Università degli Studi di Camerino – Laurea in Informatica
Prima Prova Parziale di **Algoritmi e Strutture Dati**

Docente: Emanuela Merelli

24 novembre 2009

Nome: Cognome: N.Matricola:

1. (3 points) Dimostrare, utilizzando il principio di induzione, che

$$\sum_{i=1}^{n+1}(i-1)^2 = \frac{n(n+1)(2n+1)}{6}$$

Soluzione:

caso base $n = 1$

$$\sum_{i=1}^{2}(i-1)^2 = 1 \quad \frac{1(1+1)(2+1)}{6} = 1 \ ok$$

passo induttivo, affermiamo che la proprietà è vera per n e dimostriamo che per $n+1$ vale

$$\sum_{i=1}^{n+2}(i-1)^2 = \frac{(n+1)(n+2)(2(n+1)+1)}{6} = \frac{(n+1)(n+2)(2n+3)}{6}$$

$$\sum_{i=1}^{n+2}(i-1)^2 = \sum_{i=1}^{n+1}(i-1)^2 + (n+1)^2 = \frac{n(n+1)(2n+1)}{6} + (n+1)^2 = \frac{n(n+1)(2n+1)+6(n+1)^2}{6}$$

$$\frac{2n^3+n^2+2n^2+n+6n^2+6+12n}{6} = \frac{2n^3+9n^2+13n+6}{6} = \frac{(n+1)(n+2)(2n+3)}{6}$$

2. (5 points) Mettere in ordine di velocità di crescita le seguenti funzioni in n e determinare il valore di n_0 per il quale l'ordinamento proposto è valido per ogni $n \geq n_0$:

$$\left(\log n^2, \quad n!, \quad n^n, \quad n^3, \quad \log 2n, \quad 2^n, \quad \log n^n, \quad 2n \right)$$

Soluzione:

$$\left(\log 2n \leq \quad \log n^2 \leq \quad 2n \leq \quad \log n^n \leq \quad n^3 \leq \quad 2^n \leq \quad n! \leq \quad n^n \right) \quad n \geq 10$$

3. (3 points) Che cosa è una invariante di ciclo e come si utilizza per capire se un algoritmo è corretto, commentare in termini di invariante di ciclo l'esempio dell'ordinamento.

4. (3 points) Data $f(n) = n \log n + n$ e $g(n) = (n^2)/2$ dimostrare se $f(n) = O(g(n))$ oppure $g(n) = O(f(n))$?

Soluzione 1:

Proviamo a dimostrare che $g(n)$ è un limite superiore di $f(n)$.
$g(n)$ è un limite superiore di $f(n)$, indicato con $f(n) \leq O(g(n))$, se esistono due costanti positive c, n_0 tali che

$$f(n) \leq cg(n) \quad per \ \ ogni \quad n \geq n_0$$

Quindi, posta

$$n \log n + n \leq c\frac{n^2}{2}$$

$$\log n + 1 \leq c\frac{n}{2}$$
$$2\log n + 2 \leq cn$$
$$2(\log n + 1) \leq cn$$

poniamo $c = 2$ e valutiamo la seguente disuguaglianza

$$(\log n + 1) \leq n$$

che risulta essere vera per ogni $n \geq 1$. Pertanto possiamo affermare che esistono $c = 2$ e $n_0 = 1$ per cui

$$n\log n + n \leq c\frac{n^2}{2}$$

da cui $f(n) = O(g(n))$

NB: si potevano proporre altre dimostrazioni.

5. (3 points) Indicare quali delle seguenti approssimazioni è vera o falsa

k^3	$=$	$O(k^2)$	$falsa$	k^3	$=$	$O(k + k\sqrt(k) + \sqrt(k)$	$falsa$
$\log k$	$=$	$O(k^{-\frac{1}{2}})$	$falsa$	k^2	$=$	$O(k^3)$	$vera$
$2k^2$	$=$	$O((k^2(1+k^2))$	$vera$	$3k^2 + \sqrt(k)$	$=$	$O(n^2)$	$vera$
$\log k$	$=$	$O(1/k)$	$falsa$	$1/k$	$=$	$O(\log k)$	$vera$

6. (4 points) Applicare il metodo iterativo alla seguente equazione di ricorrenza

$$T(n) = \left\{ \begin{array}{cc} n^3 + 3T(\frac{n}{3}) & n > 1 \\ \\ 1 & n = 1 \end{array} \right)$$

Soluzione:

$$\begin{array}{ll} T(n) & = n^3 + 3T(\frac{n}{3}) = \\ & = n^3 + 3((\frac{n}{3})^3 + 3T(\frac{n}{3^2})) = \\ & = n^3 + \frac{n^3}{3^2} + 3^2T(\frac{n}{3^2}) = \\ & = n^3 + \frac{n^3}{3^2} + 3^2((\frac{n}{3^2})^3 + 3T(\frac{n}{3^3})) = \\ & = n^3 + \frac{n^3}{3^2} + \frac{n^3}{9^2} + 3^3T(\frac{n}{3^3}) = \\ & = n^3 + \frac{n^3}{3^2} + \frac{n^3}{9^2} + 3^3((\frac{n}{3^3})^3 + 3T(\frac{n}{3^4}) = \\ & = n^3 + \frac{n^3}{3^2} + \frac{n^3}{9^2} + \frac{n^3}{27^2} + 3^4T(\frac{n}{3^4}) = \\ & = n^3 + \frac{n^3}{3^2} + \frac{n^3}{9^2} + \frac{n^3}{27^2} + 3^4((\frac{n}{3^4})^3 + 3T(\frac{n}{3^4})) = \\ & = \ldots \\ & = \sum_{i=0}^{k-1}(\frac{n^3}{9^i}) + 3^kT(\frac{n}{3^k}) = \\ & = n^3\sum_{i=0}^{k-1}(\frac{1}{9})^i + 3^kT(\frac{n}{3^k}) = \end{array}$$

L'iterazione termina quando $\frac{n}{3^k} = 1 \;\rightarrow\; n = 3^k \;\rightarrow\; k = \log_3 n$.

$$\begin{array}{ll} T(n) & = n^3\sum_{i=0}^{\log_3 n - 1}(\frac{1}{9})^i + n = \\ & n^3\frac{1-(\frac{1}{9})^{\log_3 n}}{1-\frac{1}{9}} + n = \\ & n^3\frac{1-(\frac{1}{9})^{\log_3 n}}{\frac{8}{9}} + n = \\ & \frac{8}{9}n^3(1 - \frac{1}{9^{\log_3 n}}) + n = \frac{8}{9}n^3 - \frac{8}{9}n + n = \frac{8}{9}n^3 + \frac{1}{9}n = \Theta(n^3) \end{array}$$

7. (3 points) Dimostrare per induzione la seguente eguaglianza $k! = 2^{k-1}$

Soluzione:

La disuguaglianza è falsa, infatti basta prendere $k = 3$ per verificare che $3! \neq 2^2$ essendo $3! = 6$.

La dimostrazione per induzione avrebbe condotto ad un assurdo.
caso base, $k = 1 \quad \rightarrow \quad 1! = 2^{1-1} = 1$ la relazione è vera
passo induttivo, assumiamo vera la relazione per k dimostriamo essere vera per $k+1$, cioè dimostriamo che $(k+1)! = 2^k$.

$$\begin{array}{ll} (k+1)! = k!(k+1) = 2^{k-1}(k+1) & \textit{per ipotesi induttiva} \\ (k+1)! = k!(k+1) = 2^{k-1}(k+1) = 2^k & \textit{solo per } k = 1 \end{array}$$

Affinchè l'uguaglianza sia vera per ogni k dovrebbe risultare $k + 1 = 2$, il che è assurdo essendo k una variabile.

NB: Se si fosse richiesto di dimostrare la seguente disuguaglianza $k! \geq 2^{k-1}$. Allora ... La dimostrazione per induzione sarebbe stata la seguente:
caso base, $k = 1 \quad \rightarrow \quad 1! = 2^{1-1} = 1$ la relazione è vera
passo induttivo, assumiamo vera la relazione per k dimostriamo essere vera per $k+1$, cioè dimostriamo che $(k+1)! \geq 2^k$.

$$\begin{array}{ll} (k+1)! = k!(k+1) = 2^{k-1}(k+1) & \textit{per ipotesi induttiva} \\ (k+1)! = k!(k+1) = 2^{k-1}(k+1) \geq 2^{k-1}(1+1) & \textit{per riduzione del termine a destra} \\ (k+1)! = k!(k+1) = 2^{k-1}(k+1) \geq 2^{k-1}(1+1) = 2^{k-1}2 = 2^k & \textit{per } \ k \geq 0 \end{array}$$

8. (3 points) Si consideri una tabella hash di dimensione $m = 10$ inizialmente vuota. Si mostri il contenuto della tabella dopo aver inserito la seguente sequenza di valori 14, 82, 36, 59, 26, 84, 17, 27, 15, 55. Si assuma che le collisioni vengano gestite mediante indirizzamento aperto utilizzando come funzione hash $h(k, i) = (h'(k) + 3i + i^2)$ mod m dove la funzione hash ordinario $h'(k) = k$ mod m.

   Soluzione:

$$\begin{array}{llll} h(14,0) & = (14 \; mod \; 10) \; mod \; 10 & = 4 & per \text{ i=0} \\ h(82,0) & = (82 \; mod \; 10) \; mod \; 10 & = 2 & per \text{ i=0} \\ h(36,0) & = (36 \; mod \; 10) \; mod \; 10 & = 6 & per \text{ i=0} \\ h(59,0) & = (59 \; mod \; 10) \; mod \; 10 & = 9 & per \text{ i=0} \\ h(26,0) & = (26 \; mod \; 10) \; mod \; 10 & = 6 & per \text{ i=0} \\ & = (26 \; mod \; 10 + 3 + 1) \; mod \; 10 & = 0 & per \text{ i=1} \\ h(84,0) & = (84 \; mod \; 10) \; mod \; 10 & = 4 & per \text{ i=0} \\ & = (84 \; mod \; 10 + 3 + 1) \; mod \; 10 & = 8 & per \text{ i=1} \\ h(17,0) & = (17 \; mod \; 10) \; mod \; 10 & = 7 & per \text{ i=0} \\ h(27,0) & = (27 \; mod \; 10) \; mod \; 10 & = 7 & per \text{ i=0} \\ & = (27 \; mod \; 10 + 3 + 1) \; mod \; 10 & = 1 & per \text{ i=1} \\ h(15,0) & = (15 \; mod \; 10) \; mod \; 10 & = 5 & per \text{ i=0} \\ h(55,0) & = (55 \; mod \; 10) \; mod \; 10 & = 5 & per \text{ i=0} \\ & = (55 \; mod \; 10 + 3 + 1) \; mod \; 10 & = 9 & per \text{ i=1} \\ & = (55 \; mod \; 10 + 6 + 4) \; mod \; 10 & = 5 & per \text{ i=2} \\ & = (55 \; mod \; 10 + 9 + 9) \; mod \; 10 & = 3 & per \text{ i=3} \end{array}$$

9. (3 points) Qual è il tempo richiesto nel caso peggiore per la ricerca di una chiave in una tabella hash con n elementi che gestisce liste di collisioni. Giutificare la risposta. (a) $O(1)$ (b) $O(\log n)$ (c) $O(n)$ (d) $O(n2)$

   Soluzione:

   Il tempo necessario, nel caso peggiore, per ricercare un elemento con chiave k in una tabella hash con n elementi che utilizza liste di collisioni per gestire i conflitti, è

$$O(n)$$

   Il caso peggiore si riscontra quando tutti gli n elementi collidono nella stessa casella.

10. (5 points) Trovare la soluzione delle seguenti relazioni di ricorrenza, disegnando l'albero della ricorsione, per iterazione, per sostituzione o usando, se possibile, il Teorema Master:

    (i) $T(n) = n + 4T(\frac{n}{2})$. (ii) $T(n) = n^2 + 4T(\frac{n}{2})$ (iii) $T(n) = n^3 + 4T(\frac{n}{2})$.

Soluzione:

(i) $T(n) = n + 4T(\frac{n}{2})$ per il Teorema Master abbiamo $a = 4$, $b = 2$, $\log_2 4 = 2$ e $n^{log_2 4} = n^2$.

1. se $n^{\log_b a}$ cresce più velocemente di $f(n)$, allora $T(n) = \Theta(n^{\log_b a})$
2. se $n^{\log_b a}$ cresce esattamemente come $f(n)$, allora $T(n) = \Theta(n^{\log_b a} \log_2 n)$
3. se $n^{\log_b a}$ cresce più lentamemente di $f(n)$, allora $T(n) = \Theta(f(n))$

Per il caso 1 abbiamo che

$$\begin{aligned} f(n) &= n = O(n^{2-\epsilon}) \ \ \textit{per} \ \ \epsilon = 1 \ \ \textit{si ha} \\ T(n) &= \Theta(n^2) \end{aligned}$$

(ii) $T(n) = n^2 + 4T(\frac{n}{2})$ per il Teorema Master abbiamo $a = 4$, $b = 2$, $\log_2 4 = 2$ e $n^{log_2 4} = n^2$.
Per caso 2 si ha che

$$\begin{aligned} f(n) &= n^2 = \Theta(n^2) \ \ \textit{si ha quindi} \\ T(n) &= \Theta(n^2 \log_2 n) \end{aligned}$$

(iii) $T(n) = n^3 + 4T(\frac{n}{2})$ per il Teorema Master abbiamo $a = 4$, $b = 2$, $\log_2 4 = 2$ e $n^{log_2 4} = n^2$.
Per caso 3 si ha che

$$\begin{aligned} f(n) &= n^3 = \Omega(n^{2+\epsilon}) \ \ \textit{per} \ \ \epsilon = 1 \ \ \textit{si ha} \\ T(n) &= \Omega(n^3) \end{aligned}$$

11. (3 points) Si scriva un algoritmo ricorsivo di *visita in profondità* per calcolare la profondità di un albero binario.

Soluzione:

Question:	1	2	3	4	5	6	7	8	9	10	11	Total
Points:	3	5	3	3	3	4	3	3	3	5	3	38
Score:												