

Informatica Generale II - Prova teorica

A.A. 20052006

Esame: 27 aprile 2007

Codice: HXRE

1. In quale ordine partendo dalla radice vengono visitati i nodi dell'albero in figura 1 da un algoritmo di attraversamento pre-order?

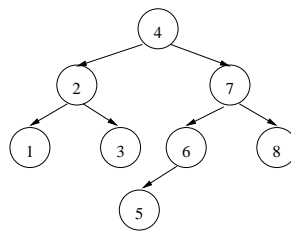
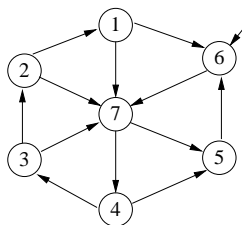


Figura 1: Albero

- (a) 12345678
- (b) 85673124
- (c) 87654321
- (d) 42137658
- (e) non rispondo

2. Dato il grafo e la sua rappresentazione con matrice delle adiacenze riportati in figura 2:



	1	2	3	4	5	6	7
1	0	0	0	0	0	1	1
2	1	0	0	0	0	0	1
3	0	1	0	0	0	0	1
4	0	0	1	0	1	0	0
5	0	0	0	0	0	1	0
6	0	0	0	0	0	0	1
7	0	0	0	1	1	0	0

Figura 2:

- (a) La visita in ampiezza dal vertice 6 produce la seguente sequenza d'uscita: 6 1 5 7 4 2 3
 - (b) La visita in ampiezza dal vertice 6 produce la seguente sequenza d'uscita: 6 7 4 5 3 2 1
 - (c) Nessuna delle risposte è accettabile;
 - (d) La visita in ampiezza dal vertice 6 produce la seguente sequenza d'uscita: 6 7 4 3 2 1 5
 - (e) non rispondo
3. Quale tipo di attraversamento d'albero implementa il seguente codice?

```
void foo-order(Node* l, void visit(Node*))  
{  
    Coda q;
```

```

put(q, l);
while (coda_piena(q)) {
    l = get(q);
    visit(l);
    if (l->left != NULL) put(q, l->left);
    if (l->right != NULL) put(q, l->right);
}
}

```

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

4. Si consideri il seguente codice:

```

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

```

La seguente dichiarazione:

```
data oggi;
```

- (a) produce un errore a tempo di compilazione in quanto l'allocazione di una variabile avente come tipo una struttura definita dal programmatore può essere allocata solo dinamicamente tramite l'operatore new;
 - (b) produce un errore a tempo di compilazione. Infatti, la suddetta dichiarazione risulta nell'invocazione del costruttore standard senza argomenti che però è stato inibito automaticamente dal compilatore a fronte della dichiarazione del costruttore a tre argomenti;
 - (c) è corretta
 - (d) produce un errore a tempo d'esecuzione. Infatti, la suddetta dichiarazione risulta nell'invocazione del costruttore standard senza argomenti che però è stato inibito automaticamente dal compilatore a fronte della dichiarazione del costruttore a tre argomenti;
 - (e) non rispondo
5. Si supponga di avere la sequenza di numeri 1,6,4,2,3. Dopo 3 iterazioni soltanto di Bubblesort su tale sequenza, il risultato sarà:
- (a) 1,6,2,3,4
 - (b) 1,2,6,4,3
 - (c) 1,2,6,3,4
 - (d) 1,2,3,6,4
 - (e) non rispondo
6. Si consideri un grafo: sia n il numero di nodi e m il numero di archi. Quale tra le seguenti affermazioni è falsa?
- (a) l'algoritmo DFS richiede tempo proporzionale a $(n + m)$;
 - (b) l'algoritmo BFS si ottiene dall'algoritmo DFS iterativo sostituendo lo Stack con una Coda;
 - (c) l'algoritmo BFS richiede tempo proporzionale a $(n + m)$
 - (d) l'algoritmo BFS si ottiene dall'algoritmo DFS iterativo sostituendo la Coda con una Stack;
 - (e) non rispondo

7. La seguente funzione `f00`:

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

- (a) inserisce il nodo y tra il nodo x e il successore di x
 - (b) concatena due liste concatenate x e y
 - (c) inserisce la lista puntata da y dopo il nodo x
 - (d) inserisce la lista puntata da x dopo il nodo y
 - (e) non rispondo
8. 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) ha complessità $\mathcal{O}(n^2)$;
 - (d) 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;
 - (e) non rispondo
9. Si considerino i seguenti due frammenti di codice:

```
const int dim=10;
...
int vett[dim];

int i;
...
i=10;
...
int vett [i];
```

- (a) Il primo frammento è corretto mentre il secondo produce un errore a tempo di compilazione poiché la dimensione dell'array deve essere rappresentata da una costante;
 - (b) I due frammenti sono equivalenti quanto a risultato poiché , sebbene il primo frammento rappresenti una allocazione statica di memoria per l'array, mentre il secondo una allocazione dinamica di memoria per l'array, lo spazio allocato si trova in entrambi i casi all'interno dello stack;
 - (c) I due frammenti non sono equivalenti in quanto rappresentano rispettivamente una allocazione statica di memoria all'interno dello stack e una allocazione dinamica all'interno dello heap;
 - (d) I due frammenti di codice sono esattamente equivalenti e hanno come risultato la allocazione statica di un vettore di 10 elementi;
 - (e) non rispondo
10. Quale delle seguenti frasi caratterizza meglio l'albero di figura 3:

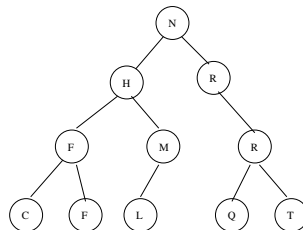


Figura 3: Albero

- (a) è un albero binario di ricerca bilanciato
- (b) nessuna delle altre risposte proposte è accettabile
- (c) è un albero binario di ricerca
- (d) è un albero binario
- (e) non rispondo