

cc00A

# Diseño y Analisis de Algoritmos

Note Title

7/28/2009

Jérémy BARBAY

## Estructura del Curso

- |                                    |   |
|------------------------------------|---|
| ① Fundamentos básicos, Complejidad | ⑦ |
| ② Técnicas básicas                 | ⑥ |
| ③ ————— avanzadas                  | ⑧ |
| ④ Algoritmos no convencionales     | ⑨ |

# Cotas Inferiores en teoría de la información

## I Codificación

① Como codificar un entero  $x$ ?

— elegir un conjunto de enteros  
(de tamaño  $n$ )

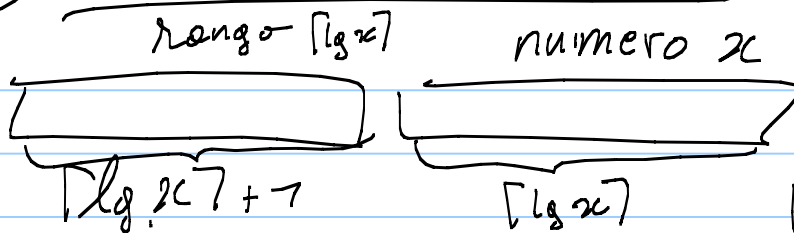
→  $\lceil \lg_2 n \rceil$  bits

$$\lg x = \log_2 x$$

## ② Codificación Unario (Unary Encoding)

$$\begin{array}{lcl} x & = & \underbrace{11\dots1}_x 0 \\ 0 & = & 0 \end{array} \left. \vphantom{\begin{array}{l} x \\ 0 \end{array}} \right\} x+1 \text{ bits}$$

③ Codificación ✓



$$1024 \leq 2009 < 2048$$

$$\lg 1024 < \lg 2009 < \lg 2048$$

$$10 < \quad < 11$$

$$x = 2009$$

$$u = \lceil \lg 2009 \rceil$$

00 --- 01  $\overline{x}^2$

$$u \text{ bits} = \lceil \lg 2009 \rceil$$

tamaño total  $2\lceil \lg x \rceil + 1$

## ⑤ Recursion

(Esta sección es opcional)

$$x \rightsquigarrow \boxed{\lceil \lg x \rceil} + \boxed{x}$$

$$2 \lg(\lceil \lg x \rceil) + 1 \quad \lg x$$

$$= \underbrace{2 \lg \lg x + 1}_1 + \underbrace{\lg x}_2$$

$$x \rightsquigarrow \boxed{\lceil \lg \lg x \rceil} \boxed{\lceil \lg x \rceil} \boxed{x}$$

$$\lg \lg x + 1 + \lg x$$

La recursion se puede repetir  
hasta que  $\lg \dots \lg x$  es pequeño.  
(en practica, muy rapidamente)

Formalmente, es definido por

$$\lg^* x \triangleq k \quad \text{if} \quad \underbrace{\lg \lg \lg \lg \dots \lg x}_k = 1$$

## II Búsqueda

① Búsqueda binaria (en arreglo ordenado)  
= codificación binaria

⑤ Busqueda en un conjunto infinito de enteros

⇒ Busqueda secuencial = Codificación Unaria  
 $0, 1, 2, 3, 4, \dots, x$  ⇒  $x + 1$  cmps

⇒ Busqueda geometrica = Doubling Search  
 $0, 1, 2, 4, 8, 16, \dots, 2^{\lceil \lg x \rceil - 1}, 2^{\lceil \lg x \rceil}$   
 $2^0, 2^1, 2^2, 2^3, 2^4, \dots$   
+ Busqueda binaria ⇒  $2^{\lceil \lg x \rceil + 1}$  cmps

### III Aplicación Cotas Inferiores

① Cuánto comparaciones necesitan para ordenar un arreglo de  $n$  enteros todos distintos, en el peor caso?

$n \lg n \rightarrow$  Porque?

$\Rightarrow$  Buscando a dentro de todas las permutaciones de  $n$ , para una en particular

$\Rightarrow$  (a) Cuántas permutaciones de  $n$ ?  
 $\Rightarrow n!$

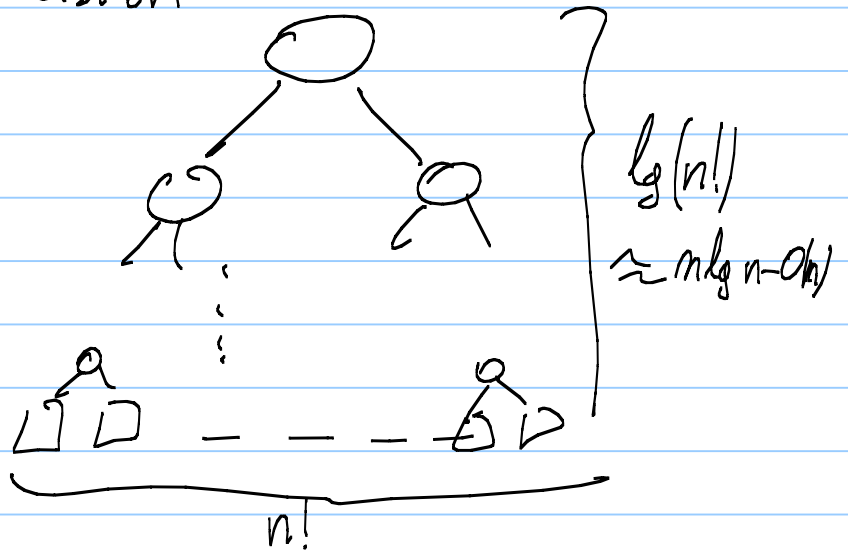
(b) Cada comparación es eliminando una mitad de los considerados

(a+b)

$\Rightarrow \lg(n!) \approx n \lg n - O(n)$

Cota(s) inferiores: dos técnicas distintas,  
que dan el mismo resultado aquí

① Árbol de decisión



## ② Teoría de la información

- Hay un conjunto de  $n!$  permutaciones
- necesitamos  $\lg(n!)$  bits para codificar una en particular
- necesitamos  $\lg(n!)$  comparaciones para identificar una.

## IV Con probabilidades

Queremos usar el hecho que algunos símbolos/objetos son más frecuente.

Busqueda  $\leadsto$  no ejemplos conocidos

Codificación  $\rightarrow$  conocemos Huffman

## Recuerdo: Huffman

Input  $n$  frecuencias  $p_1, \dots, p_n$

Output Código  $c_1, \dots, c_n$ , de longitudes  $l_1, \dots, l_n$   
libre de prefijos (prefix free)

### Algoritmo

- 1  $n$  hojas de pesos  $p_1, \dots, p_n$
- 2 tomar las dos hojas de peso minimal  $p_a, p_b$
- 3 reemplazarlas para el nodo  $c$  que tiene hijos  $a$  &  $b$ , peso  $p_c = p_a + p_b$
- 4 si hay más que un nodo  $\Rightarrow$  2

Tarea Pensar a si y como se puede usar tecnicas parecidas a Huffman para buscar en un Conjunto, cuando conocemos las probabilidades de cada objeto.

