Lucido 0: convenzioni.

Dott. Ferruccio Guidi

http://www.cs.unibo.it/~fguidi

Esercizi su O, Ω, Θ.

Notazioni.

Qui $\equiv$ indica "uguale per definizione".

$$\mathtt{R}_+ \equiv \{x \in \mathtt{R} \mid x \geq 0\};$$

$$\mathtt{R}_+^0 \equiv \{x \in \mathtt{R} \mid x > 0\};$$

$$(f + g)(x) \equiv f(x) + g(x);$$

$$(fg)(x) \equiv f(x)g(x);$$

$$(f^g)(x) \equiv f(x)^{g(x)}.$$

Quindi f^{-1} è $(1/f)$ e non $f^{\leftarrow}$.

Convenzioni.

$$f, g, h, k \in \mathtt{N} \to \mathtt{R}_+;$$

$$m, n \in \mathtt{N};$$

$$c, d, e \in \mathtt{R}_+^0.$$

Lucido 1: notazione asintotica O.

Definizione di O.

$$f \in O(g) \equiv (\exists c, m)(\forall n \geq m)[f(n) \leq cg(n)].$$

Proprietà generali di O.

$$h \in O(h);$$

se $f \in O(g)$ *e se* $g \in O(h)$ *allora* $f \in O(h)$;

se $f, g \in O(h)$ *allora* $(f + g) \in O(h)$;

se $f \in O(h)$ *allora* $(df) \in O(h)$;

se $f \in O(h)$ *e se* $g \in O(k)$ *allora* $(fg) \in O(hk)$;

se $f \in O(h)$ *allora* $(f^d) \in O(h^d)$.

Se $f \in O(h)$ *allora* $(f/g) \in O(h/g)$;

Proprietà particolari di O.

Qui $1 < b$, $0 < c < d$.

$$\log_b n \in O(n^c);\ n^c \in O(n^d);\ n^d \in O(b^n).$$

mentre NON valgono le trasposte:

$$b^n \in O(n^d);\ n^d \in O(n^c);\ n^c \in O(\log_b n).$$

Lucido 2: esercizi su O [1-4].

Esercizio 1.

$$(n + \log n)(2n + 4 \log n) \in O(n^2)?$$

Entrambi i fattori appartengono a $O(n)$ quindi il prodotto appartiene a $O(n^2)$.

Esercizio 2.

$$3n \log n + n \in O(2^n)?$$

$3n \log n + n \in O(n^2)$ perché $\log n \in O(n)$ e $O(n^2)$ è contenuto in $O(2^n)$.

Esercizio 3.

$$5n^2 \log n + n \in O(n^{\log n})?$$

$5n^2 \log n + n$ sta in $O(n^3) \subseteq O(n^{\log n})$ perché $3 \leq \log n$ quando n è grande.

Esercizio 4.

$$3^{2n} \in O(3^n)?$$

No, perché $3^{2n} = 3^n 3^n$ e quindi, se fosse vero, si avrebbe $3^n \in O(1)$ (dividendo per 3^n) che è certamente falso.

L'asserto è falso pur essendo $2n \in O(n)$.

Lucido 3: notazione asintotica $\mathbf{\Omega}$.

Definizione di $\mathbf{\Omega}$.

$$f \in \Omega(g) \equiv (\exists c, m)(\forall n \geq m)[f(n) \geq cg(n)].$$

Proprietà generali di $\mathbf{\Omega}$.

Sono le stesse che valgono per O e inoltre:

$f \in \Omega(g)$ sse $g \in O(f)$.

$f \in \Omega(g)$ sse $(1/f) \in O(1/g)$.

Transitiva mista: se $f \in \Omega(h)$ e $f \in O(k)$ allora $h \in O(k)$ oppure $k \in \Omega(h)$.

Se $f \in O(h)$ e $g \in \Omega(k)$ allora $(f/g) \in O(h/k)$ e $(g/f) \in \Omega(k/h)$.

Proprietà particolari di $\mathbf{\Omega}$.

Sono le trasposte di quelle che valgono per O mentre NON valgono per Ω quelle che valgono per O.

Lucido 4: esercizi su Ω [1-3].

Esercizio 1.

$$\frac{n^3+5n+1}{3n+2} \in O(n^3)?$$

Il numeratore (num) sta in $O(n^3)$ mentre il denominatore sta in $\Omega(1)$ quindi la frazione sta in $O(n^3)$.

Esercizio 2.

$$\frac{n^2+n+1}{\sqrt{n^3}+3} \in O(n)?$$

Il numeratore è in $O(n^2)$, $\sqrt{n^3} \equiv n^{3/2}$ quindi il den. è in $\Omega(n^{3/2})$, per cui la frazione è in $O(n^{1/2}) \subseteq O(n)$.

Esercizio 3.

$$\frac{\sqrt{n}\log_2 2^{(n^2)} + 5n^2\sqrt[4]{n} + n}{4n + \sqrt[3]{n^5} + 11} \in O(n)?$$

Num: $n^{2+1/2} + 5n^{2+1/4} + n$: è in $O(n^{5/2})$;
den: $4n + n^{5/3} + 11$: è in $\Omega(n^{5/3})$;
frazione: $O(n^{5/2-5/3}) = O(n^{5/6}) \subseteq O(n)$.

Qui abbiamo usato $\log_b b^a = a$.

Lucido 5: esercizi su $\mathbf{\Omega}$ [4-6].

Esercizio 4.

$$\frac{n^2 \log n + n\sqrt{n} + 5n + 1}{3n+2} \in \Omega(n \log n)?$$

Il numeratore sta in $\Omega(n^2 \log n)$,
il denominatore sta in $O(n)$,
quindi la frazione sta in $\Omega(n \log n)$.

Esercizio 5.

$$\frac{n(n+1)(n+2)}{2n + 5\sqrt{n} + 4} \in \Omega(n \log n)?$$

Il numeratore sta in $\Omega(n^3)$,
il denominatore sta in $O(n)$,
quindi la frazione sta in $\Omega(n^2) \subseteq \Omega(n \log n)$.

Esercizio 6.

$$\frac{4\sqrt{n}\log(2n^2) + 12n\sqrt[3]{n} + 20\sqrt{n}}{2\sqrt{5n} + 10n + 1} \in \Omega(\sqrt[3]{n})?$$

Il num. è in $\Omega(n\sqrt[3]{n})$, il den è in $O(n)$,
quindi la frazione sta in $\Omega(\sqrt[3]{n})$.
Si è usato $\log(2n^2) = 2\log(\sqrt{2}n)$.

Lucido 6: esercizi su $\boldsymbol{\Omega}$ [7-8].

Esercizio 7.

$$\frac{n^4 \log n + 17n\sqrt{n}}{(\sqrt{n}\log(n^6) + 3)n^2} \in \Omega(\sqrt{n})?$$

Semplificando la frazione per n
e ambo il lati per $\sqrt{n}$ si ha:

$$\frac{n^2\sqrt{n}\log n + 17}{(6\sqrt{n}\log n + 3)n} \in \Omega(1)$$

dove abbiamo usato $\log_b(x^a) = a\log_b x$.
Il numeratore sta in $\Omega(n^2\sqrt{n}\log n)$,
il denominatore sta in $O(n\sqrt{n}\log n)$,
per cui la frazione è in $\Omega(n) \subseteq \Omega(1)$.
Quindi l'asserto è vero.

Esercizio 8.

$$\frac{n}{\log n + 1} \in \Omega(\sqrt{n})?$$

Dividendo ambo i membri per n si ha:

$$(\log n + 1)^{-1} \in \Omega(n^{-1/2})$$

e passando ai reciproci:

$$\log n + 1 \in O(n^{1/2})$$

che certamente vale.

Lucido 7: esercizi su $\boldsymbol{\Omega}$ [9-10].

Esercizio 9.

$$\frac{n}{\sqrt{n}+\sqrt[3]{n}} \in O(\sqrt[3]{n})?$$

Il num. è in $\Omega(n)$ e il den. è in $O(\sqrt{n})$
quindi la frazione è in $\Omega(\sqrt{n})$
allora applicando la transitiva mista si ha:

$$\sqrt{n} \in O(\sqrt[3]{n})$$

che è falso quindi l'asserto non vale.

Esercizio 10.

$$\frac{5n+4}{\log n+1} \in O(\sqrt{n})?$$

Il num. è in $\Omega(n)$ e il den. è in $O(\log n)$
quindi la frazione è in $\Omega(n/(\log n))$
allora applicando la transitiva mista si ha:

$$\frac{n}{\log n} \in O(\sqrt{n})$$

e moltiplicando per $(\log n)/\sqrt{n}$:

$$\sqrt{n} \in O(\log n)$$

che è falso quindi l'asserto non vale.

Lucido 8: notazione asintotica Θ.

Definizione di Θ.

$$f \in \Theta(g) \equiv (\exists c_1, c_2, m)(\forall n \geq m)$$
$$[c_1 g(n) \leq f(n) \leq c_2 g(n)].$$

Proprietà generali di Θ.

Sono le stesse che valgono per O e inoltre:

$f \in \Theta(g)$ *sse* $f \in O(g)$ *e* $f \in \Omega(g)$.

$f \in \Theta(g)$ *sse* $g \in \Theta(f)$.

$f \in \Theta(g)$ *sse* $(1/f) \in \Theta(1/g)$.

Transitiva mista: se $f \in \Theta(h)$ *e* $f \in \Theta(k)$ *allora* $h \in \Theta(k)$ *oppure* $k \in \Theta(h)$.

Se $f \in \Theta(h)$ *e* $g \in \Theta(k)$ *allora* $(f/g) \in \Theta(h/k)$ *e* $(g/f) \in \Theta(k/h)$.

Proprietà particolari di Θ.

Per Θ NON valgono
nè quelle di O, nè quelle di Ω.

Lucido 9: esercizi su $\boldsymbol{\Theta}$ [1-2].

Esercizio 1.

$$(n + \log n)(13 + \log n^{10}) \in \Theta(n \log n)?$$

Il primo fattore sta in $\Theta(n)$
mentre il secondo fattore sta in $\Theta(\log n)$
perché $\log(n^{10}) = 10 \log n$
quindi il prodotto sta in $\Theta(n \log n)$.

Esercizio 2.

$$\frac{n \log \sqrt{n} + 3n^2 + 2n + 12344}{2n + 10^5\sqrt{n} + 1} \in \Theta(n)?$$

Il numeratore sta in $\Theta(n^2)$
mentre il denominatore sta in $\Theta(n)$
quindi la frazione sta in $\Theta(n)$.

Si noti che $\log \sqrt{n} = \log(n^{1/2}) = \dfrac{1}{2} \log n$.

Lucido 10: ricorrenza.

Dott. Ferruccio Guidi

`http://www.cs.unibo.it/~fguidi`

Esercizi sulla complessità delle funzioni ricorrenti.

Metodi di soluzione.

Per sostituzione: *si formula un'ipotesi di soluzione che poi si verifica induttivamente.*

Per iterazione: *si svolge la ricorrenza allo scopo di dedurre una forma iterativa equivalente.*

Mediante il "master theorem": *esiste una classe di funzioni ricorrenti la cui complessità si calcola in base ad un teorema generale.*

Qui vedremo in dettaglio i primi due metodi, mentre il terzo verrà affrontato in seguito.

Lucido 11: ricorrenza [1].

Esercizio 1.

Dire se la seguente funzione sta in $\Omega(n \log n)$:

$$f(n) = \begin{cases} 1 & \textit{per } n = 0 \\ f(n-1) + 4n & \textit{per } n > 0 \end{cases}$$

Soluzione per sostituzione.

Osservando che $f(n)$ è somma di n fattori che stanno in $\Omega(n)$, si ipotizza $f \in \Omega(n^2)$;

allora bisogna verificare per induzione che esiste una costante c per cui $f(n) \geq cn^2$ quando n è grande.

Svolgendo f e usando l'ipotesi induttiva si ha:

$$f(n) = f(n-1) + 4n \geq c(n-1)^2 + 4n =$$

$$= cn^2 + 2n(2-c) + 1 \geq cn^2$$

quando $2 - c \geq 0$ cioè per $c \leq 2$;

dunque $f \in \Omega(n^2) \subseteq \Omega(n \log n)$.

Lucido 12: ricorrenza [1].

Soluzione per iterazione.

Sviluppando la ricorrenza si ha:

$$\begin{aligned} f(n) &= f(n-1) + 4n \\ f(n-1) &= f(n-2) + 4(n-1) \\ f(n-2) &= f(n-3) + 4(n-2) \\ \cdots &= \cdots \\ f(1) &= f(0) + 4 \cdot 1 \\ f(0) &= 1 + 4 \cdot 0 \end{aligned}$$

Dunque sommando i due lati, si ha:

$$f(n) = 1 + \sum_{i=0}^{n} 4i = 1 + 4 \sum_{i=0}^{n} i$$

e maggiorando ogni i con n, si trova

$f(n) \leq 1 + 4n^2$ da cui $f \in O(n^2)$.

Minorando $i < \frac{n}{2}$ con 0 e $i \geq \frac{n}{2}$ con $\frac{n}{2}$, si ha:

$f(n) \geq 1 + 4\dfrac{n}{2}\dfrac{n}{2} = 1 + n^2$ da cui $f \in \Omega(n^2)$.

Quindi si procede come sopra.

Verificare che $f(n) = 1 + 2n(n+1)$.

Lucido 13: ricorrenza [2].

Esercizio 2.

Calcolare la complessità computazionale di:

```
int B(int n)
{
   if (n == 0)
      return 1;
   else
      return 3 * B(n - 1) * B(n - 1)   \
               +2 * B(n - 1)/B(n - 1);
}
```

Il primo ramo dell'`if` ha complessità costante e il secondo contiene quattro valutazioni di $B(n-1)$, quindi la complessità è descritta da:

$$f(n) = \begin{cases} 1 & \text{per } n = 0 \\ 4f(n-1) + 1 & \text{per } n > 0 \end{cases}$$

Si noti l'importanza di sommare le costanti nei due rami della definizione di f; senza le costanti si avrebbe $f(n) = 0$.

Lucido 14: ricorrenza [2].

Soluzione per sostituzione.

Siccome $f(n)$ è somma di n fattori in $\Theta(1)$
in cui è f è via via moltiplicata per 4,
si ipotizza $f \in \Theta(4^n)$.

Allora cerchiamo tre costanti c, d, e tali che

$$4^n c \leq f(n) \leq 4^n d - e$$

Svolgendo f e usando l'ipotesi induttiva si ha:

$$f(n) = 4f(n-1) + 1 \geq 4(4^{n-1}c) + 1 =$$
$$= 4^n c + 1 \geq 4^n c \text{ per ogni } c > 0;$$

e per l'altra disuguaglianza si ha:

$$f(n) = 4f(n-1) + 1 \leq 4(4^{n-1}d - e) + 1 =$$
$$= 4^n d - 4e + 1 \leq 4^n d - e$$

quando $d > 0$ e $-3e + 1 \leq 0$ cioè $e \geq \frac{1}{3}$;
con ciò abbiamo $f(n) \in O(4^n d - e) \subseteq O(4^n)$.

dunque si può affermare $f \in \Theta(4^n)$.

Lucido 15: ricorrenza [2].

Soluzione per iterazione.

Sviluppando la ricorrenza si ha:

$$\begin{aligned} f(n) &= 4f(n-1)+1 \\ 4f(n-1) &= 4^2 f(n-2)+4 \\ 4^2 f(n-2) &= 4^3 f(n-3)+4^2 \\ \cdots &= \cdots \\ 4^{n-1} f(1) &= 4^n f(0) + 4^{n-1} \\ 4^n f(0) &= 1 \cdot 4^n \end{aligned}$$

e sommando i due lati, si ha: $f(n) = \sum_{i=0}^{n} 4^i$,

per cui $f(n) \geq 4^n$ e quindi $f \in \Omega(4^n)$.

Sfruttando invece $4^i + 4^i \leq 4^{i+1}$ si trova:

$$\begin{aligned} f(n) = {} & 1 + 4 + 4^2 + \ldots + 4^{n-1} + 4^n \leq \\ & \leq 4 + 4 + 4^2 + \ldots + 4^{n-1} + 4^n \leq \\ & \leq 4^2 + 4^2 + \ldots + 4^{n-1} + 4^n \leq \\ & \leq 4^3 + \ldots + 4^{n-1} + 4^n \leq \ldots \leq \\ & \leq 4^{n-1} + 4^{n-1} + 4^n \leq 4^n + 4^n = 2 \cdot 4^n \end{aligned}$$

per cui vale anche $f \in O(4^n)$.

Verificare che $f(n) = \frac{4^{n+1}-1}{3}$.

Lucido 16: ricorrenza [2].

Consideriamo la seguente variante di B:

```
int B(int n)
{
  int m;
  if (n == 0)
    return 1;
  else
  {
    m = B(n - 1);
    return 3 * m * m + 2 * m/m;
  }
}
```

Qui il ramo `else` contiene una sola valutazione di $B(n-1)$, quindi la complessità è descritta da:

$$f(n) = \begin{cases} 1 & \text{per } n = 0 \\ f(n-1) + 1 & \text{per } n > 0 \end{cases}$$

che risulta in $\Theta(n)$ come si vede direttamente con calcoli analoghi a prima, o dimostrando:

$$f(n) = 1 + n.$$

Lucido 17: ricorrenza [3].

Esercizio 3.

Scrivere una funzione ricorrente che calcola:

$$g(n) = \begin{cases} 1 \ se \ n = 0 \\ 3 \ se \ n = 1 \ o \ n = 2 \\ 2 * g(n-1) * g(n-2) * g(n-3) \end{cases}$$

e valutarne la complessità computazionale.

Per prima cosa scriviamo il codice che implementa g usando una catena di `else if`

```
int G(int n)
{
if (n == 0)
  return 1;
else if (n == 1 || n == 2)
  return 3;
else
  return 2 * G(n - 1) * G(n - 2) * G(n - 3);
}
```

Il primi due rami dell'`if` hanno complessità costante e il terzo ha tre valutazioni di G.

Lucido 18: ricorrenza [3].

Allora la complessità di G è descritta da:

$$f(n) = \begin{cases} 1 \text{ per } n \leq 2 \\ f(n-1) + f(n-2) + f(n-3) + 1 \end{cases}$$

f è non decrescente quindi $f_1 \leq f \leq f_2$ con:

$$f_1(n) = \begin{cases} 1 & \text{per } n \leq 2 \\ 3f_1(n-3) + 1 & \text{per } n > 2 \end{cases}$$

$$f_2(n) = \begin{cases} 1 & \text{per } n \leq 2 \\ 3f_2(n-1) + 1 & \text{per } n > 2 \end{cases}$$

Ora si vede con calcoli simili ai precedenti che $f_2 \in \Theta(3^n)$ e quindi, essendo $f \in O(f_2)$, si trova $f \in O(3^n)$.

Studiamo la complessità di f_1 che è $\Theta(3^{\frac{n}{3}})$ quindi, essendo $f \in \Omega(f_1)$, si trova $f \in \Omega(3^{\frac{n}{3}})$.

Lo studio è analogo a quello dell'esercizio 2 con entrambi i metodi risolutivi. Però lo ripresentiamo per via dell'esponente $\frac{n}{3}$.

Lucido 19: ricorrenza [3].

Soluzione per sostituzione.

Siccome $f_1(n)$ è somma di $\frac{n}{3}$ fattori in $\Theta(1)$ in cui è f_1 è via via moltiplicata per 3, si ipotizza $f_1 \in \Theta(3^{\frac{n}{3}})$.

Al solito cerchiamo tre costanti c, d, e tali che

$$3^{\frac{n}{3}}c \leq f_1(n) \leq 3^{\frac{n}{3}}d - e$$

Svolgendo f_1 e usando l'ipotesi induttiva si ha:

$$f_1(n) = 3f_1(n-3) + 1 \geq 3(3^{\frac{n-3}{3}}c) + 1 =$$
$$= 3^{\frac{n}{3}}c + 1 \geq 3^{\frac{n}{3}}c \text{ per ogni } c > 0;$$

e per l'altra disuguaglianza si ha:

$$f_1(n) = 3f(n-3) + 1 \leq 3(3^{\frac{n-3}{3}}d - e) + 1 =$$
$$= 3^{\frac{n}{3}}d - 3e + 1 \leq 3^{\frac{n}{3}}d - e$$

quando $d > 0$ e $-2e + 1 \leq 0$ cioè $e \geq \frac{1}{2}$; con ciò abbiamo $f_1(n) \in O(3^{\frac{n}{3}}d - e) \subseteq O(3^{\frac{n}{3}})$.

Dunque si può affermare $f_1 \in \Theta(3^{\frac{n}{3}})$.

Lucido 20: ricorrenza [3].

Soluzione per iterazione.

Sviluppando la ricorrenza
e imponendo $n - 3(i+1) = 0$ si ha:

$$\begin{aligned}
f_1(n) &= 3f_1(n-3) + 1 \\
3f_1(n-3) &= 3^2 f_1(n-3\cdot 2) + 3 \\
3^2 f_1(n-3\cdot 2) &= 3^3 f_1(n-3\cdot 3) + 3^2 \\
\cdots &= \cdots \\
3^i f(n-3i) &= 3^{i+1} f_1(n-3(i+1)) + 3^i \\
\cdots &= \cdots \\
3^{\frac{n}{3}-1} f_1(1) &= 3^{\frac{n}{3}} f_1(0) + 3^{\frac{n}{3}-1} \\
3^{\frac{n}{3}} f_1(0) &= 1 \cdot 3^{\frac{n}{3}}
\end{aligned}$$

e sommando i due lati, si ha: $f(n) = \sum_{i=0}^{n/3} 3^i$,
per cui $f_1(n) \geq 3^{\frac{n}{3}}$ e quindi $f_1 \in \Omega(3^{\frac{n}{3}})$.
Sfruttando invece $3^i + 3^i \leq 3^{i+1}$ si trova,
come per l'esercizio precedente:

$$\sum_{i=0}^{\frac{n}{3}-1} 3^i \leq 3^{\frac{n}{3}}$$

per cui $f_1(n) \leq 3^{\frac{n}{3}} + 3^{\frac{n}{3}} = 2 \cdot 3^{\frac{n}{3}}$
ovvero $f_1 \in O(3^{\frac{n}{3}})$.

Lucido 21: ricorrenza [4].

Esercizio 4.

Calcolare la complessità della funzione:

$$f(n) = \begin{cases} 1 & per\ n \leq 1 \\ 2f(n/2) + n & per\ n > 1 \end{cases}$$

<u>Soluzione per iterazione.</u>

Sviluppando la ricorrenza
e imponendo $\frac{n}{2^{i+1}} = 1$ si ha:

$$\begin{aligned} f(n) &= 2f(n/2) + n \\ 2f(n/2) &= 2^2 f(n/2^2) + n \\ 2^2 f(n/2^2) &= 2^3 f(n/2^3) + n \\ \cdots &= \cdots \\ 2^i f(n/2^i) &= 2^{i+1} f(n/2^{i+1}) + n \\ \cdots &= \cdots \\ \frac{n}{2} f(2) &= n f(1) + n \\ n f(1) &= n \end{aligned}$$

Dunque sommando i due lati, si ha:

$$f(n) = \sum_{i=0}^{\log n} n = n(\log n + 1) \in \Theta(n \log n).$$

Lucido 22: ricorrenza [4].

Soluzione per sostituzione.

Vogliamo verificare che $f \in \Theta(n \log n)$;
allora cerchiamo due costanti c, d tali che

$$cn \log n \leq f(n) \leq dn \log n$$

Svolgendo f e usando l'ipotesi induttiva si ha:

$$\begin{aligned} f(n) &= 2f(\tfrac{n}{2}) + n \leq 2d\tfrac{n}{2} \log \tfrac{n}{2} + n = \\ &= dn \log \tfrac{n}{2} + n = dn \log n - dn + n \\ &\leq dn \log n \end{aligned}$$

quando $n(1-d) \leq 0$ cioè per $d \geq 1$

e per l'altra disuguaglianza si ha:

$$\begin{aligned} f(n) &= 2f(\tfrac{n}{2}) + n \geq 2c\tfrac{n}{2} \log \tfrac{n}{2} + n = \\ &= cn \log \tfrac{n}{2} + n = cn \log n - cn + n \\ &\geq cn \log n \end{aligned}$$

quando $n(1-c) \geq 0$ cioè per $c \leq 1$;

con ciò abbiamo appunto $f(n) \in \Theta(n \log n)$.

Lucido 23: ricorrenza [5].

Esercizio 5.

Calcolare la complessità della funzione:

$$f(n) = \begin{cases} 1 & \textit{per } n \leq 1 \\ 2f(\sqrt{n}) + \log n & \textit{per } n > 1 \end{cases}$$

<u>Soluzione per iterazione.</u>

Ponendo $g(n) = f(2^n)$

si ha $g(\frac{n}{2}) = f(2^{\frac{n}{2}}) = f(\sqrt{2^n})$ per cui

$$g(n) = \begin{cases} 1 & \text{per } n = 0 \\ 2g(n/2) + n & \text{per } n > 0 \end{cases}$$

che differisce da quella dell'esercizio precedente perché $g(1) = 3$; ne segue che:

$$g(n) = 3n + \sum_{i=0}^{\log n - 1} n = 2n + (\log n + 1)n =$$

$$= (\log n + 3)n \in \Theta(n \log n)$$

per cui, essendo $f(n) = g(\log n)$, si trova:

$$f(n) \in \Theta(\log n \log \log n).$$

Lucido 24: funzioni non ricorrenti [1-2].

Dott. Ferruccio Guidi

http://www.cs.unibo.it/~fguidi

Esercizi di ricapitolazione.

Esercizio 1.

$$3^{\log_9 n} \in \Omega(\sqrt{n})?$$

Si tratta di semplificare $3^{\log_9 n}$ che è:
3^x con $x = \log_9 n$
per cui $n = 9^x = (3^2)^x = 3^{2x} = (3^x)^2$
che dà $3^x = \sqrt{n}$ per cui la risposta è sì.

Esercizio 2.

$$\frac{n^3 + 12n + 11}{3\sqrt{n} + 2} \in \Omega(n^2)?$$

Il numeratore sta in $\Theta(n^3)$;
il denominatore sta in $\Theta(n^{1/2})$;
la frazione è in $\Theta(n^{5/2}) \subseteq \Omega(n^{5/2}) \subseteq \Omega(n^2)$.

Qui si è usata la transitiva normale
e non quella mista.

Lucido 25: funzioni non ricorrenti [3-4].

Esercizio 3.

$$\frac{7n\sqrt{n}\log n + 4n^2}{8\sqrt{n} + 5\log_3(2^n)} \in O(n)?$$

Il numeratore sta in $\Theta(n^2)$;
il denominatore sta in $\Theta(n)$
perchè $\log_3(2^n) = n\log_3 2$;
la frazione è in $\Theta(n) \subseteq O(n)$.

Esercizio 4.

Scrivere in C una procedura Primi che stampa i divisori primi di un intero in input senza ripetizioni, e valutarne la complessità.

Il listato dela procedura è nel lucido 26;
la semantica è di provare a dividere l'intero n per tutti gli interi fra 2 e n (variabile i) e di stampare il primo per cui il resto della divisione è nullo. A questo punto n viene sostituito col quoziente della divisione e si continua.
La variabile t tiene l'ultimo fattore primo trovato e serve per controllare (ed evitare) la stampa dei fattori ripetuti.

Lucido 26: funzioni non ricorrenti [4].

Listato relativo all'esercizio 4:

```
int Primi(int n)
{
   int i,t;
   i = 2;  t = 1;
   while (i <= n)
   {
      if (n%i == 0)
      {
         if (t != i) printf("%u", i);
         n = n/i;  t = i;
      }
      else i = i + 1;
   }
   printf("\n");
   return 0;
}
```

Passando da n ad n/i il test di divisibilità non riparte da 2 perchè nessun intero minore di i è fattore di n e quindi neanche di n/i.

Lucido 27: funzioni non ricorrenti [4].

Per studiare la complessità si osserva che il corpo del *while* è eseguito in tempo costante e quindi e quindi ci interessa stimare il numero di cicli che vengono eseguiti per i vari n.
Nel caso pessimo, quando n è primo, occorre provare tutti gli i fra 2 e n per cui la complessità è $\Theta(n)$;
nel caso ottimo però, quando $n = 2^k$, i cicli sono solo $k = \log n$ quindi la complessità è $\Theta(\log n)$.
Per calcolare un limite superiore al numero di cicli si considera la funzione di stato $s(i, n) = n - i$ che all'inizio dell'esecuzione vale $n - 2$ mentre alla fine dell'esecuzione vale -1 (perchè $n/i \geq i$); ora si deve notare che s decresce ad ogni ciclo (perchè aumenta i o perché diminuisce n) e quindi i cicli non possono essere più di n:
più precisamente $(n - 2) - (-1) = n - 1$;
Quindi la complessità è $O(n)$.
La funzione s, essendo una funzione di stato decrescente e inferiormente limitata, garantisce la terminazione della procedura.

Lucido 28: funzioni non ricorrenti [5].

Esercizio 5.

Si considerino le seguenti funzioni in C:

```
int F(int n)
{
    int i, z;
    z = 1;
    For (i = 1; i <= n; i = i + 1)
        z = 2 * z;
    return z;
}

int A(int n)
{
    int i, z;
    z = 1;
    For (i = 1; i <= F(n); i = i + 1)
        z = 2 + F(n);
    return z;
}
```

e si calcoli la complessità di A.

Lucido 29: funzioni non ricorrenti [5].

Occupiamoci prima della funzione $F(n)$: essa calcola 2^n mediante un ciclo for che itera n volte un corpo eseguito in tempo costante, quindi $F \in \Theta(n)$.

Vediamo ora la funzione $A(n)$: essa contiene un ciclo for che itera $F(n) = 2^n$ volte un corpo che contiene due istanze di $F(n)$ (una nell'assegnamento a z, l'altra nella condizione di iterazione che viene valutata ad ogni ciclo); il corpo ha quindi complessità $\Theta(n)$ per cui $A \in \Theta(n \cdot 2^n)$.

Per togliere il fattore n si usa la variante:

```
int A(int n)
{
    int i, j, z;
    j = F(n);  z = 1;
    For (i = 1;  i <= j;  i = i + 1)
        z = 2 + j;
    return z;
}
```

per cui si ha $A \in \Theta(n + 1 \cdot 2^n) = \Theta(2^n)$.

Lucido 30: funzioni ricorrenti [1].

Esercizio 1.

Calcolare la complessità della funzione:

$$f(n) = \begin{cases} 1 \; per \; n = 0 \\ 2 \; per \; n = 1 \\ 3f(n-1) + 5f(n-2) + 27 \end{cases}$$

Notando che f è crescente si ha $f(n-2) \leq f(n-1)$ e quindi $f_1(n) \leq f(n) \leq f_2(n)$ dove:

$$f_1(n) = \begin{cases} 1 & \text{per } n \leq 1 \\ 8f(n-2) + 27 & \text{altrimentri} \end{cases}$$

$$f_2(n) = \begin{cases} 2 & \text{per } n \leq 1 \\ 8f(n-1) + 27 & \text{altrimentri} \end{cases}$$

Svolgiamo per prima la ricorrenza f_1:

$$\begin{aligned} f_1(n) &= 8f_1(n-2) + 27 \\ 8f_1(n-2) &= 8^2 f_1(n - 2 \cdot 2) + 8 \cdot 27 \\ 8^2 f_1(n - 2 \cdot 2) &= 8^3 f_1(n - 2 \cdot 3) + 8^2 \cdot 27 \\ 8^i f(n-2i) &= 8^{i+1} f_1(n - 2(i+1)) + 8^i \cdot 27 \\ 8^{\frac{n-2}{2}} f_1(2) &= 8^{\frac{n}{2}} f_1(0) + 8^{\frac{n-2}{2}} \cdot 27 \\ 8^{\frac{n}{2}} f_1(0) &= 1 \cdot 8^{\frac{n}{2}} \end{aligned}$$

Lucido 31: funzioni ricorrenti [1].

Si noti che la condizione di arresto è data da:

$$n - 2(i+1) = 0 \text{ ovvero } i = \frac{n-2}{2} = \frac{n}{2} - 1$$

quando n è pari (caso considerato sopra), mentre per n dispari la condizione è data da:

$$n - 2(i+1) = 1 \text{ ovvero } i = \frac{n-3}{2}$$

Sommando le equazioni precedenti si ottiene:

$$f_1(n) = 8^{\frac{n}{2}} + \sum_{i=0}^{\frac{n}{2}-1} 27 \cdot 8^i$$

L'esponente maggiore è $\frac{n}{2}$ per cui $f_1 \in \Theta(8^{\frac{n}{2}})$.

Svolgiamo ora la ricorrenza f_2:

$$\begin{aligned}
f_2(n) &= 8f_2(n-1) + 27 \\
8f_2(n-1) &= 8^2 f_2(n-2) + 8 \cdot 27 \\
8^2 f_2(n-2) &= 8^3 f_2(n-3) + 8^2 \cdot 27 \\
8^i f(n-i) &= 8^{i+1} f_2(n-(i+1)) + 8^i \cdot 27 \\
8^{n-2} f_2(2) &= 8^{n-1} f_2(1) + 8^{n-2} \cdot 27 \\
8^{n-1} f_2(1) &= 2 \cdot 8^{n-1}
\end{aligned}$$

Lucido 32: funzioni ricorrenti [1-2].

Qui la condizione di arresto è data da:

$n - (i + 1) = 1$ ovvero $i = n - 2$

quando $n \geq 2$ che è il caso considerato sopra. Sommando le equazioni precedenti si ottiene:

$$f_1(n) = 2 \cdot 8^{n-1} + \sum_{i=0}^{n-2} 27 \cdot 8^i$$

L'esponente maggiore è $n - 1$ per cui $f_1 \in \Theta(8^{n-1}) = \Theta(8^n \cdot 8^{-1}) = \Theta(8^n)$.

Sfruttando ora: $f_1(n) \leq f(n) \leq f_2(n)$

possiamo concludere $f \in \Omega(8^{\frac{n}{2}}) \cap O(8^n)$.

Esercizio 2.

calcolare la complessità della funzione:

$$f(n) = \begin{cases} 1 & \textit{per } n \leq 1 \\ 2f(n/2) + n \log n & \textit{per } n > 1 \end{cases}$$

<u>Test per il Master Theorem.</u>

nel caso ricorrente $f(n)$ ha la forma $af(n/b) + g(n)$ con $a = b = 2$, $g(n) = n \log n$; inoltre si ha: $n^{\log_b a} = n^{\log_2 2} = n$.

Lucido 33: funzioni ricorrenti [2].

però per $\epsilon > 0$, $n \log n \in O(n \cdot n^{-\epsilon})$, $n \log n \in \Theta(n)$ e $n \log n \in \Omega(n \cdot n^{\epsilon})$ sono falsi per cui il Master Theorem non si applica.

<u>Soluzione per iterazione.</u>

Svolgendo la ricorrenza e imponendo $\frac{n}{2^{i+1}} = 1$ ovvero $2^i = n/2$, si ha:

$$\begin{aligned}
f(n) &= 2f(n/2) + n \log n \\
2f(n/2) &= 2^2 f(n/2^2) + n \log \tfrac{n}{2} \\
2^2 f(n/2^2) &= 2^3 f(n/2^3) + n \log \tfrac{n}{2^2} \\
2^i f(n/2^i) &= 2^{i+1} f(n/2^{i+1}) + n \log \tfrac{n}{2^i} \\
\tfrac{n}{2} f(2) &= n f(1) + n \log 2 \\
n f(1) &= n \cdot 1
\end{aligned}$$

Sommando ora le uguaglianze si ha:

$$f(n) = n + \sum_{i=0}^{\log \frac{n}{2}} n \log \frac{n}{2^i} = n + n \sum_{i=0}^{\log n - 1} (\log n - i)$$

$$= n + n \log^2 n - \frac{1}{2}(\log n - 1) \log n$$

per cui $f \in \Theta(n \log^2 n)$.

Lucido 34: funzioni ricorrenti [3].

Esercizio 3.

Calcolare la complessità della funzione:

$$
\begin{array}{l}
\texttt{int}\ A(\texttt{int}\ n)\\
\{\\
\quad \texttt{int}\ i, z;\\
\quad z = 1;\\
\quad \texttt{For}\ (i = 1;\ i <= n;\ i = i + 1)\\
\qquad z = 100 * A(n/2) + z;\\
\quad \texttt{return}\ z;\\
\}
\end{array}
$$

La funzione è composta da un ciclo for che itera n volte un corpo in cui compare la chiamata ricorrente, allora la funzione che descrive la complessità di A è:

$$f(n) = \begin{cases} 1 & \text{se } n \leq 0 \\ nf(n/2) + n & \text{se } n > 0 \end{cases}$$

dove manca l'addizionale termine costante nel ramo ricorrente perchè di ordine inferiore al termine n che rappresenta la somma dei tempi costanti spesi all'interno del for.

Lucido 35: funzioni ricorrenti [3].

Per calcolare la complessità di f svolgiamo la ricorrenza imponendo $\frac{n}{2^{i+1}} = 1$ ovvero $2^i = \frac{n}{2}$:

$$f(n) = n[f(n/2) + 1]$$
$$nf(n/2) = (n^2/2)[f(n/2^2) + 1]$$
$$(n^2/2)f(n/2^2) = (n^3/2^3)[f(n/2^3) + 1]$$
$$(n^i/2^{a_i})f(n/2^i) = (n^{i+1}/2^{a_{i+1}})[f(n/2^{i+1}) + 1]$$
$$(n^{\log \frac{n}{2}}/2^{a_{\log \frac{n}{2}}})f(2) = (n^{\log n}/2^{a_{\log n}})[f(1) + 1]$$
$$(n^{\log n}/2^{a_{\log n}})f(1) = 2(n^{\log n}/2^{a_{\log n}})$$

Per $a_i = \frac{i(i-1)}{2}$. Sommando le uguaglianze:

$$f(n) = 2(n^{\log n}/2^{a_{\log n}}) + \sum_{i=1}^{\log n}(n^i/2^{a_i})$$

e la somma di potenze ha l'ordine della potenza maggiore, dunque in questo caso:

$$f \in \Theta(n^{\log n}/2^{a_{\log n}}) = \Theta(n^{\log n}/n^{\frac{\log(n/2)}{2}})$$
$$\Theta(n^{\log n - \frac{\log(n/2)}{2}}) = \Theta(n^{\frac{\log 2n}{2}})$$

Si noti che:

$\log \frac{n}{2} + 1 = \log \frac{n}{2} + \log 2 = \log(\frac{n}{2} \cdot 2) = \log n.$