

# CI2612: Algoritmos y Estructuras de Datos II

Blai Bonet

Universidad Simón Bolívar, Caracas, Venezuela

## Problema de ordenamiento y cálculo de estadísticos

© 2016 Blai Bonet

### Objetivos

- Introducir los problemas de ordenamiento y cálculo estadístico
- Discutir su relevancia y resultados

© 2016 Blai Bonet

### Ordenamiento

Dada una secuencia  $\{a_1, a_2, \dots, a_n\}$  de  $n$  números, un algoritmo de ordenamiento produce una permutación (reordenamiento)  $\langle a'_1, a'_2, \dots, a'_n \rangle$  de la entrada tal que  $a'_1 \leq a'_2 \leq \dots \leq a'_n$

En la práctica, pocas veces tenemos que ordenar una secuencia de números. Lo que se ordena es una **secuencia de registros** cada uno con una **clave**. Los datos que acompañan a la clave en un registro se llaman **datos satélite**

Aunque presentamos algoritmos de ordenamiento para números, un algoritmo de ordenamiento debe lidiar con los datos satélite

© 2016 Blai Bonet

## Propiedades de algoritmos de ordenamiento

Un algoritmo de ordenamiento es:

- **“in place”**: si a lo sumo un número constante de elementos de la entrada son guardados fuera del arreglo de entrada durante la ejecución
- **estable**: si dos elementos  $a_i$  y  $a_j$  que sean iguales aparecen en la salida en el mismo orden relativo al que aparecen en la entrada (i.e. si  $a_i = a_j$  e  $i < j$ , entonces los elementos  $a'_{i'}$  y  $a'_{j'}$  en la salida que refieren a  $a_i$  y  $a_j$  son tales que  $i' < j'$ )

## Importancia del problema de ordenamiento

- Muchos algoritmos fundamentales requieren hacer algún tipo de ordenamiento dentro de su ejecución
- La información se guarda usualmente de forma ordenada para hacer búsqueda de forma eficiente (cf. búsqueda binaria)
- Ordenamiento también se utiliza para remover duplicados y para construir formas canónicas de objetos
- ...

## Comparación de algoritmos de ordenamiento

Algoritmo	Tiempo en el peor caso	Tiempo promedio/esperado
Insertion-Sort	$\Theta(n^2)$	$\Theta(n^2)$
Mergesort	$\Theta(n \log n)$	$\Theta(n \log n)$
Heapsort	$\Theta(n \log n)$	$\Theta(n \log n)$
Quicksort	$\Theta(n^2)$	(promedio) $\Theta(n \log n)$
Randomized-Quicksort	$\Theta(n^2)$	(esperado) $\Theta(n \log n)$
Counting-Sort	$\Theta(k + n)$	$\Theta(k + n)$
Radix-Sort	$\Theta(d(n + k))$	$\Theta(d(n + k))$
Bucket-Sort	$\Theta(n^2)$	(promedio) $\Theta(n)$

## Cálculo de estadísticos de orden

El  $i$ -ésimo estadístico de orden de un conjunto de  $n$  elementos es el  $i$ -ésimo menor elemento del conjunto

Ordenando el conjunto podemos seleccionar el  $i$ -ésimo estadístico en tiempo  $O(n \log n)$

Más adelante veremos dos algoritmos eficientes para calcular el  $i$ -ésimo estadístico:

- **algoritmo randomizado** simple que corre en tiempo esperado  $O(n)$  y tiempo  $\Theta(n^2)$  en el peor caso
- **algoritmo determinístico** que corre en tiempo  $O(n)$  en el peor caso