

Informatica Generale II - Prova teorica

A.A. 2005/2006

Esame: 19 aprile 2006

Codice: DJHL

1. Si consideri l'algoritmo Mergesort, quale di queste affermazioni è *falsa*?
 - (a) generalmente è più efficiente di Bubblesort
 - (b) deve essere sempre implementato mediante programmazione ricorsiva
 - (c) è un esempio tipico di algoritmo *divide et impera*
 - (d) il limite asintotico superiore è $\mathcal{O}(N \log N)$
 - (e) non rispondo
2. Un albero binario è:
 - (a) un albero i cui nodi hanno sempre 2 figli;
 - (b) nessuna delle altre risposte proposte è accettabile;
 - (c) un albero con cammino minimo 2 dalla radice alle foglie;
 - (d) un albero i cui nodi hanno al minimo 2 figli;
 - (e) non rispondo

3. Data la seguente dichiarazione:

```
double vett[5];
```

Le istruzioni `sizeof(vett)` e `sizeof(double)`

- (a) restituiscono rispettivamente la dimensione dell'intero array e il numero di byte necessari a rappresentare una variabile di tipo `double`
 - (b) sono rispettivamente corretta ed errata perché l'operatore `sizeof()` non può ricevere come parametro il nome di un tipo
 - (c) la dimensione del puntatore all'array e la dimensione in byte di un puntatore a una variabile di tipo `double`
 - (d) restituiscono rispettivamente la dimensione del puntatore all'array e la dimensione di un singolo elemento dell'array
 - (e) non rispondo
4. 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 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;
- (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) è corretto;
- (d) 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;
- (e) non rispondo

5. La seguente funzione `foo`:

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

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

6. Dato l'albero n-ario in figura 1:

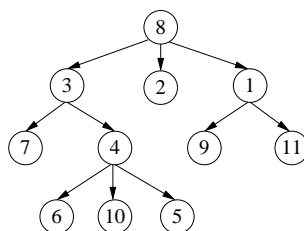


Figura 1:

- (a) La visita in postordine dell'albero produce in uscita la seguente sequenza: 7 6 10 5 4 3 2 9 11 1 8
- (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: 8 3 7 4 6 10 5 2 1 9 11
- (d) La visita in postordine dell'albero produce in uscita la seguente sequenza: 7 6 10 5 2 9 11 4 1 3 8
- (e) non rispondo

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

- (a) A,B

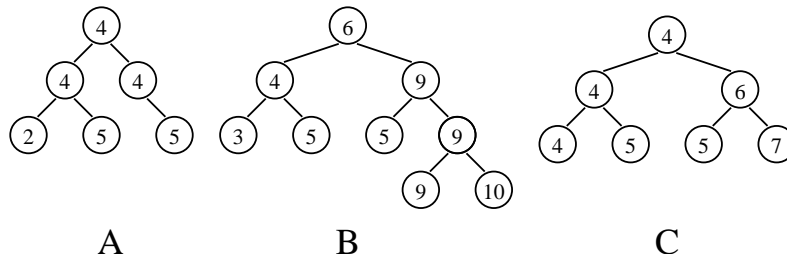


Figura 2:

- (b) B
 - (c) A
 - (d) C
 - (e) non rispondo
8. La complessità asintotica dell'algoritmo di ricerca su un BST (albero binario di ricerca) è:
- (a) $\mathcal{O}(N^2)$ dove N è il numero di nodi del BST
 - (b) $\mathcal{O}(N)$ dove N è il numero di nodi esterni del BST
 - (c) $\mathcal{O}(N \log(N))$ dove N è il numero di nodi del BST
 - (d) $\mathcal{O}(H)$ dove H è l'altezza del BST
 - (e) non rispondo
9. La ricerca sequenziale e la ricerca binaria hanno rispettivamente le seguenti complessità:
- (a) $\mathcal{O}(n^2)$ e $\mathcal{O}(n * \log n)$
 - (b) $\mathcal{O}(n^2)$ e $\mathcal{O}(n)$
 - (c) $\mathcal{O}(n)$ e $\mathcal{O}(\log n)$
 - (d) $\mathcal{O}(n)$ e $\mathcal{O}(n * \log n)$
 - (e) non rispondo
10. Quale tipo di attraversamento d'albero implementa il seguente codice?
- ```

void foo-order(Node* l, void visit(Node*))
{
 StackPtr s = new Stack();

 Push(s, l);
 while(! StackIsEmpty(s)) {
 Node * h = Pop(s);

 if (h->right != NULL) Push(s, h->right);
 if ((h->left == NULL) && (h->right == NULL)) visit(h);
 else Push(s, new Node(h->data, true));
 if (h->left != NULL) Push(s, h->left);
 if (h->flag == true) delete h;
 }
 delete s;
}

```
- (a) preorder
  - (b) inorder
  - (c) postorder
  - (d) level-order
  - (e) non rispondo