Algoritmi e Strutture Dati

Lezione 11

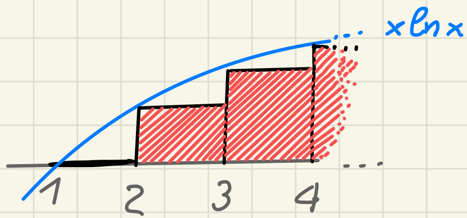
21 ottobre 2022

Problema

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

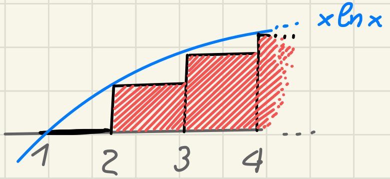
Problema

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



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

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



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

$$= \frac{x^2}{2} \ln x - \int \frac{x}{2} \, dx = \frac{x^2}{2} \ln x - \frac{x^2}{4}$$

integrazione per parti

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

$f(x) = \ln x$ $\quad$ $f'(x) = \frac{1}{x}$

$g'(x) = x$ $\quad$ $g(x) = \frac{x^2}{2}$

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

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

QuickSort

- caso semplice (≤ 1 elemento) $\rightarrow$ sol. immediata
- altrimenti
  - dividi l'array in 2 parti
  - ordinale separatamente
  - combina le due soluzioni

22 11 47 54 23 96 1 100 24

perno
pivot

54

< pivot

> pivot

22 11 47 54 23 1 24

96 100

1 11 22 23 24 47 54

96 100

ALGORITMO quickSort (array A)

IF lunghezza di $A > 1$ THEN

scegli un elemento x di A

$B \leftarrow \{ y \in A \mid y \leq x \}$

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

$\Big\}$ $\rightarrow$ partizionamento

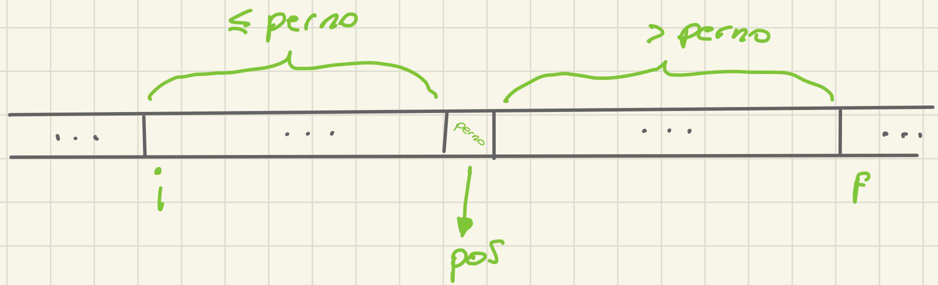
quickSort(B)

quickSort(C)

$A \leftarrow$ concatenazione di B e di 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:

$$\cdots \overbrace{12 \mid 8 \mid 2 \mid 1}^{\le \text{perno}} \mid \underbrace{17}_{\text{pos}} \mid \overbrace{46 \mid 33 \mid 24 \mid 22 \mid 45}^{> \text{perno}} \cdots$$

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

$$\cdots \overbrace{24 \mid 8 \mid 2 \mid 1 \mid 22 \mid 12 \mid 17}^{\le \text{perno}} \mid \underbrace{33}_{\text{pos}} \mid \overbrace{46 \mid 45}^{> \text{perno}} \cdots$$

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

7	2	9	12		15		33
~~12~~	~~35~~	~~15~~	~~7~~	19	~~9~~	56	~~2~~
perno			sx dx				

perno: 1° elemento

scansioni da dx fino al 1° elemento ≤ perno

scansioni da sx fino al 1° elemento > perno

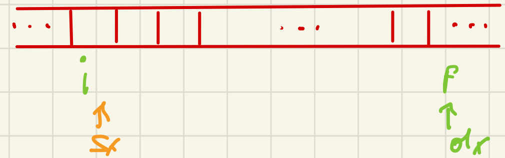
scambio

ferma a quando le due scansioni si incontrano

scambia elemento di posto di dx con perno

restringi dx

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



perno ← A[i]

dx ← f, sx ← i

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

A[i] ... A[sx−1] ≤ perno

A[dx+1] .. A[f−1] > perno

```
ALGORITMO partiziona (Array A, indice i, indice f) → indice
  perno ← A[i]
  dx ← f,  sx ← i
  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
```

12 13 14 15

dx sx

17 23 24 31

~~24~~ 8 ~~31~~ 16 ~~17~~ 40 29 ~~25~~ 28

sx dx

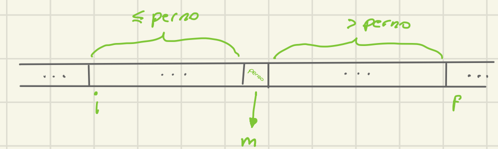
2 12 23 33

~~25~~ 7 4 ~~33~~ ~~2~~ 27 55 ~~12~~ 24 42

sx dx

\# confronti è $n-1$ o n

```
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)
```

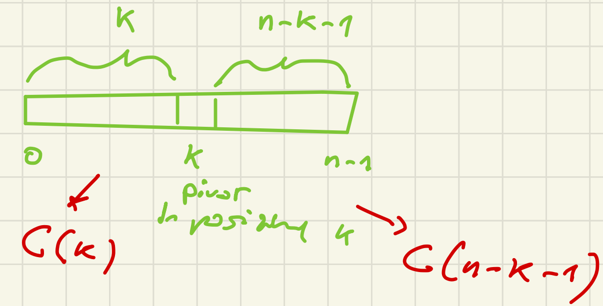



```
ALGORITMO quickSort(Array A[0..n-1])
    quickSort(A, 0, n)
```

Calcolo del n° di confronti $C(n)$ $\quad$ $n = \#$ chiavi

· Partiziona $\qquad C_{part}(n) = n$

· Chiamate ricorsive



$$C(n) = \underbrace{n}_{C_{part}} + \underbrace{C(k)}_{1^a \text{ chiamata ric}} + \underbrace{C(n-k-1)}_{2^a \text{ chiamata ricorsiva}}$$

```
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 $C_w(n)$

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

max per $k=0$ o $k=n-1$

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

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

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

$$= \Theta(n^2)$$

CASO PEGGIORE

3 | 5 10 15 20

Caso migliore

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

partizione bilanciata

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

$$C_b(n) \approx n \log_2 n \quad \text{caso migliore}$$

8 1 6 4 10 9 15