

Informatica Generale II - Prova teorica

A.A. 2005/2006

Esame: 19 aprile 2006

Codice: WCVW

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

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

3. In quale ordine partendo dalla radice vengono visitati i nodi dell'albero in figura 1 da un algoritmo di attraversamento pre-order?

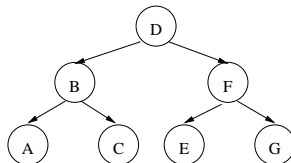


Figura 1: Albero

- (a) CABGEDD
- (b) GFEDCBA
- (c) DBACFEG
- (d) ABCDEFG
- (e) non rispondo

4. Una pila è:

- (a) un insieme di elementi gestiti secondo la politica *lifo* (last in first out);
- (b) un multiinsieme di elementi gestiti secondo la politica *lifo* (last in first out);
- (c) un multiinsieme di elementi gestiti secondo la politica *fifo* (first in first out);
- (d) un multiinsieme di elementi in cui ogni eliminazione ha per oggetto l'elemento inserito per primo;
- (e) non rispondo

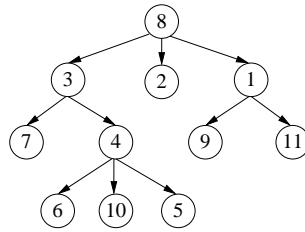


Figura 2:

5. Dato l'albero n-ario in figura 2:

- (a) La visita in preordine dell'albero produce in uscita la seguente sequenza: 7 6 10 5 4 3 2 9 11 1 8
- (b) La visita in preordine dell'albero produce in uscita la seguente sequenza: 7 6 10 5 2 9 11 4 1 3 8
- (c) Non si può effettuare la visita in preordine di un albero non binario;
- (d) La visita in preordine dell'albero produce in uscita la seguente sequenza: 8 3 7 4 6 10 5 2 1 9 11
- (e) non rispondo

6. 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

7. Si consideri il seguente frammento di codice:

```

struct Tipo2;

struct Tipo1 {
    Tipo2* a;
};

struct Tipo2 {
    Tipo1 a;
    int b;

    Tipo2(int _b) {
        b = _b;
    }
};

```

- (a) è corretto, ma Tipo2 non è istanziabile.
- (b) è errato a causa di una mutua dipendenza tra Tipo1 e Tipo2.
- (c) è corretto e compilabile.
- (d) è errato perché Tipo2 non ha un costruttore di default.

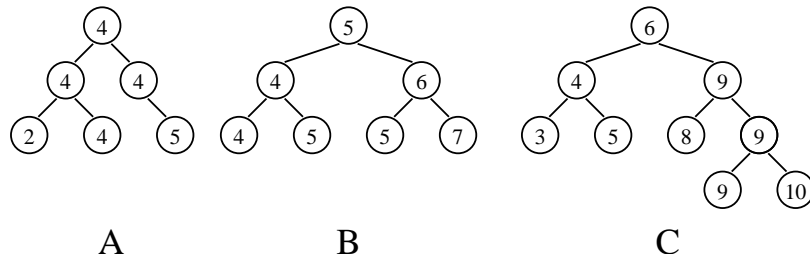


Figura 3:

- (e) non rispondo
8. Quali degli alberi in figura 3 *sono* alberi binari di ricerca, mentre gli altri non lo sono?
- C
 - A,B
 - nessuno è un albero binario di ricerca.
 - A,B,C
 - non rispondo
9. 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:
- `int i=10, j=5; funzEsame(i+1, j);`
  - `int i=10, j=5; funzEsame(5, j);`
  - `int i=10, j=5; funzEsame(i, j);`
  - `int i=10, j=5; funzEsame(i, j+1);`
  - non rispondo
10. Si supponga di avere una funzione `visit` di visita di un albero che inserisce il valore del nodo visitato in fondo ad una lista. Per ottenere una lista ordinata in senso crescente dall' attraversamento di un BST utilizzando la funzione di visita `visit`:
- non è possibile ottenere una lista ordinata in senso crescente;
  - bisogna attraversare il BST in preorder;
  - nessuna risposta tra quelle presenti è accettabile;
  - bisogna attraversare il BST in postorder;
  - non rispondo