# Analisis y Diseño de Algoritmos

Juan Gutiérrez

September 6, 2021

# Resumen

# ¿Qué es un algoritmo?

# Técnicas a estudiar en el curso

# Introducción al análisis de eficiencia



**Ejemplo 2.** *Tenemos dos computadores A y B con las siguientes características y algoritmos a ejecutar:*

- *Computador A:*
  - *Velocidad:* $10^{10}$ *instrucciones/s,*
  - *Algoritmo a ejecutar: Insertion sort, cuyo tiempo de ejecucion es* $2n^2$ *(es decir, con entrada de tamaño* $n$*, ejecuta* $2n^2$ *instrucciones),*
- *Computador B:*
  - *Velocidad:* $10^7$ *instrucciones/s,*
  - *Algoritmo a ejecutar: Merge sort, cuyo tiempo de ejecucion es* $50n\lg(n)$

- Si ordenamos $10^7$ números tenemos que ...
- Si ordenamos $10^8$ números tenemos que ...

# Ejercicios



**Ejercicio 1.** *Suponga que en una misma computadora se corre Insertion sort y Merge sort, donde Insertion sort corre en* $8n^2$ *pasos con entrada de tamaño* $n$ *y Merge sort corre en* $64n \lg n$ *pasos. Para que valores de* $n$*, es más eficiente el Insertion sort?*

# Ejercicios



**Ejercicio 2.** *Cual es el menor valor de* $n$ *tal que un algoritmo con tiempo de ejecucion* $100n^2$ *es más rapido que uno con tiempo de ejecucion* $2^n$ *en la misma máquina.*

**Ejercicio 3.** *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 $t$, suponiendo que el algoritmo para resolver el problema toma $f(n)$ $\mu s$*

| | 1 second | 1 minute | 1 hour | 1 day | 1 month | 1 year | 1 century |
|---|---|---|---|---|---|---|---|
| $\lg n$ | | | | | | | |
| $\sqrt{n}$ | | | | | | | |
| $n$ | | | | | | | |
| $n \lg n$ | | | | | | | |
| $n^2$ | | | | | | | |
| $n^3$ | | | | | | | |
| $2^n$ | | | | | | | |
| $n!$ | | | | | | | |

# Propiedades de logaritmos

- $\log_a b = x$ si y solo si $b = a^x$.

# Propiedades de logaritmos

- $\log_a b = x$ si y solo si $b = a^x$.
- $\log_a x^y = y \log_a x$

# Propiedades de logaritmos



- $\log_a b = x$ si y solo si $b = a^x$.
- $\log_a x^y = y \log_a x$
- $\log_a xy = \log_a x + \log_a y$

# Inducción

## Propiedad

*Para todo número natural $n$, $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$.*

# Inducción

**Propiedad**

*Para todo número natural $n$,* $1 + 2 + 2^2 + \cdots 2^n = 2^{n+1} - 1$.

# Inducción



**Propiedad**

*Para todo número natural $n$,* $\frac{1}{2} + \frac{1}{6} + \cdots + \frac{1}{n(n+1)} = \frac{n}{n+1}..$

# Inducción

**Propiedad**

*Para todo número natural n*, $\lg n \leq n$.

**Propiedad**

*Para todo número natural* $n \geq 44$, $8 \lg n \leq n$.

# Sumatorias



**Definición 5.** *Dada una secuencia* $a_1, a_2, \ldots, a_n$ *de números, donde* $n$ *es un entero no negativo, podemos escribir la suma finita* $a_1 + a_2 + \cdots + a_n$ *como*

$$\sum_{k=1}^{n} a_k.$$

# Sumatorias

**Definición 6.** *Dada una secuencia infinita $a_1, a_2, \ldots$ de números, podemos escribir la suma infinita $a_1 + a_2 + \cdots$ como*

$$\sum_{k=1}^{\infty} a_k = \lim_{n \to \infty} \sum_{k=1}^{n} a_k.$$

*Si el límite no existe, la serie diverge, si no, esta converge.*

# Sumatorias



**Propiedad de Linealidad**

Para cada número real $c$ y cualesquier secuencias $a_1, a_2, \ldots, a_n$ y $b_1, b_2, \ldots, b_n$, tenemos que

$$\sum_{k=1}^{n}(ca_k + b_k) = c\sum_{k=1}^{n} a_k + \sum_{k=1}^{n} b_k.$$

# Sumatorias

**Series aritméticas**

Son series en donde la resta de cada dos términos consecutivos es la misma. Ejemplo:

$$\sum_{k=1}^{n} k = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

# Sumatorias



**Suma de cuadrados y cubos**

$$\sum_{k=0}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}.$$

$$\sum_{k=0}^{n} k^3 = \frac{n^2(n+1)^2}{4} = (\frac{n(n+1)}{2})^2.$$

**Series geométricas**

Son series en donde la división de cada dos términos consecutivos es la misma.

Por ejemplo, para cada número real $x \neq 1$,

$$\sum_{k=0}^{n} x^k = 1 + x + x^2 + \cdots + x^n = \frac{x^{n+1} - 1}{x - 1}.$$

# Sumatorias



**Serie ármónica**

Es la serie

$$H_n = \sum_{k=1}^{n} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n}.$$

# Acotando sumatorias por inducción



Probaremos por inducción en $n$ que

$$\sum_{k=1}^{n} k \leq \frac{1}{2}(n+1)^2.$$

# Acotando sumatorias por inducción



Probaremos por inducción en $n$ que

$$\sum_{k=0}^{n} 3^k \leq c3^n$$

# Acotando sumatorias por inducción



Probaremos por inducción en $n$ que

$$\sum_{k=0}^{n} 3^k \leq \frac{3}{2} 3^n.$$

# Acotando sumatorias por términos

Podemos acotar superiomente cada término de una serie, por ejemplo

$$\sum_{k=1}^{n} k \leq \sum_{k=1}^{n} n = n^2$$

# Acotando sumatorias por términos



En general, sea $a_{\max} = \max_{1 \le k \le n} a_k$, tenemos que

$$\sum_{k=1}^{n} a_k \le \sum_{k=1}^{n} a_{\max} = n \cdot a_{\max}$$

# Acotando sumatorias por división



Acotaremos $\sum_{k=1}^{n} k$.

# Acotando sumatorias por división



Acotaremos $\sum_{k=0}^{\infty} k^2/2^k$.

# Acotando sumatorias por división



Acotaremos $H_n = \sum_{k=1}^{n} \frac{1}{k}$.

# Árboles binarios



Un *árbol binario* $T$ es definido recursivamente según:

- si $T$ no tiene nodos entonces es un árbol binario
- si $T$ tiene nodos, está compuesto por un nodo *raíz*, un árbol binario llamado *subárbol izquierdo* y un árbol binario llamado *subárbol derecho*

# Árboles binarios

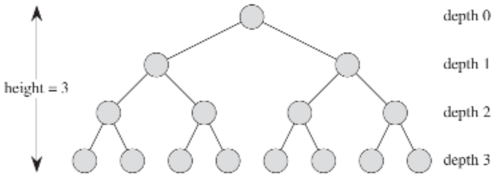


**Figure B.8** A complete binary tree of height 3 with 8 leaves and 7 internal nodes.

Figure 1: Tomada del libro Cormen, Introduction to Algorithms

Gracias