

Martínez Casimiro Nestor Hugo 2AM1

Un algoritmo de impresión de información tiene como peor caso de complejidad $T(4n^2)$, ¿Cuál es el tiempo de procesamiento en horas en un procesador con velocidad de 5^{10} instrucciones por segundo?

Suponiendo $n = 1$ para poder dar paso a las instrucciones

$4(10)^2 = 400$ Instrucciones, tomando en cuenta que el procesador resuelve 5^{10} instrucciones por segundo tomamos una regla de 3

$$5^{10} \text{ ----- } 1s \quad (400 \cdot 1) / 5^{10} = 0.00004096 \text{ s}$$

$$400 \text{ ----- } ?$$

Esto siendo equivalente a $1.13777777778e-8$ horas o en decimal:

0.0000000113777777778 horas

Compare el resultado contra el tiempo (en horas) que toma un algoritmo que resuelve el mismo problema, pero que tiene como peor caso de complejidad $T(75n \log n)$ y que se ejecuta sobre el mismo procesador

Suponiendo $n = 10$

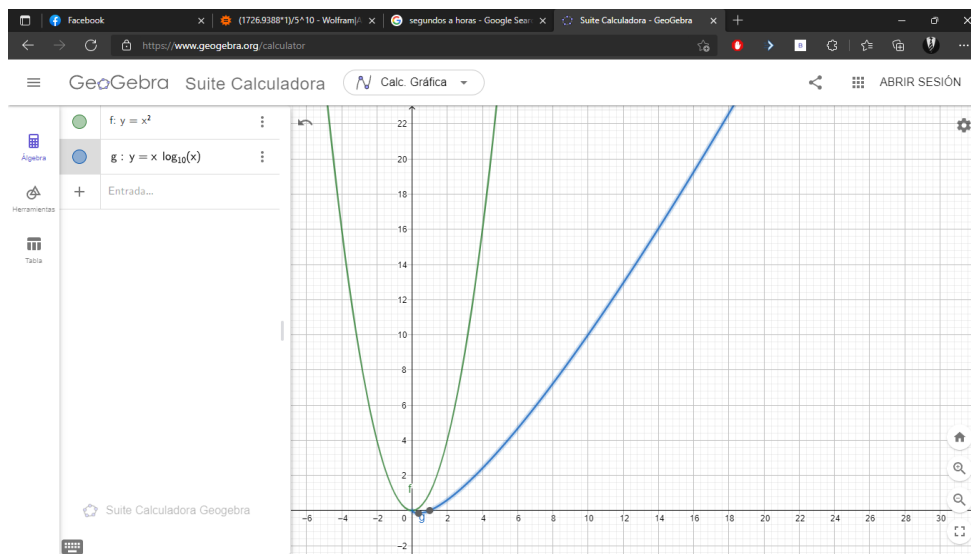
$$75(10)\log(10) = 1726.9388 \text{ instrucciones}$$

$$5^{10} \text{ ----- } 1s \quad (1726.9388 \cdot 1) / 5^{10} = 0.00017683853312 \text{ s}$$

$$1726.9388 \text{ ----- } ?$$

Siendo equivalente en horas a $4.91218147555555611e-8$

Graficando los dos algoritmos tenemos

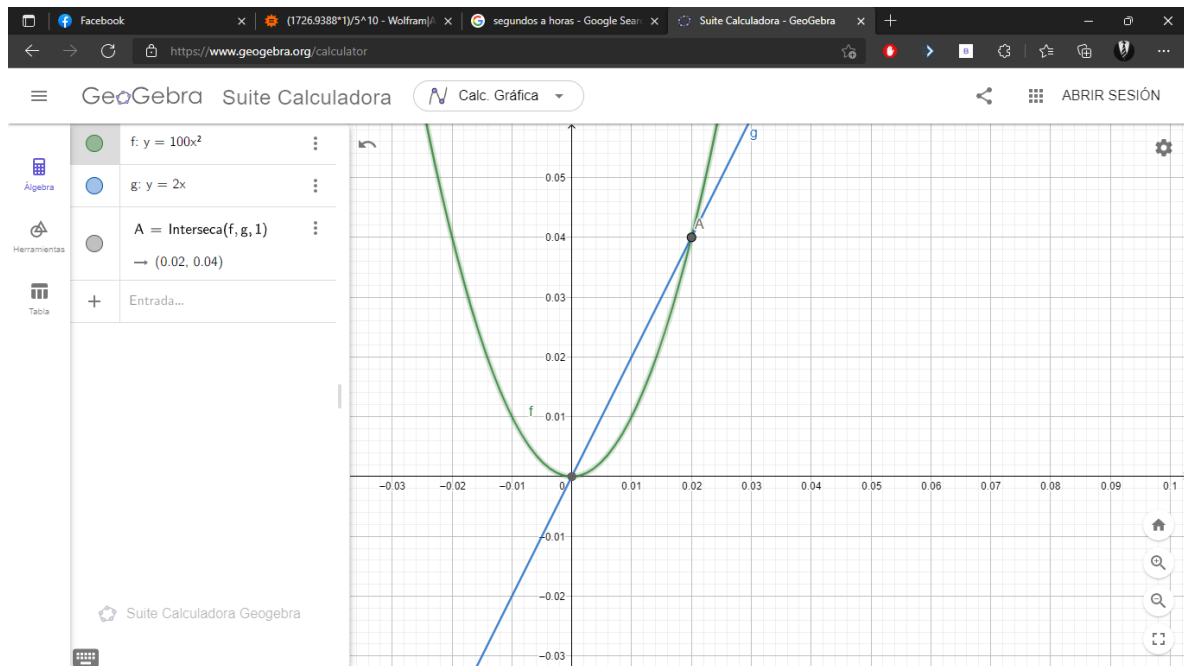


Martínez Casimiro Nestor Hugo 2AM1

Como podemos ver entre más datos las instrucciones son de un número más razonable en la complejidad logarítmica, lo que significa que el segundo algoritmo es más eficiente.

• ¿Cuál es el mínimo valor de “n” tal que un algoritmo que tiene complejidad $O(100n^2)$ se ejecute más rápido que un algoritmo que tiene complejidad $T(2n)$ sobre el mismo equipo de cómputo?

Graficando y tomando detalle tenemos



Donde n va de $[0, 0.02)$

• Para cada complejidad $O(n)$ y tiempo “t” de la siguiente tabla, determine el valor más grande para “n” de un problema que puede ser resuelto en tiempo “t”, asuma que el algoritmo que resuelve el problema le toma $O(n)$ segundos.

$O(n)/t$	1 minuto (60)	1 día (86400)	1 año (3.156×10^7)	1 siglo (3.154×10^9)
$n \log n$	245.330	982086.591	5.44959×10^8	6.8984×10^{10}
n	60	86400	3.156×10^7	3.154×10^9
n^3	216000	$6.44972544 \times 10^{14}$	3.143×10^{22}	3.137×10^{28}
$n!$	60!	(86400)!	$(3.156 \times 10^7)!$	$(3.154 \times 10^9)!$
$n^2 + n$	$60^2 + 60$	$86400^2 + 86400$	$3.156 \times 10^7^2 + 3.156 \times 10^7$	$3.154 \times 10^9^2 + 3.154 \times 10^9$

Determine las complejidades algorítmicas de las funciones implementadas en la practica 1 (crearTDA,destruirTDA, ImprimirTDA, copiarTDA).

(La práctica es mal xd)

1.- Crear TDA

```
void crearTDA(colonia *apuntador){
    int i; //1
    printf("Llenar casa 1 o 2?: "); //1
    scanf("%d",&i); //1

    printf("\nIngrese la altura: "); //1
    scanf("%d",&apuntador->casas[i-1].medias[0]); //1
    printf("Ingrese metros cuadrados: "); //1
    scanf("%d",&apuntador->casas[i-1].medias[1]); //1

    printf("Precio contado: "); //1
    scanf("%f",&apuntador->casas[i-1].precios[0]); //1
    printf("Precio credito: "); //1
    scanf("%f",&apuntador->casas[i-1].precios[1]); //1

    printf("Recamaras: "); //1
    scanf("%d",&apuntador->casas[i-1].recamaras); //1
    printf("\n"); //1
    //Total: 14
}
```

$T(n) = 14$ $O(n) = 1$

2.- Destruir TDA

```
void destruirTDA(colonia *original, colonia *borradora){
    *original = *borradora; //1
    printf("Estructura destruida\n\n"); //1
    //Total 2
}
```

$T(n) = 2$ $O(n) = 1$

3.- Imprimir TDA

```
void imprimirTDA(colonia *pasado){
    int i; //1
    printf("\nImprimir casa 1 o 2?: "); //1
    scanf("%d",&i); //1

    printf("\nAltura: %d",pasado->casas[i-1].medias[0]); //1
    printf("\nMetros cuadrados: %d",pasado->casas[i-1].medias[1]); //1
    printf("\nPrecio contado: %.3f",pasado->casas[i-1].precios[0]); //1
    printf("\nPrecio a credito: %.3f",pasado->casas[i-1].precios[1]); //1
    printf("\nRecamaras: %d",pasado->casas[i-1].recamaras); //1
    printf("\n\n"); //1
    //Total: 9
}
```

$T(n) = 9$ $O(n) = 1$

Martínez Casimiro Nestor Hugo 2AM1

Copiar TDA

```
void copiarTDA(colonia *original,colonia *copia){  
    *copia = *original; //1  
}
```

$T(n) = 1$ $O(n) = 1$

La siguiente practica la “rehíce” a partir de la que usted mando para entenderle mejor.

```
void construirTDA(salon *recibido, biblioteca* recibida, escuela* instituto){ //total: 2+n  
    int i; //1  
    for(i=0;i<cantidadSalones;i++) //  
        instituto->salones[i] = recibido[i]; //n*1  
  
    instituto->biblio = recibida[i]; // 1  
}
```

$T(n): 2 + n$ $O(n): n$

```
void imprimirTDA(escuela universidad,int a){ // total: 4 + n  
  
    int i; //1  
    printf("\nEscuela:"); //1  
    for(i=0;i<a;i++) //n*1  
    {  
        printf("\nSalon %d\nAncho: %.3f\tAlto: %.3f\tMesas: %d\tSillas %d",i,universidad.salones[i].ancho,universidad.salones[i].alto,unive  
    }  
  
    printf("\n\nBiblioteca:\n"); //1 y la de abajo 1  
    printf("Anaqueles: %d\tSillas: %d\tComputadoras: %d\n",universidad.biblio.anaqueles,universidad.biblio.sillas,universidad.biblio.computad
```

$T(n): 4+n$ $O(n): n$

```
void destruirTDA(escuela* instituto){ //1  
    free(instituto);  
}
```

$T(n): 1$ $O(n): 1$