

API Parte 2: Informatica 3

Prof. Giovanni Agosta

Prova scritta – Appello del 25/02/2011¹

COGNOME e NOME:..... Matricola:.....

¹Tempo: 1 ora e 30 minuti. Possono essere consultati libri di testo e slide del corso. È consentito scrivere a matita. Scrivere il proprio nome sugli eventuali fogli aggiuntivi. I punti bonus sono attribuiti per *un solo esercizio! Rispondere punto per punto alle domande!*

Esercizio 1 (10 punti)

Si consideri un array bidimensionale A di m righe e n colonne, in cui ogni riga è ordinata in ordine crescente. Descrivere un algoritmo in grado di riunire le m righe di A in un unico array ordinato B di nm elementi in tempo $O(nm \log m)$.

[Suggerimento: mantenere il primo elemento non ancora copiato in B di ciascuna riga in uno heap H di m elementi].

Bonus (3 punti): dimostrare la correttezza dell'algoritmo implementato.

Esercizio 2 (10 punti)

Sia dato un array A di n elementi interi. Considerate le seguenti operazioni definite su A :

- $Insert(A, i, x)$: assegna il valore x all'elemento $A[i]$;
- $Sum(A, l, m)$: calcola la somma degli elementi di A con indice compreso tra l e m .

Implementare in pseudocodice le due operazioni in modo da far sì che $Sum(A, l, m)$ abbia complessità dell'ordine di $O(\log|A|)$, eventualmente introducendo strutture dati di supporto (anche persistenti fra invocazioni distinte delle varie funzioni). Dire se e come cambia la complessità dell'operazione di inserimento, nel caso in cui risulti necessario l'aggiornamento di strutture dati di supporto, rispetto al normale inserimento in un array ($O(1)$).

Bonus (3 punti): dimostrare la correttezza dell'algoritmo Sum implementato.

Esercizio 3 (10 punti)

Risolvere le seguenti equazioni alle ricorrenze:

- $T(n) = T(n/3) + T(2n/3) + \log 3n.$

- $T(n) = T(n/2) + \Theta(n)$

- $T(n) = T(n - 3) + n + \log n$

- $T(n) = 7T(n/5) + n^4 + n^2$

- $T(n) = T(\sqrt{n}) + \Theta(1)$

Si specifichi inoltre in pseudocodice un'ipotetica funzione *foo* che abbia una complessità pari all'equazione del punto 1.

Bonus (3 punti): specificare la funzione *foo* attraverso una macchina RAM invece che in pseudocodice.

