Punto 1

#########################################

#### Calculo de notacion O(n) ###########

# T(n) = 1+2+....+n = (n/2)(n+1) = (n^2/2)(n/2)

# O(n^2)

########################################

#### Pruebas ###########################

## n = 2 f(n) = 3

## n = 3 f(n) = 6

## n = 4 f(n) = 10

def algoritmo(n,A):

fn = 0

j = 0

suma = 0

for i in range (n):

for j in range (n):

suma += A[i][j]

fn += 1

j = i

print("La suma de la submatriz triangular es ", suma)

print("Minimo de operaciones requeridas para la matriz de tamano ", n, " es ", fn)

print("En notacion es O(n)")

if \_\_name\_\_== '\_\_main\_\_':

A = []

print("Ingrese el valor n ")

n = int(input(">"))

print("Ingrese los valores de la matriz de tamano n por filas")

for i in range (n):

A.append([])

for j in range (n):

aux = int(input(">"))

A[i].append(aux)

algoritmo(n,A)

Punto 2

import math

def f(x):

suma = x + 2

log = math.log(suma)

return log

def g(x):

return math.sin(5)

def formula\_recursiva():

error = 1\* pow(10,-7)

erroract = 0

xn = 1

xn1 = 0

xn2 = 0

while(erroract < error):

xn2 = xn1

xn1 = xn

aux1 = (f(xn1)\*(xn1-xn2))

aux2 = (f(xn1) - f(xn2))

division = aux1 / aux2

xn = xn-1 - (division)

erroract = (xn1-xn)

print(erroract)

print(xn)

if \_\_name\_\_== '\_\_main\_\_':

formula\_recursiva()

Punto 3

from simpy import \*

x = Symbol('x')

#funciones

f = 3 + (3 + x)\* maht.exp(3+x)

g = x + (4)

Matriz = Matrix([f,g,h,i])

InversaJacobiano = (Matriz.jacobian([x,y,z]))\*\*(-1)

Solucion = Matrix([1,1,1,1])

while (Matriz.subs([(x,Solucion[0]),(y,Solucion[1]),(z,Solucion[2])]).norm()>1E-10):

Solucion = Solucion - (InversaJacobiano.subs([(x,Solucion[0]),(y,Solucion[1]),(z,Solucion[2])]))\*(Matriz.subs([(x,Solucion[0]),(y,Solucion[1]),(z,Solucion[2])]))

print("x = ", str(s[0]))

print("y = ", str(s[1]))