

Matricola_____ Cognome_____ Nome_____

Fondamenti di Informatica

Prova d'esame del 24/6/2014

Regole d'esame:

1. È vietato parlare con altri studenti
2. È vietato consultare appunti, dispense, libri, in qualunque formato
3. È vietato tenere accesi i telefoni cellulari

Esercizio 1

Si consideri il seguente metodo che riceve in input un intero n . Indicare, spiegandola brevemente, la complessità asintotica di caso peggiore in funzione di n .

```
public static void esercizio(int n) {
    int i=0;
    while (i<n) {
        for (int j=i; j>=0; j--)
            System.out.println("j: "+j);
        i++;
    }
}
```

Il corpo del ciclo while viene ripetuto n volte (per i che assume tutti i valori interi da 0 a $n-1$). Ad ogni iterazione di tale ciclo si esegue il ciclo for il cui corpo si ripete $i+1$ volte (per j che assume tutti i valori interi da 0 a i). Pertanto l'istruzione di stampa (che è l'istruzione dominante del metodo) viene eseguita

$\sum_{i=0}^{n-1} (i+1) = \sum_{i=0}^{n-1} i + \sum_{i=0}^{n-1} 1 = \frac{n(n-1)}{2} + n = O(n^2)$ volte. La complessità del metodo è quindi $O(n^2)$.

Esercizio 2

Scrivere un metodo di classe ricorsivo che riceve in input un array di interi a ed altri parametri se necessario, e restituisce la stringa che si ottiene concatenando tutti gli elementi pari di a . Se, ad esempio, l'array a fosse

2	5	3	6	8	1
---	---	---	---	---	---

il metodo dovrebbe restituire "268".

Note.

1. Una soluzione non ricorsiva non sarà considerata corretta
2. Se ritenuto utile o necessario è possibile scrivere un metodo per l'avvio della ricorsione
3. Il metodo non deve utilizzare variabili che non siano locali

```

public static String concatenaPari(int[] a, int i){
    String s;
    if(i==a.length)
        s="";
    else{
        if(a[i]%2==0)
            s=a[i]+concatenaPari(a,i+1);
        else
            s=concatenaPari(a,i+1);
    }
    return s;
}

public static String concatenaPari(int[] a){
    return concatenaPari(a,0);
}

```

Esercizio 3

Supponendo di aver definito la seguente classe **Esempio**, mostrare l'evoluzione della pila di attivazione a seguito dell'avvio del programma **Esempio**.

```

public class Esempio{

    public static int metodo1(int a){
        int m;
        m = 2*a;    // metodo1, 1
        return m;   // metodo1, 2
    }

    public static int metodo2(int c){
        int i;
        i = metodo1(c); // metodo2, 1
        i = metodo1(i); // metodo2, 1
        return i;       // metodo2, 2
    }

    public static void main(String[] args){
        int x, y;
        x = 1;          // main, 1
        y = metodo2(x);  // main, 2
    }
}

```

Esercizio 4

Si consideri il seguente array di interi. Dire quale algoritmo tra BubbleSort e InsertionSort effettua meno modifiche dell'array quando eseguito sull'array stesso. Si mostrino tutte le modifiche effettuate da ciascuno dei due algoritmi.

4	2	3	6	5
---	---	---	---	---

BUBBLE SORT

Input: 4 2 3 6 5

Confronto: |4|2|3 6 5

Scambio: 2 4 3 6 5 ultimoScambio: 1

Confronto: 2|4|3|6 5

Scambio: 2 3 4 6 5 ultimoScambio: 2

Confronto: 2 3|4|6|5

Confronto: 2 3 4|6|5|

Scambio: 2 3 4 5 6 ultimoScambio: 4

Fine passata-----daOrdinare: 4

Confronto: |2|3|4 5 6

Confronto: 2|3|4|5 6

Confronto: 2 3|4|5|6

Fine passata-----daOrdinare: 0

INSERTION SORT

Input: 4 2 3 6 5

Elem da inserire: 2

Confronto: |4|2 3 6 5

Modifica: 4 4 3 6 5

Inserimento: 2 4 3 6 5

Elem da inserire: 3

Confronto: 2|4|3 6 5

Modifica: 2 4 4 6 5

Confronto: |2|4 4 6 5

Inserimento: 2 3 4 6 5

Elem da inserire: 6

Confronto: 2 3|4|6 5

Inserimento: 2 3 4 6 5

Elem da inserire: 5

Confronto: 2 3 4|6|5

Modifica: 2 3 4 6 6

Confronto: 2 3|4|6 6

Inserimento: 2 3 4 5 6

Esercizio 5

Il seguente codice contiene quattro errori che possono essere errori di logica, di sintassi, o errori a tempo di esecuzione. Indicarli spiegando brevemente perché sono errori.

```
public static int esercizio(String s){  
    int a;  
    for(int i=0;i<s.length;i++){  
        a+=s.charAt(i+1);  
    }  
}
```

1. Poiché il tipo restituito è int dovrebbe esserci un'istruzione di ritorno (oppure il tipo di ritorno dovrebbe essere void)
2. L'istruzione `a+=s.charAt(i+1)` genera un'eccezione in fase di compilazione perché la variabile `a` non viene inizializzata
3. `s.length` dovrebbe essere rimpiazzato da `s.length()`. `s` è infatti una stringa e `length()` è un metodo della classe `String`
4. Poiché per ogni `i` (da 0 a `s.length-1`) si accede al carattere in posizione `i+1`, quando `i` assume il valore `s.length-1` si tenta di accedere al carattere in posizione `s.length` che non esiste. Si avrà quindi un errore in fase di esecuzione.