	1	1	0	0	1	1
0	1	1	1	1	1	0	0	1	1
0	0	1	1	1	1	0	0	0	0
1	0	1	1	1	1	1	1	0	0
0	0	0	1	0	0	0	0	0	0

Example


0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 1 0 0 0 0
0 1 1 0 0 0 0
0 1 1 1 1 0 0
0 1 1 1 1 0 0
0 0 1 0 0 1 1
0 1 0 0 1 1
0 0 0 0 0 0
1 0 1 1 1 0 0
0 0 0 1 0 0 0 0 0 0
3x3
3x3

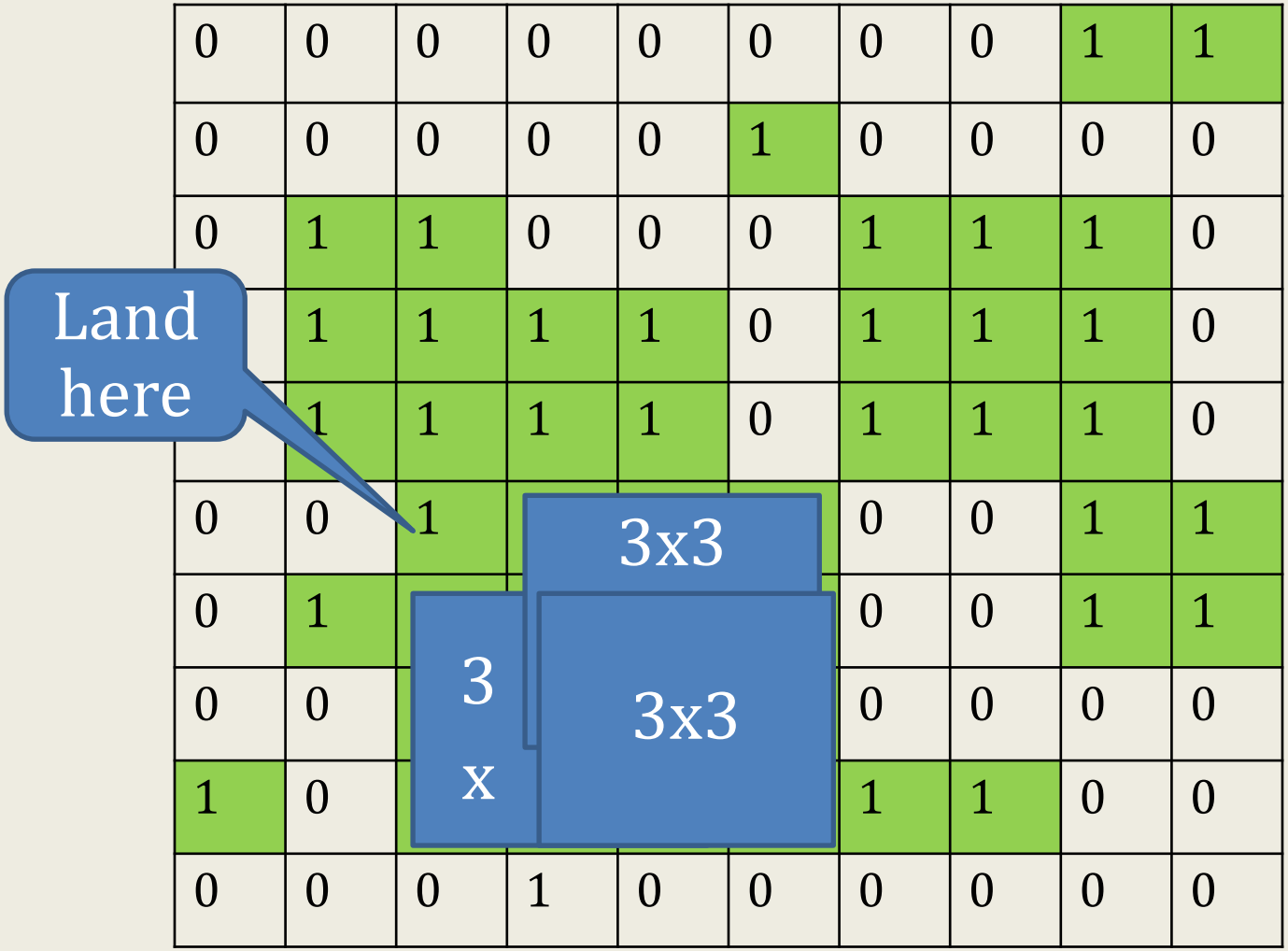
Sub-problem

- What should be our subproblem?
  - Smaller maps?
  - Divide maps by half?
  - Smaller square?

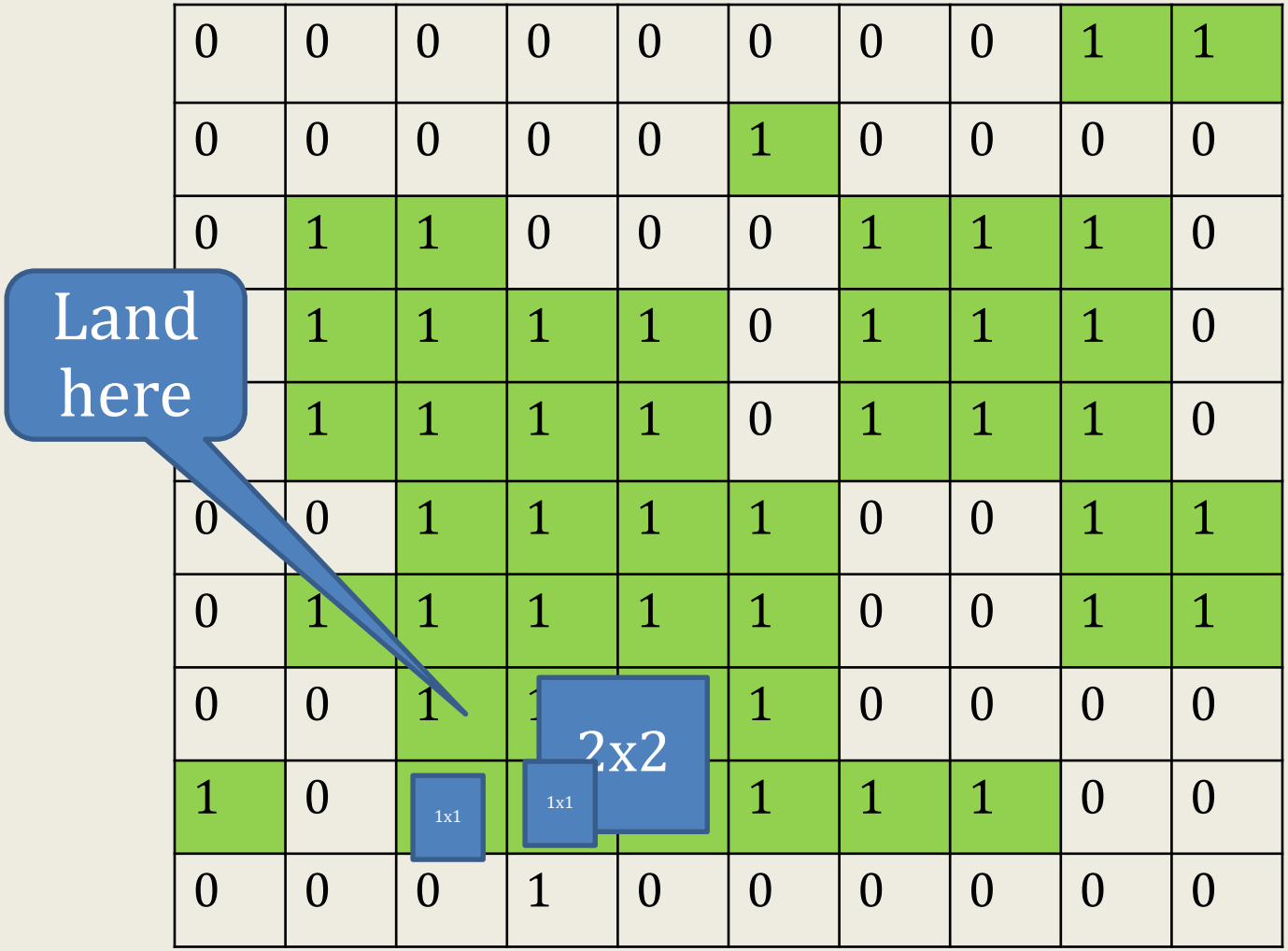
Sub-problem

- What if we find all possible square?
  - How the solution of n x n square constitute the solution of (n+1) x (n+1)?

Sub-problem

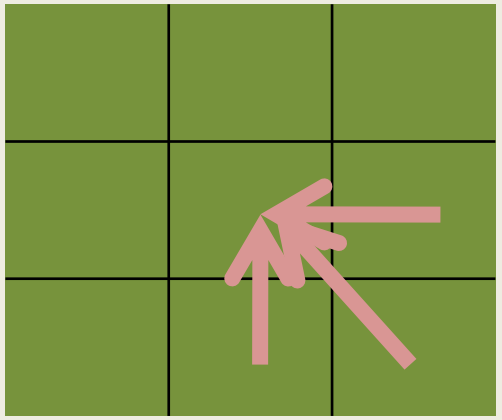

0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 1 0 0 0 0
0 1 1 0 0 0 1 1 1 0
1 1 1 1 0 1 1 1 0
1 1 1 1 0 1 1 1 0
0 0 1 0 0 1 1
0 1 0 0 1 1
0 0 0 0 0 0
1 0 1 1 0 0
0 0 0 1 0 0 0 0 0 0
Land here
3x3
3
x
3x3

Sub-problem


0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 1 0 0 0 0
0 1 1 0 0 0 1 1 1 0
1 1 1 1 0 1 1 1 0
1 1 1 1 0 1 1 1 0
0 0 1 1 1 1 0 0 1 1
0 1 1 1 1 1 0 0 1 1
0 0 1 1 1 0 0 0 0
1 0 1 1 1 1 1 0 0
0 0 0 1 0 0 0 0 0 0
Land here
2x2
1x1
1x1

Recursion

- b2[x][y] ขนาดของสี่เหลี่ยมจัตุรัสใหญ่ที่สุด ที่มีมุมล่างขวาอยู่ที่ตำแหน่ง x,y (input คือ Land[x][y], m, n)
- Biggest[x,y] =
  - Min(Biggest[x+1,y+1],
    Biggest[x ,y+1],
    Biggest[x+1,y]) + 1 if Land[x,y]
  - 0 if not land[x,y]



Sub-problem

0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0
0	2	1	0	0	0	3	2	1	0
0	2	3	2	1	0	2	2	1	0
0	1	3	2	1	0	1	1	1	0
0	0	4	3	2	1	0	0	2	1
0	1	3	3	2	1	0	0	1	1
0	0	2	2	2	1	0	0	0	0
1	0	1	1	1	1	1	1	0	0
0	0	0	1	0	0	0	0	0	0