

Algoritmi I (Informatica + Tecnologie, A.A. 2008-2009)

Prova di esame del 10/9/2009

Proff. Finocchi / Galesi

Problema 1

Supporre di avere una sequenza ordinata di n chiavi, $k_1 \dots k_n$, e di volerle inserire in un albero binario di ricerca.

- 1) Qual è la minima altezza h di un albero binario che contiene n nodi? Dare una stima esatta (non asintotica) del valore di h in funzione di n , motivando la risposta.
- 2) Progettare un algoritmo per inserire le chiavi $k_1 \dots k_n$ in un albero binario di ricerca in modo che l'albero risultante abbia altezza minima. Analizzare la correttezza e il tempo di esecuzione dell'algoritmo proposto.

Problema 2

Il Bubblesort è un noto algoritmo di ordinamento:

```
void Bubblesort(int A[])
  for (i = 1; i <= length[A]; i++)
    for (j=length[A]; j >= i+1; j--)
      if (A[j] < A[j - 1])
        then swap(A[j], A[j - 1])
```

Il *cocktail shaker sort* è una variante del bubblesort che alterna passate sull'array da sinistra verso destra con passate da destra verso sinistra.

- 1) Dare lo pseudocodice del cocktail shaker sort.
- 2) Qual è la proprietà invariante mantenuta durante l'esecuzione dell'algoritmo? Perché l'algoritmo è corretto?
- 3) Qual è il numero di passate eseguite dal cocktail shaker sort in funzione di n ? Qual è il tempo di esecuzione asintotico dell'algoritmo?

Algoritmi I (Informatica + Tecnologie, A.A. 2008-2009)

Prova di esame del 10/9/2009

Proff. Finocchi / Galesi

Problema 3

Si considerino le seguenti due funzioni:

<pre>void f(int n) { int i=1, j=n; while (i<=j) do if (TEST) { I; i=i+1; } else { J; j = j-1; } }</pre>	<pre>void g (int n) { f(n); while (n>=1) { f(n); n = n/2; f(n); } }</pre>
--	--

dove $n > 0$ e TEST, I e J sono sequenze di istruzioni che si eseguono in tempo $O(1)$. Studiare il tempo di esecuzione della funzione g.

Problema 4

Sia M la matrice di adiacenza di un grafo orientato G con n vertici. Progettare un algoritmo che, dati n e M, stampi tutti i vertici *sorgente* e tutti i vertici *pozzo* di G. Un vertice è una sorgente se non ha archi entranti, ed è un pozzo se non ha archi uscenti. L'algoritmo deve operare in modo ricorsivo. Analizzare il tempo di esecuzione dell'algoritmo proposto.