Algoritmi e Strutture Dati

Lezione 11

17 ottobre 2025

Quicksort

QuickSort

$n = \#$ elementi

- Caso semplice → sol. immediata $n \leq 1$
- Altrimenti

  divido l'array in 2 parti

  ordina le due parti separatamente

  combina le soluzioni

44 55 12 42 94 9 18 64

perno o pivot

< 42 >

12 9 18

44 55 94 64

ordina

ordina

9 12 18

42

44 55 64 94

ALGORITMO quickSort (array A)

IF lunghezza di A > 1 THEN

scegli un elemento x in A

$B \leftarrow \{ y \in A \mid y < x \}$ ← partizionamento

$C \leftarrow \{ y \in A \mid y > x \}$

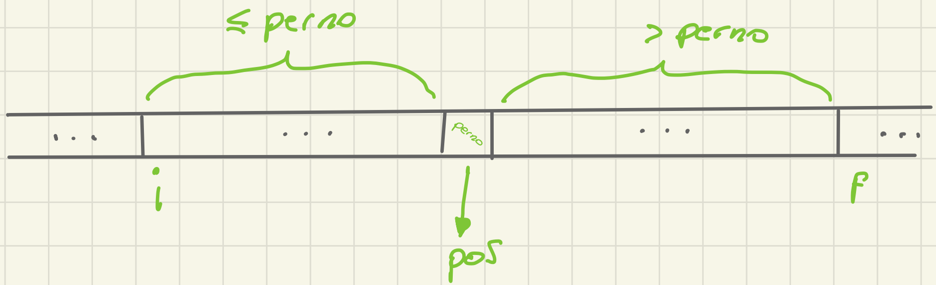
quickSort(B)

quickSort(C)

$A \leftarrow B \, ; \, x \, ; \, C$

ALGORITMO partiziona (Array A, indice i, indice f) → indice

- Partiziona $A[i..f-1]$ rispetto a un elemento di $A[i..f-1]$ scelto come perno, spostando gli elementi, in modo che alla fine



- Restituisce la posizione finale del perno (pos)

ALGORITMO partiziona (Array A, indice i, indice f) → indice

Esempio:

dato

...	12	45	17	22	46	1	33	24	2	8	...
	i										f

due delle molte soluzioni possibili:

	≤ perno					> perno					
...	12	8	2	1	17	46	33	24	22	45	...
	i				pos						f

	≤ perno								> perno		
...	24	8	2	1	22	12	17	33	46	45	...
	i							pos			f

7	2	9	15		15		33
~~12~~	~~33~~	~~15~~	~~7~~	19	~~9~~	56	~~2~~

perno

sx

dx

perno ← 1° elemento

scansiona da dx fino a raggiungere un elemento ≤ perno

scansiona da sx fino a raggiungere un elemento > perno

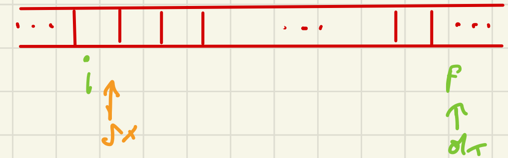
scambia i 2 elementi

termina quando le due scansioni si incontrano

scambia il perno con l'elemento di posto dx

RETURN dx

ALGORITMO partiziona (Array A, indice i, indice f) $\rightarrow$ indice




```
perno ← A[i]
sx ← i
dx ← f
WHILE sx < dx DO
    DO dx ← dx - 1 WHILE A[dx] > perno
    DO sx ← sx + 1 WHILE sx < dx AND A[sx] ≤ perno
    IF sx < dx THEN
        scambia A[sx] con A[dx]
scambia A[i] con A[dx]
RETURN dx
```

perno ← 1° elemento

scansiona da dx fino a raggiungere un elemento ≤ perno

scansiona da sx fino a raggiungere un elemento > perno

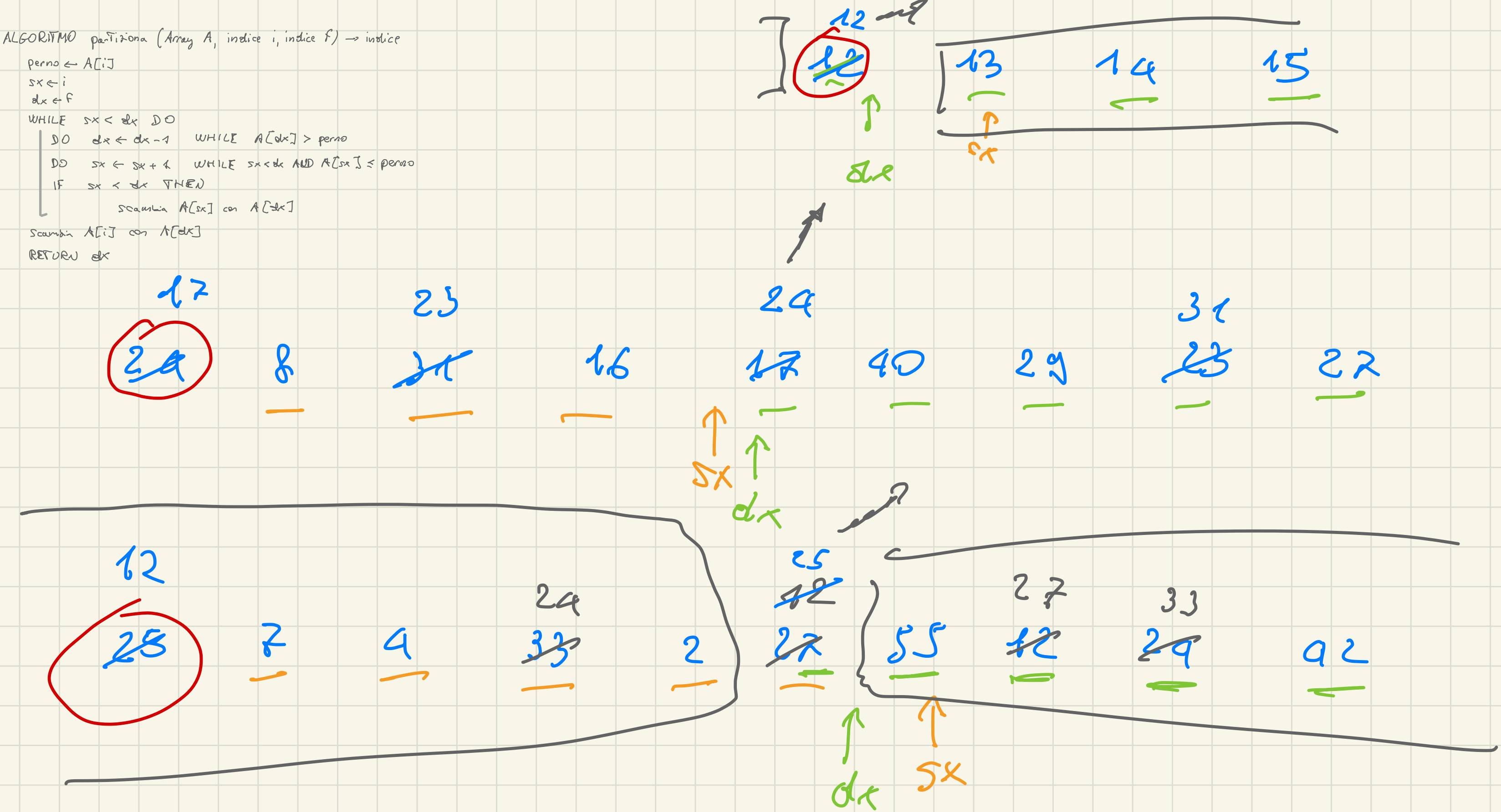
scambia i 2 elementi

termina quando le due scansioni si incontrano

scambia il perno con l'elemento di posto dx

RETURN dx

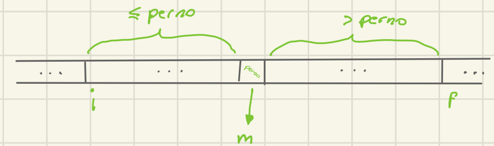
```
ALGORITMO partiziona (Array A, indice i, indice f) → indice
  perno ← A[i]
  sx ← i
  dx ← f
  WHILE sx < dx DO
  |  DO dx ← dx − 1    WHILE A[dx] > perno
  |  DO sx ← sx + 1    WHILE sx < dx AND A[sx] ≤ perno
  |  IF sx < dx THEN
  |        scambia A[sx] con A[dx]
  scambia A[i] con A[dx]
  RETURN dx
```

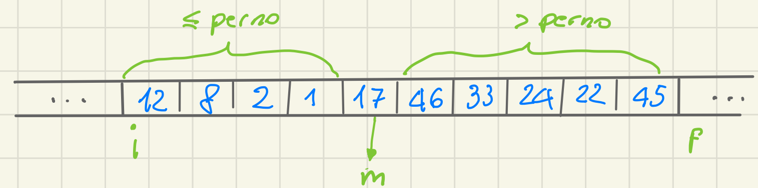


$\# cfr = n$ o $n-1$

<u>in loco</u>

```
PROCEDURA quickSort (Array A, indice i, indice f)
    IF f - i > 1 THEN
        m ← partiziona (A, i, f)
        quickSort(A, i, m)
        quickSort(A, m+1, f)
```





1 2 8 12 22 24 33 45 46

```
ALGORITMO quickSort ( Array A[0 .. n-1] )
    quickSort(A, 0, n)
```

QuickSort: numero di confronti $C(n)$

se $n \leq 1 \longrightarrow 0$ confronti

altrimenti

confronti per calcolare partizione $\quad C_{part}(n) = n$

$+ \quad\quad +$

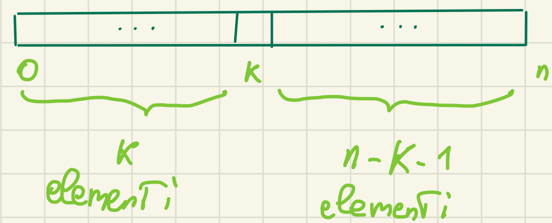
confronti per ordinare parte sx $\quad C(k)$

$+ \quad\quad +$

confronti per ordinare parte dx $\quad C(n-k-1)$

se il pivot finisce in posizione k

```
PROCEDURA quickSort(Array A, indice i, indice f)
   IF f - i > 1 THEN
      m ← partiziona(A, i, f)
      quickSort(A, i, m)
      quickSort(A, m+1, f)
```



Caso peggiore

$$C_w(n) = \begin{cases} 0 & \text{se } n \le 1 \\ n + \underbrace{\max}_{\text{si ottiene per } k=0 \text{ o } k=n-1} \{ C_w(k) + C_w(n-k-1) \mid k = 0..n-1 \} & \text{altrimenti} \end{cases}$$

caso peggiore

$$C_w(n) = n + C_w(n-1)$$

$$= n + n - 1 + C_w(n-2) = n + n - 1 + n - 2 + C_w(n-3)$$

$$= n + n - 1 + n - 2 + \dots + 2 + \underbrace{C_w(1)}_{=0}$$

$$= \sum_{i=2}^{n} i = \frac{n(n+1)}{2} - 1 = \Theta(n^2) \quad \text{CASO PEGGIORE}$$

3 5 15 18 20 25

dx sx

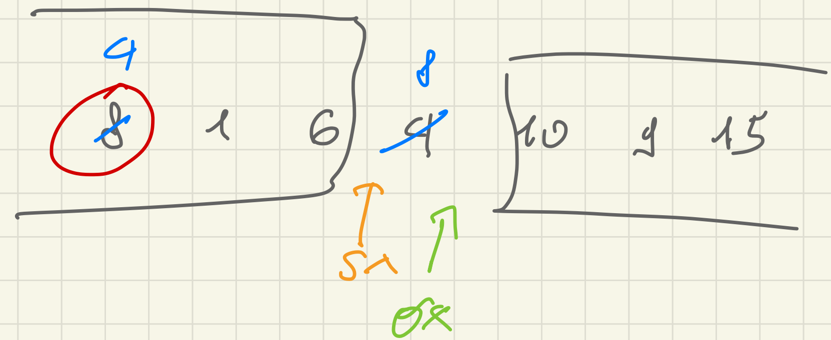
Caso migliore

$$C_b(n) = \begin{cases} 0 & \text{se } n \le 1 \\ n + \min\{ C(k) + C(n-k-1) \mid k = 0 .. n-1 \} \end{cases}$$

partizione bilanciata $\quad k \approx \frac{n}{2}$

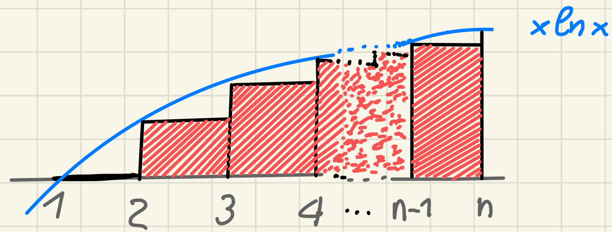
$$C_b(n) = n + 2C_b\left(\frac{n}{2}\right)$$

$$C_b(n) \approx n \log_2 n$$



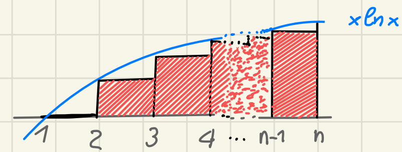
Problema

Trovare una limitazione superiore per $\sum_{i=2}^{n-1} i \ln i$



$$\sum_{i=2}^{n-1} i \ln i \leq \int_2^n x \ln x \, dx$$

$$\sum_{i=2}^{n-1} i \ln i \le \int_2^n x \ln x \, dx$$



$$\int x \ln x \, dx = \frac{x^2}{2} \ln x - \int \frac{1}{x} \frac{x^2}{2} \, dx$$

$$= \frac{x^2}{2} \ln x - \int \frac{x}{2} \, dx$$

$$= \frac{x^2}{2} \ln x - \frac{x^2}{4} + C$$

integrazione per parti

$$\int f(x) g'(x) \, dx = f(x) \cdot g(x) - \int f'(x) g(x) \, dx$$

$f(x) = \ln x \qquad g'(x) = x$

$f'(x) = \frac{1}{x} \qquad g(x) = \frac{x^2}{2}$

$$\left[\frac{x^2}{2} \ln x - \frac{x^2}{4} \right]_2^n = \frac{n^2}{2} \ln n - \frac{n^2}{4} - 2 \ln 2 + 1 \le \frac{n^2}{2} \ln n - \frac{n^2}{4}$$

$$\sum_{i=2}^{n-1} i \ln i \le \frac{n^2}{2} \ln n - \frac{n^2}{4}$$

Caso medio

$$C(n) = \begin{cases} 0 & \text{se } n \leq 1 \\ \dfrac{1}{n} \displaystyle\sum_{k=0}^{n-1} \left[n + C(k) + C(n-k-1) \right] & \text{altrimenti} \end{cases}$$

$n \geq 2$

$$C(n) = \frac{1}{n} \sum_{k=0}^{n-1} n + \underbrace{\frac{1}{n} \sum_{k=0}^{n-1} C(k)} + \underbrace{\frac{1}{n} \sum_{k=0}^{n-1} C(n-k-1)}$$

$$= n + \frac{2}{n} \sum_{i=0}^{n-1} C(i)$$

$$C(n) = \begin{cases} 0 & \text{se } n \leq 1 \\ n + \frac{2}{n} \sum_{i=2}^{n-1} C(i) & \text{altrimenti} \end{cases}$$

$C(0) = C(1) = 0$

Verifichiamo che $C(n) \leq 2n \ln n$ per $n \geq 1$

Dim induzione

Base $n = 1$ $\quad C(1) = 0 \quad 2n \ln n = 2 \ln 1 = 0$ ok

Induzione $\quad C(n) = n + \frac{2}{n} \sum_{i=2}^{n-1} C(i) \underset{\text{ip. ind}}{\leq} n + \frac{2}{n} \sum_{i=2}^{n-1} (2 i \ln i)$

$$= n + \frac{2}{n} \cdot 2 \sum_{i=2}^{n-1} i \ln i \leq n + \frac{4}{n} \left[\frac{n^2}{2} \ln n - \frac{n^2}{4} \right]$$

$$= \cancel{n} + 2n \ln n - \cancel{n} = 2n \ln n \approx 1.39\, n \log_2 n$$

cfr in media $\quad C(n) \approx 1.39\, n \log_2 n$