

# Università Degli Studi dell'Aquila

I parziale del modulo di Laboratorio di Algoritmi e Strutture Dati - A.A. 2020/2021

Venerdì 20 Novembre 2020 – Dott.ssa Giovanna Melideo (Durata: 1:15 h)

Svolgere i seguenti esercizi avendo come riferimento il linguaggio JAVA.

*Vista la modalità di svolgimento, il compito è volutamente sovradimensionato rispetto al tempo disponibile. Pertanto si richiede agli studenti di svolgere un sottoinsieme di esercizi a scelta, sfruttando tutto il tempo a disposizione.*

Classi Termine ed Espressione:

<https://univaq.sharepoint.com/:f:/s/Lab.AlgoritmieStruttureDati/ErcashArqAJGmCZZw64Vs4cBxKKHY90p64gmf2PW1gUg5Q?e=IKdHGR>

## ESERCIZIO 1

Considerare le classi Termine ed Espressione. Implementare il metodo `public double calcola(double valx)`, interno alla classe `Espressione`, che calcola il valore di questa espressione algebrica ad una variabile (`this`) per il dato valore di `valx`.

## ESERCIZIO 2

Considerare le classi Termine ed Espressione. Implementare un metodo statico che, data una lista di espressioni algebriche ad una variabile (oggetto di tipo `ArrayList<Espressione>`), ordina la lista rispetto al grado dell'espressione (il grado dell'espressione rispetto alla variabile `x` è l'esponente più alto con cui `x` compare in essa). Definire tutte le classi e/o metodi aggiuntivi ritenuti utili o necessari.

## ESERCIZIO 3

Implementare il metodo `public boolean addAll(Collection<? extends E> c)`, interno alla classe `SortedDoublyLinkedList`, che inserisce nella lista corrente (`this`) tutti gli elementi nella collezione specificata in modo da non violare l'ordinamento.

## ESERCIZIO 4

Risolvere almeno il quesito a)

- Scrivere un metodo `minoriDi` che riceve in ingresso un array di elementi di tipo `String` e un elemento di tipo `String`, e restituisce un oggetto di tipo `ArrayList` con elementi di tipo `String` contenente tutti gli elementi dell'array che sono minori dell'elemento dato.
- Come cambia la soluzione se il metodo `minoriDi` diventa generico? Scrivere un metodo generico `minoriDi` che riceve in ingresso un array di elementi di un tipo `E` e un elemento di tipo `E` e restituisce un oggetto di tipo `ArrayList` con elementi di tipo `E` contenente tutti gli elementi dell'array che sono minori dell'elemento dato. Suggestivo: serve un meccanismo che consenta di confrontare gli elementi dell'array.

## ESERCIZIO 5

Risolvere almeno il quesito a)

- Scrivere un metodo `LinkedList<Integer> union(LinkedList<Integer> L1, LinkedList<Integer> L2)` che riceve come argomenti due liste di oggetti di tipo `Integer` prive di elementi duplicati e restituisce la lista unione, priva di elementi duplicati, contenente gli elementi di entrambe le liste. Qual è la complessità di tempo del metodo proposto? Giustificare la risposta

- b) Come cambia la soluzione se il metodo union diventa generico? Scrivere un metodo generico `<void> union (List<E> L1, List<E> L2, Comparator<E> c)` che riceve come argomenti due liste L1 e L2 di oggetti di un tipo E prive di elementi duplicati e restituisce in L1 la lista priva di elementi duplicati, contenente gli elementi di entrambe le liste. I confronti tra gli elementi devono essere effettuati utilizzando il comparatore passato come terzo argomento.