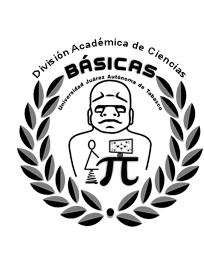
******UNIVERSIDAD JUÁREZ AUTONOMA DE TABASCO**

**DIVISIÓN ACADÉMICA DE CIENCIAS BÁSICAS**

PROGRAMA EDUCATIVO

**LIC. CIENCIAS COMPUTACIONALES**

PROFESOR

**DR. ABDIEL EMILIO CACERES GONZALEZ**

EXPERIENCIA EDUCATIVA

**ANALIZIS DE ALGORITMOS**

TRABAJO

**TAREA 2**

ESTUDIANTE

**RODRIGUEZ TORRES KEVIN NICK**

**CARDENAS, TAB. 20 DE MARZO DEL 2021**

**“EJERCICIOS”**

1. Los siguientes ejercicios se tratan de logaritmos:

a) Escribe la forma logarítmica de = 64, menciona cual es la base, la potencia

y el exponente.

2 = base de logaritmo, 6 = potencia del logaritmo, 64 = exponente

b) Escribe la forma exponencial de , menciona la base.

= 36 6 = base.

= 1.817

c) Aplica las leyes de los logaritmos para resolver log3(x/x+2)

Aplicamos la propiedad división de logaritmos la cual es: -

2. Calcula el valor de las siguientes sumas:

3. Mediante un programa en Python, grafica las funciones 4n y 3n + 8 en una sola

grafica. Permite que n tome valores desde 0 a 100. Coloca etiquetas para cada

función, y coloca un punto en las coordenadas h8; 32i, que representa la constante

n0. Consejo: Puedes utilizar matplotlib para hacer esta gráfica, puedes guiarte de

<https://matplotlib.org/tutorials/introductory/pyplot.html>

CODIGO:

import matplotlib.pyplot as plt

import matplotlib.patches as mpatches

x = []

y = []

y2 = []

proxies = []

legend\_names = ['4n','3n+8']

for n in range(0,100):

    x.append(n)

    y.append(4\*n)

    y2.append((3\*n)+8)

plt.plot([8],[32],'ro')

plt.stem(x,y2, markerfmt = "green")

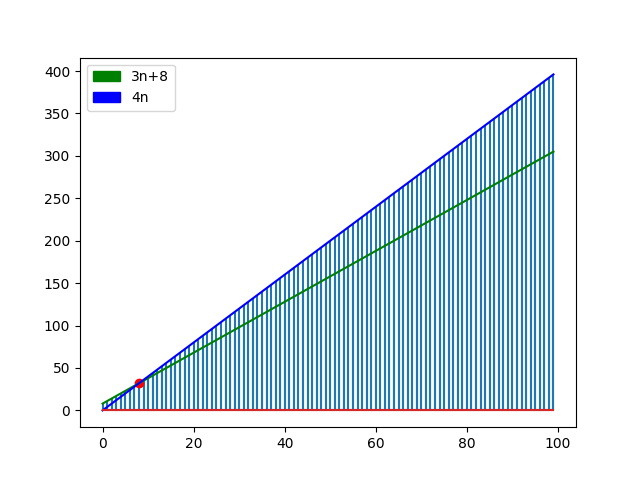
plt.stem(x, y,  markerfmt = "blue",label="4n")

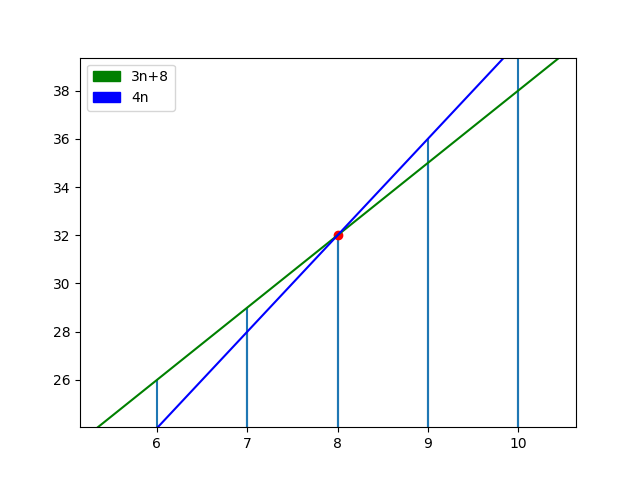
red\_patch = mpatches.Patch(color='green', label='3n+8')

red\_patch2 = mpatches.Patch(color='blue',label='4n')

plt.legend(handles=[red\_patch,red\_patch2])

plt.show()





4. Demuestra que f (n) = 6n - 9 pertenece a la clase O(n).

6n-9 7n, 6n-5n 9, n 9, C = 5, n0 = 9

5. Demuestra que f (n) = 2 pertenece a la clase ().

cg(n) 2

6. Demuestra que f (n) = 􀀀 pertenece a la clase ().

7. Considera la siguiente función en Python:

def busqueda\_lineal (k, lst):

j = len (lst) - 1

encontrado = False

while not (encontrado) and j > 0:

if lst[j] == k:

encontrado = True

j = j - 1

return encontrado

¿En qué condiciones se realizan más operaciones?

Si el valor buscado no esta o esta al inicio

¿En qué condiciones se realizan menos operaciones?

Si el valor buscado esta al final.

Si |lts| = n y considerando el peor caso:

1. ¿Cuántas asignaciones se hacen?

3 + n

1. ¿Cuántas operaciones aritméticas se hacen?

1 + n

1. ¿Cuantas operaciones lógicas se hacen?

3 + n