Complessità del problema

Problema dell'ordinamento

Input: sequenza $a_1, a_2, \ldots, a_n$ di elementi su cui è definito un ordine

Output: $a'_1, a'_2, \ldots, a'_n$ permutazione di $a_1, a_2, \ldots, a_n$ tale che $a'_1 \leq a'_2 \leq \ldots \leq a'_n$

Se non facciamo ipotesi sul tipo degli elementi della sequenza le uniche operazioni permesse sono confronti e assegnazioni.

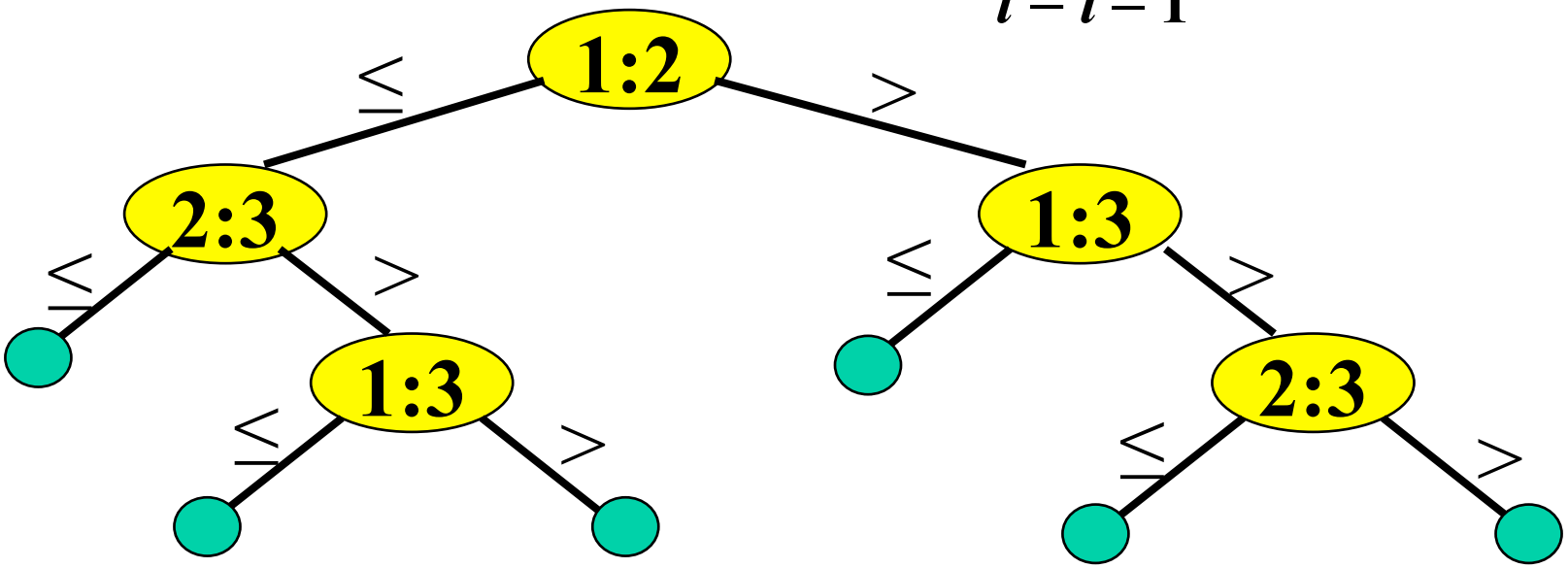
Siccome siamo interessati ad un limite inferiore possiamo contare solo alcune delle operazioni. Se un certo limite inferiore vale per il tempo richiesto per eseguire tali operazioni a maggior ragione varrà per il tempo calcolo totale.

Noi conteremo solo i confronti e dimostreremo che nel caso pessimo il numero di confronti è $\Omega(n \log n)$.

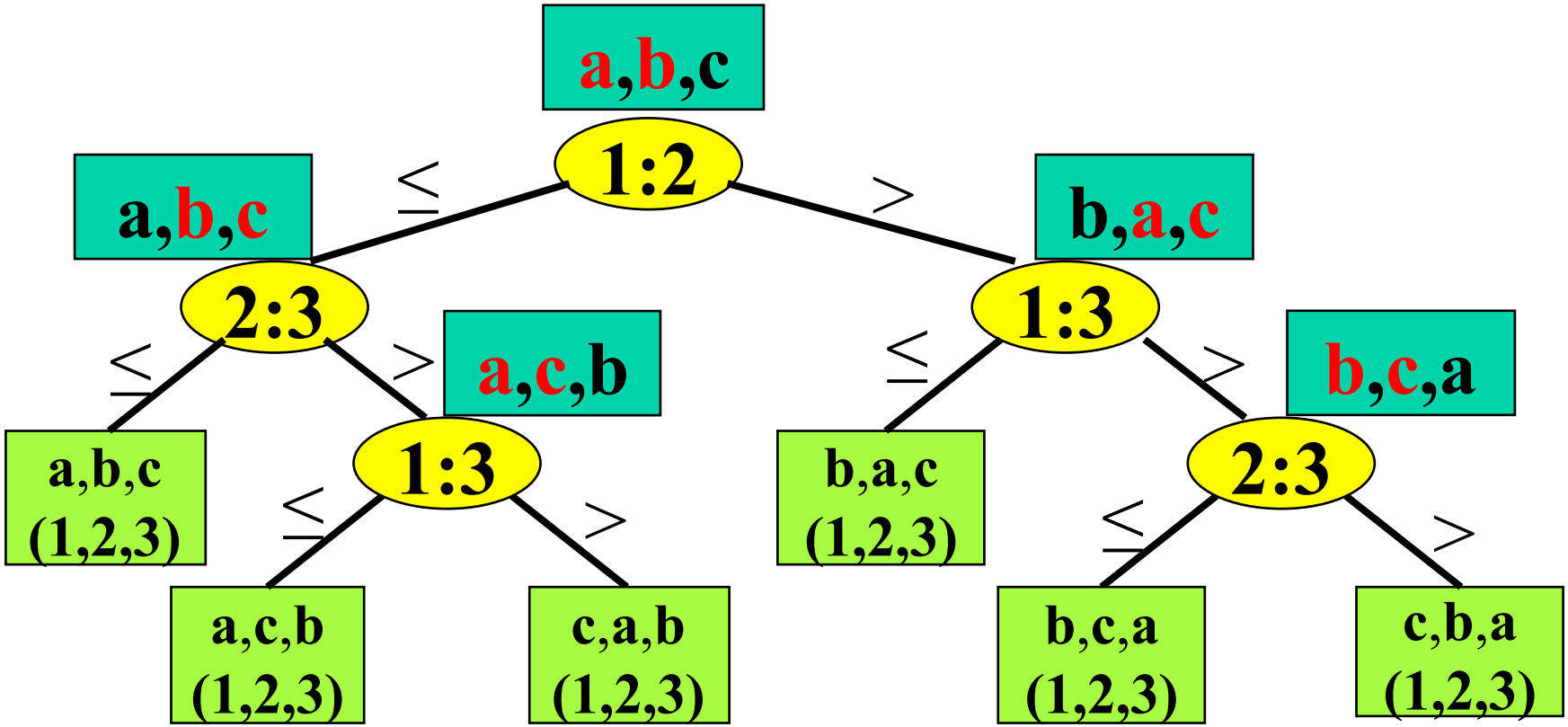
Per fare questo è utile rappresentare la struttura di un algoritmo mediante un albero delle decisioni.

Esempio. Albero delle decisioni di Insertion-Sort con un array di 3 elementi.

```
Insertion-Sort(A)
  n = A.length
  for j = 2 to n
    i = j – 1
    while i ≥ 1 and A[i]>A[i+1]
      scambia A[i] con A[i+1]
      i = i – 1
```



Esempio. Albero delle decisioni di Insertion-Sort con un array di 3 elementi.



Se l'algoritmo è corretto le foglie devono essere etichettate con ogni permutazione possibile dell'input. Perché?

Le permutazioni di 1,2,...,n sono $n!$ e quindi l'albero delle decisioni deve avere almeno $n!$ foglie.

Ma un albero binario con N foglie deve avere altezza almeno pari a $\log_2(N)$.

Esercizio: Dimostrarlo per induzione su N.

Dunque nel caso pessimo l'algoritmo deve eseguire almeno $\log_2(n!)$ confronti.

Ma $\log_2(n!) \geq \log_2\left(\frac{n}{2}\right)^{\frac{n}{2}}$

$$= \frac{n}{2}\log_2\frac{n}{2} = \frac{1}{2}n\log_2 n - \frac{n}{2}$$

$$= \Theta(n\log n)$$

e quindi $T_{\max}^{Alg}(n) = \Omega(n\log n)$ per ogni algoritmo generale di ordinamento.

Possiamo concludere che $\Omega(n \log n)$ è un limite inferiore per la complessità del problema dell'ordinamento.

L'algoritmo di ordinamento ***Heapsort*** risolve il problema dell'ordinamento con complessità massima

$$T_{\max}^{HS}(n) = O(n \log n)$$

Dunque $O(n \log n)$ è limite superiore per la complessità del problema dell'ordinamento.

Siccome limite superiore e inferiore coincidono $\Theta(n \log n)$ è limite stretto per il problema dell'ordinamento.

Considerazione sul limite inferiore $\Omega(n \log n)$ per l'ordinamento

ATTENZIONE:

Il limite inferiore $\Omega(n \log n)$ da noi dimostrato vale solo per algoritmi di ordinamento *generali*, ossia algoritmi che non fanno alcuna ipotesi sul tipo degli elementi da ordinare: le uniche operazioni ammesse su tali elementi sono confronti e assegnazioni.

Il limite inferiore $\Omega(n \log n)$ vale anche per ordinare numeri reali sui quali, oltre a confronti ed assegnazioni, si possono usare anche le quattro operazioni aritmetiche.

In questo caso la dimostrazione del limite inferiore è molto più difficile e si basa su alcuni risultati di geometria algebrica.

La dimostrazione si può trovare nel testo di Geometria Computazionale di F. Preparata.