Laurea in "Scienze di Internet"
Corso di "Algoritmi e Strutture Dati"
14 Giugno 2005

1. *Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti.*
2. *Sono ammessi al più 3 scritti consegnati per l'A.A. 2004/05 (Gennaio-Settembre 2005)*
3. *Non è possibile consultare appunti, libri o persone, né uscire dall'aula.*
4. *Ogni esercizio conta 6 punti. Si raggiunge il 18 con 3 esercizi risolti correttamente.*
5. *Le soluzioni degli esercizi devono:*

a. *spiegare a parole l'algoritmo usato (anche con eventuali disegni)*
b. *commentare l'eventuale procedura Pascal (dettagliando il significato delle variabili)*
c. *giustificare la correttezza e tutti i passaggi matematici*
d. *dimostrare la complessità (con equazioni di ricorrenza se necessario)*

1. Dato un albero binario di ricerca T di n nodi, in cui ogni nodo contiene un valore intero, lo si vuole modificare in modo che ogni nodo u contenga anche il numero di valori di T minori di quello contenuto in u. Si scriva una procedura Pascal di complessità ottima assumendo che l'albero sia *realizzato con puntatori* e che i valori interi memorizzati siano tutti distinti.

2. Data una lista $L = a_1, \ldots, a_n$ di n valori interi distinti, si vuole costruire una nuova lista $M = b_1, \ldots, b_n$ tale che ciascun elemento b_i sia uguale al numero di valori di L minori di a_i. Si scriva una procedura Pascal utilizzando gli operatori visti a lezione per le liste e se ne analizzi la complessità, assumendo che gli stessi n elementi siano memorizzati, oltre che in L, anche nell'albero binario di ricerca T dell'Esercizio 1.

3. Una sequenza $a_1, \ldots, a_n$ di n interi distinti è detta *unimodale* se esiste un indice m tale che $a_1 > \ldots > a_m < \ldots < a_n$ (cioè la sequenza è dapprima decrescente e poi crescente ed a_m è il minimo; se m=1 allora la sequenza è crescente, mentre se m=n allora è decrescente). Data una sequenza unimodale, si vuole trovare m. Si scriva una procedura di complessità O (log n) modificando opportunamente una ricerca binaria.

4. Si scriva la procedura HEAPSORT vista a lezione e la si esegua (a mano) per ordinare i 12 elementi: 8, 10, 2, 4, 7, 1, 12, 5, 3, 6, 11, 9. Si illustri con disegni, passo dopo passo, il contenuto dello heap durante l'esecuzione

5. Un dizionario è realizzato mediante una tabella hash con liste (bidirezionali) di trabocco. Si usi la funzione hash H(k) = k mod 7 per la k-esima lettera dell'alfabeto italiano e si assuma l'inserimento in coda alle liste. Si indichi il contenuto della tabella dopo avervi inserito, nell'ordine, le chiavi: C, H, U, R, C, H, T, U, R, I, N, G. Si indichi poi il contenuto della tabella dopo avervi cancellato, nell'ordine: R, I, C, E, e indi inserito, nell'ordine: H, I, L, B, E, R, T. Si discuta la complessità di tale realizzazione della struttura di dati.

6. Dato un grafo non orientato G=(N,A), un sottoinsieme S di nodi è *totalmente dominante* se ogni nodo u di N è adiacente ad almeno un nodo v di S tale che v sia diverso da u. Nel grafo dell'Esercizio 5, l'insieme {1, 5} è totalmente dominante? E l'insieme {2, 5}? E l'insieme {3, 4}? Dati G ed un intero k, si scriva un algoritmo *non deterministico* di complessità polinomiale per trovare un sottoinsieme totalmente dominante di al più k nodi.