

Informatica Generale II - Prova teorica

A.A. 2005/2006

Esame: 19 aprile 2006

Codice: XKZV

1. 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);

        if (h->right != NULL) Push(s, h->right);
        if ((h->left == NULL) && (h->right == NULL)) visit(h);
        else Push(s, new Node(h->data, true));
        if (h->left != NULL) Push(s, h->left);
        if (h->flag == true) delete h;
    }
    delete s;
}
```

- (a) inorder
(b) level-order
(c) preorder
(d) postorder
(e) non rispondo
2. Qual'è l'altezza dell'albero rappresentato in figura 1?

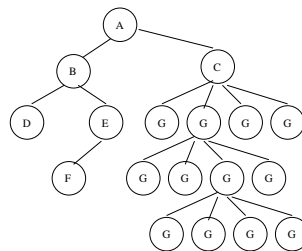


Figura 1: Albero

- (a) nessuno degli altri valori
(b) 3
(c) 4
(d) 5
(e) non rispondo
3. Si consideri l'algoritmo Mergesort, quale di queste affermazioni è falsa?
- (a) è un esempio tipico di algoritmo *divide et impera*

- (b) generalmente è più efficiente di Bubblesort
 - (c) deve essere sempre implementato mediante programmazione ricorsiva
 - (d) il limite asintotico superiore è $\mathcal{O}(N \log N)$
 - (e) non rispondo
4. Quali degli alberi in figura 2 *non sono* alberi binari di ricerca, mentre gli altri lo sono?

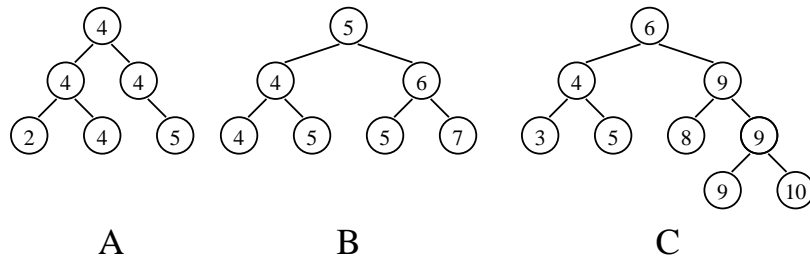


Figura 2:

- (a) A
 - (b) nessuno
 - (c) A,C
 - (d) B
 - (e) non rispondo
5. Si consideri il seguente frammento di codice:

```
struct Tipo1 {
    int a;

    Tipo1() { a = 0; }
    Tipo1(int _a) { a = _a; }
};

struct Tipo2 {
    Tipo1 a;

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

- (a) è errato perché Tipo2 non ha un costruttore di default.
 - (b) è errato perché Tipo1 e Tipo2 contengono due campi con lo stesso nome.
 - (c) è corretto e compilabile.
 - (d) è corretto, ma Tipo2 non è istanziabile.
 - (e) non rispondo
6. La seguente funzione foo:

```
void foo(Node * x, Node * y) {
    y->next = x->next;
    x->next = y;
}
```

- (a) inserisce la lista puntata da x dopo il nodo y
- (b) inserisce il nodo y tra il nodo x e il successore di x
- (c) inserisce la lista puntata da y dopo il nodo x

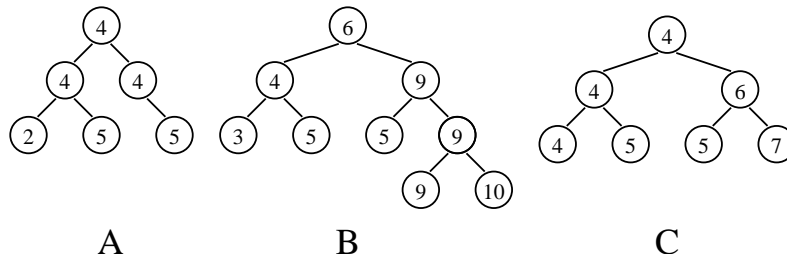


Figura 3:

- (d) concatena due liste concatenate x e y
 (e) non rispondo
7. Quali degli alberi in figura 3 hanno un livello maggiore di 15?
- (a) B
 (b) C
 (c) A
 (d) B,C
 (e) non rispondo
8. 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+1, j);`  
 (c) `double i=1.0, j=5.0; funzEsame(i, j);`  
 (d) `double i=1.0, j=5.0; funzEsame(5.0, j);`  
 (e) non rispondo
9. Dato l'albero n-ario in figura 4:

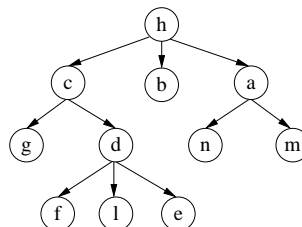


Figura 4:

- (a) La visita in postordine dell'albero produce in uscita la seguente sequenza: g f l e b n m d a c h  
 (b) Non si può effettuare la visita in postordine di un albero non binario;  
 (c) La visita in postordine dell'albero produce in uscita la seguente sequenza: h c g d f l e b a n m

- (d) La visita in postordine dell'albero produce in uscita la seguente sequenza: g f l e d c b n m a h
- (e) non rispondo

10. Si consideri l'algoritmo di fusione (merge) tra due array ordinati aventi  $n$  elementi ciascuno:

- (a) si implementa sempre tramite programmazione ricorsiva;
- (b) ha complessità lineare poiché ogni volta che si inserisce un elemento nell'array risultato si esegue un numero costante di operazioni;
- (c) 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;
- (d) ha complessità  $\mathcal{O}(n^2)$ ;
- (e) non rispondo