

Laurea in “Informatica”
Corso di “Algoritmi e Strutture Dati”
16 Settembre 2008

1. Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti.
2. Sono ammessi al più 3 scritti consegnati per l'A.A. 2007/08 (Giugno 2008-Febbraio 2009)
3. Non è possibile consultare appunti, libri o persone, né uscire dall'aula.
4. 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, lo si vuole modificare in modo che ogni nodo u contenga anche il numero di valori di T maggiori 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 $L = a_1, \dots, a_n$ di n valori interi (non necessariamente distinti), si vuole costruire una nuova lista $M = b_1, \dots, b_n$ tale che ciascun elemento b_i sia uguale al numero di valori di L minori uguali ad a_i . Si scriva una procedura Pascal utilizzando gli operatori visti a lezione per le liste e se ne analizzi la complessità.

3. Siano dati n intervalli $I_1=[a_1,b_1], \dots, I_n=[a_n,b_n]$ della retta reale, tali che $a_1 < \dots < a_n$. La *distanza* tra due intervalli I_i e I_j con $i < j$ è uguale a 0, se i due intervalli hanno intersezione non vuota, mentre è uguale ad $a_j - b_i$, se i due intervalli non si intersecano. Si progetti una procedura Pascal efficiente per trovare la coppia di intervalli, che non si intersecano, più vicini tra loro.

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 $H(k) = k \bmod 7$ per la k -esima lettera dell'alfabeto italiano e si assuma l'inserimento in testa alle liste. Si indichi il contenuto della tabella dopo avervi inserito, nell'ordine, le chiavi: C, I, P, R, O, G, E, O, R, G, I, A. Si indichi poi il contenuto della tabella dopo avervi cancellato, nell'ordine: L, I, P, P, I, e indi inserito, nell'ordine: B, U, F, F, O, N. 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 . 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.