# MTH 3270 Final Project Weekly Report

Carlos Galvan Jackeline Escalante

## Week of 04/13/2020

Click to visit GitHub page.

### What did you try?

For this week's report, we followed up with the previous program we've created, and used it to generate a scatter plot. The main function **recursion()** is what generates m (least amount of 1s needed to generate n). It follows the same logic as **recursionPrinting()** function displayed 2 reports ago, however it is much lighter, since it only returns a single int value as compared to the list value that **recursionPrinting()** returned. This function has been tested to generate the correct value.

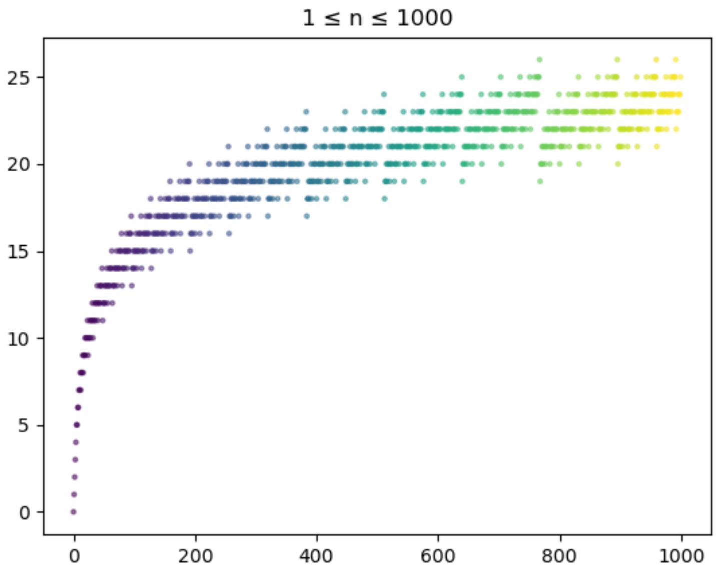
```python
def recursion(n, c):
    if n == 0:  # if no number, then 0
        c += 0
        return c
    elif n == 1:  # once n has been broken down to 1
        if n > c:  # this only applies to 1 other wise
    C1 --> 0
            c += 1
        return c
    elif (n % 2) == 0:  # is even
        n /= 2  # n/2
        c += 2
        return recursion(n, c)
    elif (n % 2) == 1:  # is odd
        n = (n - 1) / 2  # (n - 1) + 2
        c += 3  # (-1) and /2 all result to the use of
    3 Ones
        return recursion(n, c)
    else:
        print("Not a valid input")
```

By installing an external library called *matplotlib*, we were able to generate a comprehensive scatter plot. To do so, it required 2 same sized list that contained the x coordinate and the y coordinate, x representing n, and y representing m. To create these list, we used a simple for loop with its range being determined by user input. The for loop appends the incremental numbers in list one, and uses the current instance of x (or i in this case) and passed it through the recursion() function to get y.

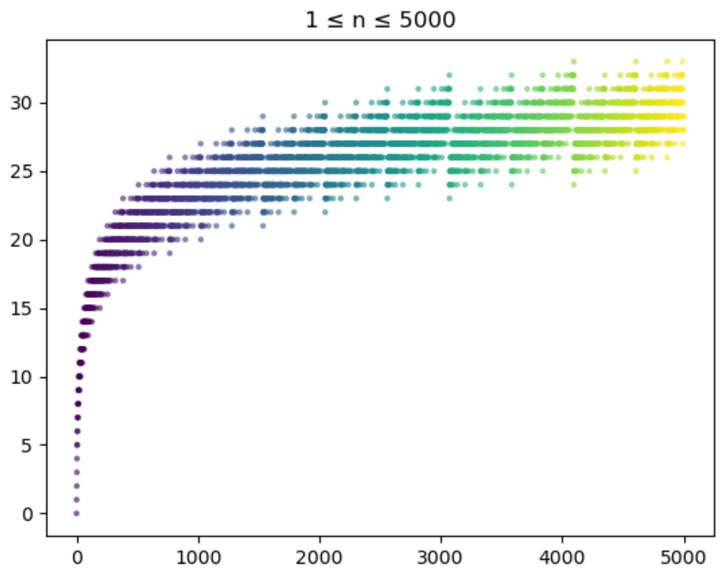
```

x = []
y = []
max = int(input("Enter Number: "))
N = max
colors = []
area = []
diction = {}

for i in range(max):
    x.append(i)
    y.append(recursion(i, 0))
    colors.append(i)
    area.append(i / 10)

title = "1     n      " + str(max)
plt.title(title)
plt.scatter(x, y, s=5, c=colors, alpha=0.5)
plt.show()
```

These are the results.

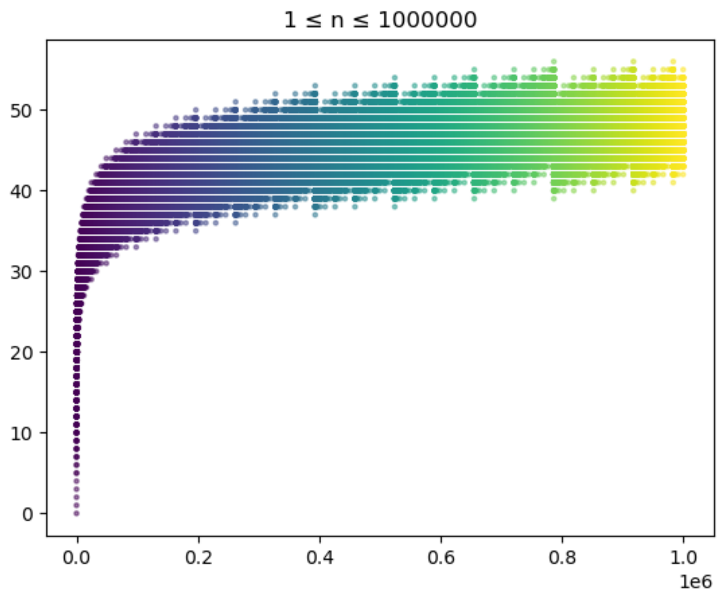
```
Enter  Number: 1000
```



```
Enter  Number: 5000
```



```
Enter  Number: 1000000
```



## What did you observe?

There is a curve when the number of n values increases.With smaller n values, 1<= n <= 10 the relationship is more linear. There is a strong and positive correlation. Although it showed in n values 1<= n <= 50 and higher we saw that there is no clear correlation with the n values and number of ones. We also notice that as the n values increase, x-axis.The more number of one values is repeated, y-axis. Took a while to generate 1,000,000 but even then the curve wasn't flattened it kept gradually increasing. What is surprising to see is how the scatter plot started off linear with smaller n values and gradually curved with n values $< 10$.

## What are your next steps?

For next week, we would like to take all of the work we done before and synthesize it to a concrete formula. We also want to start to edit the beamer template to fit with our work. Other than that, we would love to get a solid idea on how the final project should look like, and what other steps should we take to prepare ourselves.