Laurea in "Informatica"
Corso di "Algoritmi e Strutture di Dati"
12 Giugno 2012

1. *Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti e in tal caso verrà verbalizzato "ritirato". Un compito insufficiente verrà verbalizzato "respinto".*
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, assumendo $n > 0$, e si dica se tale complessità è polinomiale o esponenziale nella *dimensione* dell'input.

```
integer mistery(integer [ ] A, integer n)
    integer k ← n
    for integer i ← n downto ⌊n/2⌋ do
        ⌊ k ← ⌊i/2⌋
    if n < 11 then
        | return ⌊n/2⌋
    else
        ⌊ return 17 · mistery(A, ⌊n/4⌋) + mistery(A, k) + ⌊n/k⌋
```

2. Siano A e B due vettori rispettivamente di n ed m interi, entrambi ordinati in senso non decrescente, e sia k un intero. Scrivere lo pseudo-codice di un algoritmo efficiente che trovi, se esiste, una coppia di indici i e j tali che $A[i] + B[j] = k$ (nota: esiste un algoritmo di complessità $O(n+m)$).

3. Si scriva lo pseudo-codice di un algoritmo che, dati un albero binario T di n nodi ed un intero k, cancella in tempo *ottimo* ogni foglia di livello k, assumendo T realizzato con puntatori.

4. In un grafo orientato e *non pesato* $G = (N, A)$, la *distanza* tra due nodi u e v è il numero di archi in un cammino minimo tra u e v. Il *diametro* di G è la distanza tra la coppia di nodi u e v più lontani tra loro, ovvero il massimo tra i cammini minimi tra tutte le coppie di nodi. Scrivere lo pseudo-codice di un algoritmo efficiente che calcoli il diametro di G (nota: ne esiste uno di complessità $O(n^2+nm)$).

5. Si descriva l'algoritmo di Prim 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 esiste una *partizione* in tre sottoinsiemi *non vuoti* S, T e U tali che la cardinalità di S sia doppia di quella di T ed il prodotto degli elementi di S sia uguale alla somma degli elementi di U. Scrivere lo pseudo-codice di un algoritmo *non deterministico* che richieda tempo *polinomiale*.