Árboles de decisión

Gregorio Hernández Peñalver

UPM

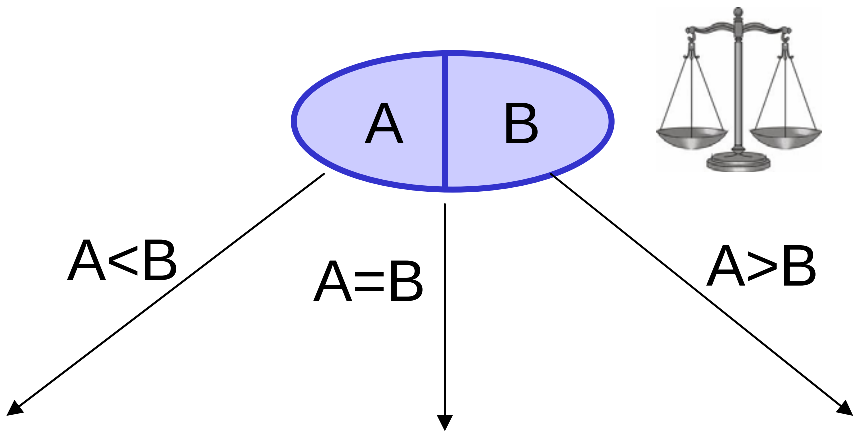
Matemática Discreta II

(MI)

Árbol de decisión

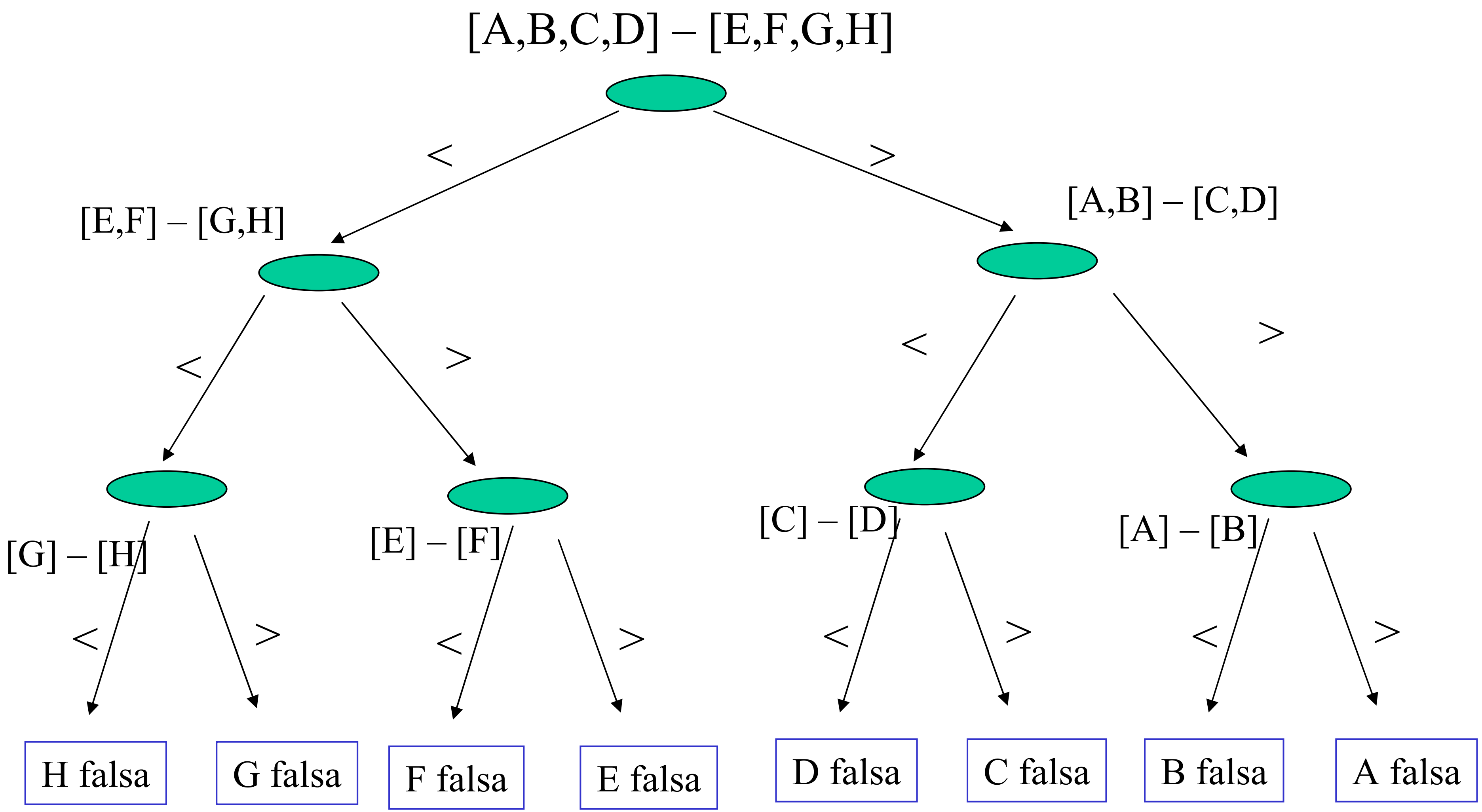
En cada vértice se toma una decisión D. Los arcos que salen de D corresponden a cada una de las opciones posibles de D

Por ejemplo, se pesan dos monedas A y B en una balanza



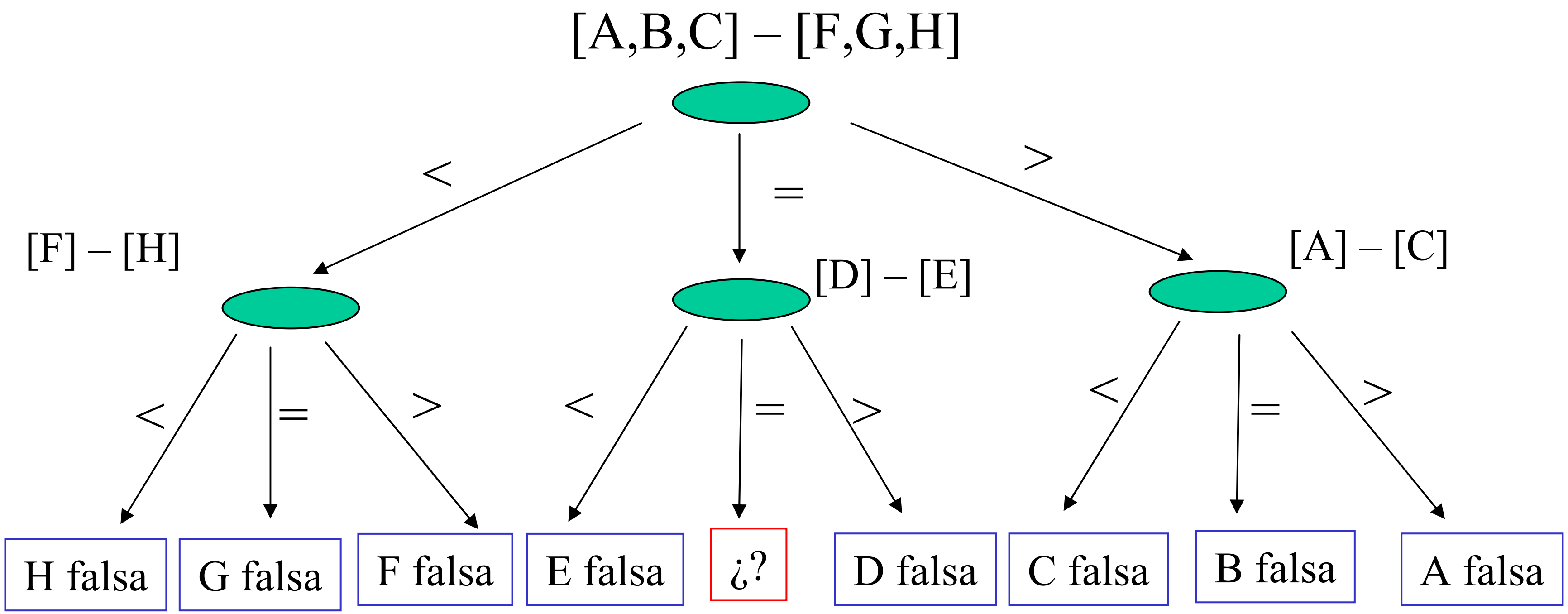
PROBLEMA

En una bolsa hay 8 monedas de las que se sabe que una es falsa por pesar más que las siete restantes. Averiguar cuál es la falsa con una balanza efectuando el menor número posible de pesadas.



ÁRBOL DE DECISIÓN BINARIO

Altura = 3 = Nº de pesadas



ÁRBOL DE DECISIÓN TERNARIO

Altura = 2 = Nº de pesadas

Las hojas de un árbol de decisión corresponden a las posibles soluciones del problema

ÁRBOLES Y ALGORITMOS DE ORDENACIÓN

Un árbol de decisión representa las comparaciones y movimientos de datos en un algoritmo de ordenación

Ordenación por inserción (Insertion sort)

Ordenación por comparación de los elementos de una lista L. En cada iteración un elemento se inserta en su lugar definitivo en una lista ya ordenada
http://en.wikipedia.org/wiki/Insertion_sort

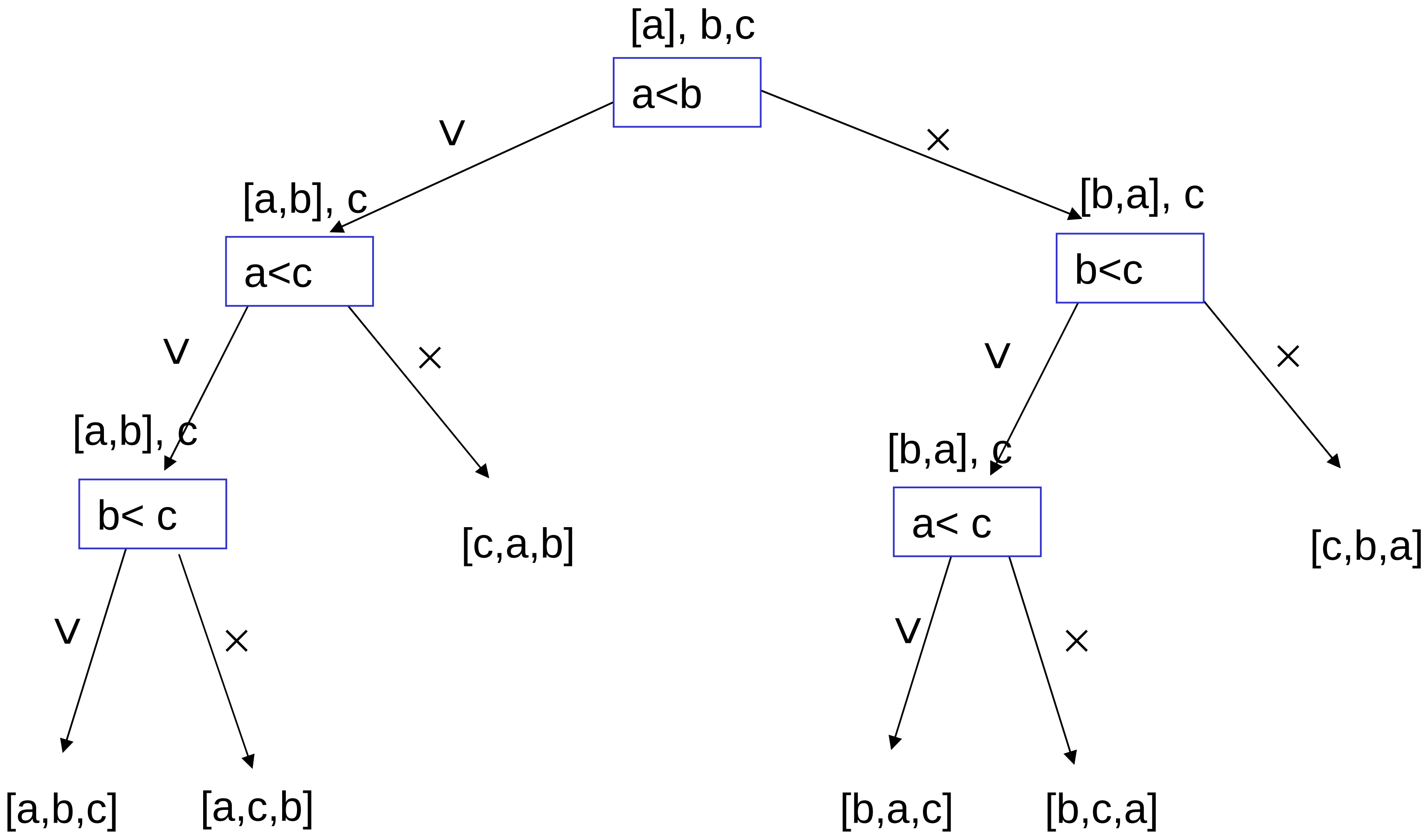
Inicialmente, paso 1, tenemos un elemento que ya está ordenado.
Paso (k). Tenemos una lista ordenada con los $k-1$ primeros elementos de L.
El elemento k se inserta en su posición ordenada en la lista anterior, efectuando a lo sumo $k-1$ comparaciones.

El número total de comparaciones efectuadas es, por tanto,

$$T(n) = 1 + 2 + 3 + \ldots + (n-1) \in O(n^2)$$

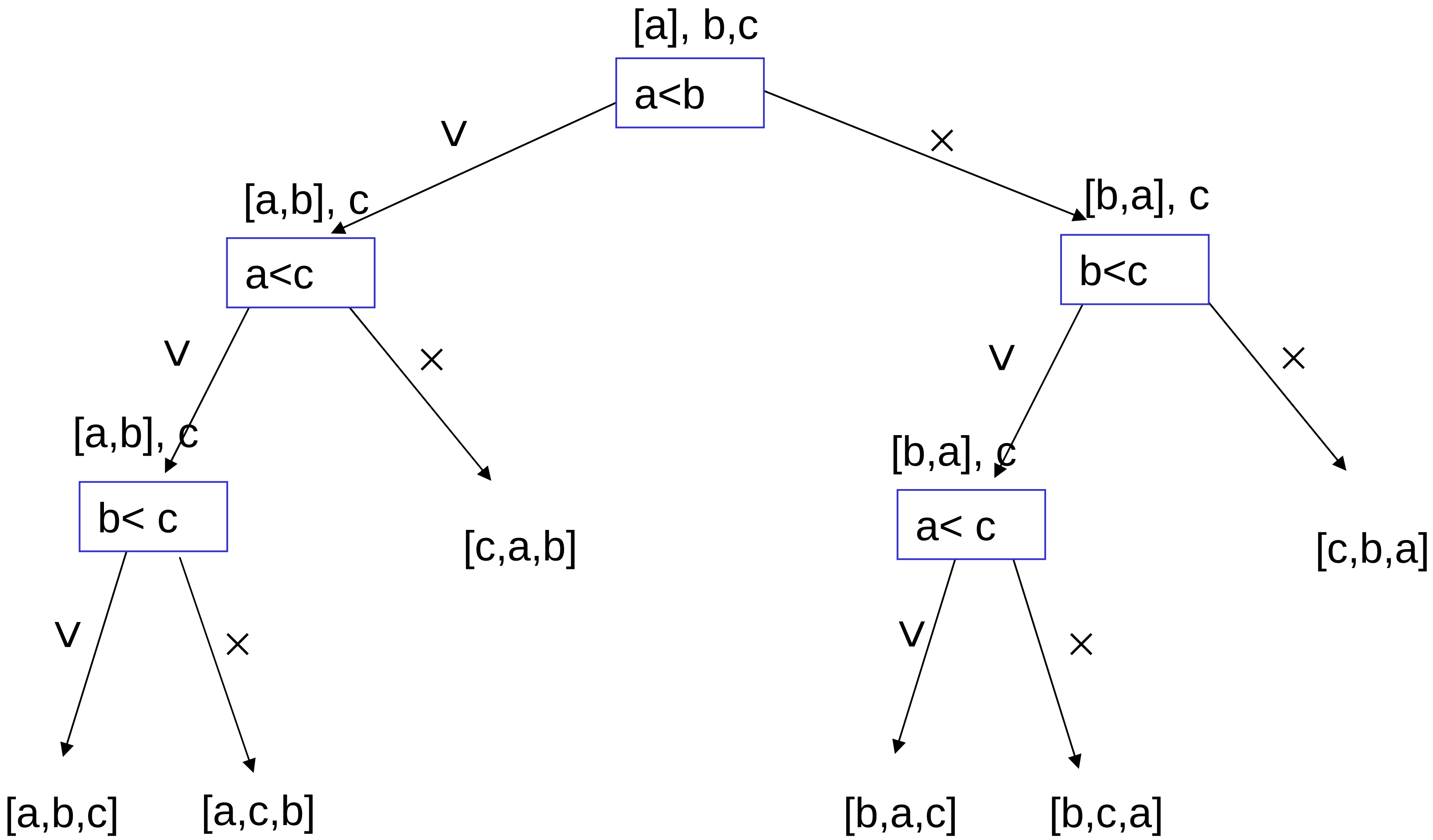
ÁRBOL DE DECISIÓN Y ALGORITMO DE ORDENACIÓN

El árbol de decisión que representa las comparaciones y movimientos de datos en el algoritmo de ordenación por inserción (lista de tamaño 3)



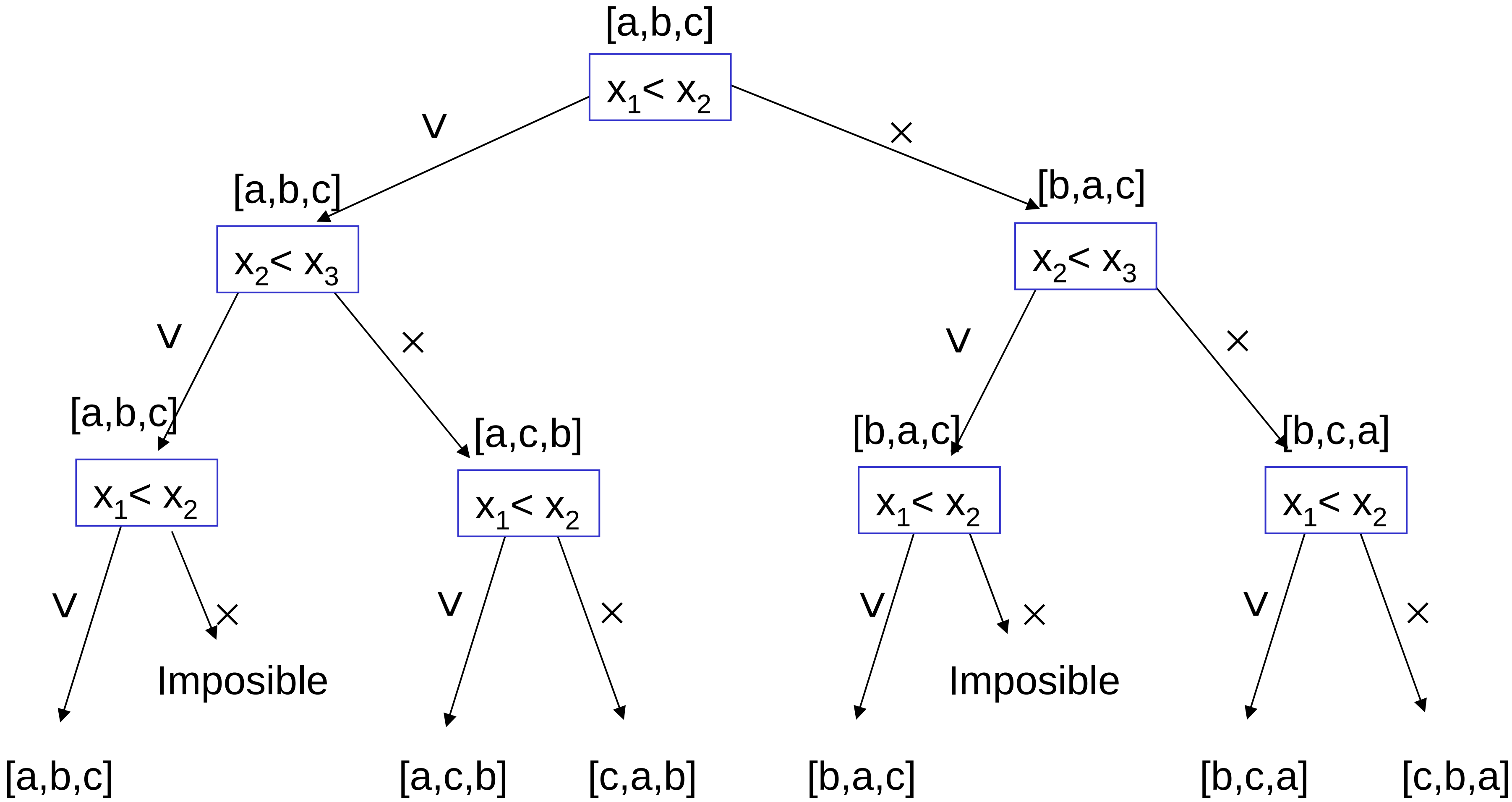
ÁRBOL DE DECISIÓN Y ALGORITMO DE ORDENACIÓN

Las hojas del árbol son todas las posibles ordenaciones, en total n!



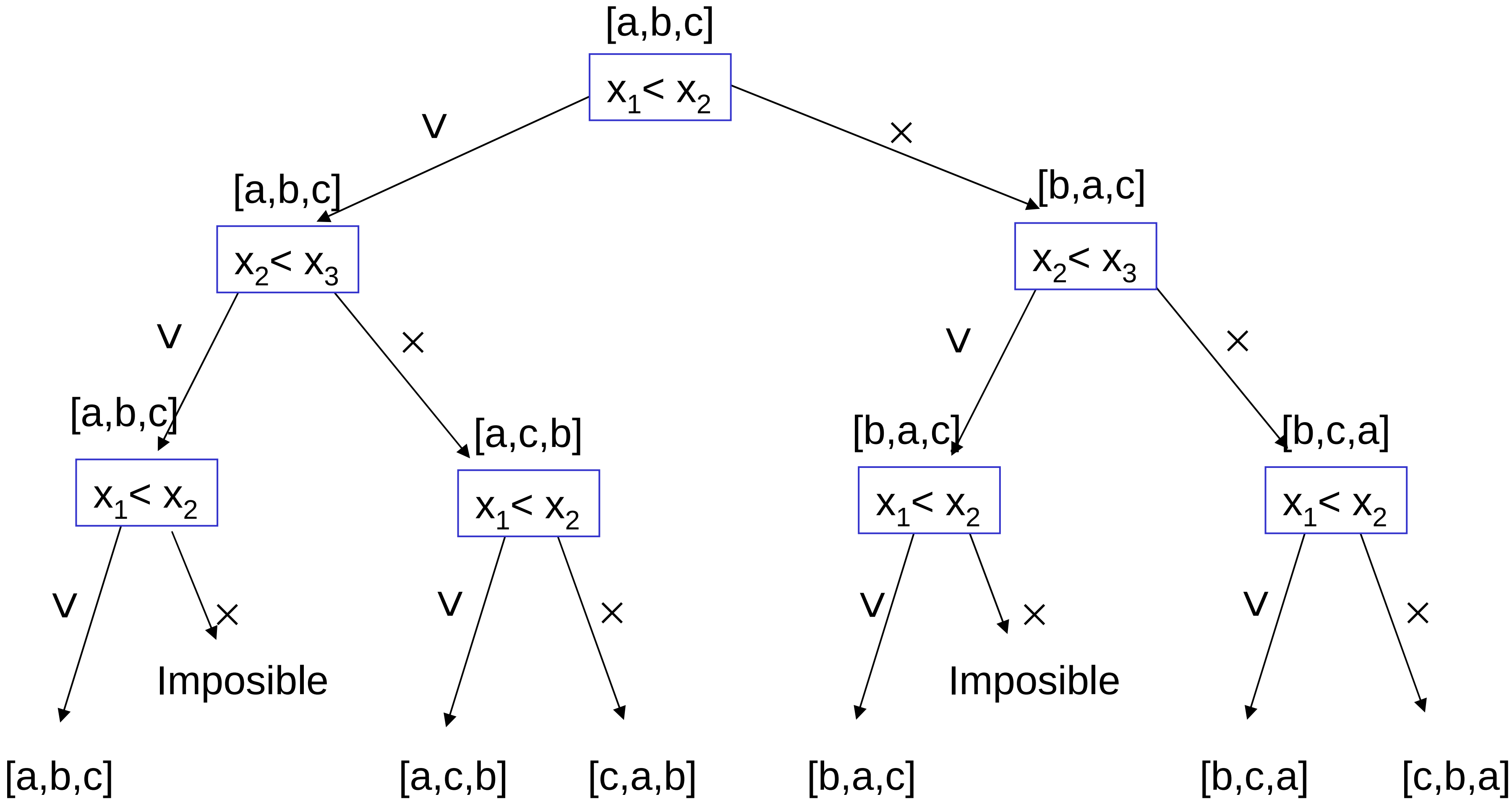
ÁRBOL DE DECISIÓN Y ALGORITMO DE ORDENACIÓN

El árbol de decisión que representa las comparaciones y movimientos de datos en el algoritmo de ordenación por burbuja (lista de tamaño 3)



ÁRBOL DE DECISIÓN Y ALGORITMO DE ORDENACIÓN

Las hojas del árbol son todas las posibles ordenaciones, en total n!



Complejidad de un problema (Ordenación)

Las hojas del árbol son todas las posibles ordenaciones, en total n!

La altura del árbol es el número de comparaciones que se deben realizar para ordenar una lista de tamaño n

Recordando la relación entre hojas y altura en un árbol binario,

$$|\text{comparaciones}| \geq \log_2 n!$$

Por tanto, el número T(n) de comparaciones realizadas por CUALQUIER algoritmo de ordenación es

$$T(n) \in \Omega(\log_2 n!)$$

$$\boxed{T(n) \in \Omega(n \log_2 n)}$$

El problema de ordenación tiene una cota inferior $\Omega(n \log_2 n)$

Complejidad de un problema

$\Omega(\log_2 n!) = \Omega(n \log_2 n)$

Dem.

(si n es par)

$n! = 1 \times 2 \times \ldots \times (n - 1) \times n = (1 \times n) \times (2 \times (n - 1)) \times \ldots \times (n/2) \times (n - n/2 + 1)$

Cada factor de este producto es $\geq n$, porque si $1 \leq i \leq n$, entonces

$i\,(n - i + 1) - n = (i - 1)(n - i) \geq 0$

Luego, $n! \geq n^{\frac{1}{2}n}$ porque hay n/2 factores

Por tanto,

$$\frac{1}{2} n \log_2 n \leq \log_2(n!) \leq \log_2 1 + \log_2 2 + \cdots + \log_2 n \leq n \log_2 n$$