El problema de Ordenar

¿Qué tan difícil es ordenar un arreglo con n datos?

¿Cambia la respuesta si sabemos algo de los datos?

Según sea el caso, puede convenir usar diferentes algoritmos

Instancias de ordenación

¿Qué tan rápido podemos ordenar un arreglo si…

los datos ya están ordenados?

Instancias de ordenación

¿Qué tan rápido podemos ordenar un arreglo si…

los datos ya están ordenados, pero al revés?

Instancias de ordenación

¿Qué tan rápido podemos ordenar un arreglo si…

los datos están *casi* ordenados?

Instancias de ordenación

¿Qué tan rápido podemos ordenar un arreglo si…

los datos están separados en dos secuencias ordenadas?

Merge

Para dos secuencias ordenadas, A y B

1. Sea C una secuencia ordenada, inicialmente vacía
2. Sean a y b el primer elemento de A y B respectivamente
3. Extraer de su respectiva secuencia el menor entre a y b
4. Insertar ese elemento extraído al final de C
5. Si quedan elementos en ambos A y B, volver a 2.
6. Concatenar C con la secuencia que aún tenga elementos

Merge

¿Como demostramos que Merge es correcto?

¿Cuál es su complejidad?

Finitud

En cada paso el algoritmo extrae un elemento de A o B y se pone en C

Cuando una de las secuencias se vacía, se toma todo lo de la otra secuencia y se pone en C

En total se hacen $|A| + |B|$ pasos, y como tanto A como B son finitos, el algoritmo es finito

Correctitud

PD: Luego de insertar el último elemento en C, esta está ordenada

Por **inducción** sobre las inserciones en C

Caso Base: Luego de la primera inserción, C tiene un solo elemento x_1, por lo que está ordenada.

Hipótesis Inductiva: Luego de la i-ésima inserción del elemento x_i, C está ordenada.

Ahora toca la siguiente inserción.

- Si quedan elementos en A y en B, sea x_{i+1} el más pequeño entre las cabezas de A y de B.

- Si solo quedan elementos en una de las dos secuencias, sea x_{i+1} la cabeza de esta.

Se elimina x_{i+1} de su respectiva secuencia y se inserta al final de C.

$x_i \leq x_{i+1}$. De no ser así x_{i+1} habría salido antes, ó A y B no habrían estado ordenadas.

Como C estaba ordenada, $x_1 \leq x_2 \leq \cdots \leq x_i$, y como $x_i \leq x_{i+1}$, entonces C está ordenada.

Por lo tanto, luego de insertar el último elemento x_n, C está ordenada.

Merge

¿Podremos usar **merge** para ordenar una secuencia arbitraria?

Si de algún modo podemos crear dos secuencias ordenadas…

…podemos combinarlas, ordenando la secuencia completa

MergeSort

Para una secuencia A

1. Si A tiene un solo elemento, terminar en este paso
2. Dividir la secuencia en dos mitades iguales
3. Ordenar cada mitad recursivamente usando MergeSort
4. Combinar las mitades usando Merge

MergeSort

Demuestra que MergeSort es correcto

Calcula y justifica su complejidad

El problema de Ordenar

¿Qué tan difícil es ordenar un arreglo con n datos?

¿Existirá algún límite a qué tan rápido podemos ordenar?

Centrémonos en los algoritmos que ordenan **comparando** datos

Ordenación por comparación

Un algoritmo de ordenación por **comparación** sigue el siguiente esquema:

1. Comparar dos de los datos a ordenar
2. Reorganizar los datos según el resultado de la comparación
3. Si aun quedan comparaciones por hacer, volver a 1.

Todos los algoritmos que hemos visto son por **comparación**

Comparaciones

¿Cuantas comparaciones es necesario realizar como mínimo en el peor caso?

¿Qué nos dice ese número sobre los algoritmos por **comparación**?

¿Será posible encontrar la cantidad exacta?

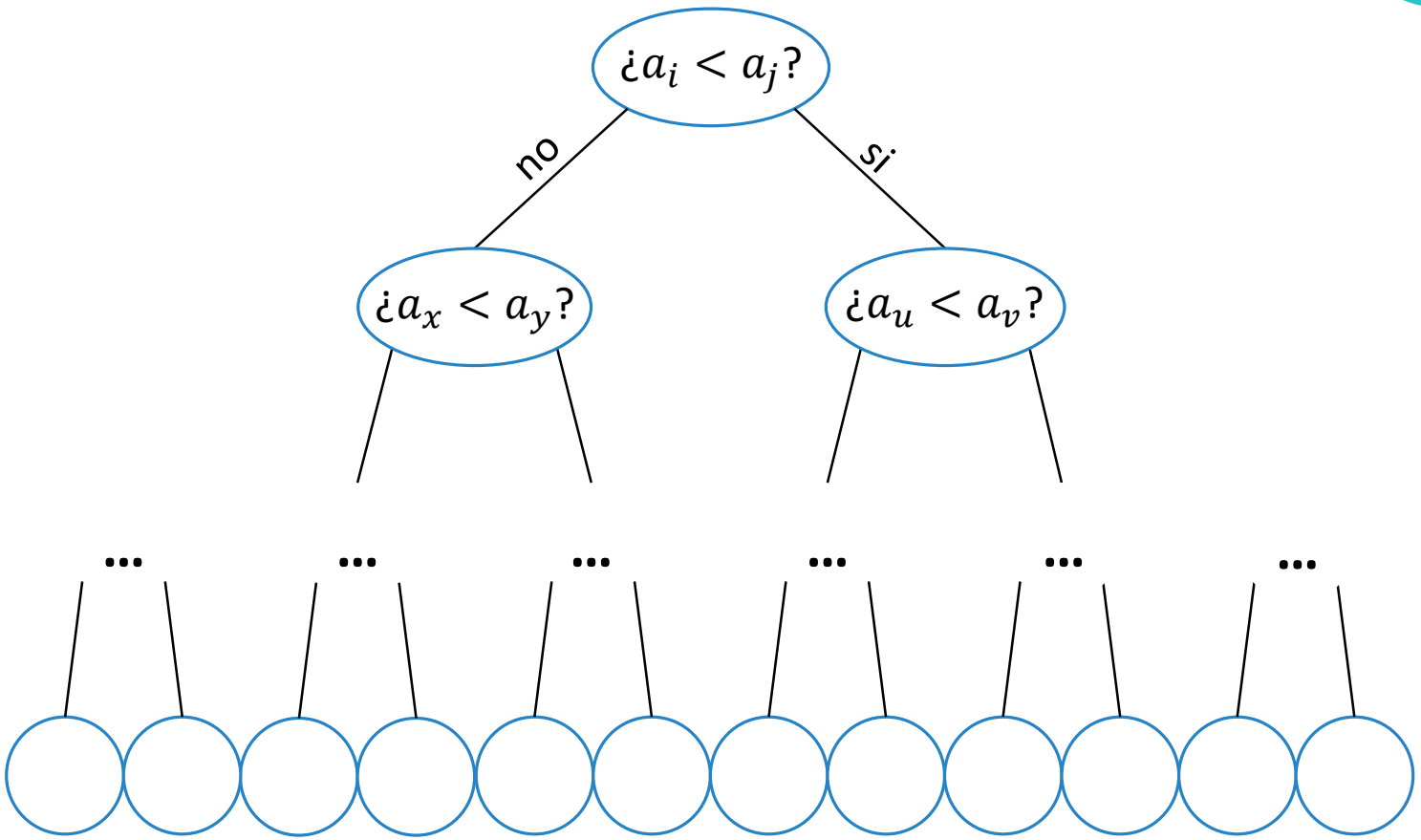
Comparaciones

Pensemos en un hipotético ***mejor algoritmo*** por comparación

¿Cuántas comparaciones debe realizar en el peor caso?

Analicemos cómo el algoritmo toma las decisiones

Árbol de decisión



¿Qué hay en las hojas?

Permutaciones

El algoritmo busca la **permutación** ordenada de los datos

¿Cuántas permutaciones distintas hay?

¿Son todas posibles respuestas del algoritmo?

Permutaciones

Inicialmente todas las permutaciones son candidato a respuesta

El algoritmo las va descartando según las comparaciones

Considerando que es el ***mejor algoritmo***, ¿Cuántas comparaciones necesita para identificar la permutación correcta?