

Informatica Generale II - Prova teorica

A.A. 2005/2006

Esame: 19 aprile 2006

Codice: IIZC

1. Una pila è:

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

2. Si consideri il seguente frammento di codice:

```
struct Tipo1 {  
    Tipo2 a;  
};
```

```
struct Tipo2 {  
    Tipo1 a;  
};
```

- (a) è errato perché Tipo1 e Tipo2 contengono due campi con lo stesso nome.
- (b) è errato perché Tipo1 e Tipo2 non sono istanziabili.
- (c) è errato perché Tipo2 non ha un costruttore di default.
- (d) è corretto, ma Tipo2 non è istanziabile.
- (e) non rispondo

3. Quale è la lunghezza del cammino dell'albero in figura 1

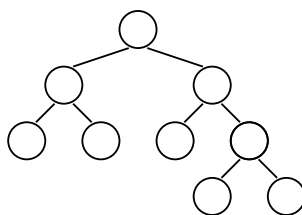


Figura 1:

- (a) 3
 - (b) 17
 - (c) 16
 - (d) 9
 - (e) non rispondo
4. Si supponga di avere la sequenza di numeri 6,11,9,7,8. Dopo 3 iterazioni soltanto di Bubblesort su tale sequenza, il risultato sarà:
- (a) 6,11,7,8,9

- (b) 6,7,11,8,9
- (c) 6,7,11,9,8
- (d) 6,7,8,11,9
- (e) non rispondo

5. Supponendo di aver definito una struttura Punto e funzione funzEsame come di seguito:

```
struct Punto {
    int x, y;
    Punto() { x = 0; y = 0; }
    Punto(int _x, int _y) { x = _x; y = _y; }
};

void funzEsame (Punto *i, double *j);
```

L'invocazione della suddetta funzione funzEsame avrà la forma:

- (a) Punto x(1,0);
double y = 0.0;
funzEsame(&x, &y);
- (b) Punto x(1,0);
double y = 0.0;
funzEsame(x, y);
- (c) Punto x(1,0);
double y = 0.0;
funzEsame(*x, *y);
- (d) Punto x(1,0);
double y = 0.0;
funzEsame();
- (e) non rispondo

6. Qual'è l'altezza dell'albero rappresentato in figura 2?

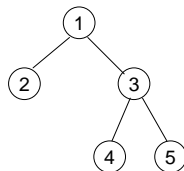


Figura 2: Albero

- (a) 2
- (b) 5
- (c) 1
- (d) 3
- (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(i, j);`
- (c) `double i=1.0, j=5.0;`
`funzEsame(5.0, j);`
- (d) `double i=1.0, j=5.0;`
`funzEsame(i+1, j);`
- (e) non rispondo

8. Quale tipo di attraversamento d'albero implementa il seguente codice?

```
void foo-order(Node* l, void visit(Node*))
{
    StackPtr s = new Stack();

    Push(s, l);
    while(! StackIsEmpty(s)) {
        Node * h = Pop(s);

        visit(h);
        if (h->right != NULL) Push(s, h->right);
        if (h->left != NULL) Push(s, h->left);
    }
    delete s;
}
```

- (a) level-order
- (b) postorder
- (c) preorder
- (d) inorder
- (e) non rispondo

9. In quale ordine partendo dalla radice vengono visitati i nodi dell'albero in figura 3 da un algoritmo di attraversamento post-order?

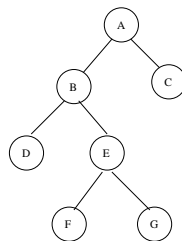


Figura 3: Albero

- (a) DBFEGCA
- (b) GEFDBCA
- (c) GFEDBCA
- (d) DFGEBCA
- (e) non rispondo

10. Sia dato un albero binario di ricerca con n nodi e si consideri un algoritmo di ricerca su esso operante:

- (a) il caso peggiore è dato dalla circostanza in cui si abbia un albero completamente bilanciato; in esso l'altezza è $\mathcal{O}(\log n)$ e da ciò segue che la ricerca in questo caso ha complessità $\mathcal{O}(n * \log n)$
- (b) il caso migliore è dato dalla circostanza in cui si abbia un albero completamente bilanciato; in esso l'altezza è $\mathcal{O}(n * \log n)$ e da ciò segue che la ricerca in questo caso ha complessità $\mathcal{O}(n * \log n)$
- (c) il caso migliore è dato dalla circostanza in cui si abbia un albero completamente bilanciato; in esso l'altezza è $\mathcal{O}(\log n)$ e da ciò segue che la ricerca in questo caso ha complessità $\mathcal{O}(\log n)$

- (d) il caso peggiore è dato dalla circostanza in cui si abbia un albero completamente bilanciato; in esso l'altezza è $\mathcal{O}(\log n)$ e da ciò segue che la ricerca in questo caso ha complessità $\mathcal{O}(\log n)$
- (e) non rispondo