

Alcuni esercizi di Algoritmi e Strutture Dati I

1 Ricorrenze

Esercizio 1. Dimostrare che la soluzione della ricorrenza $T(n) = T(\lceil n/2 \rceil) + 1$ appartiene a $O(\log n)$.

La difficoltà sta nella presenza dell'arrotondamento superiore, visto che stiamo calcolando un limite superiore.

Per avere una idea qualitativa, partiamo dal caso $n = 2^t$ in modo da non avere problemi di arrotondamento. Se poniamo ad esempio $T(1) = 1$, la soluzione esatta è:

$$\begin{aligned} T(2^t) &= 1 + T(2^{t-1}) \\ &= 2 + T(2^{t-2}) \\ &= t + T(2^0) \\ &= t + 1 \end{aligned}$$

Sulle potenze di due, dunque, $T(n) = \log_2 n + 1$. Questo suggerisce di tentare, nel caso generale, una ipotesi induttiva della forma $T(n) \leq a \log_2 n + b$. Del termine b si vede la necessità nel caso base, in quanto vogliamo che $1 = T(1) \leq a \log_2 1 + b = b$. Questo fornisce il vincolo $b \geq 1$. Per quanto riguarda il passo induttivo, sfruttando l'ipotesi induttiva otteniamo

$$T(n) \leq (a \log_2 \lceil n/2 \rceil + b) + 1,$$

ci basta che sia

$$(a \log_2 \lceil n/2 \rceil + b) + 1 \leq a \log_2 n + b.$$

Essendo in questa disequazione il valore di b indifferente, possiamo prendere $b = 1$ e, notando che $\lceil n/2 \rceil \leq \frac{n+1}{2}$, imporre il vincolo ancora più stretto

$$a \log_2 \frac{n+1}{2} + 1 \leq a \log_2 n$$

che si può riscrivere

$$a \log_2 \frac{2n}{n+1} \geq 1.$$

Siccome $\frac{2n}{n+1} = \frac{2n+2}{n+1} - \frac{2}{n+1} = 2 - \frac{2}{n+1}$ è una funzione crescente, e stiamo considerando $n \geq 2$, abbiamo $\frac{2n}{n+1} \geq 4/3$. Ora, $\log_2 \frac{4}{3} > \geq \frac{1}{3}$ da cui è sufficiente prendere $a = 3$; in definitiva, otteniamo che

$$T(n) \leq 3 \log_2 n + 1$$

che è una limitazione molto rozza ma sufficiente dato che ci disinteressiamo alla costante. Più sotto troveremo la soluzione esatta di questa ricorrenza.

Esercizio 2. Risolvere esattamente la ricorrenza

$$\begin{aligned} T(1) &= 1, \\ T(n) &= T(\lfloor n/2 \rfloor) + 1 \end{aligned}$$

Sappiamo che $T(n)$ ha un comportamento logaritmico, dobbiamo però trovarne la forma esatta. Conviene tabulare qualche valore:

n	$T(n)$
1	1
2	2
3	2
4	3
5	3
6	3
7	3
8	4

La funzione cambia in corrispondenza delle potenze di due, quindi fa certamente riferimento a $\lfloor \log_2 n \rfloor$. Se facesse riferimento a $\lceil \log_2 n \rceil$, scatterebbe infatti in corrispondenza delle potenze di due aumentate di uno.

Si vede facilmente che $T(n) = 1 + \lfloor \log_2 n \rfloor$; si tratta di dimostrarlo per induzione.

Base: $T(1) = 1 + \lfloor \log_2 1 \rfloor$, nessun problema.

Passo induttivo: dobbiamo mostrare che

$$1 + \lfloor \log_2 n \rfloor = T(n) = 1 + T(\lfloor n/2 \rfloor) = 1 + (1 + \lfloor \log_2 \lfloor \frac{n}{2} \rfloor \rfloor)$$

ossia

$$\lfloor \log_2 n \rfloor = 1 + \lfloor \log_2 \lfloor \frac{n}{2} \rfloor \rfloor$$

identità piuttosto difficile da trattare.

Meglio cambiare punto di vista: anziché esprimere $T(n)$ in funzione di n , proviamo a fare l'inverso, esprimendo n in funzione di $T(n)$. Siccome $n \rightarrow T(n)$ non è iniettiva, $T^{-1}(t)$ è un insieme. Guardando ai valori tabulati, si capisce subito quale:

$$T^{-1}(t) = [2^{t-1}, 2^t - 1].$$

Questo ci suggerisce la seguente ipotesi induttiva:

$$n \in [2^{T(n)-1}, 2^{T(n)} - 1].$$

La base è vera; entrambi gli estremi dell'intervallo sono uno. Per il passo induttivo, supponiamo dunque che

$$\lfloor \frac{n}{2} \rfloor \in [2^{T(\lfloor \frac{n}{2} \rfloor)-1}, 2^{T(\lfloor \frac{n}{2} \rfloor)} - 1].$$

Per definizione di arrotondamento inferiore, se $\lfloor \frac{k}{2} \rfloor \in [r, s]$, segue $k \in [2r, 2s + 1]$. Quindi

$$\begin{aligned} n &\in [2 \cdot 2^{T(\lfloor \frac{n}{2} \rfloor)-1}, 2 \cdot (2^{T(\lfloor \frac{n}{2} \rfloor)} - 1) + 1] \\ &= [2^{T(\lfloor \frac{n}{2} \rfloor)}, 2^{T(\lfloor \frac{n}{2} \rfloor)+1} - 1] \\ &= [2^{T(n)-1}, 2^{T(n)} - 1] \end{aligned}$$

come volevamo.

Ora, dalla proprietà dimostrata per induzione e dalla definizione di logaritmo segue la forma chiusa che avevamo intuito.

Esercizio 3. Risolvere esattamente la ricorrenza

$$\begin{aligned} S(1) &= 1, \\ S(n) &= S(\lceil n/2 \rceil) + 1 \end{aligned}$$

Affianchiamo i primi valori di $S(n)$ a quelli della $T(n)$ di sopra:

n	T(n)	S(n)
1	1	1
2	2	2
3	2	3
4	3	3
5	3	4
6	3	4
7	3	4
8	4	4
9	4	5

È facile notare che per $n \geq 2$, $S(n) = T(n-1) + 1$.

Dimostriamolo per induzione, partendo dalla base $n = 2$. $S(2)$ infatti lo possiamo calcolare a mano, verificando direttamente che $S(2) = 2 = T(1) + 1$.

Per il passo induttivo,

$$\begin{aligned}
 S(n) &= 1 + S(\lceil n/2 \rceil) \\
 &= 1 + S\left(\left\lfloor \frac{n+1}{2} \right\rfloor\right) \\
 &= 1 + \left(T\left(\left\lfloor \frac{n+1}{2} \right\rfloor - 1\right) + 1\right) \\
 &= 1 + \left(T\left(\left\lfloor \frac{n-1}{2} \right\rfloor\right) + 1\right) \\
 &= 1 + T(n-1)
 \end{aligned}$$

dove abbiamo sfruttato, nell'ordine: la definizione ricorsiva di $S(n)$; il fatto che $\lceil \frac{n}{2} \rceil = \lfloor \frac{n+1}{2} \rfloor$; un altro fatto immediato sugli arrotondamenti; infine, la definizione ricorsiva di $T(n)$. In definitiva, nota la relazione che lega $S(n)$ a $T(n)$, e nota la forma chiusa di $T(n)$, otteniamo per $n \geq 2$

$$S(n) = 2 + \lfloor \log_2(n-1) \rfloor$$

da cui si vede che è inevitabile descrivere $S(1)$ a parte, in quanto il logaritmo di zero non esiste.

Esercizio 4. Dimostrare che, detto $C(k, h)$ il numero di confronti effettuati dalla procedura merge applicata a due sequenze lunghe k e h , si ha $\min(k, h) \leq C(k, h) \leq k + h - 1$

Nell'algorithm mergesort, abbiamo $k = \lfloor n/2 \rfloor$ e $h = \lceil n/2 \rceil$ (o viceversa), da cui il numero di confronti sta nell'intervallo $[\lfloor n/2 \rfloor, n - 1]$.

Esercizio 5. Trovare in forma chiusa la soluzione della ricorrenza che descrive il caso pessimo di mergesort,

$$\begin{aligned}
 M(1) &= 0, \\
 M(n) &= M\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + M\left(\left\lceil \frac{n}{2} \right\rceil\right) + n - 1.
 \end{aligned}$$

Cominciamo a risolvere il problema nel caso delle potenze esatte, $n = 2^t$. Qui abbiamo $M(2^t) = 2M(2^{t-1}) + 2^t - 1$, quindi possiamo definire la ricorrenza ausiliaria sugli esponenti come segue: $A(t) = M(2^t)$. Di conseguenza,

$$\begin{aligned}
 A(0) &= 0 \\
 A(t) &= 2A(t-1) + 2^t - 1.
 \end{aligned}$$

È comodo disegnare l'albero di ricorsione, coi livelli numerati da t (radice) fino a 0 (foglie), e verificare che, sopra le foglie, a livello $t - j$ il lavoro di combinazione è pari a $2^t - 2^j$. Tale formula è valida anche per le foglie, dato che il lavoro compiuto al livello delle foglie è nullo. Abbiamo dunque

$$\begin{aligned}
 A(t) &= (2^t - 1) + (2^t - 2) + (2^t - 4) + \dots + (2^t - 2^t) \\
 &= (t + 1)2^t - (1 + 2 + 4 + \dots + 2^t) \\
 &= (t + 1)2^t - (2^{t+1} - 1) \\
 &= (t - 1)2^t + 1
 \end{aligned}$$

ossia, in termini di n , $M(n) = n \log_2 n - n + 1$.

Veniamo al caso generale. Come curiosità, consideriamo anche la ricorrenza relativa al caso migliore, quello in cui ogni operazione di merge esaurisce nel modo più rapido possibile la lista più corta:

$$L(1) = 0,$$

$$L(n) = L\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + L\left(\left\lceil \frac{n}{2} \right\rceil\right) + \left\lfloor \frac{n}{2} \right\rfloor.$$

Con una semplice procedura

```

void table_mergesort()
{
    int i;
    M[1]= 0;
    cout<<"      n      L(n)    M(n)    M(n)-M(n-1) "<<endl<<endl;
    for (i=2; i<18; i++)
        M[i]= M[i/2] + M[(i+1)/2] + i - 1;
    for (i=2; i<18; i++)
        L[i]= L[i/2] + L[(i+1)/2] + i/2;
    for (i=1; i<18; i++) {
        cout<<"      "<<i<<"      "<<L[i]<<"      "<<M[i];
        if (i>1)
            cout<<"      "<<M[i]-M[i-1];
        cout<<endl;
    }
}

```

possiamo tabulare i primi valori di tali ricorrenze, dove di $M(n)$ abbiamo anche considerato la successione delle differenze.

n	L(n)	M(n)	M(n)-M(n-1)
1	0	0	
2	1	1	1
3	2	3	2
4	4	5	2
5	5	8	3
6	7	11	3
7	9	14	3
8	12	17	3
9	13	21	4
10	15	25	4
11	17	29	4
12	20	33	4

13	22	37	4
14	25	41	4
15	28	45	4
16	32	49	4
17	33	54	5

Si nota che, detto $D(n) = M(n) - M(n-1)$, abbiamo $D(n) = T(n-1)$. Verifichiamolo, trovando una ricorrenza per $D(n)$:

$$\begin{aligned}
D(n) &= M(n) - M(n-1) \\
&= M\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + M\left(\left\lceil \frac{n}{2} \right\rceil\right) + n - 1 - [M\left(\left\lfloor \frac{n-1}{2} \right\rfloor\right) + M\left(\left\lceil \frac{n-1}{2} \right\rceil\right) + n - 2] \\
&= M\left(\left\lfloor \frac{n}{2} \right\rfloor\right) - M\left(\left\lfloor \frac{n-1}{2} \right\rfloor\right) + 1 \\
&= M\left(\left\lfloor \frac{n}{2} \right\rfloor\right) - M\left(\left\lfloor \frac{n-2}{2} \right\rfloor\right) + 1 \\
&= M\left(\left\lfloor \frac{n}{2} \right\rfloor\right) - M\left(\left\lfloor \frac{n}{2} - 1 \right\rfloor\right) + 1 \\
&= D\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1
\end{aligned}$$

che è la stessa ricorrenza per $T(n)$, tranne che la condizione iniziale $D(2) = 1$ sostituisce $T(1) = 1$, causando così uno shift di tutta la sequenza. Otteniamo dunque $D(n) = 1 + \lfloor \log_2(n+1) \rfloor = \lceil \log_2 n \rceil$, per ogni $n \geq 2$. Questo, assieme alla forma chiusa trovata per $M(2^t)$, ci permette di risolvere il caso generale. Detto infatti $t = \lfloor \log_2 n \rfloor$ (cosicché 2^t è la più grande potenza di due non superiore a n) abbiamo per ogni $n \geq 1$

$$\begin{aligned}
M(n) &= M(2^t) + \sum_{j=2^{t+1}}^n D(j) \\
&= t2^t - 2^t + 1 + \sum_{j=2^{t+1}}^n \lceil \log_2 j \rceil \\
&= t2^t - 2^t + 1 + (n - 2^t) \lceil \log_2 n \rceil \\
&= 2^{\lfloor \log_2 n \rfloor} (\lfloor \log_2 n \rfloor - 1) + 1 + (n - 2^{\lfloor \log_2 n \rfloor}) \lceil \log_2 n \rceil.
\end{aligned}$$