ANALISI ASINTOTICA QUICKSORT

Siccome è un algoritmo ricorsivo, scriviamo l'equazione di ricorrenza

$$T_{QS} = \begin{cases} \Theta 1 & n \leq 1 \\ T_{QS}(n_1) + T_{QS}(n_2) + \Theta 1 + T_P(n) & \end{cases}$$

Sappiamo che $n_1 + n_2 = n \rightarrow n_2 = n - n_1$ quindi scriveremo anziché $T_{QS}(n_2)$ $T_{QS}(n - n_1)$.

Sappiamo che $1 \leq n_1 < n$ (numero di elementi della partizione di sinistra) ma gli estremi di n_1 sono p e q (dell'algoritmo), in particolare $n_1 = q - p + 1$

ANALISI ASINTOTICA PARTIZIONAMENTO

Per studiare il tempo di esecuzione studiamo le operazioni che si ripetono più volte ovvero i decrementi di j e gli incrementi di di i (all'interno dei cicli), quindi possiamo dire che $T_p(n) = O(t_i + t_j)$

Sappiamo che $t_i + t_j = \Theta n$ perché la somma tra incrementi e decrementi è uguale a n o $n+1$ quindi il tempo è lineare e possiamo dedurre che $T_p = \Theta(n)$

QUICK SORT CASO SEQUENZA ORDINATA

Se la sequenza in input è ordinata, le partizioni saranno (ricorsivamente) 1/n-1 n volte. L'equazione di ricorrenza in questo caso è:

$$T_{QS} = \begin{cases} \Theta 1 & n \leq 1 \\ T_{QS}(1) + T_{QS}(n-1) + \Theta(1) + \Theta(n) \longrightarrow T_{QS}(n-1) + \Theta(n) \end{cases}$$

$T_{QS}(1) = \Theta(1)$ (caso base) che viene assorbito (assieme all'altro $\Theta(1)$) da $T(n)$

Vediamo adesso $T_{QS}(n-1)$. Il caso è analogo allo studio del fattoriale quindi possiamo studiare l'albero degenere associato

0 (n-1) – n-1

1 (n-2) – n-2

⋮

h (1) – 1

il contributo di ogni livello è sempre l'input

Il contributo totale è $\sum_{i=0}^{h} \Theta(n-i) = \Theta \sum_{i=1}^{n} i = \Theta n^2$

In definitiva $T_{QS} = \Theta(n^2)$ (in questo caso)

QUICK SORT CASO TUTTI UGUALI

Nel caso di una sequenza tutta uguale, partiziona divide sempre in 2 la sequenza, l'equazione risulta essere

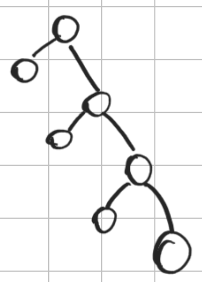
$$T_{QS} = \begin{cases} \Theta 1 & n \leq 1 \\ 2T_{QS}(n/2) + \Theta(n) \end{cases}$$

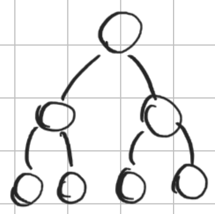
Che è proprio l'equazione di ricorrenza di merge sort quindi

$T_{QS} = \Theta n \log n$

ALBERI A CONFRONTO

Se studiamo i due alberi costruiti dalle equazioni di ricorrenza avremo i seguenti alberi:





I due alberi hanno lo stesso numero di foglie e stesso numero di nodi interni. In particolare sia f il n° di foglie e m n° di nodi avremo che $m = 2f - 1$ (vale per alberi binari che hanno ogni nodo interno di grado 2). Si osservi che le foglie rappresentano una chiamata al caso base

DIMOSTRAZIONE

Dimostriamo per induzione in cui la nostra proposizione è

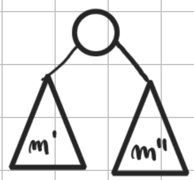
$n(f) = 2f - 1 \quad \forall f \geq 1$

L' ipotesi induttiva sarà: $\forall f > 1 \; (\forall z < f \;\; n(z) = 2z - 1) \Rightarrow n(f) = 2f - 1$

- Per $f = 1$ l' unico albero in cui tutti i nodi interni hanno grado 2 e ha solo una foglia è l' albero formato da un solo nodo

- Vediamo per $f > 1$.

La radice è un nodo interno, avremo quindi per l' albero che stiamo considerando due sottoalberi non vuoti



Il numero di nodi totale sarà $n = 1 + n' + n''$ (sottoalberi).

Se l' albero totale ha f foglie $f = f' + f''$ (la radice non è foglia)

in particolare sappiamo che f' e $f'' \geq 1$, ne deriva che $f' < f$ e $f'' < f$.

Ma scoperto ciò possiamo applicare l' ipotesi induttiva perché

supposta vera questa possiamo dire che $n' = 2f' - 1$, ma allora l'implicazione è verificata perchè:

$$n = n' + n'' = 1 + 2f' - 1 + 2f'' - 1 = 2(f' + f'') - 1 \checkmark$$

OSS

Da questa dimostrazione deduciamo che in alberi binari con stesso numero di foglie e di nodi ogni livello apporta un costo lineare all'algoritmo quindi il fattore determinante è l'ALTEZZA dell'albero.
In particolare avremo altezza minima per un albero pieno quindi prima abbiamo studiato proprio caso peggiore (albero degenere) in cui $T_{QS} = \Theta(n^2)$ e caso migliore (albero pieno) in cui $T_{QS} = \Theta(n \lg n)$

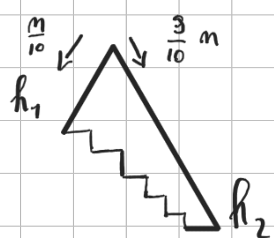
ANALISI CASISTICA

Proviamo a capire quanto è "probabile" incorrere in un caso che tende al migliore o che tende al peggiore.

Partiamo dal caso migliore e degeneriamo.

Anzichè avere $n_1 = \frac{n}{2}$ ipotizziamo $n_1 = \frac{n}{10}$ e quindi $n_1 - n = \frac{9}{10}$

$$T_{QS} = \begin{cases} \Theta(1) \\ T_{QS}(n/10) + T_{QS}(9/10\, n) + \Theta(n) \end{cases}$$



Ogni livello avrà a sinistra $\frac{n}{10^i}$ nodi mentre a destra $\left(\frac{9}{10}\right)^i n$

$\left(\frac{9}{10}\right)^i n = 1 \rightarrow$

Troviamo le due altezze:

Il cas base a sinistra si avrà quando l'input del livello è 1

$\frac{n}{10^i} = 1 \Rightarrow$ per l'altezza avremo $\Rightarrow \frac{n}{10^h} = 1 \rightarrow h = \log_{10} n$

Analogamente vale per $h_2 = \log_{\frac{10}{9}} n$

Quindi per le loro altezze avremo un apporto di

$\sum_{i=0}^{\log_{10} n} \Theta n$ e $\sum_{i=0}^{\log_{10/9} n} \Theta n$ (ogni livello sommando i rami apporta n)

Queste sono uguali a $\Theta(n \log_{10} n)$ e $\Theta(n \log_{\frac{10}{9}} n)$

Possiamo cambiare la base e portarle in base 2 per avere:

$\log_{10} n = (\log_{10} 2) \log_2 n$ e $\log_{\frac{9}{10}} n = (\log_{\frac{10}{9}} 2) \log_2 n$

Ma essendo $\log_2 n$ per una costante, asintoticamente sono uguali a $\Theta(n \lg n)$

OSS

In particolare ogni caso del tipo αn e $(1-\alpha) n$ da origine allo stesso risultato, tenderà sempre al caso migliore

Prendiamo un caso più particolare in cui si alterna un livello con scomposizione migliore e uno con peggiore.

Ricordando che $h_p = n$ e $h_M = \log_2 n$ sono le altezze, ancora una volta il carattere determinante è l'altezza.

Nel caso migliore arriviamo alle foglie dell'albero dopo $\log_2 n$ livelli, in questo caso ci impiegheremo il doppio dei livelli perché viene rallentato un livello sì e uno no quindi avrà altezza banalmente $2 \lg n$.

Ogni livello ha un apporto lineare, il risultato sarà nuovamente $\Theta(n \lg_2 n)$

Ci avviciniamo al caso peggiore quando la partizione è del tipo $k / n-k$ con k costante perché asintoticamente avrò altezza $\frac{n}{k}$ che per un apporto lineare mi darà $\Theta(n^2)$.