

## Taller 1

# Fundamentos de análisis y diseño de algoritmos

Carlos Andres Delgado S, Ing \*

Septiembre 2017

### 1. (15 %) Comparación de tiempos de ejecución

Para cada función  $f(n)$  y tiempo  $t$  en la siguiente tabla, determine el mayor tamaño de  $n$  de un problema que puede ser resuelto en tiempo  $t$ , asumiendo que el algoritmo para resolver el problema toma  $f(n)$  microsegundos.

	1 seg	1 min	1 hora	1 día	1 mes	1 año	1 siglo
$\log n$							
$\sqrt{n}$							
$n$							
$n \log n$							
$n^2$							
$n^3$							
$2^n$							
$n!$							

### 2. (25 %) Realice los siguientes ejercicios del libro de Cormen:

- 1.1-4
- 1.1-5
- 1.2-3
- 2.1-3
- 2.1-4

### 3. (25 %) Implemente el algoritmo **insertion sort**.

Utilizando el comando time de linux [https://es.wikipedia.org/wiki/Time\\_\(Unix\)](https://es.wikipedia.org/wiki/Time_(Unix)) tome el tiempo promedio de 20 ejecuciones para cada caso de ordenamiento: 10 números, 100 números, 1000 números, 10000 números y 100000 números. Genere estos números aleatoriamente para estudiar el caso promedio. Grafique los tiempos de ejecución. ¿Como crece el tiempo de ejecución de acuerdo al tamaño de la entrada?.

### 4. (35 %) Dar la complejidad en términos de $n$ . Analizar el mejor caso, caso promedio y peor caso. Suponga que $n > 0$ . Adicionalmente, encuentre

- Representación de estado
- Estado inicial
- Transformación de estados
- Invariante de ciclo
- Estado final

**Importante:** Demuestre la correctitud del algoritmo utilizando la invariante de ciclo.

```

1 //a: arreglo de tamaño n
2 //n: tamaño del arreglo a
3 //Primera posición arreglos es 0
4 algoritmo1(a, n)
5     i = 0
6     b = 0
7
8     while(i < n)
9         j = 0
10        while(j < n)
11            s = a[j]
12            b = b + s*i
13            while(a[i] < a[j])
14                t = a[j]
15                a[j] = a[i]
16                a[i] = t
17            end
18            j++
19        end
20        i++
21    end
22
23    print b
24 end

```

```

1 algoritmo2(n)
2     b = 0
3     c = 1
4     j = n
5     while j < n*n
6         a = b
7         b = c
8         c = b + a
9         j+=n
10    end
11    print c

```

\* carlos.andres.delgado@correounivalle.edu.co