NUMBER OF RECTANGLES IN A N*N GRID
PROOF BY RECURSION

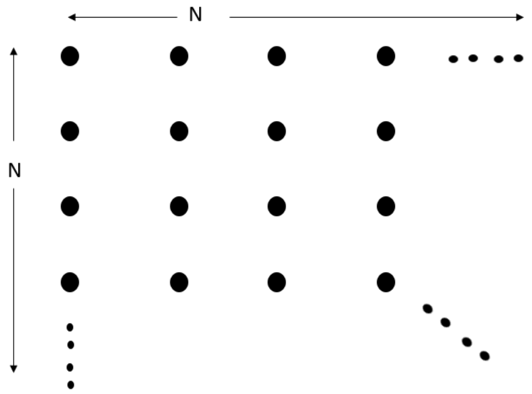
Author: Meherwan Singh Gill

1. Introduction

We will first look at base case of a N*N grid (a square grid) and then we will proceed to a general case of M*N grid (a rectangular grid). Since the base case is easier and going through that will give us a more intuitive idea for the general case

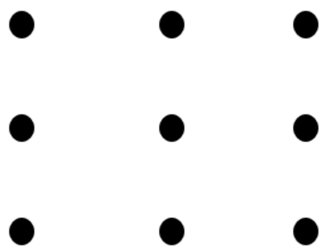
2. Base Case

A general N * N grid looks like



NOTE: (here) N corresponds to the number of segments with length of 1 unit, you can draw in one row or one column

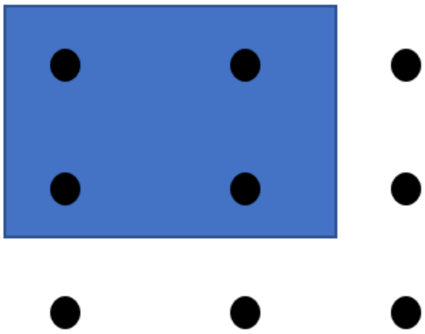
Let's look at a 2x2 case first



Brute forcing this, will easily give us 9 rectangles. 4 rectangles of size 1x1 , 2 rectangles of size 2x1 , 2 rectangles of size 1x2 and 1 rectangle of size 2x2.

$$number\ of\ rectangles\ in\ a\ 2 * 2\ grid\ = 9\ - equation\ (1)$$

Also, in that figure we see that



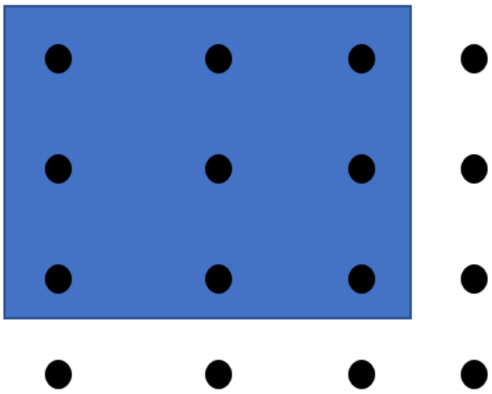
The number of rectangles in the 2x2 figure will have the same number of rectangles as the one in the shaded figure and some extra rectangles added due to 5 nodes which are not in the shaded portion. That shaded figure is nothing else but a grid of size 1*1.

In more mathematical terms, if we say that the number of rectangles in a n * n grid is T(n) then

$$T(2) = T(1) + F(2)$$

Where F(2) is the extra rectangles drawn due to the last row of 2*2 grid and the last column of 2*2 grid

Similarly, for a 3*3 grid



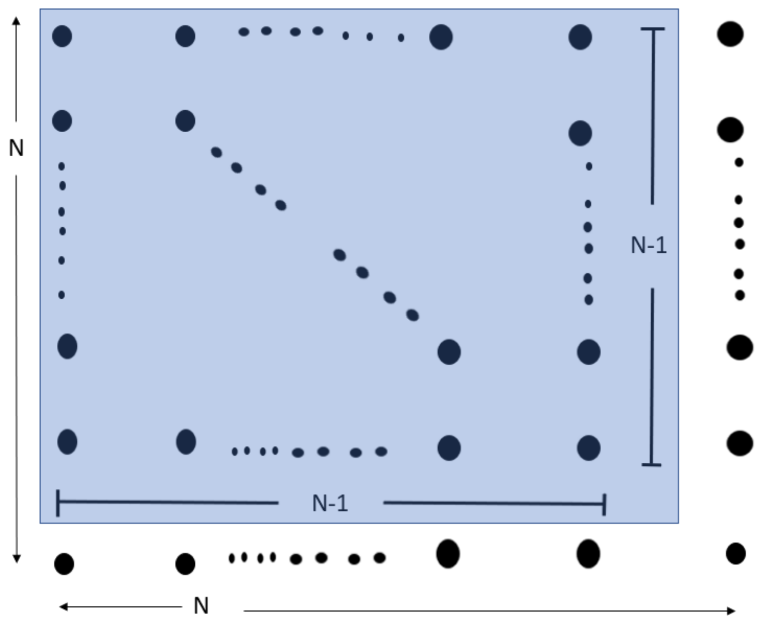
We see that it will include all the rectangles in the shaded portion, which is a 2*2 grid. This means that

$$T(3) = T(2) + F(3)$$

Since we brute forced the solution to T(2) to 9 in equation (1) , this can be further simplified to

$$T(3) = 9 + F(3)$$

We see that we can apply this to a general N*N grid too



Every N*N square grid can be divided into a (N-1) * (N-1) square grid + the last row of N*N square grid + the last column of N*N square grid

Therefore, our general formula becomes

$$T(N) = T(N-1) + F(N)$$
$$- \quad equation\ (2)$$

Now to find F(N)

The way we will approach this problem is by

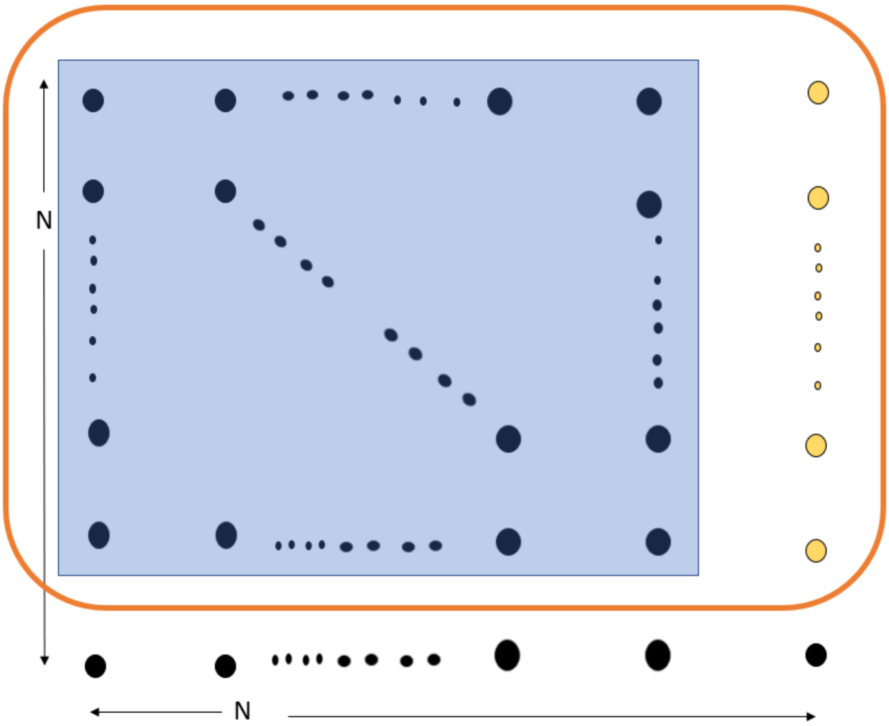
a. analysing the number of rectangles drawn by the first N – 1 rows of the last column of N*N grid, such that all the rectangles drawn DON'T use any node in the last row of the N*N grid.
b. Analysing the number of rectangles drawn by the first N-1 columns of the last row of N*N gird, such that all the rectangles drawn DON'T use any node in the last column of the N*N grid
c. Analysing the number of rectangles drawn by the node whose position is the intersection point of the last row and last column of the N*N grid

The reason I am doing this is because, the three cases described above are mutually exclusive, i.e. the rectangles drawn in any one scenario WILL NOT overlap/intersect with rectangles drawn in other scenario

Mathematically speaking, this means

$$F(N) = scenario\ a + scenario\ b + scenario\ c$$

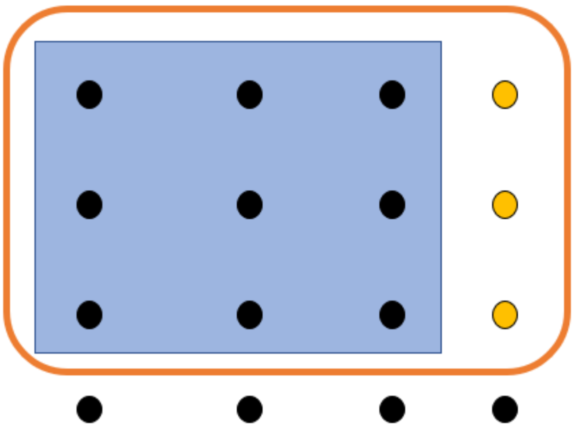
Visualising scenario a.



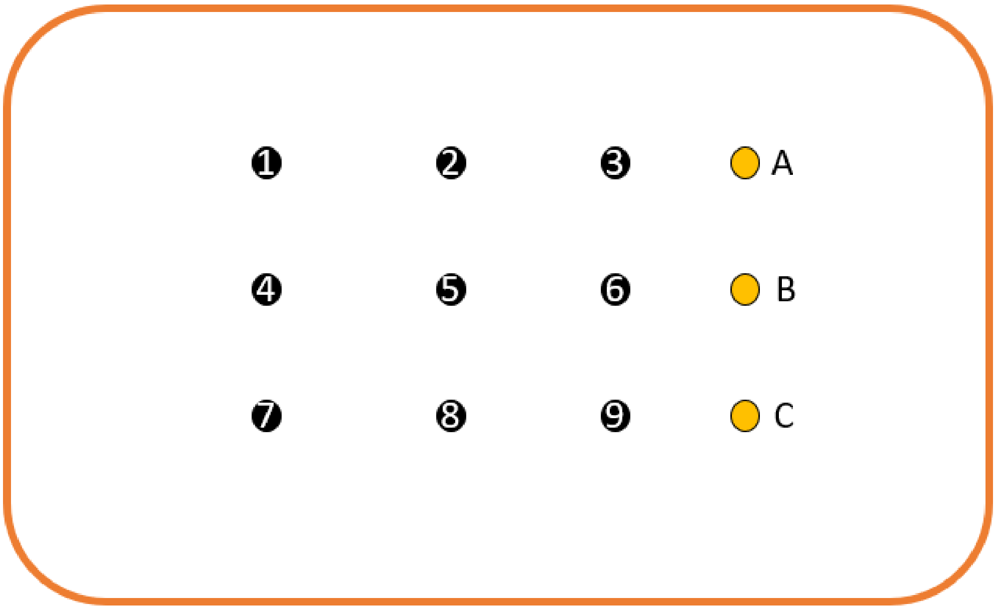
(As before, the blue shaded area represents a (N-1)*(N-1) grid)

Scenario a, tells us to analyse the number of rectangles in the area bounded by the orange bordered rounded rectangle such that it uses at least two nodes from the yellow coloured nodes.

Let's look at a small case of a 3*3 grid first to get some idea on how to tackle this problem.



Lets just look at the portion bounded by the orange box, since that is what interests us in the current scenario

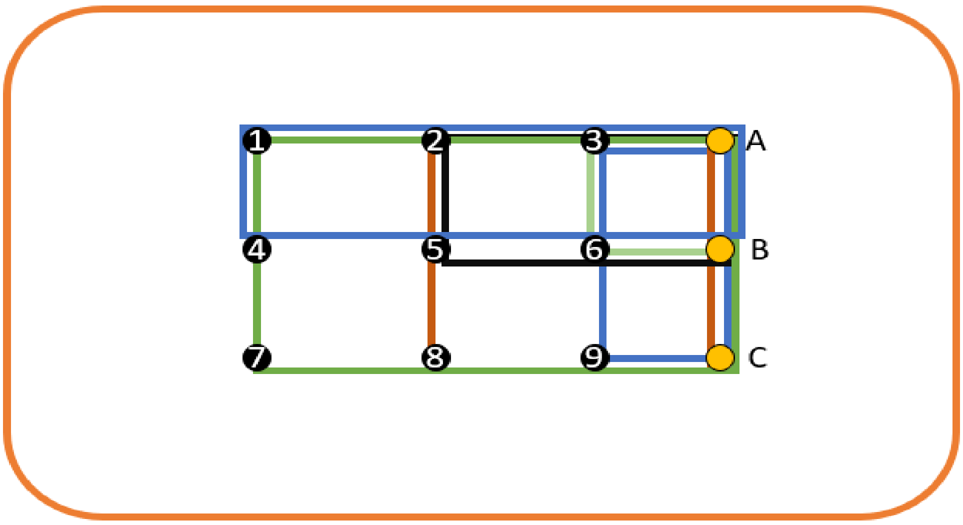


I labelled the 3 yellow nodes as A , B and C , and the rest of the nodes have been numbered through 1 to 9

Now the way we will draw the rectangles from the yellow node is by ensuring that the yellow node is the TOP vertex of the rectangle.
We can observe two facts

1. A yellow node will make at least one pair with a node in the same row. (this might seem too obvious to state, but you shall see its significance later)
2. Every pair of (a yellow node, a node in the same row) will have the same number of possible rectangles you can draw

Now let’s try to draw all the rectangles from point A to further understand the two observations



For vertex A, it makes a pair with each: node 1, node 2 and node 3

For each of these pairs, it makes two rectangles
For pair A, 3:

- A 3 6 B
- A 3 9 C

For pair A, 2:

- A 2 5 6
- A 2 8 C

For pair A, 1:

- A 1 7 C
- A 1 4 B

Similarly, for vertex B, it makes a pair with each: node 4, node 5, node 6
For each of these pair, it makes 1 rectangle (Remember, we are only drawing rectangles (considering cases) such that the vertex for which we draw is the top most vertex of the rectangle, there when we consider vertex B, B A 3 6 will not be a valid case, since A is the top vertex and not B. This helps us in removing duplicate cases)
For pair B, 6:

- B 6 9 C

For pair B, 5:

- B 5 8 C

For pair B, 4

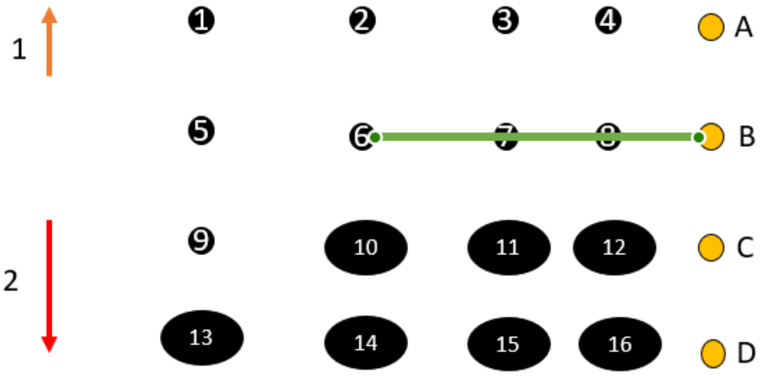
- B 4 7 C

For vertex C, there will be 0 rectangles such that it remains the top vertex in all those rectangles, since there are no nodes beneath C.

Therefore, the total number of rectangles drawn are:

- $3 * 2 + 3 * 1 + 3 * 0 = 3(0 + 1 + 2) = 9$

To try and understand a bit about where all the constants come from and why it is in such a nice form, lets look at the top vertex (A) again. It makes a pair with every node in its row except itself, and the total number of such nodes are N (For a N*N grid, the number of nodes in one row are N + 1. For example, in a 1*1 grid, the number of nodes in one row are 2) . For each node in that set, it can only draw a straight line directly beneath itself. To illustrate it better



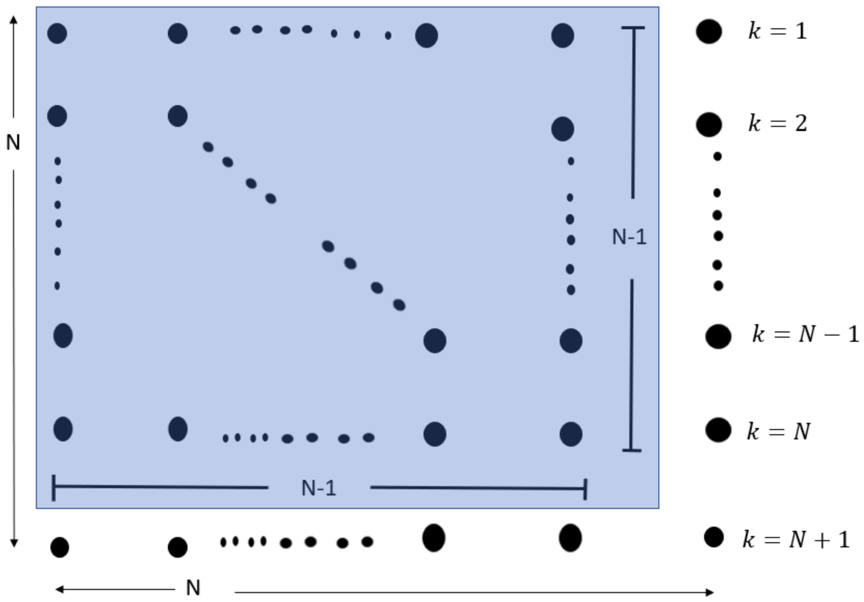
As we see in this grid, let's look at the pair (B , 6) , The line starts from B to 6. Since we are only making rectangles downwards, therefore, there are only two possible vertices where the line can go from node 6 to make a rectangle, node 10 or node 14. Reiterating our second observation, the fact that there are 2 possible ways for pair (B, 6) to make a rectangle, will be true for every pair of B with another node in the same row. Which means (B, 8) will make 2 rectangles, (B,5) will make 2 rectangles and so on.

We see that the number of rectangles such a pair can make has to be the number of rows beneath the pair. Starting from the top node (A), we have 3 rows beneath it, so it will make 3 rectangles with each of its $N = 4$ nodes.

As we step down another row to analyse the number of triangles the vertex of that row can make, the number of rows beneath that node effectively decrease by 1. For example, for Node B, we have 2 rows beneath it, for Node C we have 1 row beneath it, for Node D, we have 0 rows beneath it.

We can generalise this for a N*N grid by saying

- One yellow Node, will make a pair with N nodes (all nodes in its row except itself)
- Every pair made in the above line, will make $N - k$ rectangles, where k is the row number of the top vertex pair of the rectangle. Hence k goes from 1 to N



Remember, for a N*N grid, it will have N+1 nodes in its row.

To write what we found out mathematically,

The number of rectangles made in scenario a, is

$$\sum_{k=1}^{k=N}\big((N)*(N-k)\big) = N*\sum_{k=0}^{k=N-1} k$$

$$= N*\frac{N\,(N-1)}{2}$$

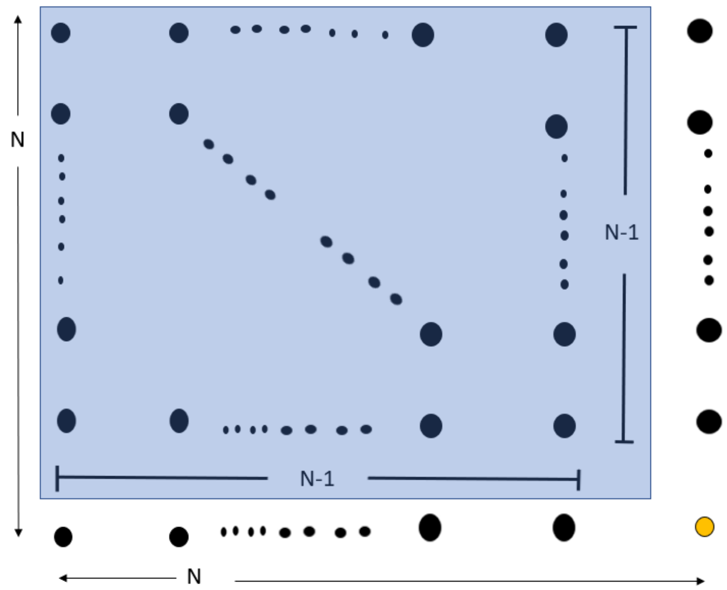
$$-equation\ (3)$$

Due to symmetry, the answer for scenario b is also

$$\frac{N\big(N(N-1)\big)}{2}$$

$$-\ equation\ (4)$$

Now we come to scenario c. That is the number of rectangles made by the bottom right most node of the grid.



i.e. The number of rectangles made by the yellow vertex here

It is easy to see that every rectangle made by that vertex will use ONLY ONE UNIQUE NODE from the blue shaded region , and every Node in the blue region will be used in making a rectangle.
To rephrase, each node in the blue region will make ONE rectangle using the yellow node and that is ALL the rectangles that the yellow node will make.

This makes our problem quite easy because now we only need to find the number of nodes in the blue shaded region. Since the blue shaded regions is a $(N-1)*(N-1)$ grid, therefore, it will have N^2 nodes.
This means that the number of rectangles made by the yellow node will be N^2

That is the answer to scenario c $-\ equation\ (5)$

Coming back to our equation for $F(N) = scenario\ a + scenario\ b + scenario$

Using equation 4, 5,6 we get

$$F(N) = \frac{N\big(N(N-1)\big)}{2} + \frac{N\big(N(N-1)\big)}{2} + N^2$$
$$= N^2(N-1) + N^2$$
$$= N^3$$

This gives us a recursive definition for $T(N)$

$$T(N) = T(N-1) + F(N)$$
$$= T(N-1) + N^3$$
$$-equation\ (6)$$

Our base case is $T(1) = 1$

This recursion can be solved in two ways

- A shorter and intuitive way
- A more general and robust way (repertoire method)

Using the shorter and more intuitive way

If we keep expanding the recursion, we see that

$$T(N) = T(N-1) + N^3$$

$$= T(N-2) + (N-1)^3 + N^3$$
$$= T(1) + 2^3 + 3^3 + 4^3 + \cdots + (N-3)^3 + (N-2)^3 + (N-1)^3 + N^3$$
$$= \sum_{k=1}^{k=N} k^3$$
$$= sum\ of\ first\ N\ cubes$$
$$= \left(\frac{N(N+1)}{2}\right)^2$$

Using the repertoire method
(NOTE: For people who are unfamiliar with repertoire method, they can look at https://www.youtube.com/watch?v=8WbpRwYcEf0**)**

$$T(N) = T(N-1) + N^3$$

Since T(N) is sum of