

Laurea in “Informatica”

Corso di “Algoritmi e Strutture di Dati”

24 Luglio 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 pseudocodice di una procedura ricorsiva che riceva come parametri di ingresso un vettore di interi e gli indici del primo e dell'ultimo elemento del vettore e tale che la sua complessità $T(n)$ verifichi la ricorrenza:

$$\begin{aligned} T(n) &= c, & \text{per } n < 13, \\ T(n) &= 4T(n/2) + dn, & \text{altrimenti,} \end{aligned}$$

dove c e d sono due costanti, ed n è il numero di elementi del vettore. Si risolva poi la ricorrenza, calcolando l'ordine di grandezza di $T(n)$.

2. Date due liste A e B (non ordinate), ciascuna delle quali contiene elementi interi distinti, si vuole determinare se esiste un elemento che compare in A ma non in B . Fornire un algoritmo efficiente, scriverne lo pseudocodice e discuterne la complessità (ne esiste uno che richiede tempo $O(n)$ nel caso medio, dove n è il numero di elementi della lista più lunga).

3. Si scriva lo pseudo-codice di una funzione che verifichi in tempo $O(n)$ se un albero binario T di n nodi è un albero di ricerca, usando gli operatori degli alberi binari.

4. Dato un vettore a_1, \dots, a_n di n interi, si vuole calcolarne la *produttoria* $a_1 a_2 \dots a_n$. Si scriva lo pseudo-codice di un algoritmo di complessità *ottima* che utilizzi la tecnica *divide-et-impera* con *partizione bilanciata dei dati*.

5. Un menu di un ristorante elenca n specialità culinarie, ognuna delle quali vale $c[i]$ calorie e costa $e[i]$ euro. Il cliente, essendo a dieta ed anche un po' parsimonioso, vuole trovare il sottoinsieme di specialità che permette di superare il fabbisogno minimo K di calorie con la minore spesa possibile.

- Posto $E(i; k)$ il costo minimo per ottenere almeno k calorie con un qualunque sottoinsieme delle prime i specialità, si scriva un'espressione ricorsiva per $E(i; k)$;
- Si progetti un algoritmo basato sulla programmazione dinamica e se ne valuti la complessità.

6. Si descriva l'algoritmo di approssimazione per il COMMESO VIAGGIATORE con disuguaglianze triangolari (Δ TSP) visto a lezione, dimostrandone la complessità e lo scarto dall'ottimo della soluzione trovata.