MTH 3270 Final Project Weekly Report

Carlos Galvan

Jackeline Escalante

Week of 03/16/2020

What did you try?

What we know: if n amount of integers added up its results equal n

Old approach: Last week we took a direct approach and calculated all the possible ways of writing n amounts of digits with either addition or multiplication. We also observed that by including grouping(parenthesis) into the equation, but it made things exponentially more difficult and convoluted.

New Approach: this week, we took a new approach. We discovered that if you divide the n amount by 2, you can drastically lower the amount of integers needed. For example we can write C_{10} by adding ones ten times, or we can split it into 5 times 2.

$$10 = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 \tag{1}$$

We can write 5, by adding one five times and put it into its own group. The same can be said for 2. So C_{10} can be written as:

$$10 = (1 + 1 + 1 + 1 + 1) * (1 + 1) = 10 \tag{2}$$

This means we have a total of 7 integers or $C_n \leq 7$ But using this logic, we can refine the numbers themselves so they break down even further.However we come across an error when we try to divide an odd number by two. So our solution was to subtract one from it to turn it into an even number. So 5 can be written as:

$$10 = (4 + 1) * (2) \tag{3}$$

We can take our previous approach and apply it to the 4 to break it down even further. So its written as:

$$10 = ((2 * 2) + 1) * (2)) \tag{4}$$

The final step is to break down the twos into ones, so our final equation looks like:

$$10 = (((1 + 1) * (1 + 1) + 1) * (1 + 1)) \tag{5}$$

$$C_{10} = 7 \tag{6}$$

This still gives us 7 integers, which sounds redundant but when you apply this logic to larger numbers, you can see the difference.

C_n	Number of One Integers
C_{10}	7
C_{11}	9
C_{12}	8
C_{13}	9
C_{14}	8
C_{15}	8
C_{16}	8
C_{17}	10
C_{18}	8
C_{19}	10
C_{20}	9

What did you observe/what did you prove?

To summarize, take n and break it down by either dividing by 2 if it is even. If n is odd we can turn it even by subtracting one from it, thus turning it even (which can now be divided by one).Each time we do this, all the integers get placed inside a group. We do this until all of the integers are 1, then count the number of integers to get our result. This seemed to work logically with anything we tried. But it can be more tedious for numbers that are larger. In addition, we tried C_{10} and going up to C_{20}. There seemed to be no pattern as the number of one integers being used(see table above).

What are your next steps?

Our next step is to take this approach and test it to make sure it works with other numbers. We also want to see if we can optimize it to be more efficient.