

**Laurea in “Informatica”**  
**Corso di “Algoritmi e Strutture di Dati”**  
**12 Luglio 2011**

1. Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti.
2. Sono ammessi al più 3 scritti consegnati per l'A.A. 2010/11 (Giugno 2011-Febbraio 2012)
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 più efficienti sono assegnati punteggi maggiori

1. Si scriva lo pseudocodice di una procedura ricorsiva PIPPO che prenda in input un vettore  $A$  di  $n$  elementi interi in modo che la sua complessità  $T(n)$  verifichi la relazione di ricorrenza  $T(n) = 4T(n/2) + cn$ , con  $c$  costante positiva. Si ricavi  $T(n)$  resolvendo la ricorrenza.

2. Date  $k$  liste ordinate, si vuole fonderle in un'unica lista ordinata. Fornire un algoritmo *efficiente*, scriverne lo pseudocodice e discuterne la complessità (nota: ne esiste uno che richiede tempo  $O(n \log k)$ , dove  $n$  è il numero totale di elementi di tutte le liste).

3. Sia  $A$  un vettore di  $n$  valori interi (non necessariamente distinti) e si consideri il problema di determinare se esistono o no due valori distinti che compaiono in  $A$  lo stesso numero di volte. Per esempio, se  $A = 3, 1, 3, 4, 1$ , allora la risposta è SI (perché 3 e 1 compaiono 2 volte), mentre se  $A = 3, 1, 3, 3, 1$ , allora la risposta è NO. Fornire un algoritmo *efficiente*, scriverne lo pseudocodice e discuterne la complessità (nota: ne esiste uno che richiede tempo  $O(n \log n)$ ).

4. Si scriva lo pseudocodice della procedura *Depth-First-Search (DFS)* vista a lezione. Si esegua la procedura *DFS* sul grafo *non orientato*  $G = (N, A)$ ,  $N = \{1, 2, 3, 4, 5\}$ ,  $A = \{[1,2], [1,3], [1,5], [2,3], [2,5], [4,5]\}$  a partire dal nodo 5, assumendo che gli insiemi di adiacenza siano ordinati in modo *crescente*, mostrando l'ordine di visita dei nodi e degli archi.

5. Dato un insieme  $A$  di  $n$  interi, si vuole decidere se esiste un sottoinsieme  $S$  di  $A$  tale che la somma degli elementi in  $S$  sia uguale al prodotto degli elementi in  $A - S$ . Scrivere un algoritmo non deterministico di complessità polinomiale.

6. A causa di un grave terremoto, nella regione colpita ci sono  $n$  persone ferite e  $k$  ospedali ancora funzionanti. Ogni persona deve essere ricoverata in un ospedale che non disti più di mezz'ora dal luogo di ferimento. Per evitare di distribuire i pazienti in maniera diseguale, ogni ospedale deve ricevere al più  $n/k$  pazienti. Formulare il problema come problema di flusso massimo su una opportuna rete di flusso e dire qual'è la complessità (in funzione di  $n$  e  $k$ ) dell'algoritmo di Ford-Fulkerson per trovare la soluzione ottima sulla rete così definita.