Introduzione agli algoritmi
Prova di esame del 7/11/2016
Prof.sse E. Fachini - R. Petreschi

Parte prima

1)
Si dimostri il teorema sulla limitazione inferiore per il tempo asintotico di esecuzione nel caso peggiore di un algoritmo di ordinamento nel modello basato sui confronti.
Sol. Si veda libro di testo.

2)
a) Dimostrare la verità o la falsità delle seguenti asserzioni:
- siano p(n) e q(n) due polinomi.Se il grado di p(n) è maggiore del grado di q(n), allora p(n) = O(q(n)).
- **Sol.** Falso. Basta considerare $p(n) = n^2$ e $q(n) = n$. Dovrebbero esistere due costanti n_0 e $c \geq 0$ tali che $n^2 \leq c\, n$, per ogni $n \geq n_0$. Dividendo per n, si ottiene $n \leq c$ che è falso essendo c una costante, per cui per ogni scelta di c posso prendere un valore di n maggiore.
- se $f(n) = O(g(n))$ allora $f(n)^k = O(g\ (n)^k)$ per ogni $k \geq 1$.
- **Sol.** Vero . Per definizione esistono due costanti n_0 e $c \geq 0$ tali che $f(n) \leq c\, g(n)$, per ogni $n \geq n_0$. Elevando alla k-sima potenza la disuguaglianza rimane vera perchè l'elevamento a potenza è una funzione crescente.

b) Se si dimostra che un algoritmo ha tempo di esecuzione in Θ(n) nel caso migliore, posso dedurne che nel caso peggiore terminerà in O(n) passi?
Sol. No, il limite superiore al tempo di esecuzione nel caso migliore può non essere affatto tale per il caso peggiore. Basta pensare all'insertionSort che nel caso migliore ha tempo di esecuzione in Θ(n), ma nel peggiore in $\Theta(n^2)$.

3)
Dati due heap minimi, H1 e H2, di n e m elementi, rispettivamente, stampare il valore del terzo elemento nell'ordinamento crescente degli n+m elementi. Si dia una valutazione asintotica del tempo di esecuzione dell'algoritmo presentato.

Sol. Si tratta di individuare la possibile posizione del terzo elemento in uno dei due Minheap. Può trovarsi nel livello 0, 1 o 2 , come mostrano i seguenti esempi.
Nel seguente esempio il terzo elemento, 3, è nel livello 2 del primo minHeap.

H1 = 1 e H2 = 5
2 20 7 10
3 30 9 8

Invece nel seguente esempio è 7 che si trova nel livello 1 di H2

H1 = 1 e H2 = 5
12 20 7 10
9 8

Nel seguente esempio è 5 che si trova nel livello 0 di H2

H1= 1 e H2 = 5
3 20 7 14
9 8

Il tempo di esecuzione è in O(1) perchè si deve confrontare un numero costante di elementi.
La soluzione più semplice è quella di copiare gli al più 7 elementi, dei due minHeap in un array d'appoggio di al più 14 elementi e ordinarli con l'insertionSort, restituendo il terzo elemento dell'array d'appoggio.
Così anche lo spazio aggiuntivo è in O(1).

Prova di esame del 7/11/2016
Prof.sse E. Fachini - R. Petreschi

Parte seconda

1)
Si presenti, facendo uso di disegni, l'idea dell'algoritmo di cancellazione in un generico albero binario di ricerca. Si dia una valutazione asintotica del suo tempo di esecuzione.
Sol. Si veda libro di testo e lucidi.

2)
Si determini il tempo di esecuzione asintotico T(n) della seguente funzione:

analizzami(A,i,j)
n = j - i+1
c = 1
h = i + (j - i +1)/3
k = h + (j - i +1)/3
d = h
while d > i do
for m = 1 to n do c++
d = d/3
if n > 1 then
return **analizzami**(A,i,h) + **analizzami**(A,h+1,k) + **analizzami**(A,k+1,j)
Sol. La relazione di ricorrenza è
$T(n) = 3T(n/3) + O(n\log_3 n)$
Ogni nodo dell'albero della ricorsione a livello i è etichettato $cn\log_3 n/3^i$, l'albero ha altezza $\log_3 n$, quindi concludiamo che
$T(n) = 3^{\log_3 n}T(1) + c\Sigma_{i=0,..,(\log_3 n)-1}\ n\log_3 n/3^i = nT(1) + cn\log_3^2 n - cn\Sigma_{i=0,\log_3 n}\ i \leq nT(1) + cn\log_3^2$, da cui $T(n)=O(n \lg^2 n)$.
Prova induttiva che esistono n_0 e $d \geq 0$ tali che $T(n) \leq dn \log_3^2 n$, per ogni $n \geq n_0$ (per semplicità prendiamo direttamente il logaritmo in base 3).
Supponiamo sia vero per ogni $m<n$ e consideriamo T(n).
Per definizione $T(n) = 3T(n/3) + cn\log_3 n$ e per ipotesi induttiva $T(n/3) \leq dn/3 \log_3^2 n/3$, quindi $T(n) \leq 3nd/3 \log_3^2 n/3 + cn\log_3 n =$
$nd(\log_3 n - \log_3/3)^2 + cn\log_3 n = nd(\log_3^2 n + 1 - 2\log_3 n) + cn\log_3 n$.

Verifichiamo se $nd(\log_3^2 n + 1 - 2\log_3 n) + cn\log_3 n \leq nd\log_3^2 n$, questo è vero se $nd + cn\log_3 n \leq 2nd\log_3 n$, considerando $n \geq 1$, dividiamo per n e otteniamo $d + c\log_3 n \leq 2d\log_3 n$. Prendendo d = c si ottiene $c \leq c\log_3 n$, che per $n \geq 3$ è vera. Quindi con le costanti $n_0 = 3$ e d = c, le cose vanno bene, per ora.
Controlliamo T(3). $T(3) = 3T(1) +3c$, dovendo essere $T(3) \leq d\ 3 \log_3^2 3 = 3d$,

dobbiamo modificare la costante in modo da catturare anche il caso base. Ponendo T(1) = a, e prendendo d = c+a, si ha T(3) ≤ 3(a+c), che è vero. Possiamo concludere che $T(n) \leq (a+c)n \log_3^2 n$, per ogni $n \geq n_0 = 3$ e quindi che l'ipotesi $T(n)=O(n \lg^2 n)$ è verificata.

3)
Dato un vettore di 3n elementi, n contenenti il valore Bianco, n il valore Rosso ed n il valore Verde. Si ordini il vettore in modo da avere prima tutti gli elementi di colore Verde, poi quelli di colore Bianco e infine quelli di colore Rosso, in O(n). I colori sono chiavi di elementi complessi, quindi si esclude la semplice riscrittura del vettore con n Verdi, poi n Bianchi e infine gli n Rossi.
Sol. Si tratta di utilizzare l'algoritmo della tripartition spiegato a lezione il 31 marzo 2016.