Stimiamo quanti accessi alla lista l vengono effettuati durante l'esecuzione di merge-sort(l), in funzione della lunghezza n di l

---

Prima di tutto, facciamo la stessa cosa con merge.

Domanda

Se l_1 è lunga n_1 l_2 è lunga n_2, quanti accessi alla lista compie la funzione merge?

Risposta

- il numero di volte che scriviamo nella lista destinazione è $n_1 + n_2$
- il numero di volte che leggiamo da l_1 per scrivere in l è n_1
- " " " " l_2 per scrivere in l è n_2
- il numero di if $l_1[i_1] < l_2[i_2]$ che eseguiamo è al massimo uguale alla lunghezza della lista destinazione, quindi $n_1 + n_2$. Ogni if causa 2 accessi alla lista

$\Downarrow$

Totale: al massimo $\boxed{4\,n_1 + 4\,n_2 \text{ accessi a lista.}}$

Domanda: quanti accessi alla lista esegue merge sort se la lunghezza della lista in input è n?

$$n + \left(4n/2 + 4n/2\right) + 2 * \text{il numero di accessi di merge-sort per liste lunghe } n/2$$

per copiare l in first e second

il tempo per il merge

Chiamiamo T_n il numero di accessi a lista che compie merge sort quando l'input è lungo n

$\to T_n = \boxed{n + (4\,n/2 + 4\,n/2)} + 2\,T_{n/2}$ per $n \geq 2$

$= 5n + 2\,T_{n/2}$

$T_1 = 0$

$T_0 = 0$

Successione definita per ricorsione

$T_2 = 5 \cdot 2 + 2\,T_1 = 10 + 0 = 10$

$T_4 = 5 \cdot 4 + 2\,T_2 = 20 + 2 \cdot 10 = 20 + 20 = 40$

$n = 2^k$

$T_n = \boxed{5 \cdot n + 2\,T_{n/2}}$

$T_x = 5x + 2\,T_{x/2}$

$= 5 \cdot n + 2\left(5 \cdot n/2 + 2\,T_{n/4}\right)$

$= \underbrace{5n + 5n}_{\text{2 copie}} + 4\,T_{n/4}$ $\quad 4 = 2^2$

$= 5n + 5n + 4\left(5 \cdot n/\cancel{4} + 2\,T_{n/8}\right)$

$= \underbrace{5n + 5n + 5n}_{\text{3 copie}} + 8\,T_{n/8}$ $\quad 8 = 2^3$

$= \underbrace{5n + 5n + 5n \ldots}_{k \text{ copie}} + n\,T_{n/n}$ $\quad n = 2^k$

$n/n = 1$

$k = \log n$

$= 5kn = 5n\log n$

$\Downarrow$

L'algoritmo merge_sort ha complessità $O(n \log n)$