

# Informatica Generale II - Prova teorica

A.A. 2005/2006

Esame: 8 settembre 2006

**Codice: MRSM**

1. Dato l'albero n-ario in figura 1:

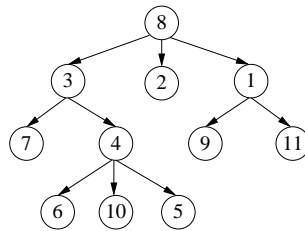


Figura 1:

- (a) La visita in postordine dell'albero produce in uscita la seguente sequenza: 7 6 10 5 2 9 11 4 1 3 8
  - (b) La visita in postordine dell'albero produce in uscita la seguente sequenza: 8 3 7 4 6 10 5 2 1 9 11
  - (c) La visita in postordine dell'albero produce in uscita la seguente sequenza: 7 6 10 5 4 3 2 9 11 1 8
  - (d) Non si può effettuare la visita in postordine di un albero non binario;
  - (e) non rispondo
2. Quali degli alberi in figura 2 *non sono* alberi binari di ricerca, mentre gli altri lo sono?

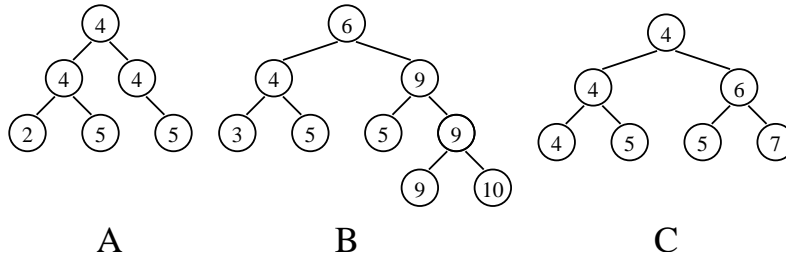


Figura 2:

- (a) A,C
  - (b) sono tutti alberi binari di ricerca.
  - (c) A,B
  - (d) A,B,C
  - (e) non rispondo
3. Il seguente frammento di codice:

```
struct data {  
    int giorno;  
    int mese;
```

```

int anno;
data(int g, int m, int a){ giorno=g; mese=m; anno=a; }
};

struct persona {
char* nome;
char* cognome;
data datanascita;
persona(char * n, char * c, data d_nasc) {
    nome=n; cognome=c; datanascita=d_nasc;
}
};

int main() {
    ...
    char n[]="Sergio";
    char c[]="Rossi";
    data datanasc(25,12,1988);
    persona io(n, c, datanasc);
    ...
}

```

- (a) produce un errore a tempo di compilazione. Infatti, all'interno della dichiarazione di `persona` si ha l'invocazione del costruttore standard senza argomenti di `data` che però è stato inibito automaticamente dal compilatore a fronte della dichiarazione del costruttore a tre argomenti;
- (b) produce un errore a tempo d'esecuzione perché l'istruzione `persona io(n, c, datanasc);` richiama il costruttore senza argomenti di `data` che non è presente all'interno della definizione di `data`;
- (c) è corretto;
- (d) produce un errore a tempo d'esecuzione. Infatti, all'interno della dichiarazione di `persona` si ha l'invocazione del costruttore standard senza argomenti di `data` che però è stato inibito automaticamente dal compilatore a fronte della dichiarazione del costruttore a tre argomenti;
- (e) non rispondo

4. Si consideri un algoritmo di visita di una grafo:

- (a) è preferibile non usare una *marca* che segni i nodi già visitati perché l'algoritmo potrebbe risultare appesantito da una tale operazione;
- (b) usare una *marca* che segni i nodi già visitati può essere utile per migliorare l'efficienza dell'algoritmo, ma non è necessario sempre;
- (c) occorre usare una *marca* che segni i nodi già visitati per essere sicuri che l'algoritmo non entri in un ciclo infinito;
- (d) nessuna delle risposte è accettabile;
- (e) non rispondo

5. Una pila è:

- (a) un multiinsieme di elementi gestiti secondo la politica *fifo* (first in first out);
- (b) un multiinsieme di elementi in cui ogni eliminazione ha per oggetto l'elemento inserito per primo;
- (c) un multiinsieme di elementi gestiti secondo la politica *lifo* (last in first out);
- (d) un insieme di elementi gestiti secondo la politica *lifo* (last in first out);
- (e) non rispondo

6. Si supponga di avere la sequenza di numeri 3,8,6,4,5. Dopo 3 iterazioni soltanto di Bubblesort su tale sequenza, il risultato sarà:

- (a) 3,4,8,5,6
- (b) 3,4,5,8,6
- (c) 3,4,8,6,5
- (d) 3,8,4,5,6
- (e) non rispondo

7. Si consideri l'algoritmo di fusione (merge) tra due array ordinati aventi  $n$  elementi ciascuno:
- (a) si implementa sempre tramite programmazione ricorsiva;
  - (b) si avvale di questo algoritmo l'ordinamento per fusione che funziona nel seguente modo: si sceglie a caso un elemento di pivot sulla base del cui valore si divide in due parti l'array, di seguito ciascuna parte viene ricorsivamente ordinata e infine i due array vengono fusi;
  - (c) ha complessità  $\mathcal{O}(n^2)$ ;
  - (d) ha complessità lineare poiché ogni volta che si inserisce un elemento nell'array risultato si esegue un numero costante di operazioni;
  - (e) non rispondo
8. Dato il grafo in figura 3:

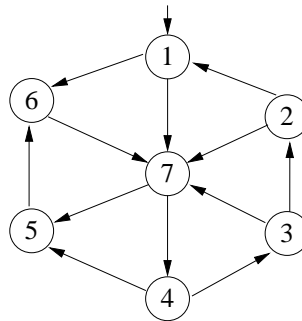


Figura 3:

- (a) La visita in profondità produce la seguente sequenza d'uscita: 1 6 7 5 3 4 2
  - (b) La visita in profondità produce la seguente sequenza d'uscita: 1 6 7 4 3 5 2
  - (c) Nessuna delle risposte è accettabile;
  - (d) La visita in profondità produce la seguente sequenza d'uscita: 1 7 4 3 2 5 6
  - (e) non rispondo
9. In quale ordine partendo dalla radice vengono visitati i nodi dell'albero in figura 4 da un algoritmo di attraversamento pre-order?

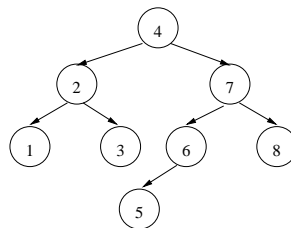


Figura 4: Albero

- (a) 12345678
  - (b) 85673124
  - (c) 42137658
  - (d) 87654321
  - (e) non rispondo
10. Il seguente codice:

```
int a[5];
++(*(&(* (a+3))) - 1));
```

- (a) ha l'effetto di incrementare di 1 il terzo elemento dell'array a
- (b) ha l'effetto di incrementare di 1 il quarto elemento dell'array a
- (c) è sbagliato perché a è un array e non un puntatore;
- (d) ha l'effetto di incrementare di 1 il secondo elemento dell'array a
- (e) non rispondo