Informatica Generale II - Prova teorica

A.A. 20052006

Esame: 25 gennaio 2007
Codice: PMNA

1. Il *frame* (o record di attivazione) di una funzione:

   (a) viene a volte deallocato automaticamente;
   (b) può essere localizzato nell'area di stack, oppure nell'area di heap;
   (c) a seconda dei casi, è necessario deallocarlo mediante l'operatore `delete`;
   (d) è sempre localizzato in un area di memoria detta stack;
   (e) non rispondo

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

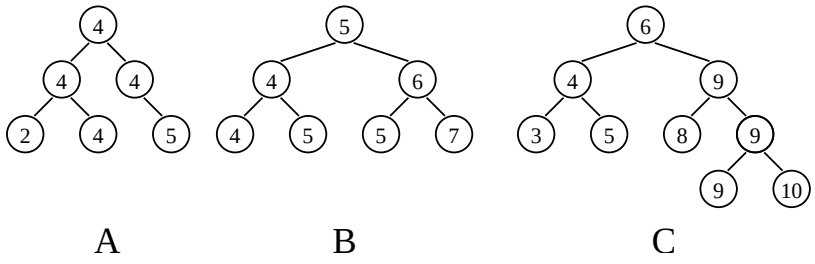

Figura 1:

   (a) A
   (b) nessuno
   (c) B
   (d) A,C
   (e) non rispondo

3. 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 corretto, ma Tipo2 non è istanziabile.
(c) è corretto e compilabile.
(d) è errato perché Tipo1 e Tipo2 contengono due campi con lo stesso nome.
(e) non rispondo

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

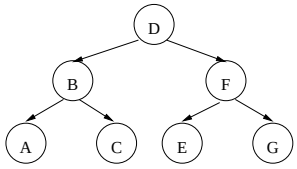

Figura 2: Albero

(a) DBFACEG
(b) ABCDEFG
(c) GFEDCBA
(d) CABGEGD
(e) non rispondo

5. 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 la lista puntata da y dopo il nodo x
(c) inserisce il nodo y tra il nodo x e il successore di x
(d) concatena due liste concatenate x e y
(e) non rispondo

6. Si consideri un algoritmo di visita di una grafo:

(a) occorre usare una *marca* che segni i nodi già visitati per essere sicuri che l'algoritmo non entri in un ciclo infinito;
(b) nessuna delle risposte è accettabile;
(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) è preferibile non usare una *marca* che segni i nodi già visitati perché l'algoritmo potrebbe risultare appesantito da una tale operazione;
(e) non rispondo

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

(a) 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 $\mathcal{O}(n * log\ n)$
(b) 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 $\mathcal{O}(log\ n)$
(c) 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 $\mathcal{O}(n * log\ n)$
(d) 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 $\mathcal{O}(log\ n)$
(e) non rispondo

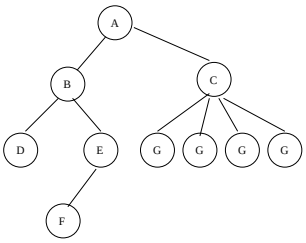

Figura 3: Albero

8. Qual'è l'altezza dell'albero rappresentato in figura 3?

   (a) 1
   (b) 4
   (c) 3
   (d) 2
   (e) non rispondo

9. Si analizzi il seguente codice:

```
void foo(int A[], int N) {
  for (int i=0; i < N-1; ++i) {
    int min = i;
    for (int j=i+1; j < N; ++j)
      if (A[j] < A[min]) min = j;
    swap(A[i], A[min]);
  }
}
```

   Esso implementa l'algoritmo:

   (a) Bubblesort
   (b) Quicksort
   (c) Selection sort
   (d) Mergesort
   (e) non rispondo

10. Dato il grafo in figura 4, quale delle seguenti visite non puó essere il risultato di una visita in depth-first (DFS):

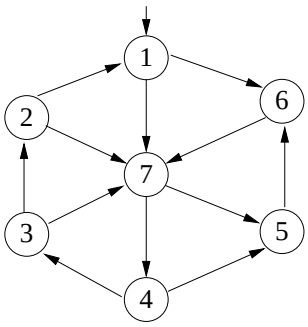

Figura 4:

   (a) 1764532
   (b) 1745632

(c) 1743256
(d) 1675432
(e) non rispondo