UNITED STATES MILITARY ACADEMY

PROGRAMMING ASSIGNMENT 1:

MA386: NUMERICAL ANALYSIS

SECTION I2

COL JOSEPH LINDQUIST

By

CADET EZRA HARRIS ‘22, CO E2

WEST POINT, NEW YORK

2 SEPTEMBER 2021

_______ MY DOCUMENTATION IDENTIFIES ALL SOURCES USED AND ASSISTANCE RECEIVED IN COMPLETING THIS ASSIGNMENT.

EH I DID NOT USE ANY SOURCES OR ASSISTANCE REQUIRING DOCUMENTATION IN COMPLETING THIS ASSIGNMENT.

SIGNATURE: Ezra Harris

# Tech Lab 1

CDT Harris, Ezra

September 1, 2021

## 1 Introduction

The objective of this project was to find the sum of the digits divisible by three of the first five hundred million terms of the Fibonacci sequence. The Fibonacci sequence begins with 0 and 1 and is constructed by adding the two previous terms together. To test whether the number in the Fibonacci sequence is divisible by three it is sufficient to divide by three and observe whether it divides evenly or yields a remainder. If it divides evenly the number is added to a variable in order to keep track of the sum of the digits divisible by three.

## 2 Methods

In order to find the sum of the numbers divisible by three in the first five hundred million terms of the Fibonacci sequence it is appropriate to use a program such as python due to the large number of computations required. Beginning with an algorithm of four variables we constructed a method for finding the Fibonacci sequence and the sum of the digits divisible by three.

```
INPUT: largest natural number $n_0$ to be considered
OUTPUT: sum of the terms less than $n_0$ in the Fibonacci sequence that are divisible by three.
Step 1: set a=0, set b=1, set fibSumThree = 0
Step 3: if b is divisible by three then:
Step 3a: add b to fibSumThree
Step 4: compute the next Fibonacci number, i.e. c=a+b
Step 5: set a = b
Step 6: set b = c
Step 7: return fibSumThree
```

In order to complete this algorithm we defined a function fibSumThree which we passed the upper bound $n_0$. Then created a loop which constructed the Fibonacci sequence, tested whether the numbers were divisible by three, and added them to variable fibSumThree while the iteration was below the upper bound. Finally, we established two arrays, x and y, to keep track of the numbers divisible by three and their sum.

## 3 Analysis and Results

The sum was 119814915. The numbers divisible by three in the first five-hundred million numbers of the Fibonacci sequence are: 3, 21, 144, 987, 6765, 46368, 317811, 2178309, 14930352, 102334155.

To compute the Fibonacci sequence we established a variable c which equaled the sum of a and b. a and b were passed initial conditions of 0 and 1. After the sum was calculated a became b and b became c. This process repeated as long as the b was less then the upper bound passed to 3. The sum of the numbers divisible by three of the first five-hundred million terms of the sequence was calculated by first testing if b is divisible by 3 using the modulo function and then upon satisfying the test added to a variable fibSum3. Upon completion of the while loop the program returned fibSum3. See Appendix A for the code.

Figure 1 is a linear graph of the sum of the numbers divisible by three of the first five-hundred million terms of the Fibonacci sequence and the Fibonacci number.

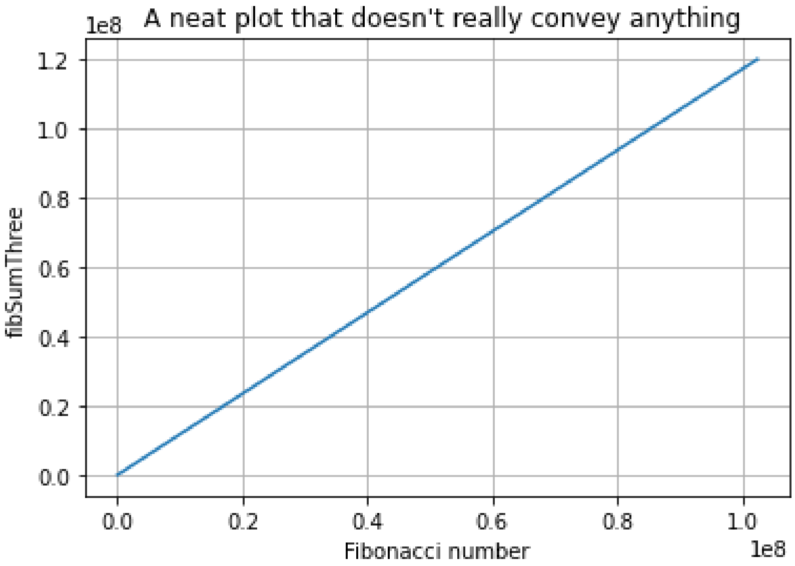

Figure 1: linear plot of sum

Figure 2 is a graph with a logarithmic scale of the sum and the Fibonacci number.

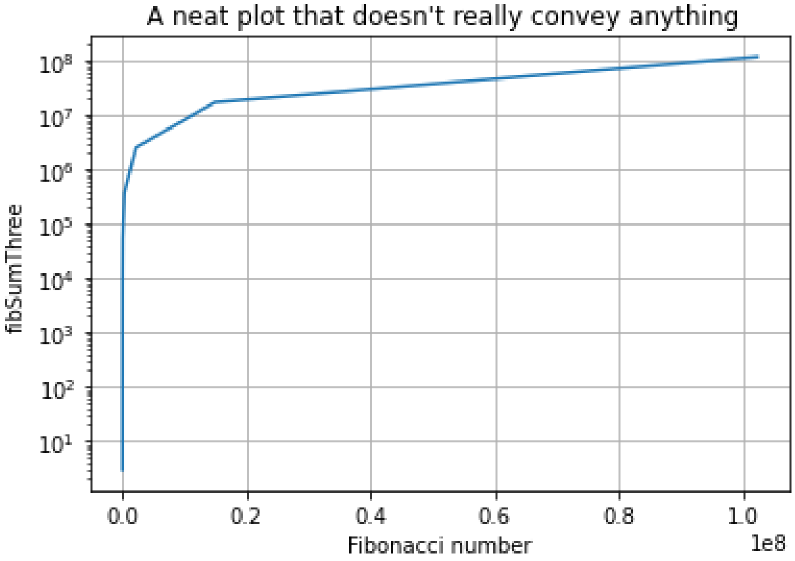

Figure 2: log plot of sum

## 4 Conclusion

This was a surprising result simply because I thought there would be more numbers divisible by three in the first five-hundred million numbers of the Fibonacci sequence. This led to a smaller sum then expected. This project gave me better insight into the Fibonacci sequence and numbers divisible by three.

# Appendix A

```
# -*- coding: utf-8 -*-
def fibSumThree(n0):

    """
Created on Mon Aug 23 07:40:53 2021

@author: Ezra

fibSumThree(n0) function to find the sum of all the terms in the
Fibonacci sequence divisible by three whih do not exceed n0.

Input: n0 is the largest natural number considered

Output: fibSumThree- the sum of the Fibonacci terms divisible by 3 that do
not exceed n0.

    """

    a=0
    b=1
    fibSum3 = 0
    while b < n0:
        if b % 3 == 0 :
            fibSum3 = fibSum3 + b

        c =a+b
        a=b
        b=c
    print("b=", "fibSum3",fibSum3)
    return fibSum3

fibSumThree(500000000)

def fibSumThreeTracker(n0):

    """
Created on Mon Aug 23 07:40:53 2021

@author: Ezra

fibSumThree(n0) function to find the sum of all the terms in the
Fibonacci sequence divisible by three whih do not exceed n0.

Input: n0 is the largest natural number considered

Output: fibSumThreeTracker-
    x: Fibonacci numbers taht are divisible by 3
    y: The sum of the Fibonacci terms divisible by 3
    that do not exceed n0
```

```
the sum of the Fibonacci terms divisible by 3 that do
not exceed n0. keeps track of intermeiate values.

    """

    a=0
    b=1
    fibSum3 = 0
    x=[]
    y=[]

    while b < n0:
        if b % 3 == 0 :
            fibSum3 = fibSum3 + b
            x.append(b)
            y.append(fibSum3)

        c =a+b
        a=b
        b=c
    return x,y
    print("b=", "fibSum3",fibSum3)
    return fibSum3

x,y = fibSumThreeTracker(500000000)
print(x)
print(y)

import matplotlib.pyplot as plt

plt.figure(0)
plt.plot(x,y)
plt.title(" A neat plot that doesn’t really convey anything")
plt.xlabel("Fibonacci number")
plt.ylabel("fibSumThree")
plt.grid()
plt.show()


plt.figure(1)
plt.plot(x,y)
plt.title(" A neat plot that doesn’t really convey anything")
plt.xlabel("Fibonacci number")
plt.ylabel("fibSumThree")
plt.grid()
plt.yscale(’log’)
plt.show()
```