Informatica Generale II - Prova teorica

A.A. 20052006

Esame: 27 aprile 2007
Codice: MXSF

1. Quale delle seguenti affermazioni è *vera*

   (a) Un albero ordinato è un albero con radice ed è specificato l'ordine dei figli della radice.
   (b) Un albero ordinato è un albero con radice ed è specificato l'ordine dei figli di ciascun nodo.
   (c) Un albero ordinato è un albero con radice ed è sepcificato l'ordine dei figli di qualche nodo.
   (d) Un albero ordinato è un albero dove è specificato l'ordine dei figli di qualche nodo.
   (e) non rispondo

2. Si consideri un algoritmo di visita di una 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) usare una *marca* che segni i nodi già visitati può essere utile per migliorare l'efficienza dell'algoritmo, ma non è necessario sempre;
   (c) occorre usare una *marca* che segni i nodi già visitati per essere sicuri che l'algoritmo non entri in un ciclo infinito;
   (d) nessuna delle risposte è accettabile;
   (e) non rispondo

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

   (a) 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 $\mathcal{O}(log\ n)$
   (b) 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 $\mathcal{O}(log\ n)$
   (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 $\mathcal{O}(n * log\ n)$ e da ciò segue che la ricerca in questo caso ha complessità $\mathcal{O}(n * log\ n)$
   (e) non rispondo

4. Le operazioni di base di stack e code sono:

   (a) push e pop
   (b) push e pop per stack, put e get per code
   (c) push e pop per code, put e get per stack
   (d) put e get
   (e) non rispondo

5. L'algoritmo selection sort ha complessità asintotica:

   (a) $\mathcal{O}(NlogN)$
   (b) $\mathcal{O}(N^{2/3})$
   (c) nessuna delle altre risposte proposte è corretta
   (d) $\mathcal{O}(N^2)$
   (e) non rispondo

6. 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) è corretto;

(b) 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`;

(c) 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;

(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

7. Dato il grafo in figura 1:

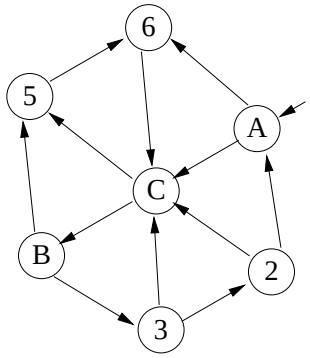

Figura 1:

(a) Nessuna delle risposte è accettabile;

(b) La visita in profondità produce la seguente sequenza d'uscita: A 6 C B 3 5 2

(c) La visita in profondità produce la seguente sequenza d'uscita: A C B 3 2 5 6

(d) La visita in profondità produce la seguente sequenza d'uscita: A 6 C 5 3 B 2

(e) non rispondo

8. Supponendo di aver definito una funzione `funzEsame` con la seguente intestazione:

```
void funzEsame (int *i, int *j);
```

L'invocazione della suddetta funzione `funzEsame` avrà la forma:

(a)
```
int i=10, j=5;
funzEsame(*i, *j);
```
(b)
```
int i=10, j=5;
funzEsame(i, j);
```
(c)
```
int i=10, j=5;
funzEsame(&i, &j);
```
(d)
```
int i=10, j=5;
funzEsame();
```
(e) non rispondo

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

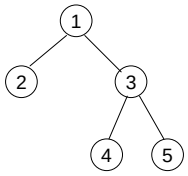

Figura 2: Albero

(a) 24531
(b) 12345
(c) 43512
(d) 21435
(e) non rispondo

10. Quali degli alberi in figura 3 hanno una lughezza del cammino maggiore di 15?

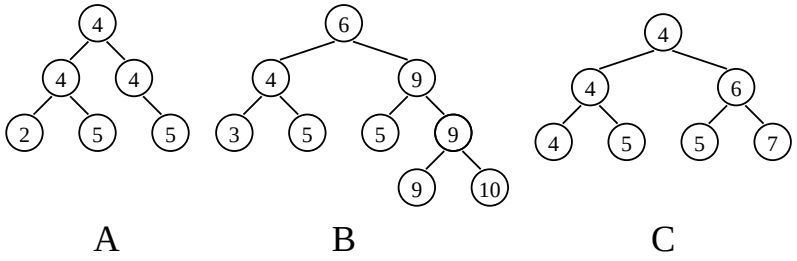

Figura 3:

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