

Cognome e Nome _____

Matricola _____

Appello del 22 febbraio 2024

Esercizio 1 [7 punti]

Si considerino le seguenti classi.

```
class A {  
    protected int n;  
  
    public int getN() {  
        return n;  
    }  
  
    public A(int n) {  
        this.n = n;  
    }  
  
    public A(int n, int m) {  
        this(n*m);  
    }  
  
    public A metodo(A a) {  
        n = n - a.n;  
        return this;  
    }  
}
```

```
class B extends A {  
    public B(int n) {  
        this(n, n-1);  
    }  
  
    public B(int n, int m) {  
        super(n-1, m);  
    }  
  
    public A metodo(A a) {  
        a.n = n + a.n;  
        return a;  
    }  
}
```

Si dica cosa viene stampato a video dal seguente codice:

```
A a = new A(5, 7);  
B b = new B(2);  
A ab = new B(3, 4);  
System.out.println(a.metodo(ab).getN());  
System.out.println(ab.metodo(b).getN());  
System.out.println(ab.metodo(ab.metodo(b)).getN());
```

Si giustifichi la risposta individuando in particolare le firme associate a tempo di compilazione e a tempo di esecuzione ad ogni chiamata di metodo (riempire le caselle della tabella con 1, 2 o "ERRORE" se non è possibile identificare nessun metodo):

	Tempo di compilazione	Tempo di esecuzione
a.metodo(ab)		
ab.metodo(b)		
ab.metodo(ab.metodo(b))		

Esercizio 2 [6 punti]

Si consideri la seguente sequenza di numeri interi:

[4, 10, 2, 16, 9, 5, 24, 3, 1]

1. **[2 punti]** Mostrare **passo-passo l'evoluzione dell'albero binario di ricerca** che si ottiene, a partire dall'albero vuoto, inserendo le chiavi nell'ordine indicato.
2. **[4 punti]** Mostrare **passo-passo l'evoluzione del min-heap** che si ottiene, a partire dall'heap vuoto, inserendo le prime sette chiavi (da 4 a 24) della sequenza nell'ordine indicato. Dopo aver inserito tali chiavi, effettuare **due estrazioni** del minimo mostrando l'heap che si ottiene dopo ogni estrazione, e successivamente inserire le ultime due chiavi (3 e 1) della sequenza.

Esercizio 3 [10 punti]

Si vogliono gestire, in Java, i movimenti associati al credito di SIM di telefonia mobile.

Si assuma, senza scrivere nulla, l'esistenza di una classe astratta **Movimento** con

- una variabile di istanza **idSim** (tipo String, private), che contiene il numero telefonico associato alla SIM a cui il movimento si riferisce;
- una variabile di istanza **tempo** (tipo long, private), che rappresenta l'istante del movimento espresso in secondi trascorsi dalle ore 00:00 del 1° gennaio 2000.

e i seguenti metodi di istanza:

- un costruttore che crea un movimento dati campi **idSim** e **tempo**;
- metodo di accesso **public String getIdSim()**.
- metodo di accesso **public long getTempo()**.
- metodo pubblico astratto **public abstract double getImporto()** che restituisce l'importo in Euro del movimento. In particolare, se il movimento è un movimento di spesa viene restituito un valore minore o uguale a zero, mentre se è un movimento di ricarica credito viene restituito un valore positivo.

La classe astratta **Movimento** è estesa dalle sottoclassi (non astratte) **Chiamata**, **Sms** e **Ricarica**.

Si tenga conto del fatto che la classe **Chiamata** ha:

- una variabile di istanza **durata** (tipo int, private) che contiene la durata in secondi della chiamata;
- una variabile di istanza **destinatario** (tipo String, private), che contiene il numero chiamato.

La classe **Sms** deve avere:

- una variabile di istanza **destinatario** (tipo String, private), che contiene il destinatario dell'SMS.

La classe **Ricarica** deve avere:

- una variabile di istanza **importo** (tipo double, private), che contiene l'importo della ricarica e deve valere almeno 0,01.

Si assuma, senza scrivere nulla, che tutte le sottoclassi abbiano i costruttori (che prendono in input gli opportuni parametri e che richiamino opportunamente il costruttore della classe **Movimento**), i metodi di accesso per tutti i campi e l'implementazione del metodo **double getImporto()**.

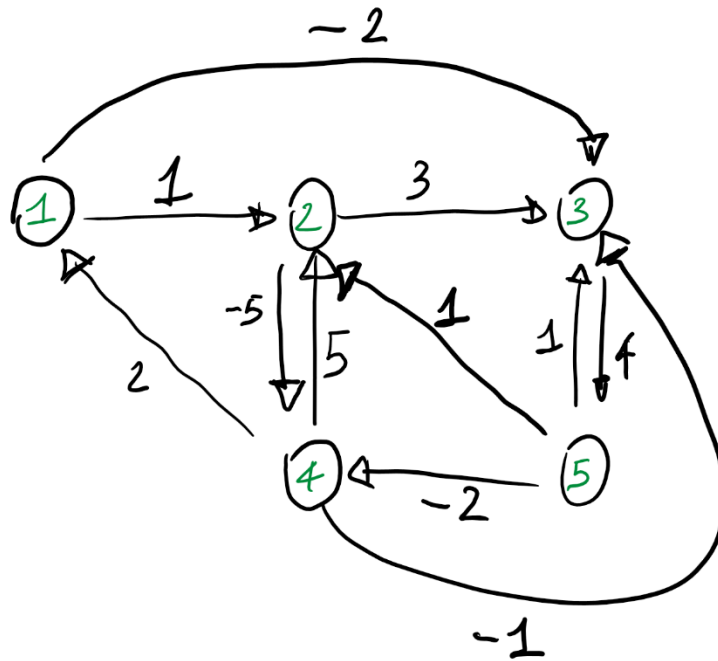
Si assuma, senza scrivere nulla, l'esistenza di una classe **Gestore** con una variabile di istanza **movimenti** (tipo `ArrayList<Movimento>`, private).

Si aggiungano alla classe **Gestore** i seguenti metodi, stimando per ciascuno di essi la complessità computazionale in termini del numero di movimenti **M** e del numero di distinti idSim **N**.

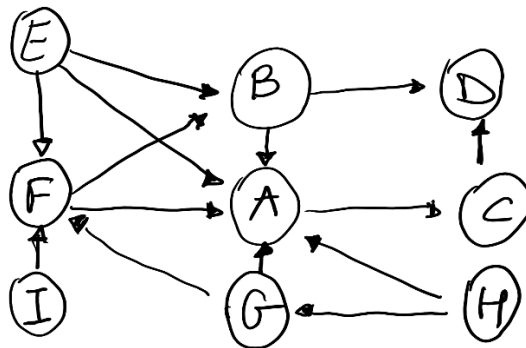
Si valuti la possibilità di utilizzare delle HashMap per avere metodi più efficienti (dà luogo ad una valutazione migliore, di 5 punti per ciascun metodo).

- **[4-5 punti]** un metodo **public ArrayList<String> listaIdSim ()** che restituisce un ArrayList contenente, senza ripetizioni, tutti gli identificativi (i numeri) delle SIM cui si riferisce almeno un movimento nell'ArrayList **movimenti**.
- **[4-5 punti]** un metodo **public ArrayList<String> simPiuAttive (double soglia)** che restituisce un ArrayList contenente tutti e soli gli **id** delle Sim che hanno effettuato chiamate e/o mandato SMS per un importo complessivo almeno uguale a **soglia** Euro.

Esercizio 4 [11 punti]



- a. [6 punti] Mostrare una possibile esecuzione **passo-passo** dell'algoritmo di **Bellman-Ford** per calcolare i cammini minimi a partire dalla sorgente **3**. Mostrare, per ogni iterazione completa che si fa sugli archi, l'evoluzione del **vettore delle distanze** e del **vettore dei padri**, assumendo di visitare, in ogni iterazione, gli archi in ordine crescente rispetto alle etichette dei nodi da cui escono. Dire, giustificando la risposta, se il grafo possiede cicli di peso negativo.



- b. [5 punti] Calcolare un ordinamento topologico del DAG in figura, usando il metodo della visita in profondità ed evidenziando per ogni nodo i timestamp di inizio e fine visita. Si assuma di iniziare e riprendere la visita sempre dal nodo (non ancora visitato) avente l'etichetta alfabeticamente più piccola.

Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **150** minuti.
- Scrivere **subito** nome, cognome, matricola su **OGNI FOGLIO (compreso questo)**.
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- È possibile consultare appunti, libri e dispense.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato** e **depositato in vista sulla cattedra**
- Mettere in vista sul banco un valido documento di identità.