

Fondamenti di Informatica

Prova d'esame del 9/9/2014

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 array di interi **a**. Indicare, spiegandola brevemente, la complessità asintotica di caso peggiore in funzione della lunghezza **n** dell'array **a**. Spiegare anche qual è la configurazione di caso peggiore.

```
public static int esercizio(int[] a){
    int n=a.length;
    int c=0;
    for(int i=0;i<n;i++){
        int j=i+1;
        while(j<n && a[j]<a[i]){
            c++;
            j++;
        }
    }
    return c;
}
```

Il corpo del ciclo for 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 while. Il numero di volte che viene ripetuto il corpo di tale ciclo dipende dal valore di i e dal contenuto dell'array. Se il contenuto dell'array è tale che la condizione $a[j]<a[i]$ è sempre verificata, allora il corpo del ciclo while viene eseguito $n-1-i$ volte (per j che assume tutti i valori interi da $i+1$ a $n-1$). Pertanto le istruzioni $c++$ e $j++$ (che sono le istruzioni dominanti del metodo) vengono eseguite $\sum_{i=0}^{n-1} (n-1-i) = \sum_{i=0}^{n-1} n - \sum_{i=0}^{n-1} 1 - \sum_{i=0}^{n-1} i = n^2 - n - \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2} = O(n^2)$ volte. La complessità del metodo è quindi $O(n^2)$.

La configurazione che dà luogo al caso peggiore si ha, come già detto, quando la condizione $a[j]<a[i]$ è sempre verificata. Ciò accade, ad esempio, se l'array è ordinato in senso decrescente.

Esercizio 2

Scrivere un metodo di classe ricorsivo che riceve in input una matrice quadrata di interi **a** ed altri parametri se necessario, e restituisce la somma degli elementi sulla diagonale principale di **a**.

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 int sommaDiagonale(int[][] a, int i){
    int somma;
    if(i==a.length)
        somma=0;
    else{
        somma=a[i][i]+sommaDiagonale(a,i+1);
    }
    return somma;
}

public static int sommaDiagonale(int[][] a){
    return sommaDiagonale(a);
}

```

Esercizio 3

Si descrivano i concetti di **record di attivazione** e **pila di attivazione**. Mostrare poi l'evoluzione della pila di attivazione a seguito dell'esecuzione del seguente programma.

***Nota:** Nella rappresentazione dei record di attivazione si può evitare di mostrare lo stack degli operandi e il campo **Classe**.*

```

public class Esercizio{

    public static int metB(int p){
        int a;
        a = 2+p;           // B, 1
        return a;          // B, 2
    }

    public static int metA(int q, int r){
        int b, c, d;
        b = metB(q);       // A, 1
        c = metB(r);       // A, 2
        d = b*c;           // A, 3
        return d;          // A, 4
    }

    public static void main(String[] args){
        int h;
        h = metA(2,3);     // main, 1
    }
}

```

Esercizio 4

Con riferimento all'algoritmo QuickSort, mostrare l'esecuzione della invocazione **partition(dati,0,10)** essendo **dati** il seguente array di interi:

Questo compito è stato discusso e definito collegialmente dalla commissione d'esame di Fondamenti di Informatica

5	10	2	17	16	14	6	12	8	11	9
---	----	---	----	----	----	---	----	---	----	---

PER RISPONDERE ALLA DOMANDA SI DEVE MOSTRARE IL VALORE ASSUNTO DI VOLTA IN VOLTA DALLE VARIABILI UTILIZZATE DAL METODO PARTITION E IL CONTENUTO DELL'ARRAY OGNI VOLTA CHE AVVIENE UNO SCAMBIO

```

partition sx: 0 dx: 10
pivot: 9
i j
5 10 2 17 16 14 6 12 8 11 9 i: -1 j: 0
i j
5 10 2 17 16 14 6 12 8 11 9 i: 0 j: 1
i j
5 10 2 17 16 14 6 12 8 11 9 i: 0 j: 2
i j
5 2 10 17 16 14 6 12 8 11 9 i: 1 j: 3
i j
5 2 10 17 16 14 6 12 8 11 9 i: 1 j: 4
i j
5 2 10 17 16 14 6 12 8 11 9 i: 1 j: 5
i j
5 2 10 17 16 14 6 12 8 11 9 i: 1 j: 6
i j
5 2 6 17 16 14 10 12 8 11 9 i: 2 j: 7
i j
5 2 6 17 16 14 10 12 8 11 9 i: 2 j: 8
i j
5 2 6 8 16 14 10 12 17 11 9 i: 3 j: 9

5 2 6 8 9 14 10 12 17 11 16

```

Esercizio 5

Dire che cosa stampa il seguente codice.

```

public class Esercizio5{
    public static void main(String[] args){

        String a="1001010";
        int v=0;

        for(int i=0;i<a.length();i++){
            v=Integer.parseInt(""+a.charAt(i))+2*v;
        }

        System.out.println(v);
    }
}

```

Stampa 74