Laurea in "Informatica"
Corso di "Algoritmi e Strutture di Dati"
10 Luglio 2012

1. *Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti.*
2. *Sono ammessi al più 3 scritti consegnati per l'A.A. 2011/12 (Giugno 2012-Febbraio 2013)*
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 calcoli la complessità $T(n)$ della seguente procedura ricorsiva, specificando se tale complessità è polinomiale o esponenziale nella *dimensione* dell'input:

```
integer mistery(integer [ ] A, integer n)
    integer k ← 4
    for integer i ← n downto n – 2 do
        ⌊ k ← k
    if n < 13 then
        | return ⌊n/k⌋
    else
        ⌊ return ⌊n/k⌋ · mistery(A, ⌊n/4⌋) + mistery(A, ⌊n/k⌋) + mistery(A, k)
```

2. Dato un vettore A di n interi tale che $A[1] < A[n]$, si vuole trovare un indice i per cui $A[i] < A[i + 1]$. Si dimostri che (almeno) un indice i in cui questo avviene esiste sempre e si scriva lo pseudo-codice di un algoritmo di complessità $O(\log n)$ che restituisca un tale indice.

3. Dati due numeri interi x ed y, si definisce la *distanza* tra x ed y come $|x - y|$. Sia T un albero binario di ricerca di n nodi le cui chiavi sono numeri interi. Si scriva lo pseudo-codice di un algoritmo di complessità $O(n)$ per determinare due chiavi di T aventi distanza minima.

4. In un grafo *non orientato* e *connesso* G, la *distanza* tra due nodi s e v è il numero di archi di un cammino tra s e v che contiene il minor numero di archi. Dati G ed un suo nodo s, si vuole calcolare la distanza media tra s e tutti gli altri nodi del grafo (quindi escluso s). Scrivere lo pseudo-codice di un algoritmo che richieda complessità $O(n + m)$, dove n è il numero di nodi ed m il numero di archi di G.

5. Si descriva l'algoritmo di Kruskal visto a lezione per calcolare il minimo albero di copertura di un grafo non orientato pesato, dimostrandone correttezza e complessità.

6. Dato un insieme A di n interi distinti, si vuole decidere se esisteno tre sottoinsiemi *non vuoti* S, T e U tali che la cardinalità di S sia uguale a quella di T ed il prodotto degli elementi di T sia il doppio della somma degli elementi di U. Si scriva lo pseudo-codice di un algoritmo *non deterministico* che richieda tempo *polinomiale*.