

Proyecto 1  
Análisis de Algoritmos  
Primer Semestre 2021, Prof. Cecilia Hernández

**Fecha Inicio: Viernes 16 de Abril 2021.**

**Fecha Entrega: Viernes 7 de Mayo 2021 (23:59 hrs).**

1. [0.5 puntos] Determine si las siguientes afirmaciones son verdaderas o falsas. En cada caso, muestre los valores de las constantes  $n_0$  y  $c$  que hacen cierta su afirmación.
  - a)  $4^n \in \omega(4^{\frac{n}{2}})$
  - b)  $5 \log_4(\log(\log(n^{100}))) \in O(\log \log(n))$
  - c)  $4^{\log(n)} \in o(n^2)$
  - d)  $2n - 2\sqrt{n} \in \Theta(n)$
  - e) Si  $f(n) \in \Theta(g(n))$  entonces  $g(n) \in \Theta(f(n))$
2. [0.5 puntos] Ordene de menor a mayor orden asintótico las siguientes funciones.
  - a)  $n\sqrt[3]{n}$
  - b)  $4000^{5^{1000000}}$
  - c)  $3^{0,001n}$
  - d)  $4^{\log(n)}$
3. [0.5 puntos] Resuelva las siguientes recurrencias
  - a)  $T(n) = 3T(n/2) + cn$
  - b)  $T(n) = 7T(n/2) + cn^2$
  - c)  $T(n) = 3T(2n/3) + cn$
  - d)  $T(n) = 16T(\sqrt{n}) + \log_5(n)$
4. [0.5 puntos] Construya los árboles recursivos para las siguientes recurrencias y estime la solución de la recurrencia. Note que se pide solo la estimación.
  - a)  $T(n) = 2T(n/2) + \log(n)$
  - b)  $T(n) = 2T(n/2) + n^2$
5. [0.5 puntos] Resuelva las siguientes recurrencias usando el método de substitución.
  - a)  $T(n) = 4T(n/2) + n$
  - b)  $T(n) = T(n/2) + c$

6. [1 punto] Proporcione un análisis asintótico de peor caso en notación  $O()$  para el tiempo de ejecución de los siguientes fragmentos de programa.

(a)

```
for( int i = 1; i <= n; i *= 2 ) {  
    for( int j = 0; j < i; j++ ) {  
        for( int k = 0; k < n; k += 2 ) {  
            f(); // O(1)  
        }  
        for( int j = 1; j < n; j *= 2 ) {  
            g() // O(1)  
        }  
    }  
}
```

(b)

```
for ( i=1; i < n; i *= 2 ) {  
    for ( j = n; j > 0; j /= 2 ) {  
        for ( k = j; k < n; k += 2 ) {  
            sum += (i + j * k );  
        }  
    }  
}
```

(c)

```
int F(int n) {  
    for (i = 0; i < n*n; i += 2) {  
        procesar(i); // O(n)  
    }  
    if (n <= 0)  
        return 1;  
    else  
        return F(n-3);  
}
```

(d)

```
void G(a[], n){  
    if( n <= 1)  
        return;  
    else  
        for(i=1; i<= n-1; i++){  
            if (a[i] > a[n] )  
                intercambiar(a[i], a[n])  
        }  
    G(a[], n-1)  
}
```

7. [1.0 punto] Asuma los algoritmos dados en las funciones  $F()$  and  $G()$ . Determine que realizan, demuestre que son correctos y determine su complejidad asintótica.

```
int F(a,b)
  x=a, y=b, z = 0
  while( x > 0)
    z = z + y
    x = x - 1
  return z
```

```
int G(a,b)
  x=a, y=b, z = 0
  while( x > 0)
    if (x % 2 == 1)
      z = z + y
    x = x>>1
    y = y<<1
  return z
```

8. [1.5 puntos] Considere el algoritmo mergesort basado en la siguiente función recursiva, donde  $A$  es un arreglo que contiene elementos de tipo float.

```
void mergesort(float *A, int l, int r) {
  if(l < r){
    int m = floor((l+r)/2);
    mergesort_serial(A, l, m);
    mergesort_serial(A, m+1, r);
    merge(A, l, m, r);
  }
}
```

Para el algoritmo dado se pide lo siguiente:

- Modifique el algoritmo mergesort dado para que en lugar de dividir el arreglo en 2 lo divida en 3 y bosqueje su operación con un ejemplo.
- Modifique el algoritmo mergesort dado para que en lugar de dividir el arreglo en 2 lo divida en 4 y bosqueje su operación con un ejemplo.
- Determine las recurrencias y resuélvalas para los dos algoritmos.
- Implemente los tres algoritmos en C/C++ y realice un análisis experimental midiendo tiempo de ejecución para distintos tamaños del arreglo  $A$ . Note que su implementación debe soportar tamaños de  $A$  grandes, es decir, sobre 10 millones de elementos. Grafique y analice sus resultados.

9. Medición de tiempos.

```
#include <chrono>

using namespace std;

auto start = chrono::high_resolution_clock::now();
auto finish = chrono::high_resolution_clock::now();
auto d = chrono::duration_cast<chrono::nanoseconds>(finish - start).count();
cout <<"total time "<< duration << " [ns]" << " \n";

// nota, si estima necesario puede usar milliseconds
// en lugar de nanoseconds
```