

# Informatica Generale II - Prova teorica

A.A. 2005/2006

Esame: 19 aprile 2006

**Codice: EMFC**

1. Quale delle seguenti affermazioni è vera

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

2. L'algoritmo selection sort ha complessità asintotica:

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

3. Quali degli alberi in figura 1 hanno un livello minore di 15?

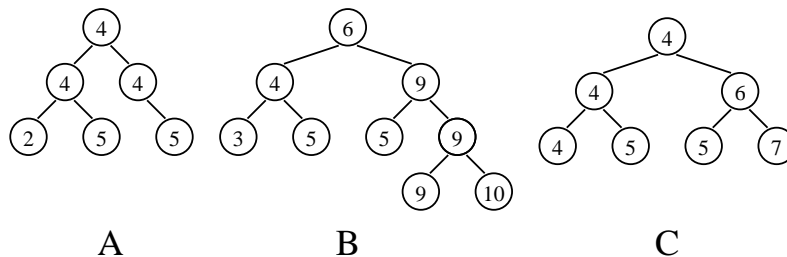


Figura 1:

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

4. La seguente funzione foo:

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

- (a) inserisce la lista puntata da y dopo il nodo x
- (b) concatena due liste concatenate x e y
- (c) inserisce la lista puntata da x dopo il nodo y

- (d) inserisce il nodo  $y$  tra il nodo  $x$  e il successore di  $x$   
 (e) non rispondo
5. Si consideri l'algoritmo di fusione (merge) tra due array ordinati aventi  $n$  elementi ciascuno:
- (a) si implementa sempre tramite programmazione ricorsiva;
  - (b) 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;
  - (c) ha complessità  $\mathcal{O}(n^2)$ ;
  - (d) ha complessità lineare poiché ogni volta che si inserisce un elemento nell'array risultato si esegue un numero costante di operazioni;
  - (e) non rispondo
6. In quale ordine partendo dalla radice vengono visitati i nodi dell'albero in figura 2 da un algoritmo di attraversamento post-order?

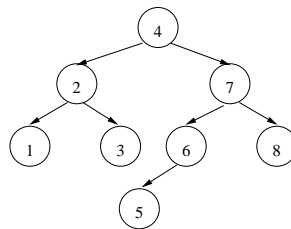


Figura 2: Albero

- (a) 12345678
  - (b) 42713685
  - (c) 13256874
  - (d) 87654321
  - (e) non rispondo
7. In quale ordine partendo dalla radice vengono visitati i nodi dell'albero in figura 3 da un algoritmo di attraversamento level-order?

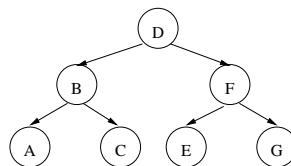


Figura 3: Albero

- (a) GFEDCBA
  - (b) CABGEGD
  - (c) ABCDEFG
  - (d) DBFACEG
  - (e) non rispondo
8. Il seguente codice:

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

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

9. 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. 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;
- (b) è corretto;
- (c) 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;
- (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

10. Il seguente codice:

```
int a[5];
++(*((&*(a+2)) + 2));
```

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