## RICERCA GENERALE DEL $\Omega$ DEGLI ALGORITMI DI ORDINAMENTO

Vogliamo cercare il limite inferiore asintotico al tempo richiesto per ordinare una sequenza arbitraria di lunghezza $n$ nel suo caso migliore e peggiore.

Per il caso migliore INSERTION SORT è ottimale perché lineare (meno di lineare non possiamo avere perché dobbiamo controllare che la sequenza sia effettivamente ordinata).

Quindi il caso migliore di ordinamento ha $\Omega(n)$, concentriamoci sul caso peggiore.

## CASO PEGGIORE

Assumiamo che ogni algoritmo utilizzi come strumento di ordinamento i CONFRONTI, quindi cerchiamo il numero minimo di confronti.

Ovviamente ci sono altri strumenti utilizzati, come gli scambi, ma questi sono direttamente collegati ai confronti quindi asintoticamente ci basta studiare i confronti.

La ricerca del numero minimo di questi si può formalizzare:

$\forall A \; \exists c>0 \mid T_A(n) \geq c \cdot f(n)$

In cui $A$ sono gli algoritmi esistenti ed $f(n)$ è il tempo impiegato.

Ovviamente il problema sta nel fatto che noi non conosciamo tutti gli algoritmi, possiamo pero' studiare i vari comportamenti (che non sono infiniti).

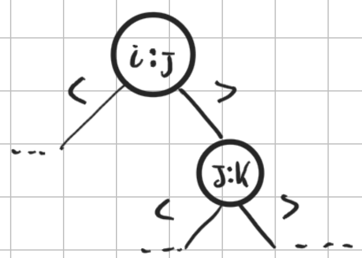
Vogliamo trovare i migliori casi peggiori (che per quello che abbiamo visto sono merge sort e heap sort).

Tutti questi algoritmi (e quindi comportamenti) vengono raccolti in STRUTTURE.

## ALBERI DI DECISIONE

Indichiamo con $i:j$ il confronto $A[i] < A[j]$?

L'esito puo' essere SI/NO, costruiamo l'albero




Il numero di foglie è pari al numero di permutazioni della sequenza (perche' arrivati alle foglie sapremo esattamente come ordinare la nostra sequenza)

La struttura risultante è detta **ALBERO DI DECISIONE**

Definiremo **ORDINE** dell'albero di decisione (in questo specifico caso) come la lunghezza della sequenza da ordinare.

**PROPOSIZIONE**

Per ogni coppia $(A, n)$ con $A$ algoritmo e $n$ lunghezza della sequenza esiste ed è unico un albero di decisione di ordine $n$ associato ad $A$

Es.

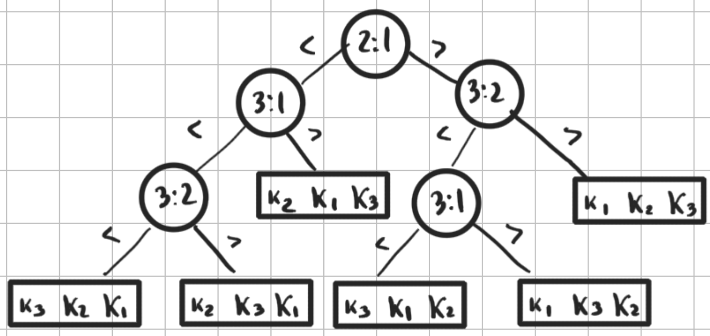
ALBERO DI DECISIONE INSERTION SORT $n=3$

Sia $\underset{1\;\;\;2\;\;\;3}{\boxed{K_1 | K_2 | K_3}}$ la successione in input e l'algoritmo

INSERTION SORT (A,n)

```
FOR J=2 TO N DO
    X=A[J]
    i=J-1
    WHILE (i>0 && A[i]>X) DO
        A[i+1]=A[i]
        i--
    A[i+1]=X
```

L'albero di decisione è




Il caso peggiore è dato dal percorso più lungo, ovvero l'altezza.

Per trovare l'algoritmo con il miglior caso peggiore in generale dobbiamo trovare l'altezza minore tra tutti gli alberi di decisione di ordine $n$.

## RICERCA DEL $\Omega T(n)$ TRAMITE ALBERI DI DECISIONE

Il limite asintotico inferiore al tempo richiesto per ordinare sequenze di lunghezza $n = \Omega(h(n))$ dove $h(n)$ è la minima altezza di un albero di decisione di ordine $n$.

Gli alberi di decisione hanno le seguenti proprietà:

- Sono binari

- Hanno almeno $n!$ foglie ($n!$ permutazioni possibili)
- Ogni nodo interno ha esattamente grado 2

Da queste ipotesi sappiamo che $nn = 2f-1$ dove $nn$ è il numero di nodi e $f$ numero di foglie.

Gli alberi con altezza minore sono quelli pieni dove l'altezza è $h = \lfloor \lg_2 nn \rfloor$ quindi sappiamo che $h(n) \geq \lfloor \lg_2 nn \rfloor$ con $n$ lunghezza della sequenza.

$nn$ dipende dal numero di foglie che è almeno $n!$, in particolare

$$nn \geq 2n!-1 \rightarrow h(n) \geq \lfloor \lg_2(2n!-1) \rfloor \geq \lfloor \lg_2(n!) \rfloor$$

Vogliamo trovare il valore di $\lg_2(n!)$.

Per Stirling vale:

$$n! \geq \left(\frac{n}{e}\right)^n$$

Sostituendo

$$\log(n!) \geq \lg_2\left(\frac{n}{e}\right)^n = n\lg_2 n - \cancel{n\lg_2 e} \rightarrow \lg(n!) \geq n\lg n$$

asintoticamente ignorabile

Quindi per transitività

$$h(n) \geq n\lg_2 n \Rightarrow h(n) = \Omega(n\lg_2 n)$$

# RICERCA DEL CASO MEDIO

Dopo l'analisi precedente, e sotto le stesse ipotesi, per trovare il caso medio ci basta fare la media dei percorsi massimali (dato un algoritmo A con ordine $n$)

$$\frac{\sum \text{Percorsi massimali}}{n^\circ \text{ percorsi massimali}}$$

La somma dei percorsi massimali (dalla radice alle foglie) e' pari alla lunghezza del PERCORSO ESTERNO, ovvero la somma degli archi necessari ad arrivare ad ogni foglia dalla radice, mentre il n° di percorsi e' pari al numero di foglie. Quindi vale

$$T_M(n) = \frac{PE}{n!}$$