

Informatica Generale II - Prova teorica

A.A. 2005/2006

Esame: 23 giugno 2006

Codice: PDJQ

1. Si consideri un algoritmo di visita di un 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) occorre usare una *marca* che segni i nodi già visitati per essere sicuri che l'algoritmo non entri in un ciclo infinito;
 - (c) usare una *marca* che segni i nodi già visitati può essere utile per migliorare l'efficienza dell'algoritmo, ma non è necessario sempre;
 - (d) nessuna delle risposte è accettabile;
 - (e) non rispondo
2. Quale tra le seguenti affermazioni relative alla ricerca binaria è *falsa*?
 - (a) essendo n il numero di elementi nell'array, nel caso peggiore il numero di confronti da effettuare è il più piccolo m che soddisfa la relazione seguente: $2^m \geq n$
 - (b) la complessità della ricerca binaria non dipende dal particolare elemento da individuare;
 - (c) la complessità della ricerca binaria è $\mathcal{O}(\log n)$
 - (d) essendo n il numero di elementi nell'array, dopo j confronti la dimensione dell'array su cui si deve continuare la ricerca è $n/(2^j)$;
 - (e) non rispondo
3. Si supponga di avere una funzione `visit` di visita di un albero che inserisce il valore del nodo visitato in fondo ad una lista. Per ottenere una lista ordinata in senso crescente dall'attraversamento di un BST utilizzando la funzione di visita `visit`:
 - (a) bisogna attraversare il BST in preorder;
 - (b) nessuna risposta tra quelle presenti è accettabile;
 - (c) non è possibile ottenere una lista ordinata in senso crescente;
 - (d) bisogna attraversare il BST in postorder;
 - (e) non rispondo
4. Un array allocato dinamicamente
 - (a) va sempre deallocato esplicitamente;
 - (b) è sempre localizzato in un'area di memoria detta heap;
 - (c) è sempre localizzato in un'area di memoria detta stack;
 - (d) può essere localizzato nell'area di stack, oppure nell'area di heap;
 - (e) non rispondo
5. Quali degli alberi in figura 1 sono alberi binari di ricerca, mentre gli altri non lo sono?
 - (a) A,B,C
 - (b) nessuno è un albero binario di ricerca.
 - (c) C
 - (d) A,B
 - (e) non rispondo
6. Dato il grafo in figura 2:

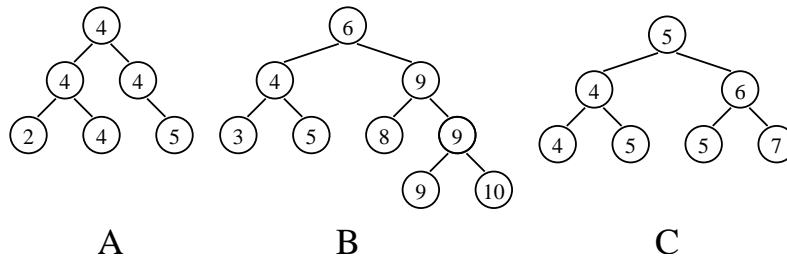


Figura 1:

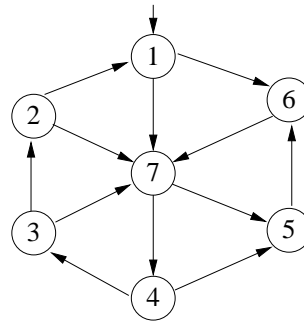


Figura 2:

- (a) Nessuna delle risposte è accettabile;
- (b) La visita in profondità produce la seguente sequenza d'uscita: 1 7 4 3 2 5 6
- (c) La visita in profondità produce la seguente sequenza d'uscita: 1 6 7 4 3 5 2
- (d) La visita in profondità produce la seguente sequenza d'uscita: 1 6 7 5 3 4 2
- (e) non rispondo

7. 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 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`;
- (b) 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;
- (c) è corretto;
- (d) 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;
- (e) non rispondo

8. Dato l'albero n-ario in figura 3:

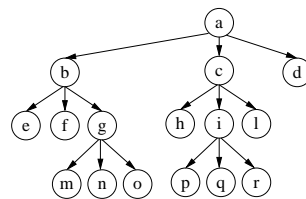


Figura 3:

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

9. Una pila è:

- (a) un insieme di elementi gestiti secondo la politica *lifo* (last 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 *fifo* (first in first out);
- (d) un multiinsieme di elementi gestiti secondo la politica *lifo* (last in first out);
- (e) non rispondo

10. Si consideri l'algoritmo Mergesort, quale di queste affermazioni è *falsa*?

- (a) il limite asintotico superiore è $\mathcal{O}(N \log N)$
- (b) deve essere sempre implementato mediante programmazione ricorsiva
- (c) generalmente è più efficiente di Bubblesort
- (d) è un esempio tipico di algoritmo *divide et impera*
- (e) non rispondo