Algoritmi I (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 9 Settembre 2008

Gli studenti esonerati devono svolgere solo gli esercizi 3 e 4. Nel caso in cui abbiano già consegnato la seconda parte del compito in un appello precedente (non ottenendo la sufficienza), il primo esonero non è più valido e l'esame deve essere sostenuto per intero.

Esercizio 1. Un array A di dimensione n è *quasi ordinato* se l'i-esimo numero più piccolo, con $1 < i < n$, occupa la posizione $i-1$, i, oppure $i+1$. Il minimo occupa la posizione 1 o 2, e il massimo la posizione $n-1$ o n. Assumendo che gli elementi siano tutti distinti, risolvere i seguenti problemi:

- Dato un array quasi ordinato, progettare un algoritmo, il più possibile efficiente, per ordinare l'array, ed analizzarne correttezza e tempo di esecuzione.
- Dato un array quasi ordinato di dimensione n dispari, trovare in tempo costante il mediano dell'array, ovvero l'elemento che occuperebbe la posizione $\lceil n/2 \rceil$ se l'array fosse ordinato. Analizzare la correttezza dell'algoritmo proposto.

Esercizio 2. Mostrare un algoritmo il cui tempo di esecuzione soddisfi l'equazione di ricorrenza:

$$T(n) = \begin{cases} 2\,T(\lfloor n/3 \rfloor) + \Theta(1) & \textit{se } n > 2 \\ \Theta(1) & \textit{altrimenti} \end{cases}$$

Esercizio 3. Proporre un algoritmo che, dato un albero AVL T, crei un albero binario di ricerca T' uguale a T. L'algoritmo, al fine di costruire T', può eseguire soltanto ripetuti inserimenti di chiavi in un albero binario di ricerca (utilizzando l'operazione `insert` degli alberi binari di ricerca) e deve essere il più possibile efficiente. Analizzare correttezza e tempo di esecuzione dell'algoritmo proposto.

Esercizio 4. Sia $G = (V, E)$ un grafo non orientato e connesso e sia $u \in V$ uno specifico vertice. Supponete di calcolare un albero di copertura di G radicato in u tramite una visita in profondità. Sia T tale albero. Supponete ora di eseguire anche una visita in ampiezza di G a partire da u , ottenendo lo stesso albero T. Dimostrare che deve essere $G = T$, ovvero: se T è un albero di copertura di G ottenibile sia da una visita in profondità che in ampiezza, allora G non può contenere archi che non appartengono a T.

Appello del 9 Settembre 2008

Nome	Cognome

Soluzione Esercizio 1:

Appello del 9 Settembre 2008

Nome	Cognome

Soluzione Esercizio 2:

Algoritmi I (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 9 Settembre 2008

Appello del 9 Settembre 2008

Nome	Cognome

Soluzione Esercizio 3:

Appello del 9 Settembre 2008

Appello del 9 Settembre 2008

Nome	Cognome

Soluzione Esercizio 4:

Appello del 9 Settembre 2008