

Introduzione agli Algoritmi

Merge Sort

Dr. Emanuela Merelli

Argomenti della lezione

- Progetto di algoritmi
 - Divide et impera
- Concetto di
 - Ricorsione

Progettazione di algoritmi

- Approccio incrementale
 - Se è ordinato $A[1..j-1]$ si inserisce $A[j]$...
- Approccio divide-et-impera
 - Divido il problema in n sottoproblemi e opero ricorsivamente sugli n problemi

Divide-et-impera

Un algoritmo si dice ricorsivo se al suo interno (tra i suoi comandi) sono presenti chiamate a se stesso per gestire sottoproblemi analoghi a quello dato

L'approccio seguito dagli algoritmi ricorsivi è quello di suddividere il problema dato in sottoproblemi simile a quello di partenza ma di dimensione inferiore
Poi, una volta che i sottoproblemi sono stati risolti ricorsivamente combinano le soluzioni trovate per creare la soluzione al problema dato

Divide
Impera
Combina

Algoritmo di ordinamento

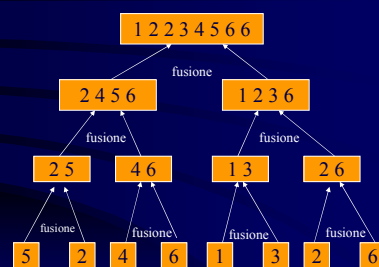
Merge Sort

Divide: gli n elementi della sequenza da ordinare in due sottosequenze di $n/2$ elementi ciascuna

Impera: ordina, usando ricorsivamente il merge sort, le due sottosequenze

Combina: fonde le due sottosequenze per produrre come risposta la sequenza ordinata

$A = \langle 5, 2, 4, 6, 1, 3, 2, 6 \rangle$



Sequenza iniziale

Merge Sort

Il processo di suddivisione si ferma quando la sequenza da ordinare ha lunghezza 1

Il passo combina fonde le due sequenze utilizzando una procedura ausiliaria merge (fusione)

$merge(A, p, q, r)$ A è un array p, q, r sono indici di elementi dell'array tali che $p \leq q < r$
merge assume che $A[p \dots q]$ e $A[q+1 \dots r]$ siano ordinati e genera $A[p \dots r]$ ordinato

Merge Sort

$p = \text{left}, r = \text{right}, q = \text{mid}$

```
MergeSort(A, left, right) {  
  if (left < right) {  
    mid =  $\lfloor ((\text{left} + \text{right}) / 2) \rfloor$ ;  
    MergeSort(A, left, mid);  
    MergeSort(A, mid+1, right);  
    Merge(A, left, mid, right);  
  }  
}  
  
// Merge() takes two sorted subarrays of A and  
// merges them into a single sorted subarray of A  
// (how long should this take?)
```

Il più grande intero
minore o uguale

Merge?

HOMEWORK

Complessità computazionale del Merge?

HOMEWORK

Esercizio

Definire un algoritmo ricorsivo che calcoli il fattoriale di n .

Si ricorda che la funzione fattoriale, indicata con **fatt**, è così definita

fatt(0) = 1

fatt(1) = 1

fatt(n) = $1 \times 2 \times \dots \times n$

Fattoriale di n

```
Function fatt (n): integer;  
begin  
  if n=0 then fatt:=1 else fatt:= fatt(n-1) * n  
end;
```