

# Esame di Algoritmi e Strutture Dati

## 17 Giugno 2002

Si prega di scrivere il proprio nome, cognome e numero di matricola in alto a destra su ogni foglio. Si prega inoltre di svolgere le parti di teoria e laboratorio su fogli separati.

### Teoria

1. Si proponga un algoritmo asintoticamente ottimo per ordinare i nodi di un grafo sulla base del loro grado. Si supponga che ogni nodo abbia un campo in cui è già riportato il grado.
2. Dato un RB-albero si proponga un algoritmo logaritmico per determinare il nodo mediano, cioè il nodo che nella linearizzazione dell'albero occupa la posizione mediana. Suggerimento: aggiungere un campo  $SUM(x)$  che contiene il numero di nodi del sottoalbero radicato in  $x$ .
3. Un evento atmosferico ha provocato la distruzione di tutti i ponti di Venezia. La ditta Pinco Palla s.r.l. ci ha chiesto di calcolare la lista di ponti da riparare secondo le seguenti specifiche:
  - (a) Non devono esistere due percorsi alternativi per andare da un punto  $x$  ad un punto  $y$ .
  - (b) Dalla stazione ferroviaria deve essere possibile raggiungere qualsiasi punto di Venezia nel più breve tempo possibile.

Si descriva formalmente il problema della ditta Pinco Palla e si proponga un algoritmo per risolvere il problema stesso.

### Laboratorio

Dato lo schema della seguente classe

```
public class Persona {  
    private String      nome, cognome;  
    private java.util.Date dataNascita;  
    private int         reddito;  
    ...  
}
```

1. Si fornisca la codifica in Java opportunamente commentata del o dei metodi necessari per poter eseguire il confronto fra oggetti di questa classe e che implementi il criterio di confronto lessicografico sul meta attributo *nome+cognome*; come si deve modificare la dichiarazione di classe? (max 7/30)
2. Si fornisca la codifica di un algoritmo di ordinamento che dato un array di object di tipo *Persona* restituisca l'array ordinato secondo il criterio di ordinamento naturale della classe. (max 7/30)
3. Si fornisca la codifica in Java opportunamente commentata delle classi esterne necessarie per implementare i seguenti criteri di confronto: *piùGiovane*, *minorReddito*. (max 8/30)
4. Si fornisca la codifica dell'algoritmo ShellSort che dato un array di object di tipo *Persona* restituisca l'array ordinato secondo il criterio di ordinamento passato come secondo parametro. (max 8/30)