

Problema 1

Si consideri il metodo `incrementa(ListIterator<Integer> cifre)` che, dato l'iteratore di una lista che contiene le cifre di generico numero intero (da quella meno significativa a quella piu' significativa), incrementa di una unita' il valore (facendo side effect sulla relativa lista):

```
public static void incrementa(ListIterator<Integer> cifre) {
    if (!cifre.hasNext()) {
        cifre.add(1);
    } else {
        Integer val = cifre.next();//valore corrente
        if (val < 9) {
            cifre.set(val + 1);//assegna un valore all'elemento corrente (costo costante)
        } else {
            cifre.set(0);//assegna il valore 0 all'elemento corrente (costo costante)
            incrementa(cifre);
        }
    }
}
```

Si richiede quanto segue.

- (a) Determinare, giustificandolo opportunamente, il costo computazionale del metodo `incrementa(ListIterator<Integer> cifre)` in funzione della dimensione dell'input.
- (b) Scrivere il metodo statico Java `decrementa(ListIterator<Integer> cifre)` che decrementa di una unita' il valore rappresentato dall'iteratore (secondo le medesime convenzioni del metodo `incrementa(...)`). Assumere che il valore passato come parametro sia strettamente positivo. Determinare, giustificandolo opportunamente, il costo del metodo.
- (c) Scrivere il metodo statico Java `ListIterator<Integer> differenza(ListIterator<Integer>a, ListIterator<Integer>b)` che, utilizzando i metodi `incrementa(...)` e/o `decrementa(...)` definiti in precedenza, calcoli e restituisca il valore $a - b$ (assumendo $a \geq b$). Determinare, giustificandolo opportunamente, il costo del metodo.

Problema 2

Con riferimento al tipo astratto Coda di Priorita' si richiede quanto segue:

- (a) Definire il tipo astratto Coda di Priorita' realizzando una possibile interfaccia Java PQ che lo descrive.
- (b) Realizzare la classe astratta Java `PQListaOrdinata` che implementa l'interfaccia PQ e che rappresenta una coda di priorita' realizzata mediante lista ordinata contenente chiavi di tipo intero. La classe deve contenere tutte le variabili di istanza e le firme dei metodi fondamentali oltre alla implementazione del metodo di inserimento di un nuovo elemento. Di ogni metodo deve infine essere indicato (e motivato) il costo computazionale.
- (c) Definire il metodo `static void PQsort(int[] array)` che ordina il vettore `array` mediante una coda di priorita' `PQListaOrdinata`. Indicarne poi, motivandolo adeguatamente, il costo computazionale.

Problema 3

Con riferimento ai grafi orientati si richiede quanto segue:

- (a) Definire il concetto di chiusura transitiva di un grafo.
- (b) Definire una possibile rappresentazione per un grafo basata su liste di adiacenza realizzando una classe astratta Java `Graph` contenente tutte le variabili di istanza e le firme di tutti i metodi pubblici che si ritiene fondamentali per risolvere il problema successivo.
- (c) Realizzare un metodo di istanza della classe `Graph` la cui firma e' `Graph chiusuraTransitiva()` e che calcola la chiusura transitiva del grafo `this`. Descrivere poi il costo computazionale dell'algoritmo, motivandolo opportunamente.

<i>nominativo e matricola</i>	<i>esame cui si partecipa</i>