

# PROGETTO ALGA

---

Angelo Di Iorio

Università di Bologna

# Progetto ALGA

- ALGA (ALGorithm Animator) è un software Java in grado di simulare e documentare l'esecuzione di un algoritmo
- Applicazione interattiva basata su Java FX
- Obiettivo: realizzare un software utile per spiegare un algoritmo a propria scelta (ad esempio al docente il giorno della discussione!)

# Quale algoritmo?

- Il progetto si svolge in gruppo
- Ogni gruppo sceglie un algoritmo da un elenco predefinito
- Algoritmi divisi in tre categorie:
  - Ordinamento di vettori
  - Visite di alberi
  - Visite di grafi
- Ogni algoritmo può essere scelto da **4 gruppi al massimo**  
(se non bastano rilasceremo il vincolo)
- Elenco completo nelle prossime slide

# Gruppi e voto

- Il progetto si svolge individualmente o in coppia
- **NON sono ammessi gruppi di più di 2 persone**
- Il progetto si consegna una volta sola e il voto è valido per l'intero anno accademico (e per gli anni successivi se le regole non cambiano)
- Per gli studenti del primo anno è obbligatorio consegnare il progetto prima dell'esame scritto
- Per gli studenti degli anni precedenti il progetto non è obbligatorio
- Il progetto “fa media” (pesata) con il voto dello scritto:  $\frac{1}{4}$  del voto complessivo
- Il voto non dipende dall'algoritmo scelto ma dallo svolgimento del progetto

# Consegna

- Il progetto si consegna il giorno della discussione (portare un laptop)
- La consegna finale deve includere:
  - Sorgenti Java
  - Eseguibile (Runnable .jar)
  - Javadoc
  - README

# Quando consegnare?

- Orientativamente 2 settimane prima di ogni appello scritto
- A giugno/luglio:
  - 23 maggio
  - 14 giugno
  - 4 luglio
- Date e aule da confermare (ma cambieranno di poco)
- Saranno pubblicate su AlmaEsami e sul sito del laboratorio
- Se ci sono molte richieste possiamo fissare un'ulteriore data straordinaria prima o dopo queste

# Requisiti

- ALGA mostra:
  - descrizione testuale e pseudocodice dell'algoritmo
  - campi/bottoni per inserire dati in input e interagire con l'applicazione
  - le strutture dati utilizzate e come queste cambiano durante l'esecuzione
- ALGA permette di scegliere i dati in input in diversi modi
- ALGA permette di eseguire l'algoritmo per passi
- ALGA gestisce correttamente gli errori e li notifica all'utente
  - input non corretto, operazioni non possibili, etc.

# Dati in input

- All'apertura ALGA permette all'utente di selezionare i dati di input dell'algoritmo
- L'utente ha tre scelte a disposizione:
  - 1) dati generati casualmente
    - l'utente deve poter scegliere le dimensioni dei dati (es. lunghezza del vettore da ordinare)
  - 2) dati inseriti interattivamente dall'utente
  - 3) dati caricati da un file di testo
    - strutturato secondo regole stabilite dall'applicazione
    - ALGA verifica se il file è valido ed è possibile caricare i dati; in caso contrario notifica all'utente la presenza di errori



## Dati in input (2)

- I valori in input possono essere interi, stringhe o altri tipi confrontabili (`Comparable`) a scelta dell'applicazione
- ALGA verifica che i dati inseriti siano del tipo corretto
- E' possibile imporre vincoli sulle dimensioni dei dati in input, purchè sia correttamente notificato all'utente

# Esecuzione

- ALGA permette all'utente di selezionare la modalità di esecuzione e animazione dell'algoritmo:
  - *Step-by-step*: l'utente clicca su un bottone per avanzare di un "passo" nell'esecuzione
  - *Motion*: l'utente lancia l'esecuzione e ALGA mostra tutti i passi, ad intervalli regolari
    - ALGA permette all'utente di selezionare la durata di ogni passo
- Ad ogni passo, ALGA mostra le strutture dati utilizzate e come queste cambiano durante l'esecuzione
  - E' possibile aggiungere messaggi di spiegazione o altri hints grafici

# Algoritmi

- Ordinamento

1. ~~CountingSort~~ pag.35
2. ~~HeapSort~~
3. ~~InsertionSort~~ pag.28
4. MergeSort pag.30
5. ~~PigeonholeSort~~ pag.87
6. ~~QuickSort~~ pag.228
7. ~~SelectionSort~~ pag.28
8. ~~ShellSort~~ pag.289

- Alberi

9. Visite in profondita (nello stesso progetto: previsita, postvisita e invisita)
10. Visita in ampiezza

- Grafi

11. Visita BFS (a partire da un nodo scelto dall'utente)
12. Visita DFS (a partire da un nodo scelto dall'utente)

# Come scegliere e prenotarsi

- Compilare form: <http://diiorio.nws.cs.unibo.it/asd1617/progetto/>
- Non è possibile cambiare gruppo o algoritmo tramite il form, contattare direttamente il docente via mail

Algoritmi e Strutture Dati - Progetto ALGA			Info
Gruppo			
Nome:			
<input type="text"/>			
Componenti			
Nome:		Cognome:	Mail Unibo:
<input type="text"/>		<input type="text"/>	<input type="text"/>
Nome:		Cognome:	Mail Unibo:
<input type="text"/>		<input type="text"/>	<input type="text"/>
Algoritmo scelto			
<input checked="" type="radio"/> Counting Sort			
<input type="radio"/> Heap Sort			
<input type="radio"/> Insertion Sort			
<input type="radio"/> Merge Sort			
<input type="radio"/> Pigeonhole Sort			
<input type="radio"/> Quick Sort			
<input type="radio"/> Selection Sort			
<input type="radio"/> Shell Sort			
<input type="radio"/> Alberi - visite profondità			
<input type="radio"/> Alberi - visita ampiezza			
<input type="radio"/> Grafi - Depth-First Search			
<input type="radio"/> Grafi - Breadth-First Search			
<input type="button" value="Invia"/>			

# Cosa ci aspettiamo dal progetto?

- Il progetto deve:
  - Implementare correttamente l'algoritmo
  - Mostrare chiaramente la sua esecuzione
  - Gestire gli errori
  - Gestire le diverse modalità di input
  - Avere un layout gradevole
- **Soprattutto: deve essere frutto del vostro lavoro, dovete essere in grado di spiegarlo e modificarlo on-the-fly il giorno della consegna!**

# Domande?

- Buon lavoro!