

Informatica Generale II - Prova teorica

A.A. 20052006

Esame: 27 aprile 2007

Codice: EDZO

1. Dato il grafo in figura 1:

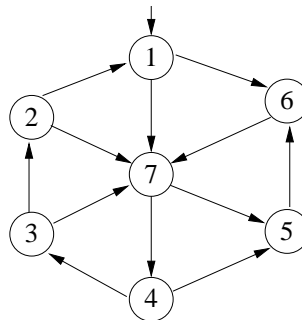


Figura 1:

- (a) La visita in profondità produce la seguente sequenza d'uscita: 1 6 7 4 3 5 2
- (b) La visita in profondità produce la seguente sequenza d'uscita: 1 7 4 3 2 5 6
- (c) La visita in profondità produce la seguente sequenza d'uscita: 1 6 7 5 3 4 2
- (d) Nessuna delle risposte è accettabile;
- (e) non rispondo

2. 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) 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 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
3. La ricerca sequenziale e la ricerca binaria hanno rispettivamente le seguenti complessità:
- (a) $\mathcal{O}(n^2)$ e $\mathcal{O}(n)$
 - (b) $\mathcal{O}(n)$ e $\mathcal{O}(n * \log n)$
 - (c) $\mathcal{O}(n)$ e $\mathcal{O}(\log n)$
 - (d) $\mathcal{O}(n^2)$ e $\mathcal{O}(n * \log n)$
 - (e) non rispondo
4. Si consideri un grafo: sia n il numero di nodi e m il numero di archi. Quale tra le seguenti affermazioni è falsa?
- (a) l'algoritmo DFS richiede tempo proporzionale a $(n + m)$;
 - (b) l'algoritmo BFS richiede tempo proporzionale a $(n + m)$;
 - (c) l'algoritmo BFS si ottiene dall'algoritmo DFS iterativo sostituendo la Coda con una Stack;
 - (d) l'algoritmo BFS si ottiene dall'algoritmo DFS iterativo sostituendo lo Stack con una Coda;
 - (e) non rispondo
5. Supponendo di aver definito una funzione `funzEsame` con la seguente definizione:
- ```
void funzEsame (int i, int & j) {
 j += i;
}
```
- L'invocazione scorretta della suddetta funzione `funzEsame` avrà la forma:
- (a) `int i=10, j=5; funzEsame(i+1, j);`
  - (b) `int i=10, j=5; funzEsame(i, j);`
  - (c) `int i=10, j=5; funzEsame(5, j);`
  - (d) `int i=10, j=5; funzEsame(i, j+1);`
  - (e) non rispondo
6. L'algoritmo selection sort ha complessità asintotica:
- (a)  $\mathcal{O}(N^2)$
  - (b) nessuna delle altre risposte proposte è corretta
  - (c)  $\mathcal{O}(N^{2/3})$
  - (d)  $\mathcal{O}(N \log N)$
  - (e) non rispondo
7. Si consideri il seguente frammento di codice:

```

{
 int x=10;
 int *j;
 j = new int[x];
 ...
}

```

Quale delle seguenti affermazioni è corretta:

- (a) le espressioni  $j[i]$  e  $*(j + i)$  sono espressioni equivalenti per indicare l'indirizzo dell' $i$ -esimo elemento dell'array
  - (b) le espressioni  $j[i]$  e  $*(j + i)$  sono espressioni equivalenti per indicare l' $(i+1)$ -esimo elemento dell'array
  - (c) le espressioni  $j[i]$  e  $*(j + i)$  sono espressioni equivalenti per indicare l' $(i)$ -esimo elemento dell'array
  - (d) le espressioni  $j[i]$  e  $*(j + i)$  sono espressioni *non* equivalenti
  - (e) non rispondo
8. In quale ordine partendo dalla radice vengono visitati i nodi dell'albero in figura 2 da un algoritmo di attraversamento pre-order?

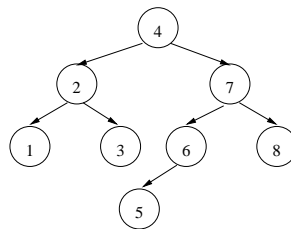


Figura 2: Albero

- (a) 12345678
  - (b) 42137658
  - (c) 87654321
  - (d) 85673124
  - (e) non rispondo
9. Si supponga di voler implementare un ADT stack di dimensione fissata. Quale di queste affermazioni è *falsa*?
- (a) le operazioni di inserimento ed estrazione hanno uguale complessità asintotica: costante
  - (b) se implementato con array piuttosto che con lista concatenata, lo stack occupa meno spazio in memoria
  - (c) se implementato con array piuttosto che con lista concatenata, l'array occupa più spazio in memoria
  - (d) la complessità asintotica delle operazioni di inserimento ed estrazione di un elemento è maggiore se lo stack è implementato con lista concatenata invece che con array
  - (e) non rispondo
10. Un albero è ordinato se
- (a) è un albero dove è specificato l'ordine dei figli di qualche nodo.
  - (b) è un albero con radice ed è specificato l'ordine dei figli di qualche nodo.
  - (c) è un albero con radice ed è specificato l'ordine dei figli della radice.
  - (d) è un albero con radice ed è specificato l'ordine dei figli di ciascun nodo.
  - (e) non rispondo