

# AlgaT

Il progetto AlgaT, scritto in linguaggio Java ed utilizzando la libreria grafica JavaFX, è un programma che ha lo scopo di mostrare ed insegnare il metodo di funzionamento di un particolare algoritmo. Il progetto è relativo al corso di “Algoritmi e Strutture Dati” dell’ Alma Mater - Studiorum Università di Bologna con docente di riferimento Angelo di Iorio.

Al lancio dell’applicazione, all’utente verrà mostrata una parte teorica ed una parte implementativa in pseudo-codice con una successiva fase di test in cui potrà verificare le nozioni appena acquisite (tutorial).

Il programma è stato suddiviso in classi per una miglior comprensione, leggibilità e portabilità del codice.

L’algoritmo scelto è MergeSort() il quale utilizza la nota tecnica del “Divide et Impera” inventata da John von Neumann nel 1945.

Di seguito mostriamo il funzionamento generale delle principali classi all’interno della cartella “src”. Affinchè il programma possa essere lanciato avviare il file “AlgaT.jar” situato all’interno la cartella “dist”.

## ***Lessons.java :***

Il file lessons.java contiene tutte le funzioni necessarie per:

- inizializzare il tutorial
- leggere le lezioni
- immissione dimensione array
- mostrare all’utente il funzionamento dell’algoritmo.

Per la finestra principale, abbiamo creato uno stage e chiamata poi

tramite la funzione ***“myStage.show();”***. A questo punto abbiamo suddiviso la finestra in due parti (Vbox) per una miglior comprensione:

1. nella parte superiore troviamo le lezioni con le spiegazioni.
2. nella parte inferiore la creazione dell'array e relativa parte di visualizzazione del metodo di funzionamento dell'algoritmo.

Per quanto riguarda la parte delle lezioni, una volta che l'utente avrà terminato di leggere il testo potrà inizializzare il tutorial tramite pulsante con label “Inizia Test”. Una volta completato e solo se superato potrà accedere alla seconda lezione sempre nel riquadro della lezione precedente.

I due test sono stati creati tramite le due funzioni

***showQuestion();*** e ***showQuestion1();*** relativamente uguali.

Le risposte poi saranno controllate tramite le funzioni

***check.setAction();*** e gli eventHandler.

Durante i due test l'utente dovrà controllare la correttezza delle sue risposte tramite il bottone relativo “Verifica Risposta”. Se la risposta è corretta potrà proseguire, altrimenti dovrà ritentare (l'utente è avvisato per ciascun caso da opportune finestre tramite la funzione citata prima).

Finito il tutorial nella parte inferiore della finestra principale, l'utente può inserire la grandezza dell'array a piacere, rispettando i limiti delle avvertenze in modo tale da non avere problemi visivi in fase di esecuzione. Nel caso egli non rispetti il vincolo, una finestra lo avviserà del mancato rispetto dello stesso.

Il tasto start infine avvierà la fase di spiegazione in modalità interattiva

dell'algoritmo.

L'utente può decidere liberamente quale delle due parti può interagire per prima (parte superiore o parte inferiore).

### ***MergeSort.java :***

Il file mergesort.java contiene la principale implementazione e funzionamento dell'algoritmo scelto per l'ordinamento. Inizialmente il programma verifica la dimensione dell'algoritmo in cui se la dimensione di input è minore di 2, l'array è già ordinato (essendo costituito da una sola unità/cella). Superato questo controllo il programma procederà all'ordinamento tramite MergeSort().

### ***Main.java :***

Questo file oltre a fissare determinate specifiche per la finestra principale dove vengono visualizzate le parti "Lezioni" e "Tutorial" gestisce le Exception di questa finestra principale.

### ***Input.java :***

Nella classe "Input" è stata definita la funzione di creazione dell'array situato nella parte inferiore della finestra principale. L'utente inserisce un valore numerico e la funzione determina un array di quel valore, creato con valori casuali.

### ***Graphics.java :***

In questo file viene creata la sezione dedicata alla spiegazione dei passi dell'algoritmo adottato. Il programma mostra a video (nella parte

superiore) l'array generato e una volta scelto un metodo tra step-by-step e play, l'array si ordinerà interattivamente. Effettuato l'ordinamento, una console al di sotto dell'array ordinato, stamperà a video i passi effettuati dell'algoritmo in modo tale da spiegare in dettaglio il procedimento.

### ***ArrayGen.java :***

Una volta inserito il valore numerico grazie alla classi input, l'array viene generato con valori casuali e senza ripetizioni grazie alla funzioni presenti nella classe ArrayGen con relative Exception.

### **Collaboratori del progetto:**

Martina dell'Elce (matricola n°801842 )

Thomas Cupertino (matricola n° 817155)