Esercizi su ricorrenze

I segienti esercizi sono proposti (prevalentemente) senza soluzioni.

Esercizio 1.1 Risolvere, verificando la correttezza della soluzione trovata, la seguente equazione di ricorrenza valida *per ogni* $n \geq 0$:

$$T(n) = T\left(n-2\right) + 4n, \; n \geq 2, \quad T(1) = 4 \;\;, \;\; T(0) = 1$$

Esercizio 1.2 Si consideri la seguente equazione di ricorrenza quando il parametro n è della forma 2^{2^i}, con $i \geq 0$:

$$T(n) = \frac{\sqrt{n}}{2} T\left(\sqrt{n}\right), \; n > 2, \quad T(2) = 2.$$

(a) Risolvere la ricorrenza, determinando la formula esatta per $T(n)$.

(b) Verificare la soluzione ottenuta utilizzando l'induzione.

Esercizio 1.3 Si consideri la seguente equazione di ricorrenza quando il parametro n è della forma 2^{2^i}, con $i \geq 0$:

$$T(n) = T\left(\sqrt{n}\right) + \left(\log n\right)^2, \; n > 2, \quad T(2) = 0.$$

(a) Risolvere la ricorrenza, determinando la formula esatta per $T(n)$.

(b) Verificare la soluzione ottenuta utilizzando l'induzione.

Esercizio 1.4 Si risolva la seguente equazione di ricorrenza per valori del parametro $n = 3^{3^i}$:

$$T(n) = \begin{cases} T\left(\sqrt[3]{n}\right) + 2\log_3 \log_3 n, & n > 3, \\ 0 & n = 3. \end{cases}$$

Esercizio 1.5 Si consideri la seguente equazione di ricorrenza per valori di n che siano potenze di due:

$$T(n) = \begin{cases} T(n/2) + T(n/4) + 2T(n/8) + n, & n \geq 8, \\ 4 & n < 8. \end{cases}$$

Si determini una costante $c > 0$ tale che $\forall n, T(n) =\geq cn \log n$.

Esercizio 1.6 Si consideri la seguente relazione di ricorrenza definita per valori arbitrari positivi del parametro n:

$$T(n) = \begin{cases} 3, & n \leq 6, \\ T\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil - 1\right) + n, & n > 6. \end{cases}$$

Si dimostri che $T(n) = O(n)$.

Esercizio 1.7 Si risolva la seguente equazione di ricorrenza per valori del parametro $n = 2^i$:

$$T(n) = \begin{cases} 3T\left(\frac{n}{2}\right) + 7n^{\log_2 3}, & n > 1, \\ 1 & n = 1. \end{cases}$$

Esercizio 1.8 Si risolva e si verifichi la seguente equazione di ricorrenza per valori del parametro $n = 2^{2^i}$:

$$T(n) = \begin{cases} \sqrt{2}T\left(\sqrt{n}\right) + \sqrt{\log n}, & n > 2, \\ 0 & n = 2. \end{cases}$$

Esercizio 1.9 Si consideri la seguente equazione di ricorrenza definita per ogni valore positivo del parametro n:

$$T(n) = \begin{cases} 7T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n^3, & n > 1, \\ 8 & n = 1. \end{cases}$$

Utilizzando l'induzione parametrica, si dimostri che $T(n) = O(n^3)$.

Esercizio 1.10 Al variare dei due parametri interi positivi a e b, si consideri la seguente famiglia di ricorrenze, definite per $n = 2^{2^i}$:

$$T_{a,b}(n) = \begin{cases} 2^a T_{a,b}\left(\sqrt{n}\right) + \log^b n, & n > 2, \\ 0 & n = 2. \end{cases}$$

Si determini l'ordine di grandezza di $T_{a,b}(n)$ in funzione di n e dei due parametri a e b. (*Suggerimento:* Si applichi la formula generale e si discutano vari casi a seconda dei valori di a e b ...)

Answer: Let us apply the general formula with $s(n) = 2^a$, $f(n) = \sqrt{n}$, $w(n) = \log^b n$, $T_0 = 0$ and $n_0 = 2$. We have $f^{(i)}(n) = n^{1/2^i}$ and $f^\star(n, 2) = \log\log n - 1$. Moreover, $\prod_{j=0}^{\ell-1} 2^a = 2^{a\ell}$ and $w(f^{(\ell)}(n)) = \log^b n^{1/2^\ell} = (1/2^{b\ell}) \log^b n$. Therefore:

$$
\begin{aligned}
T(n) &= \sum_{\ell=0}^{\log\log n-1} 2^{(a-b)\ell} \log^b n \\
&= \log^b n \sum_{\ell=0}^{\log\log n-1} (2^{a-b})^\ell.
\end{aligned}
$$

Let $S(n) = \sum_{\ell=0}^{\log\log n-1} (2^{a-b})^\ell$. We have three cases, depending on the values of a and b. If $a < b$, then $S(n) = \Theta(1)$, hence $T_{a,b}(n) = \Theta(\log^b n)$. If $a = b$, then $S(n) = \log\log n$, hence $T_{a,b}(n) = \Theta(\log^b n \log\log n)$. Finally, if $a > b$, then $S(n) = \Theta((2^{a-b})^{\log\log n}) = \Theta(\log^{a-b} n)$, hence $T_{a,b}(n) = \Theta(\log^a n)$. □

Esercizio 1.11 Si consideri la seguente equazione di ricorrenza, definita per valori arbitrari parametro $n > 0$:

$$
T(n) = \begin{cases} \lfloor\sqrt{n}\rfloor T\left(\lfloor\sqrt{n}\rfloor\right) + n^2, & n > 2, \\ 1 & n = 1, 2. \end{cases}
$$

Si dimostri, utilizzando l'induzione parametrica, che $T(n) \in O(n^2)$.

Answer:

We postulate that there exists a positive constant c for which $T(n) \leq cn^2$, for any value of $n > 0$, and then proceed to find an appropriate value for c using parametric induction. Consider the base cases first. It must be $T(1) = 1 \leq c \cdot 1^2$ and $T(2) = 1 \leq c \cdot 2^2$, whence $c \geq \max\{1, 1/4\} = 1$. For $n \geq 3$, assume that $T(k) \leq ck^2$ for $k < n$. When $k = n$ we obtain:

$$
\begin{aligned}
T(n) &\leq \lfloor\sqrt{n}\rfloor T(\lfloor\sqrt{n}\rfloor) + n^2 \\
&\leq cn\sqrt{n} + n^2
\end{aligned}
$$

It then suffices to choose a constant c such that for all $n \geq 3$,

$$
cn\sqrt{n} + n^2 \leq cn^2 \Leftrightarrow
$$

$$\begin{aligned} c+\sqrt{n} &\leq c\sqrt{n} \Leftrightarrow \\ c &\geq \sqrt{n}/(\sqrt{n}-1) \end{aligned}$$

Note that $f(n) = \sqrt{n}/(\sqrt{n}-1)$ is a strictly decreasing function over its domain, hence it suffices to choose $c \geq f(3) = \sqrt{3}/(\sqrt{3}-1) > 1$. Putting it all together, we have proved that

$$T(n) \leq \frac{\sqrt{3}}{\sqrt{3}-1} n^2.$$

$\square$

Esercizio 1.12 Sia n una potenza di due. Dato un problema Π, sia A^1_Π un algoritmo ricorsivo per Π la cui complessità in tempo $T_1(n)$ obbedisce alla seguente ricorrenza:

$$T_1(n) = \begin{cases} 1 & n = 1, \\ 2T_1\left(\frac{n}{2}\right) + 2n & n > 1. \end{cases}$$

Sia A^2_Π un ulteriore algoritmo per Π di complessità $T_2(n) = n^{3/2}$. Si determini la complessità in tempo del migliore algoritmo ibrido per Π ottenibile combinando A^1_Π e A^2_Π.

Answer: Let $i \in \mathcal{I}_\Pi$ be an instance of Π, with size$(i) = n$, and let n_0 be a constant parameter (power of two) to be determined. In order to obtain a hybrid algorithm by combining A^1_Π and A^2_Π, we change the base case in the code of A^1_Π, as follows.

HYBRID(i)
$n \leftarrow$ size(i)
if $n \leq n_0$
  then return $A^2_\Pi(i)$
...
{ same code as in A^1_Π}
...

The recurrence associated with the new algorithm is parametric in both n and n_0:

$$T_{n_0}(n) = \begin{cases} n^{3/2} & n \leq n_0, \\ 2T_{n_0}\left(\frac{n}{2}\right) + 2n & n > n_0. \end{cases}$$

Let $n \geq n_0$. By using the iterated method, we obtain:

$$T_{n_0}(n) = 2^i T_{n_0}\left(\frac{n}{2^i}\right) + i \cdot 2n, \quad 1 \leq i \leq \log(n/n_0).$$

By substituting $i = \log(n/n_0)$ in the above formula, we obtain:

$$T_{n_0}(n) = \left(\frac{n}{n_o}\right) n_0^{3/2} + \log(n/n_0) \cdot 2n = 2n \log n + n(\sqrt{n_0} - 2 \log n_0)$$

(note that $T_{n_0}(n_0) = n_0^{3/2}$). In order to obtain the optimal choice for n_0, it suffices to compute the partial derivative of $T(n, n_0)$ with respect to n_0 and equate it to zero:

$$\frac{\partial T_{n_0}(n)}{\partial n_0} = n \left(\frac{1}{2\sqrt{n_0}} - \frac{2}{n_0 \ln 2} \right) = 0 \Leftrightarrow \sqrt{n_0} = \frac{4}{\ln 2} \Leftrightarrow n_0 = \frac{16}{\ln^2 2} \simeq 33.3.$$

Therefore, the best choice is either $\bar{n}_0 = 32$ or $\bar{n}_0 = 64$. Since $\sqrt{32} - 2 \log 32 < \sqrt{64} - 2 \log 64$, we choose $\bar{n}_0 = 32$, which yields the best hybrid algorithm. $\square$