Laurea in "Informatica"
Corso di "Algoritmi e Strutture di Dati"
17 Settembre 2013

1. Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti.
2. Sono ammessi al più 3 scritti consegnati per l'A.A. 2012/13 (Giugno 2013-Febbraio 2014)
3. Non è possibile consultare appunti, libri o persone, né uscire dall'aula.
4. Le soluzioni degli esercizi devono spiegare a parole l'algoritmo usato (anche con l'aiuto di eventuali disegni); fornire e commentare lo pseudocodice (dettagliando a parole il significato delle variabili); giustificare la correttezza e la complessità (con tutti i passaggi matematici necessari)
5. Un esercizio può ammettere più soluzioni: a soluzioni computazionalmente più efficienti e/o concettualmente più semplici sono assegnati punteggi maggiori

1. Si scriva lo pseudo-codice di una procedura che riceva come parametri d'ingresso un vettore e gli indici dei suoi estremi sinistro e destro, tale che il suo tempo di esecuzione $T(n)$ verifichi la ricorrenza:

$$T(n) = c \qquad \text{per } n < 5$$
$$T(n) = 4T(n/2) + dn^2 \qquad \text{altrimenti}$$

con n numero di elementi del vettore e c e d costanti. Si calcoli anche l'ordine di grandezza di $T(n)$ risolvendo la ricorrenza.

2. Sia dato un albero binario T realizzato con puntatori, in cui ogni nodo contiene un valore intero. Scrivere lo pseudo-codice di una procedura di *complessità ottima* che modifichi T cancellando ogni foglia che ha un fratello e contiene un valore uguale a quello del fratello.

3. Data una lista L di interi, si vuole togliere da L tutti gli elementi ripetuti e inserirli in una nuova lista M, mantenendo in entrambe le liste l'ordine originario degli elementi (p.e. se L = 2, 1, 5, 4, 5, 2, 7, 4, 5, il risultato è L = 1, 7 e M = 2, 5, 4, 5, 2, 4, 5). Si scriva lo pseudo-codice di una procedura efficiente utilizzando gli *operatori* delle liste visti a lezione (ne esiste una che richiede tempo O(n) nel caso medio, con n lunghezza della lista L).

4. Si scriva l'algoritmo di Programmazione Dinamica visto a lezione per risolvere il problema dello String Matching Approssimato (date due stringhe P e T contenenti rispettivamente m ed n caratteri, tratti dallo stesso alfabeto, trovare una copia della stringa P nella stringa T avente il minor numero di errori dovuti a sostituzione, inserimento e cancellazione).

5. Si descriva la struttura di dati Merge-Find-Set (MFSET), con le operazioni tipicamente utilizzate, una sua realizzazione efficiente, e si illustri un algoritmo dove l'utilizzo di tale struttura è vantaggiosa.

6. Dato un grafo non orientato G=(N,A), un *insieme coprente* è un sottoinsieme S dei nodi in N tale che ciascun arco [i,j] in A ha almeno uno dei suoi nodi estremi i e j in S. Si scriva un algoritmo *non deterministico* di complessità polinomiale per trovare, dati G e k, un insieme coprente di al più k nodi.