

# Informatica Generale II - Prova teorica

A.A. 20052006

Esame: 25 gennaio 2007

**Codice: RZNU**

1. Dato il grafo in figura 1:

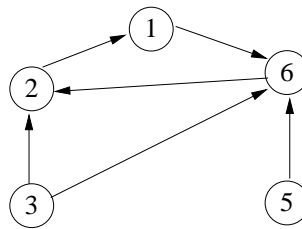


Figura 1:

- (a) Si tratta di un grafo fortemente connesso.
  - (b) Si tratta di un grafo completo.
  - (c) Nessuna delle risposte è accettabile;
  - (d) Si tratta di un grafo aciclico.
  - (e) non rispondo
2. Quale delle seguenti sequenze di complessità asintotiche rappresenta un ordinamento crescente dal più piccolo al più grande?
- (a)  $O(\log n)$   $O(n)$   $O(n^3)$   $O(n^n)$   $O(n^2 \log n)$   $O(2^n)$   $O(n^{\log n})$
  - (b)  $O(n)$   $O(n^3)$   $O(n^n)$   $O(\log n)$   $O(n^2 \log n)$   $O(2^n)$   $O(n^{\log n})$
  - (c)  $O(\log n)$   $O(n)$   $O(n^2 \log n)$   $O(n^3)$   $O(2^n)$   $O(n^{\log n})$   $O(n^n)$
  - (d)  $O(\log n)$   $O(n)$   $O(n^2 \log n)$   $O(n^3)$   $O(n^n)$   $O(2^n)$   $O(n^{\log n})$
  - (e) non rispondo
3. Si consideri il seguente frammento di codice:

```
struct Tipo1 {
    int a;

    Tipo1(int _a) { a = _a; }
};

struct Tipo2 {
    Tipo1 a;

    Tipo2(int _a) {
        a = Tipo1(_a);
    }
};
```

- (a) è errato, ma sarebbe corretto se esistesse anche il costruttore `Tipo1()` `{a = 0;}`.
- (b) è corretto e compilabile.

- (c) è errato perché Tipo1 e Tipo2 contengono due campi con lo stesso nome.
- (d) è errato perché Tipo2 non ha un costruttore di default.
- (e) non rispondo

4. Dato il grafo in figura 2:

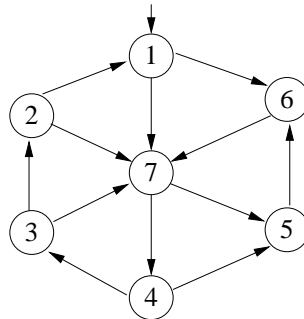


Figura 2:

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

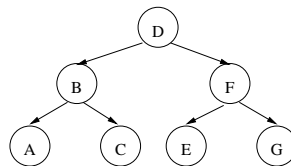


Figura 3: Albero

- (a) CABGEDD
  - (b) GFEDCBA
  - (c) ACBEGFD
  - (d) ABCDEFG
  - (e) non rispondo
6. Data la seguente dichiarazione:

```
double vett[5];
```

Le istruzioni `sizeof(vett)` e `sizeof(double)`

- (a) sono rispettivamente corretta ed errata perché l'operatore `sizeof()` non può ricevere come parametro il nome di un tipo
- (b) la dimensione del puntatore all'array e la dimensione in byte di un puntatore a una variabile di tipo `double`
- (c) restituiscono rispettivamente la dimensione dell'intero array e il numero di byte necessari a rappresentare una variabile di tipo `double`

- (d) restituiscono rispettivamente la dimensione del puntatore all'array e la dimensione di un singolo elemento dell'array
- (e) non rispondo

7. Supponendo di aver definito una funzione `funzEsame` con la seguente definizione:

```
void funzEsame (double i, double & j) {  
    j += i;  
}
```

L'invocazione *scorretta* della suddetta funzione `funzEsame` avrà la forma:

- (a) `double i=1.0, j=5.0;`  
`funzEsame(i, j+1);`
- (b) `double i=1.0, j=5.0;`  
`funzEsame(5.0, j);`
- (c) `double i=1.0, j=5.0;`  
`funzEsame(i+1, j);`
- (d) `double i=1.0, j=5.0;`  
`funzEsame(i, j);`
- (e) non rispondo

8. Quale è la lunghezza del cammino dell'albero in figura 4

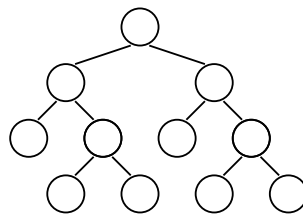


Figura 4:

- (a) 23
  - (b) 22
  - (c) 11
  - (d) 3
  - (e) non rispondo
9. 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) 3,4,8,6,5
  - (c) 3,8,4,5,6
  - (d) 3,4,5,8,6
  - (e) non rispondo
10. Le operazioni di base di stack e code sono:
- (a) push e pop per stack, put e get per code
  - (b) push e pop per code, put e get per stack
  - (c) push e pop
  - (d) put e get
  - (e) non rispondo