Prima Provetta
ASD1 2002-2003

Esercizio 1. *Dimostrare che* $f(n) = o(g(n))$ *se e solo* $g(n) = \omega(f(n))$.

comprensione

Scriviamo le definizioni di $o(g(n))$ e di $\omega(f(n))$:

$$\begin{aligned} o(g(n)) &= \{f(n) \mid \forall k > 0\, \exists n_0 \mid \forall n \geq n_0, 0 \leq f(n) \leq kg(n)\} \\ \omega(f(n)) &= \{g(n) \mid \forall h > 0\, \exists n_1 \mid \forall n \geq n_1, g(n) \geq hf(n) \geq 0\} \end{aligned}$$

Abbiamo già sistemato i nomi del generico elemento di $o(g(n))$ e di $\omega(f(n))$ in modo da attagliarsi al meglio al nostro problema; l'unica difficoltà che ci rimane è di mostrare che le due condizioni sono equivalenti. Ma ciò segue dall'arbitrarietà di k e di h, ad esempio come h nella seconda condizione possiamo prendere $1/k$, al che essa diventa formalmente uguale alla prima condizione.

soluzione formale

$$\begin{aligned} f(n) \in o(g(n)) &\Leftrightarrow \forall k > 0\, \exists n_0 \mid \forall n \geq n_0, 0 \leq f(n) \leq kg(n) \\ &\Leftrightarrow \forall \frac{1}{h} > 0\, \exists n_0 \mid \forall n \geq n_0, 0 \leq f(n) \leq \frac{1}{h} g(n) \\ &\Leftrightarrow \forall h > 0\, \exists n_0 \mid \forall n \geq n_0, 0 \leq hf(n) \leq g(n) \\ &\Leftrightarrow g(n) \in \omega(f(n)) \end{aligned}$$

Esercizio 2. *Si assuma* $f(n) = O(g(n))$. *Dimostrare che* $5f(n) + 2g(n) = \Theta(g(n))$.

comprensione

Essendo $f(n) \in O(g(n))$, sappiamo che $\exists k > 0, \exists \bar{n}_0$ tali che $\forall n \geq \bar{n}_0, 0 \leq f(n) \leq kg(n)$. Segue per le proprietà delle disequazioni che

$$5 \cdot 0 + 2g(n) \leq 5f(n) + 2g(n) \leq 5kg(n) + 2g(n).$$

Ricordiamo che

$$\Theta(g(n)) = \{h(n) \mid \exists \alpha > 0\, \exists \beta > 0\, \exists n_1 \mid \forall n \geq n_1, 0 \leq \alpha g(n) \leq f(n) \leq \beta g(n)\}$$

Ma allora possiamo prendere $\alpha = 2, \beta = 5k + 2, n_1 = n_0$ e l'appartenenza di $h(n) = 5f(n) + 2g(n)$ a $\Theta(g(n))$ è dimostrata. Si noti come, per la limitazione inferiore, abbiamo usato in maniera essenziale la non negatività asintotica di $f(n)$.

Abbiamo detto infatti che $f(n), g(n) \geq 0$ per ogni $n \geq n_0$. (D'altronde $f(n) = X(g(n))$, qualunque sia il simbolo X tra quelli da noi visti, implica che $f(n)$ e $g(n)$ sono definitivamente non negative).

soluzione formale

Essendo $f(n) \in O(g(n))$, sappiamo che $\exists k > 0, \exists \bar{n}_0$ tali che $\forall n \geq \bar{n}_0, 0 \leq f(n) \leq kg(n)$. In particolare, $f(n), g(n) \geq 0$ per ogni $n \geq n_0$.

Dimostriamo che $5f(n) + 2g(n) = O(g(n))$.
Dobbiamo pertanto mostrare che esistono costanti $c > 0$ ed n_0 tali che $5f(n) + 2g(n) \leq cg(n)$ per ogni $n \geq n_0$. Si consideri $c = 1$ ed $n_0 = \bar{n}_0$

$$5f(n) + 2g(n) \le 0 + 2g(n) \le g(n) = cg(n) \qquad \text{per ogni } n \ge n_0$$

Dimostriamo che $5f(n) + 2g(n) = \Omega(g(n))$.

Dobbiamo pertanto mostrare che esistono costanti $c > 0$ ed n_0 tali che $0 \le cg(n) \le 5f(n) + 2g(n)$ per ogni $n \ge n_0$. Si consideri $c = 5k + 2 > 0$ ed $n_0 = \bar{n}_0$

$$0 \le cg(n) = (5k+2)g(n) = 5kg(n) + 2g(n) \le 5f(n) + 2g(n) \qquad \text{per ogni } n \ge n_0$$

Esercizio 3. *Ordinare le seguenti funzioni per ordine di crescita asintotico non decrescente. Ve ne sono alcune che presentano lo stesso ordine di crescita?* $f(n) = 2^{\log_2 n}$, $f(n) = n!$, $f(n) = n^{\log n}$, $f(n) = 2^n$, $f(n) = 4^n$, $f(n) = 2^{n+1}$, $f(n) = 4^{\log_2 n}$, $f(n) = (n+1)! - n!$, $f(n) = n^{\frac{n}{100}}$.

comprensione

Riscriviamole intanto nei modi più convenienti per il confronto. Di quasi tutte diamo anche il logaritmo in base due, poichè per ordinare molte di esse è sufficiente confrontarne i logaritmi. Sia infatti $s_1(n) = 2^{t_1(n)}$ e $s_2(n) = 2^{t_2(n)}$. Abbiamo allora i fatti seguenti:

a) se $t_1(n) \in \Theta(t_2(n))$ (e, quindi, anche viceversa) in genere non si può dire nulla del confronto tra le funzioni originarie. Ma se, ad esempio, $t_1(n) - t_2(n) \to \infty$, allora $s_2(n) \in o(s_1(n))$.

b) supponiamo che si abbia $t_1(n) \ge \alpha, t_2(n) \ge \alpha$ per un certo $\alpha > 0$. Allora, se ad esempio, $t_2(n) \in o(t_1(n))$, segue $s_2(n) \in o(s_1(n))$.

Dimostriamo a): se $t_1(n) - t_2(n) \to \infty$, allora

$$\frac{s_1(n)}{s_2(n)} = 2^{t_1(n) - t_2(n)} \to \infty.$$

Per dimostrare b), notiamo che $t_2(n) \in o(t_1(n))$ è equivalente a dire

$$\lim_{n\to\infty} \frac{t_1(n)}{t_2(n)} = \infty$$

da cui anche

$$\lim_{n\to\infty} \frac{t_1(n) - t_2(n)}{t_2(n)} = \infty.$$

Dato che, definitivamente, $t_2(n) \ge \alpha$, abbiamo

$$\lim_{n\to\infty} t_1(n) - t_2(n) = \infty$$

da cui la tesi segue per il punto a).

Veniamo alle dieci funzioni. Per i fattoriali, sfruttiamo il fatto che $\log(k!) \in \Theta(k \log k)$.

$$\begin{aligned}
f_1(n) &= 2^{\log_2 n} = n = 2^{\Theta(\log n)} \\
f_2(n) &= n! = 2^{\Theta(n \log n))} \\
f_3(n) &= (\log_2 n)! = 2^{\Theta(\log n \log\log n)} \\
f_4(n) &= n^{\log_2 n} = (2^{\log_2 n})^{\log_2 n} = 2^{\Theta(\log^2 n)} \\
f_5(n) &= 2^n \\
f_6(n) &= 4^n = 2^{2n} \\
f_7(n) &= 2^{n+1} = 2 \cdot 2^n \\
f_8(n) &= 4^{\log_2 n} = (2^2)^{\log_2 n} = (2^{\log_2 n})^2 = n^2 = 2^{\Theta(\log n)} \\
f_9(n) &= (n+1)! - n! = [(n+1) - 1]n! = n \cdot n! = 2^{\Theta(n \log n)} \\
f_{10}(n) &= n^{n/100} = 2^{\frac{n \log_2 n}{100}}
\end{aligned}$$

Le due funzioni con logaritmo minore sono $f_1(n)$ e $f_8(n)$; chiaramente, $f_1(n) < f_8(n)$. Usiamo la notazione del confronto fra numeri seguendo l'ovvia analogia. Subito dopo vengono $f_3(n)$ e $f_4(n)$. Seguono le funzioni che hanno logaritmo lineare; vediamo che $f_5(n) = f_7(n) < f_6(n)$. Vi sono poi tre funzioni il coi logaritmo è $\Theta(n \log n)$. Se è chiaro che $f_2(n) < f_9(n)$, bisogna classificare $f_{10}(n)$. Mostriamo che essa cresce meno del fattoriale. Ricordiamo che $n! \geq (\frac{n}{2})^{\frac{n}{2}}$ da cui

$$\log_2(n!) \geq \frac{n}{2} \log_2 n - \frac{n}{2}$$

e

$$\log_2(f_2(n)) - log_2(f_{10}(n)) \geq \frac{49}{100} n \log_2 n - \frac{n}{2}$$

per cui possiamo sfruttare il punto b) di sopra. Abbiamo in defnitiva

$$f_1(n) < f_8(n) < f_3(n) < f_4(n) < f_5(n) = f_7(n) < f_6(n) < f_{10}(n) < f_2(n) < f_9(n).$$

soluzione formale

$$f_1(n) <_1 f_8(n) <_2 f_3(n) <_3 f_4(n) <_4 f_5(n) =_5 f_7(n) <_6 f_6(n) <_7 f_{10}(n) <_8 f_2(n) <_9 f_9(n).$$

Dimostro $f_1(n) <_1 f_8(n)$:

$$\lim_{n\to\infty} \frac{f_1(n)}{f_8(n)} = \lim_{n\to\infty} \frac{2^{\log_2 n}}{4^{\log_2 n}} = \lim_{n\to\infty} \frac{n}{n^2} = 0$$

Dimostro $f_8(n) <_2 f_3(n)$:

$$\lim_{n\to\infty} \frac{f_8(n)}{f_3(n)} = \lim_{n\to\infty} \frac{4^{\log_2 n}}{(\log_2 n)!} \leq \lim_{n\to\infty} \frac{4^{\log_2 n}}{\frac{(\log_2 n)}{2}^{\frac{(\log_2 n)}{2}}}$$

Dove abbiamo usato il fatto che $n! \geq \frac{n}{2}^{\frac{n}{2}}$ per $n > 0$. Pertanto

$$\lim_{n\to\infty} \frac{f_8(n)}{f_3(n)} = \lim_{n\to\infty} \left(\frac{32}{\log_2 n}\right)^{\frac{\log_2 n}{2}} = 0$$

Dimostro $f_3(n) <_3 f_4(n)$:

$$\lim_{n\to\infty} \frac{f_3(n)}{f_4(n)} = \lim_{n\to\infty} \frac{(\log_2 n)!}{n^{\log_2 n}} \leq \lim_{n\to\infty} \frac{(\log_2 n)^{\log_2 n}}{n^{\log_2 n}}$$

Dove abbiamo usato il fatto che $n! \leq n^n$ per $n > 0$. Pertanto

$$\lim_{n\to\infty} \frac{f_3(n)}{f_4(n)} \leq \lim_{n\to\infty} \frac{(\log_2 n)^{\log_2 n}}{n^{\log_2 n}} = \left(\frac{\log_2 n}{n}\right)^{\log_2 n} = 0.$$

Dimostro $f_4(n) <_4 f_5(n)$:

$$\lim_{n\to\infty} \frac{f_4(n)}{f_5(n)} = \lim_{n\to\infty} \frac{n^{\log_2 n}}{2^n} = \lim_{n\to\infty} \frac{(2^{\log_2 n})^{\log_2 n}}{2^n} = \lim_{n\to\infty} \frac{2^{\log_2^2 n}}{2^n} = \lim_{n\to\infty} 2^{\log_2^2 n - n} = 2^{\lim_{n\to\infty}(\log_2^2 n - n)} = 2^{-\infty} = 0.$$

Dimostro $f_5(n) =_5 f_7(n)$:

$$\lim_{n\to\infty} \frac{f_5(n)}{f_7(n)} = \lim_{n\to\infty} \frac{2^n}{2^{n+1}} = \frac{1}{2} > 0.$$

Dimostro $f_7(n) <_6 f_6(n)$:

$$\lim_{n\to\infty} \frac{2^{n+1}}{4^n} = \lim_{n\to\infty} 2\frac{2^n}{4^n} = 2 \lim_{n\to\infty} \left(\frac{2}{4}\right)^n = 2 \lim_{n\to\infty} \left(\frac{1}{2}\right)^n = 2 \cdot 0 = 0.$$

Dimostro $f_6(n) <_7 f_{10}(n)$:

$$\lim_{n\to\infty} \frac{f_6(n)}{f_{10}(n)} = \lim_{n\to\infty} \frac{4^n}{n^{n/100}} = \lim_{n\to\infty} \left(\frac{4^{100}}{n}\right)^{n/100} = 0.$$

Dimostro $f_{10}(n) <_8 f_2(n)$:

$$\lim_{n\to\infty} \frac{f_{10}(n)}{f_2(n)} = \lim_{n\to\infty} \frac{n^{n/100}}{n!} \leq \lim_{n\to\infty} \frac{n^{n/100}}{\left(\frac{49}{50}n\right)^{\frac{n}{50}}}$$

Dove abbiamo usato il fatto che

$$n! \geq n(n-1)\cdots\left(n-\frac{n}{50}\right) \geq \left(n-\frac{n}{50}\right)^{\frac{n}{50}} = \left(\frac{49}{50}n\right)^{\frac{n}{50}} =$$

Pertanto

$$\lim_{n\to\infty} \frac{f_{10}(n)}{f_2(n)} \leq \lim_{n\to\infty} \frac{(n^{1/2})^{n/50}}{\left(\frac{49}{50}n\right)^{\frac{n}{50}}} = \lim_{n\to\infty} \left(\frac{(n^{1/2})}{\frac{49}{50}n}\right)^{\frac{n}{50}} = \lim_{n\to\infty} \left(\frac{\sqrt{n}}{\frac{49}{50}n}\right)^{\frac{n}{50}} = \lim_{n\to\infty} \left(\frac{50}{49\sqrt{n}}\right)^{\frac{n}{50}} = 0$$

Dimostro $f_2(n) <_9 f_9(n)$:

$$\lim_{n\to\infty} \frac{f_2(n)}{f_9(n)} = \lim_{n\to\infty} \frac{n!}{(n+1)n! - n!} = \lim_{n\to\infty} \frac{n!}{((n+1)-1)n!} = \lim_{n\to\infty} \frac{n!}{n \cdot n!} = \lim_{n\to\infty} \frac{1}{n} = 0$$

Esercizio 4. *Siano $f(n)$ e $g(n)$ due funzioni asintoticamente non negative. Si assuma che $\lim_{n\mapsto\infty} \frac{f(n)}{g(n)} = 5$. Dimostrare che $f(n) = O(g(n))$. Possiamo concludere che $f(n) = \Theta(g(n))$?*

Supponiamo che sia

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \alpha,$$

per un qualche $\alpha > 0$. Allora, per la definizione di limite, dato un qualsiasi $\epsilon > 0$ possiamo trovare un n_0 a partire dal quale sia

$$\alpha - \epsilon \leq \frac{f(n)}{g(n)} \leq \alpha + \epsilon.$$

Scegliendo in particolare, ad esempio $\epsilon = \alpha/2$, otteniamo

$$\frac{\alpha}{2} \leq \frac{f(n)}{g(n)} \leq \frac{3}{2}\alpha.$$

Supponiamo ora che $g(n) > 0$ a partire da un certo n_1; sia $n_2 = \max(n_0, n_1)$. A partire da n_2 possiamo scrivere

$$\frac{\alpha}{2}g(n) \leq f(n) \leq \frac{3}{2}\alpha g(n)$$

il che è equivalente a dire che $f(n) \in \Theta(g(n))$.

Esercizio 5. *Si dimostri che $\binom{n}{\frac{n}{2}} = O(2^n)$.*

Visto che n deve essere pari, possiamo scrivere direttamente $n = 2m$.
Prima soluzione: abbiamo

$$\binom{2m}{m} = \frac{(2m)!}{m!\,m!} = \frac{2m}{m}\frac{2m-1}{m}\frac{2m-2}{m-1}\frac{2m-3}{m-1}\cdots\frac{2}{1}\frac{1}{1}.$$

Si vede chiaramente che nessuna delle $2m$ frazioni qui moltiplicate supera due.
Seconda soluzione. Secondo la formula di Stirling $k! = [1+\Theta(k^{-1})]\sqrt{2\pi k}\left(\frac{k}{e}\right)^k$ da cui

$$\binom{2m}{m} = \frac{(2m)!}{m!\,m!} = [1 + \Theta(m^{-1})]\sqrt{\frac{1}{\pi m}}2^{2m}.$$

Terza soluzione: i coefficienti della riga k-esima del triangolo di Tartaglia sommano a 2^k :

$$\sum_{j=0}^{k} \binom{k}{j} = 2^k.$$

Questo si dimostra o per induzione o considerando lo sviluppo del binomio $(1+1)^k$. Allora $\binom{2m}{m}$, essendo uno solo degli $m+1$ coefficienti della riga $2m$-esima, non può superare 2^{2m}.

Tra l'altro, ricordiamo che $\binom{2m}{m}$ è il più grande dei coefficienti della riga $2m$-esima, in quanto i coefficienti binomiali crescono muovendosi dai lati verso il centro. Questo si può vedere facilmente scrivendo

$$\binom{k}{j} = \frac{k}{1}\,\frac{k-1}{2} \cdots \frac{k-j+2}{j-1}\,\frac{k-j+1}{j}$$

e vedendo quando i fattori cominciano a diventare minori di uno. Questa semplice osservazione ci permette di stabilire che

$$\binom{2m}{m} \geq \frac{2^{2m}}{m+1}.$$

Quindi, con semplici osservazioni sul triangolo di Tartaglia, abbiamo scoperto che $\binom{2m}{m} \in O(2^{2m})$ e che $\binom{2m}{m} \in \Omega(2^{2m}m^{-1})$. Il risultato che si ottiene con Stirling, $\binom{2m}{m} \in \Theta(2^{2m}m^{-1/2})$, ci dice che la verità sta proprio nel mezzo.

Esercizio 6. *Si determini l'andamento asintotico nel caso peggiore di un tempo di calcolo T_n soggetto alla seguente ricorrenza*

$$T_n = T_k + T_{n-k} + \Theta(\min\{k, n-k\}) \qquad \textit{per } k \geq 1,\ k < n$$

E nel caso migliore?

soluzione in breve

Nel caso si abbia sempre $k = 1$, allora otteniamo la ricorrenza

$$T_n = T_1 + T_{n-1} + \Theta(1) = T_{n-1} + \Theta(1)$$

Che ha soluzione $T(n) = \Theta(n)$.

Nel caso si abbia sempre $k = \lfloor \frac{n}{2} \rfloor$, allora otteniamo la ricorrenza

$$T_n = T_{\lfloor \frac{n}{2} \rfloor} + T_{\lceil \frac{n}{2} \rceil} + \Theta(\frac{n}{2})$$

Che ha soluzione $T(n) = \Theta(n \log n)$ (è la stessa ricorrenza che per il *MergeSort*), come si può dedurre dal Master Theorem, e verificare per induzione.

Vorremmo ora evidenziare che i due casi esaminati sono effettivamente due casi estremi (in senso asintotico, si vedano gli approfondimenti per un'analisi più sottile). Per fare ciò verificheremo che, comunque vengano scelti i valori di k, avremo $c_1 n \leq T_n \leq c_2 n \log_n$, per induzione. Prima di partire con le 2 verifiche formali ci conviene però osservare una volta per tutte che una definizione equivalente della T_n è la seguente.

$$T_n = T_k + T_{n-k} + \Theta(k) \qquad \text{per } k \geq 1,\ k < \lfloor \tfrac{n}{2} \rfloor$$

Verifichiamo $T_n \geq c_1 n$, per induzione:

$$T_n = T_k + T_{n-k} + \Theta(k) \geq c_1 k + c_1(n-k) + \Theta(k) \geq c_1(k+n-k) = c_1 n$$

Verifichiamo $T_n \leq c_2 n \log_2 n$, per induzione:

$$\begin{aligned}
T_n &= T_k + T_{n-k} + \Theta(k) \leq c_2 k \log_2 k + c_2(n-k)\log_2(n-k) + \Theta(k) \leq c_2 k \log_2 k + c_2(n-k)\log_2 n + \Theta(k) \\
&= (c_2 k \log_2 n - c_2 k \log_2 \frac{n}{k}) + c_2(n-k)\log_2 n + \Theta(k) \leq c_2 n \log_2 n - c_2 k \log_2 2 + \Theta(k) \\
&= c_2 n \log_2 n - c_2 k + \Theta(k)
\end{aligned}$$

Resta pertanto induttivamente verificato che $T_n \leq c_2 n \log_2 n - c_2 k + \Theta(k) \leq c_2 n \log_2 n$ non appena la c_2 scelta sia sufficientemente grande da dominare la costante nascosta nella notazione $\Theta(k)$.

approfondimenti

Sia $T(n) = T(n-k) + T(k) + \Theta(\min(k, n-k))$. Questa ricorrenza non si può in generale risolvere con il Master Theorem. Bisogna quindi indovinare qualitativamente la soluzione e poi verificare per sostituzione diretta.

Ovviamente concediamo che k sia una funzione di n, $k = k(n)$. Supponiamo, senza perdita di generalità, che sia $k \leq n-k$. Un caso estremo che può capitare è quello in cui $k = n/2$ (lasciamo perdere in prima approssimazione gli arrotondamenti). Qui otteniamo $T(n) = 2T(n/2) + \Theta(n)$ che è la ricorrenza di mergesort, dunque con soluzione $T(n) \in \Theta(n \log n)$. L'altro caso estremo è quello in cui k sia costante, ad esempio $k = 1$. In tal caso, essendo $T(k) = T(1) = \Theta(1)$, abbiamo $T(n) = T(n-1) + \Theta(1)$ che ovviamente ha la soluzione $T(n) = \Theta(n)$.

Per il caso generale $1 < k < n/2$, intuiamo che l'andamento sia un qualcosa di intermedio tra questi due. Vediamo di formalizzare questa intuizione, per farlo in maniera pulita mettiamo in chiaro che cosa significa avere il $\Theta()$ entro la ricorrenza, e perchè non abbiamo specificato una condizione iniziale, ad esempio, su $T(1)$.

Con maggiore dettaglio, la ricorrenza di sopra si potrebbe scrivere, sempre supponendo $k(n) \leq n/2$,

$$\begin{aligned}
T(n) &\geq T(k(n)) + T(n-k(n)) + \alpha k(n) \\
T(n) &\leq T(k(n)) + T(n-k(n)) + \beta k(n) \\
T(1) = \gamma &
\end{aligned}$$

per opportune costanti $\alpha > 0, \beta > 0, \gamma > 0$. Notare che potremmo avere anche $\gamma = 0$; in tal caso meglio calcolarsi i primi $T(p)$ finchè non sia $T(p) > 0$, e usare quella come condizione iniziale. Supponiamo comunque per ora, senza perdita di generalità, che sia $p = 1$. A questo punto, per ottenere un limite superiore ci interesseremo alla seconda ed alla terza riga; studiando la ricorrenza

$$S(n) = S(k(n)) + S(n-k(n)) + k(n), \ S(1) = 1$$

è facile vedere che $T(n) \leq \max(\beta, \gamma) S(n)$. Per il limite inferiore, analogamente, ci interesseremo alla prima e terza riga; in questo caso avremo $T(n) \geq \min(\alpha, \gamma) S(n)$. Per questo possiamo supporre che tutte le costanti coinvolte siano pari a uno, e studiare la ricorrenza scritta sopra per $S(n)$. È facile rendersi conto che deve essere $S(n) \geq n$; essendo $k(n) \geq 1$, infatti,

$$S(n) \geq S(n-k(n)) + k(n)$$

(abbiamo addirittura trascurato un termine positivo); con la condizione iniziale data, è facile dimostrare la tesi per induzione.

Più complesso è dimostrare la limitazione superiore.

Esercizio 7. *Si consideri il seguente algoritmo* Goloson.

Input: un sacchetto di cioccolatini.

1. *se il sacchetto contiene al più un cioccolatino, allora mangiatelo e termina.*
2. *prendi un cioccolatino dal sacchetto ricevuto in input, leccalo, e mettilo in un sacchetto di cioccolatini leccati.*
3. *se il sacchetto ricevuto in input non è ancora vuoto, e finchè non ti assale lo scrupolo per il fatto di essere in dieta, puoi ripetere il passo 2.*
4. *preso dallo scrupolo (o dai rimpianti), conta i cioccolatini leccati, siano essi x.*
5. *nel tentativo di nascondere la tua debolezza, sposta $\lfloor \frac{x}{2} \rfloor$ cioccolatini dal sacchetto dei leccati al sacchetto ricevuto in input.*
6. *agisci da* Goloson *sul sacchetto dei cioccolatini leccati.*
7. *agisci da* Goloson *sul sacchetto dei cioccolatini ricevuto in input.*

Sia $G(n)$ il numero massimo di degustazioni di cioccolatino (leccatine + magnate) in cui si può incorrere partendo da un sacchetto di n cioccolatini. Descrivere $G(n)$ tramite una ricorrenza. Condurre analisi asintotica per $G(n)$. Riesci ad estrapolare od intuire la politica da tenere (quando conviene farsi prendere da scrupolo) qualora si voglia massimizzare il numero di degustazioni? Quale potrà essere invece il minor numero di degustazioni possibili partendo da un sacchetto di n cioccolatini? Riesci ad estrapolare anche formalmente quale politica porti a questo minimo.

Chiaramente, $G(0) = 0$ e $G(1) = 1$ sono le nostre condizioni al contorno. Indicato con l il numero di cioccolatini leccati ad una generica chiamata della procedura, otteniamo facilmente la seguente ricorrenza.

$$G(n) = G(\lceil \frac{l}{2} \rceil) + G(n - \lceil \frac{l}{2} \rceil) + l \qquad \text{con } l \geq 1,\ l \leq n$$

Nel caso si abbia sempre $l = 1$, allora otteniamo la ricorrenza

$$G(n) = G(1) + G(n-1) + 1 = G(n-1) + 2$$

Che ha soluzione $G(n) = 2n - 1$, vista la condizione al contorno $G(1) = 1$.

Nel caso si abbia sempre $l = n$, allora otteniamo la ricorrenza

$$G(n) = G(\lceil \frac{n}{2} \rceil) + G(\lfloor \frac{l}{2} \rfloor) + n$$

Che ha soluzione $G(n) = \Theta(n \log n)$ (è la stessa ricorrenza che per il *MergeSort*), come si può dedurre dal Master Theorem, e verificare per induzione.

Vorremmo ora evidenziare che i due casi esaminati sono effettivamente due casi estremi (in senso asintotico, ma provate voi a vedere se sia possibile condurre un'analisi più sottile). Per fare ciò verificheremo che, comunque il nostro *Goloson* scelga i valori di l, avremo $c_1 n \leq G(n) \leq c_2 n \log_n$, per induzione.

Verifichiamo $G(n) \geq 2n - 1$, per induzione:

$$G(n) = G(\lceil \frac{l}{2} \rceil) + G(n - \lceil \frac{l}{2} \rceil) + l \geq 2\lceil \frac{l}{2} \rceil - 1 + 2(n - \lceil \frac{l}{2} \rceil) - 1 + 1 = 2n - 1$$

Dove nella diseguaglianza abbiamo poggiato sull'ipotesi induttiva ma abbiamo sfruttato anche il fatto che nessuno può trattenere *Goloson* dal dare almeno una leccatina ($l \geq 1$).

Verifichiamo $G(n) \leq c_2 n \log_2 n$, per induzione:

$$\begin{aligned}
G(n) &= G(\left\lceil \frac{l}{2} \right\rceil) + G(n - \left\lceil \frac{l}{2} \right\rceil) + l \leq c_2 \left\lceil \frac{l}{2} \right\rceil \log_2 \left\lceil \frac{l}{2} \right\rceil + c_2(n - \left\lceil \frac{l}{2} \right\rceil) \log_2(n - \left\lceil \frac{l}{2} \right\rceil) + l \\
&\leq c_2 \left\lceil \frac{l}{2} \right\rceil \log_2 \left\lceil \frac{l}{2} \right\rceil + c_2(n - \left\lceil \frac{l}{2} \right\rceil) \log_2 n + l \\
&= (c_2 \left\lceil \frac{l}{2} \right\rceil \log_2 n - c_2 \left\lceil \frac{l}{2} \right\rceil \log_2 \frac{n}{\lceil \frac{l}{2} \rceil}) + c_2(n - \left\lceil \frac{l}{2} \right\rceil) \log_2 n + l \leq c_2 n \log_2 n - c_2 \left\lceil \frac{l}{2} \right\rceil \log_2 \frac{3}{2} + l \\
&\leq c_2 n \log_2 n - c_2 \frac{l}{2} \log_2 \frac{3}{2} + l
\end{aligned}$$

E l'induzione si chiude non appena $c_2 \geq 2 \log_{\frac{3}{2}} 2$.