# MTH 3270 Final Project Weekly Report

Carlos Galvan Jackeline Escalante

## Week of 04/06/2020

### What did you try?

This week we attempted to look at new approach of finding the least number of ones to make 10. Did these to see if there was a different approach that would be more efficient. We figured to do factor families from one through ten(Shown in Figure 1). Using these factor families we broke them down to ones.(Shown in Figure 2). This would hopefully let us see if there is a pattern in the number of ones being used. Adding on to using the factor families as a table, we looked at how many different ways can we add numbers to equal 10 and we came up with 11 ways...

- 0 + 10
- 10 + 0
- 1 + 9
- 9 + 1
- 2 + 8
- 8 + 2
- 3 + 7
- 7 + 3
- 4 + 6
- 6 + 4
- 5 + 5

We will be using this to break each equation down to the least number of ones using our factor families table.

### What did you observe/what did you prove?

What I took from this table of factor families is that a number multiplied by itself and one generated the most amount of ones(Shown in Figure 2). We also noticed that in a factor family, for example 1 * 3 generated a 4 ones. The opposite order of 3 * 1 also generated 4 ones. The same goes for all of the numbers, so in this case order didn't really mattered. There is also 4 steps needed to follow in order to generate the least number of ones.

Step 1: Use table in Figure 1 to break down the equation, use factor families corresponding to the number with the least number of ones.

Step 2: Avoid choosing the factor family where the number multiplies itself and 1, from what we saw it just generated a lot more ones.(E.g. 1*2,1*3,1*4, etc.)

Step 3: When breaking down the equation and you get 2. Don't use the factor family of (1*2), because this generates 3 ones. Instead just do (1+1), because this generates 2 ones.

Step 4: If the number is odd, use its closest lower even number plus one. For example if you have...

- 3 -> 2+1
- 5 -> 4+1
- 7 -> 6+1
- 9 -> 8+1

From what we found the least number of ones generated to make 10 was when we did $1 + 9$ which would give you 7 ones, . Overall, this approach compared to our old approach is less efficient. The reason being, if we were to do this with numbers greater than 10 it would take a long process to list all ways that would add up to that number and follow all the steps given in order to know what $C_n$ is equal to.

## What are your next steps?

The next step would be to generate a scatter plot of the upper bounds produced by our Code vs. n values from last week. As well is to generate a code for this new approach and compare it to our other approach to see which is more efficient.

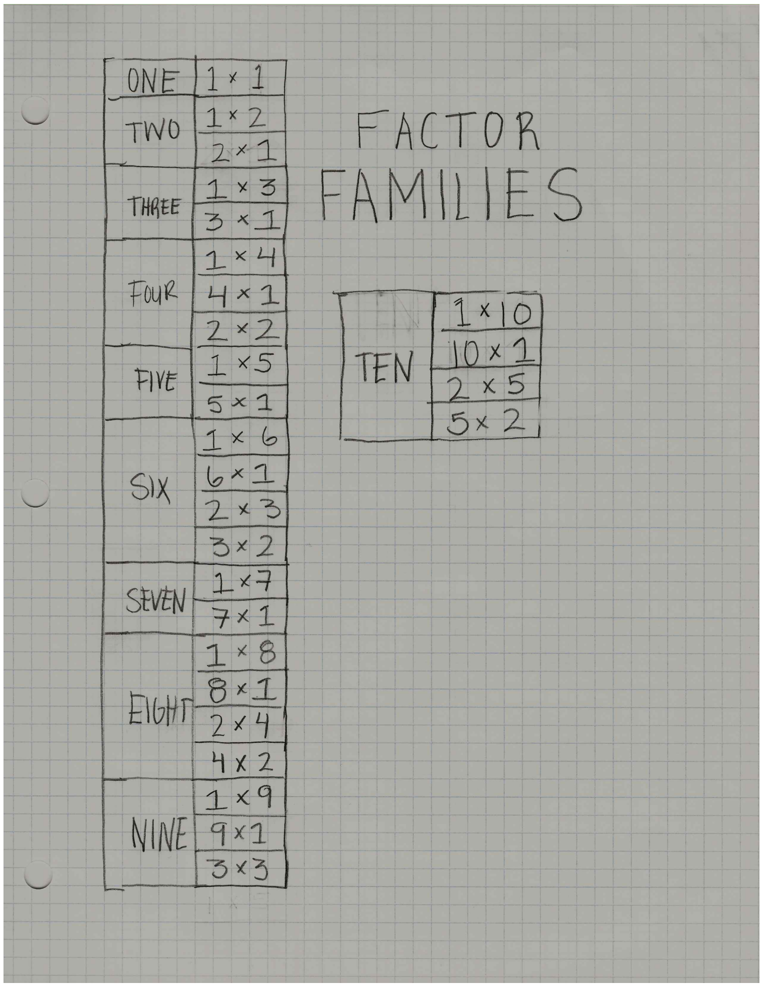


Figure 1: Factor Families

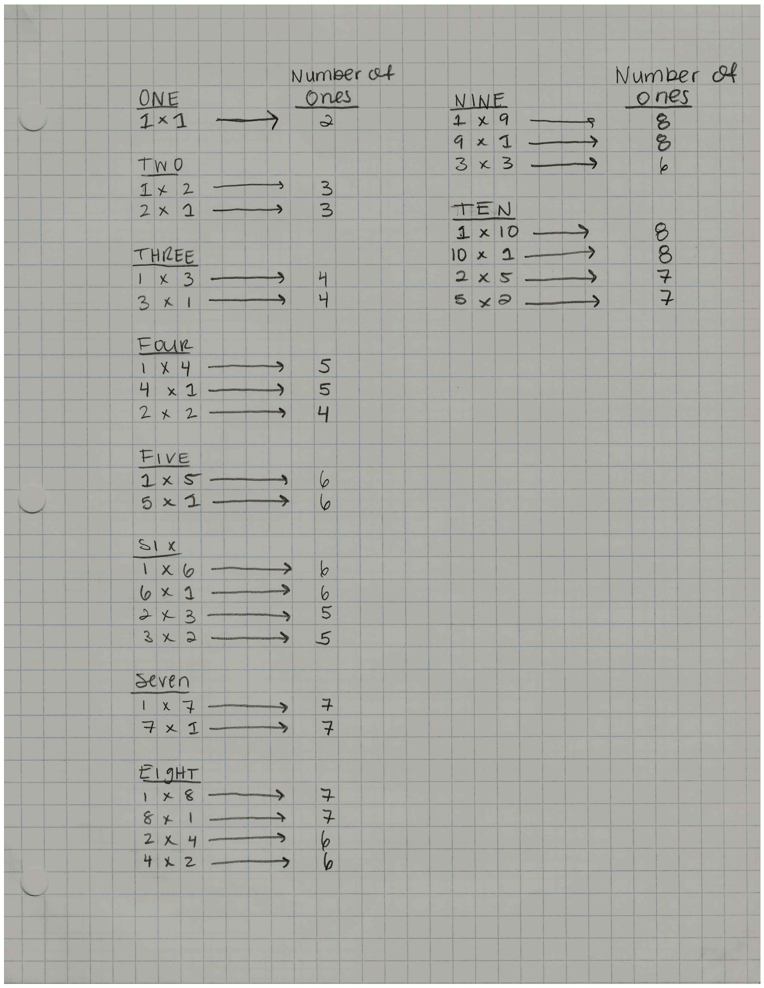


Figure 2: Number of Ones