

REPORTE DE PRIMOS Y FIBONACCI

Luis Daniel Honorato Hernández

Licenciatura en Matemáticas

1729448

INTRODUCCION

En este presente reporte abordaremos acerca del tema de los números primos y Fibonacci en el cual se determinará una breve descripción de lo que se quiere trabajar en sus respectivos programas además de indicar lo que se realiza en cada programa y para hacer un poco más de comprensión se añadirá una gráfica para revelar los resultados ya realizados y finalmente una conclusión acerca de lo aprendido en este reporte.

1 En este programa se trata de buscar los números primos (es decir son los números que son divisibles entre sí mismos y el uno y obtiene un residuo igual a cero) que están en el rango del 1 al 50 y además el número de operaciones que se realiza para encontrar dichos primos que este será con un contador.

cnt=0 #Se pone un contador igual a cero ya que se debe comenzar así al momento en que se realizan las operaciones y permite reanudar los contadores

def es_primo(n) :#se define el número primo

 global cnt # es una variable global ya que esta se ocupara durante todo el programa

 for i in range(2,round(n**0.5)): #aquí se empieza el ciclo con el número dos ya que si se empezara con el uno resultaría ser primo más este no lo es y termina en raíz de n ya que es más fácil comparar sus divisores que se encuentran en la raíz de n que ir comparándolos normalmente ya que se alargaría infinitamente nuestro proceso

 cnt+=1 # se van realizando la suma del contador hasta que encuentre un número primo al momento de ser encontrado se para la suma

 if n%i==0: #aquí como se explicó en el razonamiento de problema un número es primo si es divisible entre sí mismo y el uno y en especial en esta línea se da cuenta que es primo o no si el residuo da igual a cero

 break #si no es así se rompe el ciclo

 return cnt #se regresa el contador

for i in range(1,50): #se define un ciclo para encontrar los primos y sus contadores en el intervalo apreciado

 cnt=0 #se reanuda el contador

```
es_primo(i) #se guarda el número primo
```

```
print (cnt,i)# se imprimen los primos y sus respectivos contadores
```



2 En este programa se calcula los números Fibonacci (es decir una serie infinita de números como un patrón que sigue cada uno) este además trabaja con números naturales es decir números que van del 1 al infinito, se van sumando de dos a dos inicialmente desde 0 y 1 es decir $x_n = x_{n-1} + x_{n-2}$ y además se mostrara la cantidad de operaciones que se realiza con un contador (se realiza de forma recursiva)

```
def fibonacci (n):# se define nuestro fibonacci y numero a evaluar
```

```
    global cnt #se crea una variable global que se ocupara durante todo el programa
```

```
    cnt+=1 # se van realizando las operaciones del contador tomando en cuenta que este inicia en cero
```

```
    if n==0 or n==1: # si nuestro número está en el rango ya sea de 0 o 1 por regla de la sucesión Fibonacci este tendrá el valor de uno ya que si se sustituye en la formula previamente vista dará como resultado uno y son solo esos los dos números que lo cumplen
```

```
        return (1)
```

```
    return fibonacci (n-2)+fibonacci(n-1) #se regresa el valor de nuestro numero
```

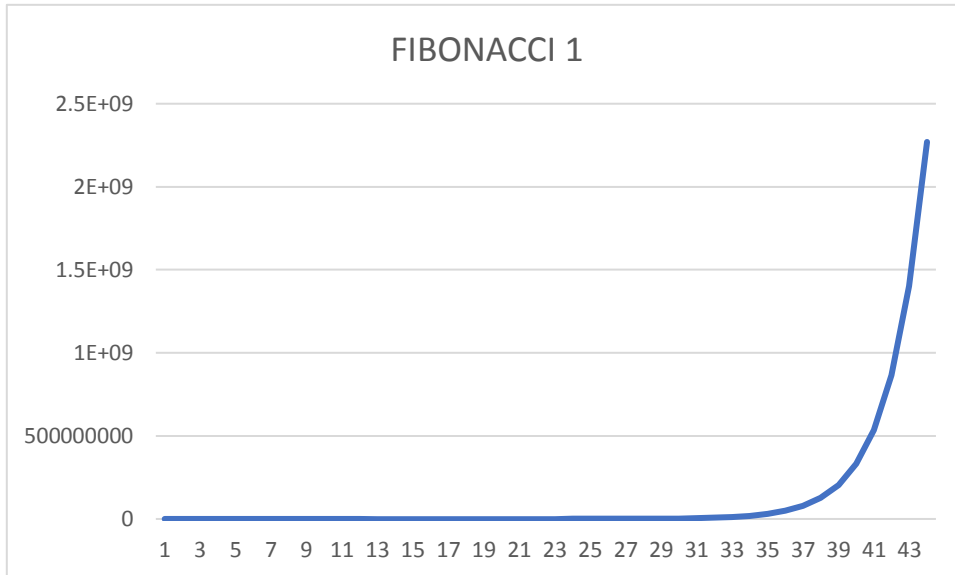
```
for i in range (1,51): #se declara un ciclo desde ese intervalo para encontrar nuestros valores de fibonacci y operaciones y se termina en 51 porque si se terminara en 50
```

no tomaría el número 50 sino hasta el 49 y es por eso que también inicia en 1 ya que si no tomaría de comienzo el número 2

```
cnt=0 #se declara nuestro contador que va a ir realizando las operaciones
```

```
fibonacci(i) #se guardan los valores de cada número
```

```
print (i,cnt) #se imprimen los valores y sus contadores de los números
```



3. Este programa es lo mismo que el anterior solo que ahora se realizara de la forma iterativa (un poco más compleja de dificultad)

```
cnt=0 # se inicializa nuestro contador igual a cero para realizar el proceso de nuestro programa además de que sirve como auxiliar para la variable global
```

```
def fibonacci(n):# se define nuestro número a evaluar en la serie Fibonacci
```

```
    global cnt #se define nuestra variable global inicializada en cero
```

```
    if n==0 or n==1:#si nuestro número se encuentra en el intervalo de 0 a 1 este tendrá un valor de 1 por lo antes dicho en el programa anterior
```

```
        return(1)
```

```
    r,r1,r2=0,1,1 #se define nuestras variables r para realizar las operaciones en el ciclo
```

```
    for i in range (2,n+1):#el ciclo empieza en dos porque si iniciara en uno este tendría valor uno y este ciclo finaliza en el número a trabajar es decir si fuese 6 sería de 2 al 6 es decir se trabajan 5 números
```

```
        cnt+=1 # se realizan las operaciones en nuestros contadores siempre iniciándose en cero
```

`r=r1+r2` #se suman los números que se encuentran en el intervalo del ciclo y de la variable ya final guardada hasta llegar al número final indicado

`r2=r1` #se guarda el valor en nuestra variable auxiliar

`r1=r` #se guarda (un cambio) en nuestra variable final

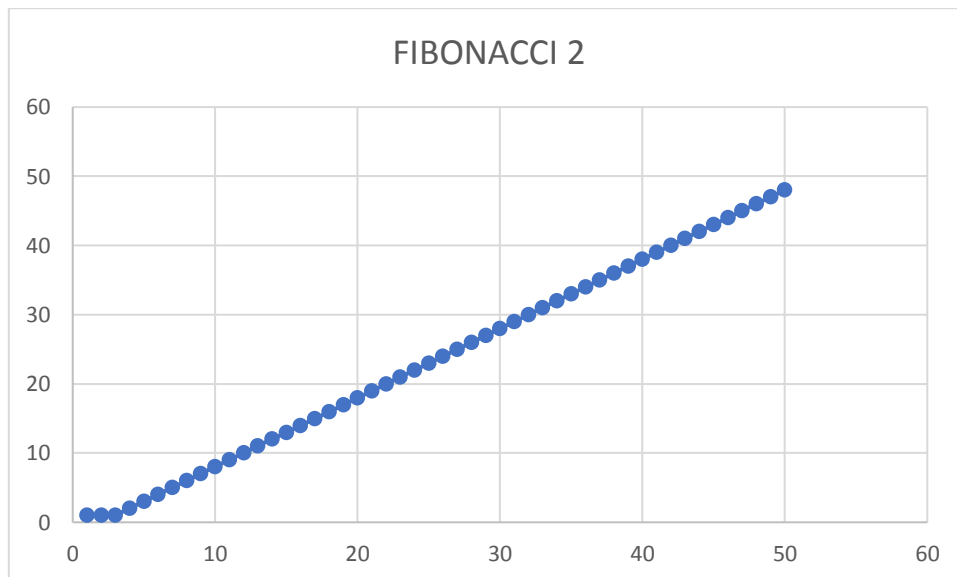
`return cnt` # se regresa el contador

`cnt=0` # y se reanuda nuestro contador

`for i in range(0,50):` #se trabaja nuestro ciclo de 0 a 50 ya que si se iniciara en 1 el primer número sería el 0 y el último el 49 y nosotros deseamos solo los primeros 50

`cnt=0` #se va reanudar nuestro contador

`print(i+1,fibonacci(i))` #se inicializa en `i+1` ya que al momento de imprimir el 1 sería `0+1=1` y se imprimen los números Fibonacci con sus valores y respectivo número de operaciones



4. Este último programa busca lo mismo que los dos anteriores acerca de los números Fibonacci solamente que estos se guardarán en un arreglo

`memo={}` #se realiza un arreglo vacío donde van guardando los números Fibonacci

`cnt=0` #se define nuestro contador igual a cero

`def fibonacci(n):` #se define nuestro número a evaluar

`global memo,cnt` #se define nuestra variable global donde esta se ocupara durante todo el programa y ocupara la variable auxiliar de contador

`cnt+=1` #se realizan las operaciones indicadas

```
if n==0 or n==1: # si el número dado esta en este intervalo dará uno como lo  
describimos en el problema 2 y 3
```

```
    return(1)
```

```
if n in memo: # si el número se encuentra en memo es decir del paso anterior
```

```
    return memo[n] #se guardará en el arreglo memo
```

```
else:
```

```
    val=fibonacci(n-2)+fibonacci(n-1)#si no es así se guarda en una variable  
    auxiliar
```

```
    memo[n]=val # y después esta estará en el arreglo memo
```

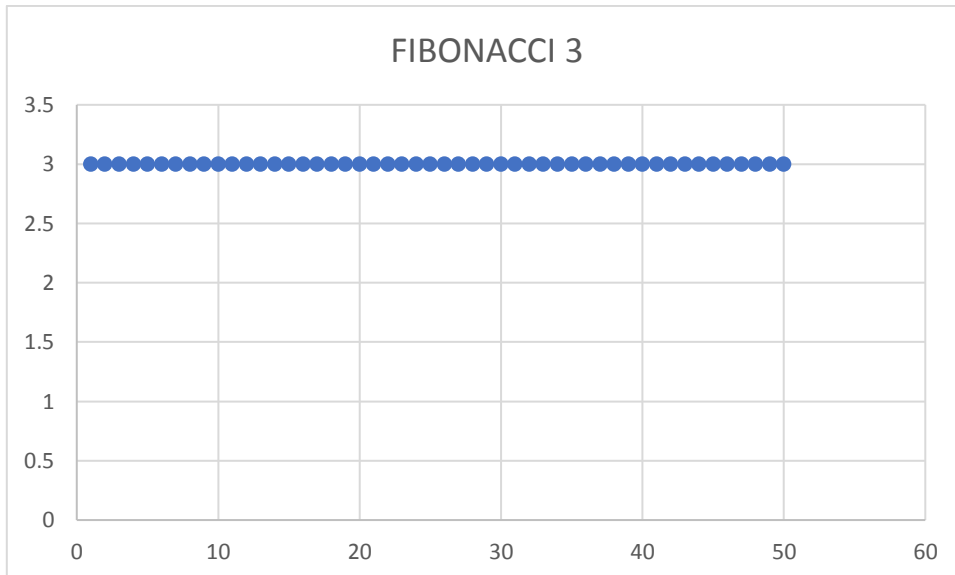
```
    return val # se regresa val
```

```
for i in range(0,50): # se define nuestro ciclo de 0 a 50
```

```
    cnt=0 #se define el contador igual a cero
```

```
    fibonacci(i)# se guarda el valor de fibo
```

```
    print(i,cnt) #se imprime el valor y el número de operaciones
```



CONCLUSION

En este presente trabajo he podido retroalimentar algunas cosas vistas acerca de la materia de Inducción a la teoría de números además ahora ya sé cómo estos se trabajan en el momento de la programación y antes yo pensaba cuando al momento de que estaba viendo lo de la sucesión Fibonacci que al momento de hacerla de manera recursiva pensé que sería más fácil que al momento de hacerlo en forma iterativa pero al momento de hacerlos fue al revés pero hay algunas excepciones