

Esame di Fondamenti di Informatica L-B

Ingegneria Gestionale e dei Processi Gestionali (L-Z)

Appello del 16/9/2008

Esercizio 1 (4 punti)

Gestione ed utilizzo di insiemi in Java.

Esercizio 2 (6 punti)

Siano date le seguenti funzioni C:

```
int f(int V[], int M) {  
    int i=0, sum=0;  
    while(i<2*M) {  
        sum+=V[i];  
        i+=2;  
    }  
    return sum;  
}  
  
int g(int V[], int N) {  
    int j, res=1;  
    for(j=0; j<N; j+=2)  
        res*=f(V, j);  
    return res;  
}
```

1. Calcolare la complessità in passi base della funzione f nei termini del parametro M .
2. Calcolare la complessità in passi base della funzione g nei termini del parametro N (suggerimento: si supponga N pari e si esprima $j=2 \cdot i$)
3. Calcolare la complessità asintotica della funzione g nei termini del parametro N .

Esercizio 3 (7 punti)

A chi si fosse perso le dirette delle olimpiadi di Pechino, il network satellitare “Earth” propone alcuni canali che trasmettono le repliche delle gare. Ogni trasmissione è caratterizzata dal titolo, da una breve descrizione, dall’orario di inizio e da quello di fine (le trasmissioni iniziano e finiscono sempre allo scadere di ogni ora). Si scriva una classe `Trasmissione` per il network “Earth” che:

1. Possieda un opportuno costruttore con parametri.
2. Presenti opportuni metodi che permettano di accedere alle variabili di istanza della classe.
3. Presenti il metodo `toString` che fornisca la descrizione della trasmissione.
4. Possieda il metodo `equals` per stabilire l’uguaglianza con un altro oggetto `Trasmissione` (l’uguaglianza va verificata unicamente sul titolo).
5. Implementi l’interfaccia `Comparable`, definendo il metodo `compareTo` per stabilire la precedenza con una `Trasmissione` passata come parametro; l’ordine è quello di inizio trasmissione e, se due trasmissioni si sovrappongono come orario, devono essere considerate equivalenti.

Esercizio 4 (8 punti)

Si scriva una classe `Canale` che memorizzi le informazioni relative agli eventi trasmessi su un dato canale in una giornata. In particolare, oltre al nome del canale, le trasmissioni in palinsesto vanno memorizzate in una lista che va tenuta ordinata in base al punto 5. dell’esercizio 3. La classe `Canale` deve inoltre:

1. Possedere un opportuno costruttore (inizialmente il palinsesto del canale non contiene alcuna trasmissione).
2. Presentare un metodo `trovaTrasmissione` che, dato il titolo di una trasmissione, restituisca l’oggetto `Trasmissione` con tale titolo, se questo esiste.
3. Possedere un metodo `aggiungiTrasmissione` che, dato un oggetto `Trasmissione`, lo inserisca all’interno del palinsesto; nel caso che tale trasmissione si sovrapponga ad una già esistente, l’inserimento non va effettuato (suggerimento: si utilizzi il metodo `add(int i, Trasmissione t)` della classe `List<Trasmissione>`).
4. Possedere il metodo `toString` che restituisca una stringa che fornisca una descrizione del palinsesto del canale, comprendendo anche i titoli delle trasmissioni.

Esercizio 5 (7 punti)

Si scriva un’applicazione per il network “Earth” che:

1. Crei un insieme di oggetti `Canale`.
2. Crei un oggetto `Canale`, letto da tastiera le informazioni necessarie.
3. Provveda ad inserire all’interno dell’insieme di cui al punto 1. l’oggetto creato al punto 2., controllando che un oggetto uguale non esista già.
4. Letto da tastiera un titolo di una trasmissione, provveda ad inserire in una lista tutte le trasmissioni disponibili con tale titolo sui vari canali del network.
5. Stampi l’orario di inizio della prima trasmissione disponibile nella lista di cui al punto 4.

Per la lettura di dati da tastiera è possibile utilizzare l’oggetto `Lettore.in`, definito all’interno del package `fiji.io`, che possiede i seguenti metodi:

- `char leggiChar()` Legge un singolo carattere.
- `double leggiDouble()` Legge un numero razionale (delimitato da spazi).
- `float leggiFloat()` Legge un numero razionale (delimitato da spazi).
- `int leggiInt()` Legge un intero (delimitato da spazi).
- `String leggiLinea()` Legge una linea di testo.
- `String leggiString()` Legge una parola senza spazi al suo interno.

Soluzione Esercizio 2

Domanda 1:

| | |
|----------------------|----------|
| 2 assegnamenti | 2 |
| while | $M + 1$ |
| $\text{sum} += V[i]$ | M |
| $i += 2$ | M |
| Totale | $3M + 3$ |

Domanda 2:

| | |
|------------------|---------------------|
| 2 assegnamenti | 2 |
| for | $N/2 + 1$ |
| chiamata di f | $N/2$ |
| complessità di f | $3N^2/4$ |
| $j += 2$ | $N/2$ |
| Totale | $3N^2/4 + 3N/2 + 3$ |

Domanda 3:

Complessità asintotica: $O(N^2)$

Soluzione Esercizio 3

```
class Trasmissione implements Comparable<Trasmissione> {
    private String titolo, descr;
    private int iora, fora;

    public Trasmissione(String titolo, String descr, int iora, int fora) {
        this.titolo=titolo;
        this.descr=descr;
        this.iora=iora; this.fora=fora;
    }

    public String getTitolo() { return titolo; }
    public String getDescr() { return descr; }
    public int getOrarioInizio() { return iora; }
    public int getOrarioFine() { return fora; }

    public String toString() { return descr; }

    public int compareTo(Trasmissione t) {
        if(this.fora<=t.iora) return -1;
        if(this.iora>=t.fora) return 1;
        return 0;
    }

    public boolean equals(Object o) { return equals((Trasmissione) o); }
    public boolean equals(Trasmissione t) {
        return (titolo.equals(t. titolo));
    }
}
```

Soluzione Esercizio 4

```
import java.util.*;
class Canale {
    private String nome;
    private List<Trasmissione> palinsesto;

    public Canale(String nome) {
        this.nome=nome;
        palinsesto=new ArrayList<Trasmissione>();
    }

    public Trasmissione trovaTrasmissione(String titolo) {
        for(Trasmissione t:palinsesto)
            if(t.getTitolo().equals(titolo)) return t;
        return null;
    }

    public boolean aggiungiTrasmissione(Trasmissione t) {
        int i=0;
        while((i<palinsesto.size())&&(palinsesto.get(i).compareTo(t)<0))
            i++;
        if((i==palinsesto.size())||(palinsesto.get(i).compareTo(t)>0)) {
            palinsesto.add(i, t);
            return true;
        }
        return false;
    }

    public String toString() {
        String s="Canale: "+nome+"\n";
        for(Trasmissione t:palinsesto) s+=t.getTitolo()+"\n";
        return s;
    }
}
```

Soluzione Esercizio 5

```
import fiji.io.*;
import java.util.*;
class Applicazione {
    public static void main(String[] args) {
        Set<Canale> s=new TreeSet<Canale>(); // domanda 1
        Canale c=new Canale(Lettore.in.leggiString()); // domanda 2
        s.add(c); // domanda 3
        List<Trasmissione> l=new LinkedList<Trasmissione>();
        String titolo=Lettore.in.leggiLinea();
        for(Canale ch:s) {
            Trasmissione t=ch.trovaTrasmissione(titolo);
            if(t!=null) l.add(t);
        } // domanda 4
        int primaOra=25;
        for(Trasmissione t:l)
            if(t.getOrarioInizio()<primaOra) primaOra=t.getOrarioInizio();
        System.out.println(primaOra); // domanda 5
    }
}
```