Project Euler 87

Let’s consider some more examples, in x, y dimensional plane:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | |  | |  | |  | |  |   10 | |  |  | | --- | --- | |  |  | |  |  | |  |  | |  |  |   30 | |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | |  |  |  |   60 | |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  |   100 |
| |  | | --- | |  | |  | |  |   6 | |  |  | | --- | --- | |  |  | |  |  | |  |  |   18 | |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  |   36 | |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | |  |  |  |  | |  |  |  |  |   60 |
| |  | | --- | |  | |  |   3 | |  |  | | --- | --- | |  |  | |  |  |   9 | |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  |   18 | |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | |  |  |  |  |   30 |
| |  | | --- | |  |   1 | |  |  | | --- | --- | |  |  |   3 | |  |  |  | | --- | --- | --- | |  |  |  |   6 | |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  |   10 |

We hope to find a function nBoxes, a function of the x and y dimension of the grid.

The first thing to notice is that the above is a product table, so the function we’re searching for is a product of two (identical) functions:

This seems a sensible start point.

Now let’s actually think about the problem. We really want to generalise this so let’s consider an grid ( units and ), all we’re doing here is choosing the position of two vertical lines, once we have the two lines we join them with horizontal ones on either side.

Now, physicists who’ve studied thermodynamics andor statistical mechanics are going to instantly think of the statistical weight,, which is the number of ways of placing atoms in a lattice. We have lattice sites (in one dimension) and two atoms. If you’re a mathematician you’ll probably call it the binomial coefficient but that’s rubbish, it’s the statistical weight and I won’t hear otherwise. In general:

And specifically we’re looking for:

Well that’s fine so far, now if we expand the factorials as products we can cancel terms:

So, and similarly for . Therefore:

Testing this formula confirms that it works for all the examples I bothered to count above. So now suppose we have a grid, we can draw 25502500 boxes. This is approximately 26 million and is way above out 2 million limit in the problem so let’s use this as the limit to the search.

**def** **twoOmega**(x):

**return** (x\*x)+x

**def** **nBoxes**(x,y):

**return** twoOmega(x)\*twoOmega(y)/4

xy,=20,20

best=25502500

bestX,bestY=100,100

**for** x **in** **range**(100):

**for** y **in** **range**(100):

nB=nBoxes(x,y)

**if** **abs**(2000000-nB)<**abs**(2000000-best):

best=nB

bestX=x

bestY=y

**print** bestX\*bestY