Divide et Impera

1) Dividere in sottoproblemi
2) Risolvere i sottoproblemi
3) Combinare le soluzioni dei sottoproblemi per risolvere il problema di partenza

Come risolvo 2) ? Riapplico la tecnica fino a quando ottengo sottoproblemi la cui risoluzione è immediata

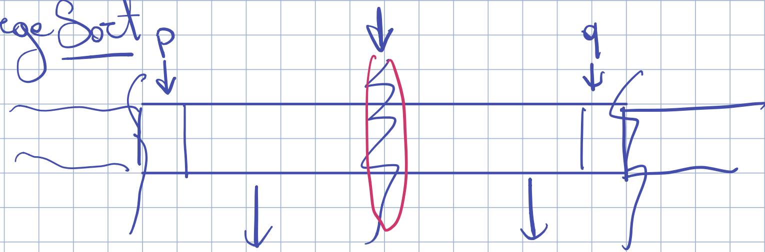
↓

Ricorsione

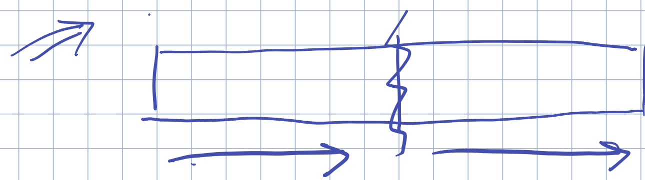
<u>Esempi</u>

- Fattoriale Ricorsiva vs Iterativa
- Fibonacci Ricorsiva

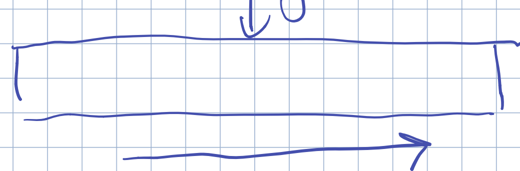
MergeSort



ordino ordino ← fino a che arrivo a vettori di 1 elem.

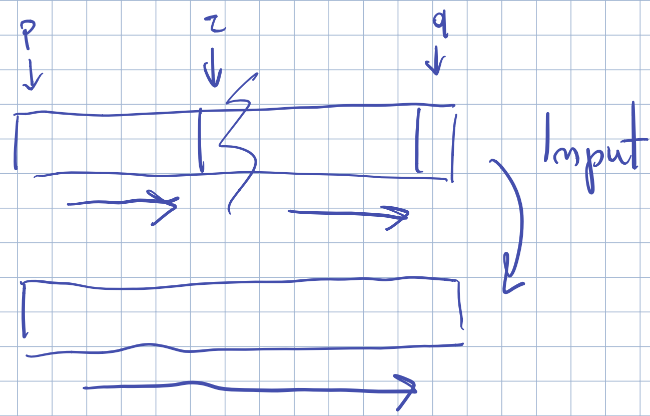


Merge fonde ← come funziona?



```
MergeSort (A, p, q) {
    if (p < q) {
        r ← ⌊(p+q)/2⌋
        MergeSort (A, p, r)
        MergeSort (A, r+1, q)
        Merge (A, p, r, q)
    }
}
```

Merge



Input Merge — Precondizione

$A[p..r]$ e $A[r+1..q]$ sono ordinati

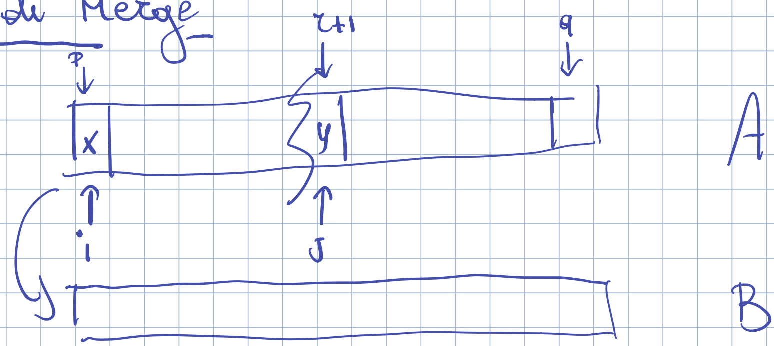
Output Merge — Postcondizione

$A[p..q]$ è ordinato

Correttezza Merge

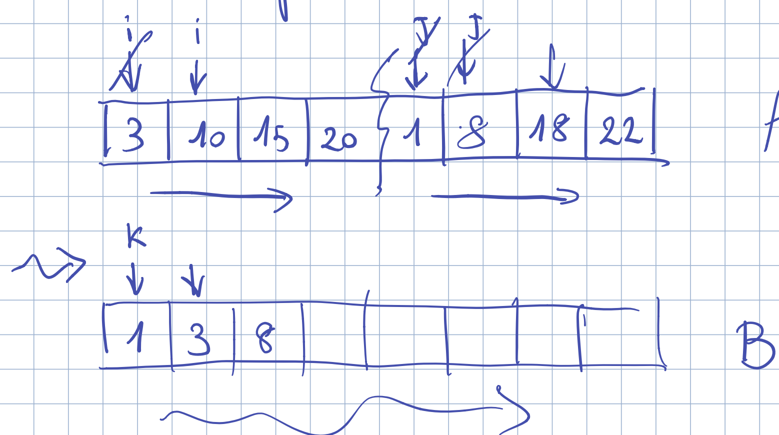
Se $A[p..r]$ e $A[r+1..q]$ sono ordinati allora $Merge(A, p, r, q)$ termina con $A[p..q]$ ordinato

Idea di Merge



copio il più piccolo e incremento

Poi ricopio tutto in A



Esercizio

Scrivere Pseudocodice Merge (A, p, r, q)

$\boxed{\Theta(n)}$ Non In-Place Merge $\perp$

```
MergeSort (A, p, q) {
  → if (p < q) {
        r ← ⌊(p+q)/2⌋
        MergeSort (A, p, r)
    ⇝   MergeSort (A, r+1, q)
    ⇝   Merge (A, p, r, q)
    }
}
```

Correttezza MergeSort

MergeSort(A, p, q) termina sempre e $A[p..q]$ al termine è ordinato

Dim.

Per induzione su $n = q - p + 1$ lunghezza della porzione da ordinare

BASE $n = 1$ $\leadsto$ $p = q$ $A[p..p]$ un solo elemento ✓

PASSO

HpInd) MergeSort (A, p, q) termina con $A[p..q]$ ordinato se $q-p+1 < n$

Ts) MergeSort (A, p, q) termina con $A[p..q]$ ordinato nel caso $q-p+1 = n$

$n > 1$ if $(p<q)$ ha successo

$p \leq r < q$

MergeSort (A, p, r) $\quad r-p+1 < q-p+1 = n$

per Hp Ind

* Termina con $A[p..r]$ ordinato

MergeSort $(A, r+1, q)$ $\quad q-(r+1)+1 < q-p+1 = n$

per Hp Ind

* Termina con $A[r+1..q]$ ordinato

valgono le precondizioni $A[p..r]$ e $A[r+1..q]$ ordinate

Correttezza di Merge

Merge termina con $A[p..q]$ ordinato

Complessità Merge Sort

MergeSort (A, p, q) { $T(n)$

$(\Theta(1))$ c if $(p < q)$ {

$(\Theta(1))$ c $r \leftarrow \left\lfloor \frac{p+q}{2} \right\rfloor$

$T(n/2)$ MergeSort (A, p, r)

$T(n/2)$ MergeSort $(A, r+1, q)$

$(\Theta(n))$ $c \cdot n$ Merge (A, p, r, q)

}

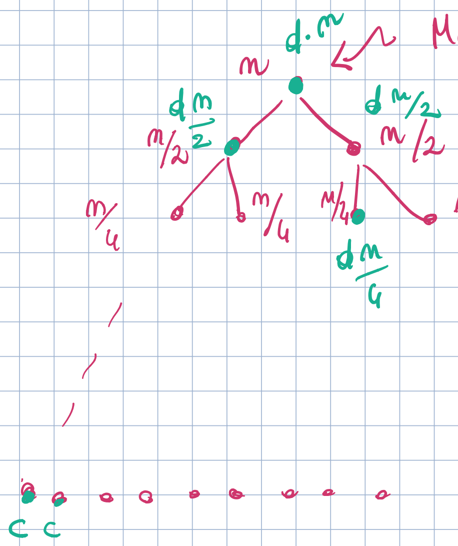
}

$n = q - p + 1$

$$T(n) = \begin{cases} c & \text{se } n = 1 \\ 2T(n/2) + d \cdot n & \text{se } n > 1 \end{cases}$$

Equazione nell'incognita $T : \mathbb{N} \rightarrow \mathbb{R}^+$

MergeSort (A, p, q)



dn

dn

dn

Albero Binario completo

dn

cn

Foglie Casi Base sono n

(Vettori da 1 elemento)

$$\sum_{i=0}^{h-1} dn + cn$$

$h \simeq \log_2 n$

$$= d \cdot n \log_2 n + cn$$

$$\boxed{= O(n \cdot \log n)}$$

MergeSort