



Sommario

- Definizione di ricorsione e strategie *divide et impera*
- Semplici algoritmi ricorsivi
- Merge Sort
- Quicksort
- Esempi più complessi di algoritmi ricorsivi



Descrizione

Quicksort è un algoritmo Divide et Impera in cui il partizionamento viene fatto in base al *valore* (e non alla posizione) degli elementi del vettore.

Ad ogni passo, viene scelto un valore "pivot" (perno), e gli elementi vengono assegnati alla prima o alla seconda partizione a seconda che siano minori o maggiori del pivot.



Formulazione

- **Dividi**: partiziona $A[p..r]$, riordinando gli elementi in due sottovettori $A[p..q]$ e $A[q+1..r]$, in modo tale che gli elementi di $A[p..q]$ siano **tutti** \leq degli elementi di $A[q+1..r]$. Il valore di q è variabile.
- **Risolvi**: ordina **ricorsivamente** $A[p..q]$ e $A[q+1..r]$.
- **Combina**: poiché $A[p..q]$ e $A[q+1..r]$ sono ordinati, e i primi sono \leq ai secondi, non occorre **fare nulla**: $A[p..r]$ è ordinato.

A.A. 2001/2002

APA-ricorsione

3



Quicksort

```
QUICKSORT( $A, p, r$ )
1  if  $p < r$ 
2      then   $q \leftarrow \text{PARTITION}(A, p, r)$ 
3              QUICKSORT( $A, p, q$ )
4              QUICKSORT( $A, q+1, r$ )
```

A.A. 2001/2002

APA-ricorsione

4

Partition

```

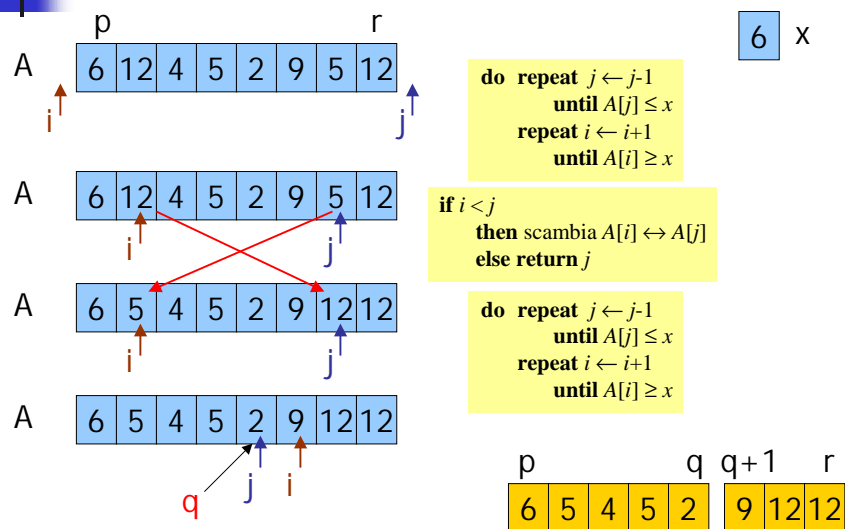
PARTITION(A, p, r)
1  x ← A[p]      ▷ elemento Pivot
2  i ← p - 1
3  j ← r + 1
4  while true
5      do repeat j ← j - 1
6          until A[j] ≤ x
7      repeat i ← i + 1
8          until A[i] ≥ x
9      if i < j
10         then scambia A[i] ↔ A[j]
11        else return j
    
```

A.A. 2001/2002

APA-ricorsione

5

Esempio



A.A. 2001/2002

APA-ricorsione

6



Analisi di Partition

La procedura fa “convergere” verso il centro gli indici i e j .

Ogni volta che i trova un elemento maggiore del pivot e j trova un elemento minore del pivot, gli elementi vengono scambiati per permettere agli indici di continuare.

Quando i “incontra” j , la partizione è completa.

Il numero complessivo di operazioni è $\Theta(r-p+1)$, nonostante i cicli annidati.

A.A. 2001/2002

APA-ricorsione

7



Esercizio

Mostrare il funzionamento di Partition sulla seguente sequenza di numeri:

$A = \{ 13, 19, 9, 5, 12, 8, 7, 4, 11, 2, 6, 1 \}$

A.A. 2001/2002

APA-ricorsione

8



Soluzione

$A = \{ 13, 19, 9, 5, 12, 8, 7, 4, 11, 2, 6, 1 \}$

$x = 13$

$A = \{ 13, \underline{19}, 9, 5, 12, 8, 7, 4, 11, 2, 6, \underline{1} \}$

$A = \{ 13, 1, 9, 5, 12, 8, 7, 4, 11, 2, \underline{6}, \underline{19} \}$

La partizione è:

$\{ \{ 13, 1, 9, 5, 12, 8, 7, 4, 11, 2, 6 \}, \{ 19 \} \}$



Esercizio proposto

Implementare in C la procedura Partition, verificando il corretto funzionamento.

Quale valore viene restituito come punto di partizione q su vettori casuali? su vettori ordinati? su vettori ordinati inversamente?



Esercizio

Sempre in riferimento al vettore

$A = \{ 13, 19, 9, 5, 12, 8, 7, 4, 11, 2, 6, 1 \}$

quale potrebbe essere una scelta migliore (“a mano”) per il pivot x ?



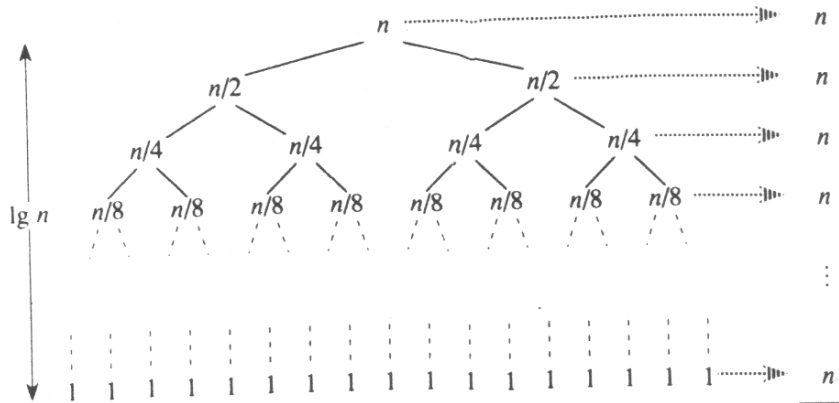
Prestazioni

L'efficienza di Quicksort dipende totalmente dalla qualità del partizionamento ottenuto.

- Se il partizionamento è ottimo ($q \sim (p+r)/2$), l'algoritmo è $O(n \log n)$
- Se il partizionamento è pessimo ($q \sim p$ o $q \sim r$), l'algoritmo degenera in un $O(n^2)$

La qualità del partizionamento dipende da una buona scelta del pivot.

Caso migliore

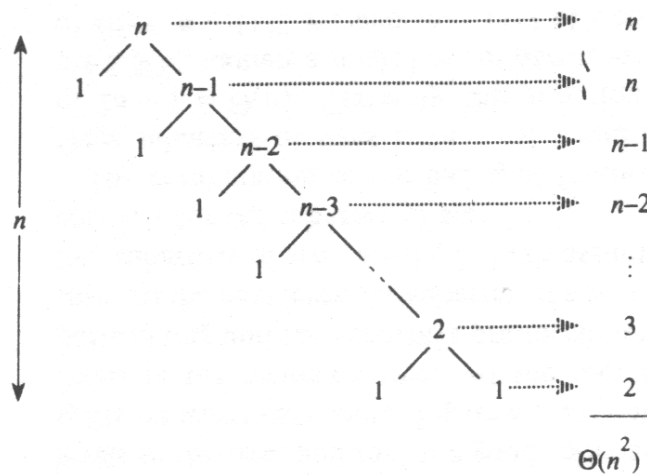


A.A. 2001/2002

APA-ricorsione

13

Caso peggiore



A.A. 2001/2002

APA-ricorsione

14



Scelta del pivot

Il caso peggiore si verifica quando le partizioni hanno 1 ed $n-1$ elementi, rispettivamente, ossia quando il pivot è il minimo o il massimo.

Un vettore già ordinato ricade quindi nel caso peggiore! Anche un vettore ordinato inversamente è nel caso peggiore.

Il caso migliore si ha quando il vettore è il più disordinato possibile.



Quicksort randomizzato

PARTITION-RANDOM(A, p, r)

▷ genera un numero casuale tra p e r

1 $i \leftarrow \text{RANDOM}(p, r)$

▷ evita il caso peggiore!

2 scambia $A[i] \leftrightarrow A[p]$

▷ applica il partizionamento

3 **return** **PARTITION** (A, p, r)



Scelte del pivot

- Scegliere un elemento a caso (PARTITION-RANDOM)
- Scegliere l'elemento nel mezzo: $x \leftarrow A[(p+r)/2]$
- Scegliere il valore medio tra min e max
- Scegliere la mediana tra 3 elementi presi a caso nel vettore
- ...

A.A. 2001/2002

APA-ricorsione

17



Complessità

Da quanto visto emerge che Quicksort ha un comportamento:

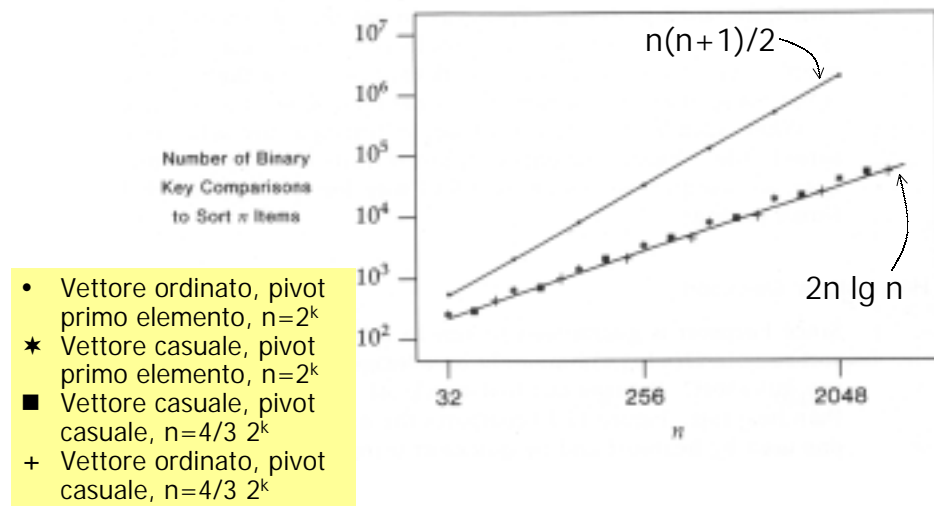
- $O(n^2)$ in generale, $O(n \log n)$ in media
- $\Theta(n^2)$ nel caso peggiore (che si deve cercare di evitare)
- $\Theta(n \log n)$ nel caso medio e nel caso migliore

A.A. 2001/2002

APA-ricorsione

18

Esempi



A.A. 2001/2002

APA-ricorsione

19

Esercizio proposto

Si implementi l'algoritmo Quicksort in C, e si sperimentino diverse scelte dell'elemento pivot.

A.A. 2001/2002

APA-ricorsione

20