Laurea in "Informatica"
Corso di "Algoritmi"
15 Febbraio 2011

1. *Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti.*
2. *Sono ammessi al più 3 scritti consegnati per l'A.A. 2009/10 (Giugno 2010-Febbraio 2011)*
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 eventuali disegni)*
   - *commentare l'eventuale procedura Pascal (dettagliando il significato delle variabili)*
   - *giustificare la correttezza e tutti i passaggi matematici*
   - *dimostrare la complessità (con equazioni di ricorrenza se necessario)*

1. Sia dato un albero binario T implementato con puntatori, in cui ogni nodo contiene un valore intero. Scrivere una funzione Pascal di *complessità ottima* che modifichi T cancellando ogni foglia che contenga un valore uguale a quello del padre.

2. Si scriva la procedura Pascal *Depth-First-Search (DFS)* vista a lezione. Si esegua la procedura DFS sul grafo *non orientato* $G = (N, A)$, $N = \{1, 2, 3, 4, 5\}$, $A = \{[1,2], [1,3], [1,5], [2,5], [3,2], [3,4]\}$ a partire dal nodo 3, assumendo che i vettori di adiacenza siano ordinati in modo *crescente* e mostrando il contenuto dei vettori di adiacenza.

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

4. Dato un vettore di n interi, si vuole trovare in tempo *ottimo* il valore massimo. Si progetti una procedura Pascal di tipo divide et impera che ad ogni ricorsione divida il vettore in tre parti (bilanciate), impostando e risolvendo l'equazione di ricorrenza della complessità (per semplicità si può assumere che n sia una potenza di tre).

5. In un grafo non orientato $G = (N, A)$, un *insieme indipendente* è un sottoinsieme S di nodi tale che per nessuna coppia di nodi i e j di S esiste un arco $[i, j]$ nel grafo G. Si scriva una procedura Pascal non deterministica di complessità polinomiale che, dati in input un intero k ed un grafo G rappresentato con *matrice di adiacenza* nodi-nodi, decide se esiste un insieme indipendente di cardinalità maggiore o uguale a k.

6. Si riconsideri la definizione di insieme indipendente dell'Esercizio 5. Trovare un insieme indipendente di massima cardinalità è un problema NP-arduo se G è un grafo qualsiasi, ma è risolubile in tempo polinomiale se G è un albero. Si dimostri che in un albero tutte le foglie fanno parte di un insieme indipendente S di massima cardinalità e poi, basandosi su questa proprietà, si progetti un algoritmo (deterministico) di complessità polinomiale per trovare S.