**UNIVERSIDAD DEL NORTE**

DIVISIÓN DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

ALGORITMOS Y COMPLEJIDAD

CÓDIGO: 200055832

NOMBRE: Luis Sebastian Caicedo Pimienta

FECHA : 11 / 02 / 2016.

**Orden de crecimiento del tiempo de corrida de un programa.**

Suponga que un problema es solucionado por tres algoritmos cuyos tiempos de corrida son T1(n)= n2; T2(n)= n \* Ln(n); y T3(n)= 2n, suponiendo que la unidad de tiempo del modelo es el µ seg.

1. Comparar los tiempos de T1(n)= n2 versus T2(n)= n \* Ln(n); para un valor de n= 1024

R/: T1(n)= n2 = 1024\*1024\*10-6 = (1024×1024)/10^6 = 1.04 seg.

T2(n)= n \* Ln(n)= 1024 \* Ln(1024) \* 10^-6 = 0.00709 seg.

1. Comparar los tiempos de T1(n)= n2 versus T2(n)= n \* Ln(n) para un valor de n= 65535.

## R/: T1(n)= n2= (65535)2 \* 10-6 = 65535^2/10^6= 4294.836225 seg / 3600s= 1.19 horas.

T2(n)= n\*Ln(n) = 65535\* Ln(65535) \* 10-6 = (65535 log(65535))/10^6 = 0.7268 seg.

1. Compara los tiempos de T2(n)= n \* Ln(n) versus T3(n)= 2n; para un valor de n= 48, n= 1024

R/:

Para n= 48

T2(n)= n \* Ln(n)= 48 \* Ln(48) \* 10-6 = (48 log(48))/10^6= 0.00018 seg.

T3(n)= 2n= 248 \* 10-6 = 2.8x108 seg =78187 horas = 3257 dias = 8.92 años

Para n = 1024

T2(n)= n \* Ln(n)= 1024 \* Ln(1024) \* 10-6 = 0.007 seg.

T3(n)= 2n= 21024 \* 10-6 = 1.79x10302 seg= 4.99x10298 horas= 2x10297 dias= 5.7x10294 años

1. definidas como:

T(n) = (n + 1)2 y f (n)= n2

Desmostrar que

R/. Se debe demostrar que P

Elegimos dos constantes c= 4 y = 1

Ordene las siguientes funciones en orden de complejidad creciente Ln(n), 100\*n, 3n, k, donde k es una constante negativa, representativa del tiempo de corrida de un conjunto de programas.

R/ No es posible ordenar debido a que las funciones no pueden ser negativa,