Informatica Generale II - Prova teorica

A.A. 2005/2006

Esame: 19 aprile 2006
Codice: UXXU

1. La seguente funzione `foo`:

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

   (a) concatena due liste concatenate x e 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
   (d) inserisce la lista puntata da x dopo il nodo y
   (e) non rispondo

2. Un albero binario è:

   (a) un albero con cammino minimo 2 dalla radice alle foglie;
   (b) un albero i cui nodi hanno al minimo 2 figli;
   (c) nessuna delle altre risposte proposte è accettabile;
   (d) un albero i cui nodi hanno sempre 2 figli;
   (e) non rispondo

3. Quali degli alberi in figura 1 *sono* alberi binari di ricerca, mentre gli altri non lo sono?

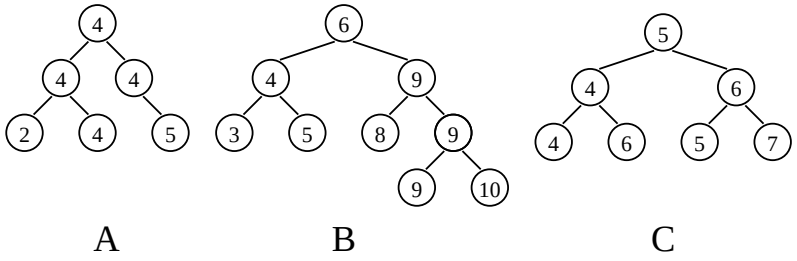

Figura 1:

   (a) nessuno è un albero binario di ricerca.
   (b) B,C
   (c) A,B
   (d) A, B, C
   (e) non rispondo

4. L'algoritmo selection sort ha complessità asintotica:

   (a) $\mathcal{O}(N^{2/3})$
   (b) $\mathcal{O}(N^2)$
   (c) $\mathcal{O}(NlogN)$

(d) nessuna delle altre risposte proposte è corretta
(e) non rispondo

5. In quale ordine partendo dalla radice vengono visitati i nodi dell'albero in figura 2 da un algoritmo di attraversamento pre-order?

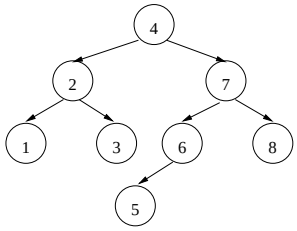

Figura 2: Albero

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

6. Il seguente frammento di codice:

```
{
 double i=1.0;
 double *j;
 j=&i;
 *j=10.0;
 ...
}
```

(a) è corretto
(b) è errato perché `j` punta a una zona di memoria arbitraria
(c) è errato perché si tenta di assegnare ad un puntatore un valore intero
(d) è errato perché l'operatore di dereferenziamento non può comparire a sinistra dell'operatore d'assegnamento
(e) non rispondo

7. Si consideri il limite superiore asintottico $\mathcal{O}(f(n))$ alla complessità di un problema. Quale tra le seguenti affermazioni è *falsa*:

(a) esiste almeno un algoritmo di soluzione avente complessità $\mathcal{O}(f(n))$;
(b) La notazione $\mathcal{O}(f(n))$ fornisce una valutazione approssimata per eccesso della complessità di un algoritmo;
(c) un algoritmo ha complessità $\mathcal{O}(f(n))$ se, per ogni *n* e per qualche input di dimensione *n*, l'algoritmo impiega una quantità di risposte proporzionali a $f(n)$;
(d) esistono tre opportune costanti a, b, c tali che il numero di istruzioni $t(n)$ che vengono eseguite nel caso peggiore con input di dimensione n verifica per ogni $n > c$ la seguente relazione: $t(n) < a * f(n) + b$;
(e) non rispondo

8. 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) è corretta
(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) 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`;
(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

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

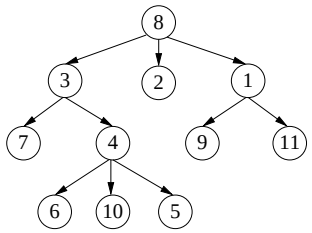

Figura 3:

(a) La visita in preordine dell'albero produce in uscita la seguente sequenza: 8 3 7 4 6 10 5 2 1 9 11
(b) La visita in preordine dell'albero produce in uscita la seguente sequenza: 7 6 10 5 2 9 11 4 1 3 8
(c) La visita in preordine dell'albero produce in uscita la seguente sequenza: 7 6 10 5 4 3 2 9 11 1 8
(d) Non si può effettuare la visita in preordine di un albero non binario;
(e) non rispondo

10. Il seguente codice:

```
int a[5];
++(*((&(*(a+3))) - 1));
```

(a) ha l'effetto di incrementare di 1 il quarto elemento dell'array `a`
(b) ha l'effetto di incrementare di 1 il terzo elemento dell'array `a`
(c) ha l'effetto di incrementare di 1 il secondo elemento dell'array `a`
(d) è sbagliato perché `a` è un array e non un puntatore;
(e) non rispondo