

Laurea in “Informatica”

Corso di “Algoritmi e Strutture di Dati”

15 Gennaio 2013

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, con $n > 0$, specificando se tale complessità è polinomiale o esponenziale nella *dimensione* dell'input:

```
integer mystery(integer [ ] A, integer n)
    integer k ← 5
    for integer i ← n to n + 2 do
        | k ← k - 1
    if n > 5 then
        | return  $\lfloor n/k \rfloor + \text{mystery}(A, \lfloor n/k \rfloor) + \text{mystery}(A, \lfloor n/2 \rfloor) + \lfloor n/2 \rfloor$ 
    else
        | return n
```

2. Dato un vettore a_1, \dots, a_n di n interi, si vuole calcolarne la *sommatoria*. Si scriva lo pseudo-codice di un algoritmo di complessità *ottima* che utilizzi la tecnica *divide-et-impera* con *partizione bilanciata dei dati*.

3. Dato un albero binario B in cui ogni nodo è colorato verde, bianco o rosso, si vuole cambiare il colore dei nodi in modo che, considerando i nodi dell'albero per livelli crescenti e, a parità di livello, da sinistra verso destra, si incontrino prima tutti i nodi verdi, poi tutti i bianchi, e infine tutti i rossi. Scrivere lo pseudo-codice di un algoritmo di complessità *ottima* assumendo che B sia realizzato con *puntatori*.

4. Si scriva lo pseudo-codice della procedura *Depth-First Search (DFS)* vista a lezione e la si esegua sul grafo $G = (N, A)$, con $N = \{1, 2, 3, 4, 5, 6\}$ ed $A = \{(1,2), (3,1), (3,2), (3,6), (4,1), (5,3), (6,4), (6,5)\}$ a partire dal nodo 3, assumendo che gli insiemi di adiacenza dei nodi siano ordinati in senso crescente.

5. Si specifichi il contenuto di una tabella hash di dimensione 12, inizialmente vuota, dopo l'inserzione, nell'ordine, delle chiavi: C, O, M, P, I, T, O, T, R, E. Si usino la funzione hash $H(k) = 1 + k \bmod 12$ per la k -esima lettera dell'alfabeto inglese, il metodo di memorizzazione interno ed il metodo di scansione lineare con passo unitario (nota: l'alfabeto inglese è: ABCDEFGHIJKLMNOPQRSTUVWXYZ).

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