

Programmazione II

Appello del 7 Giugno 2004

Per cortesia leggere attentamente:

- *Volete fare il secondo esonero?* Fare l'esercizio di sbarramento e l'esercizio 1.
- *Volete fare l'esame completo?* Fare tutti gli esercizi.

Se l'esercizio di sbarramento non viene svolto in modo soddisfacente il resto del compito non viene corretto.

Avete 1 ora e 30 minuti di tempo.

IN BOCCA AL LUPO!

Esercizio di sbarramento. Si scriva una procedura ricorsiva per effettuare il merge di due liste ordinate.

Esercizio 1. Siano dati un vettore $x[]$ di interi di dimensione nota n e due numeri interi p_1 e p_2 chiamati *pivot*. Si scriva una procedura che partizioni $x[]$ in tre zone contigue: nella prima zona si trovano gli elementi minori o uguali a p_1 , nella seconda quelli maggiori di p_1 e minori o uguali a p_2 e nella terza quelli maggiori di p_2 .

Si dia una stima della complessità computazionale della vostra procedura.

Esercizio 2. Si implementi la procedura di cancellazione di un elemento da un albero di ricerca.

Esercizio di sbarramento. Sino dati un vettore $x[]$ di dimensione nota n ed un intero v . Si scriva una funzione C ricorsiva che implementi la ricerca binaria.

Esercizio 1. Un *treap* é un albero ternario, in cui cioè un nodo può avere sino a tre figli, che rispetta la proprietà della forma e quella dell'ordine.

- Si implementi la procedura SiftDown per un treap dando una stima del tempo di esecuzione;
- Usando SiftDown si implementi TreapSort, un algoritmo per ordinare un vettore di interi dato $x[]$ di dimensione nota n .

Esercizio 2. Si implementi la procedura di cancellazione di un elemento da un albero di ricerca.

Esercizio di sbarramento. Dato un albero binario di ricerca T e due interi i e j , si scriva una funzione che restituisca il numero di elementi contenuti in T compresi tra i e j (estremi inclusi).

Esercizio 1. Un albero binario di numeri interi si dice *sinistro* se, per ogni nodo dell'albero, la somma dei valori del sottoalbero sinistro è maggiore o uguale alla somma dei valori del sottoalbero destro. L'albero nullo è sinistro. Si scriva una funzione che verifichi se un albero binario è sinistro oppure no.

Si dia una stima della complessità computazionale della vostra procedura.

Esercizio 2. Si implementi la procedura di cancellazione di un elemento da un albero di ricerca.

Esercizio di sbarramento. Definendo le opportune strutture dati si scriva una funzione C per l'inserimento di un elemento in un albero di ricerca.

Esercizio 1. Si scriva una funzione che, dato in input un albero binario, ne calcoli la *larghezza*, cioè il massimo numero di elementi appartenenti allo stesso livello. Si dia una stima della complessità computazionale della vostra procedura.

Esercizio 2. Si implementi la procedura di cancellazione di un elemento da un albero di ricerca.