Sorting Visualizer Documentation

Introduction

The Sorting Visualizer is a Java program designed to visualize various sorting algorithms in action. It provides an interactive graphical interface that allows users to input a list of numbers and select a sorting algorithm to visualize the sorting process. The program aims to help users understand how each sorting algorithm works by animating the sorting process step by step

Usage

Running the Program

To run the Sorting Visualizer program, follow these steps:

1. Compile the Java source files.

2. Execute the main class ‘SortingVisualizer’.

3. The program window will appear, allowing you to input a list of numbers and select a sorting algorithm.

Inputting Data

- Input the list of numbers in the provided text field. Separate each number by a space.

- Click the "Generate Random Array" button to generate a random list of numbers.

Selecting Sorting Algorithm

- Choose the desired sorting algorithm from the dropdown menu.

- The supported sorting algorithms are:

- Bubble Sort

- Insertion Sort

- Selection Sort

Starting, Pausing, and Resetting

- Click the "Start" button to begin the sorting process.

- Use the "Pause" button to pause/resume the sorting animation.

- Click the "Reset" button to reset the sorting process and input data.

Supported Sorting Algorithms

Bubble Sort

Bubble Sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted.

Insertion Sort

Insertion Sort is a simple sorting algorithm that builds the final sorted list one item at a time. It iterates over the input list, removing one element per iteration and inserting it into the correct position in the sorted list.

Selection Sort

Selection Sort is a simple sorting algorithm that repeatedly finds the minimum element from the unsorted part of the list and moves it to the beginning. The algorithm maintains two sublists: the sublist of items already sorted and the sublist of items remaining to be sorted.

Insertion Sort

Insertion Sort is a simple sorting algorithm that builds the final sorted list one item at a time. It iterates over the input list, removing one element per iteration and inserting it into the correct position in the sorted list. At each step, the algorithm selects an element from the unsorted part of the list and inserts it into its correct position in the sorted part of the list.

Selection Sort

Selection Sort is a simple sorting algorithm that repeatedly finds the minimum element from the unsorted part of the list and moves it to the beginning. The algorithm maintains two sublists: the sublist of items already sorted and the sublist of items remaining to be sorted. At each iteration, the algorithm selects the smallest element from the unsorted sublist and swaps it with the first element of the unsorted sublist.

Design Decisions

- Modular Design: The program is designed with modularity in mind, separating the sorting algorithms into different classes. This design choice enhances code organization, readability, and maintainability.

- Interactive GUI: The graphical user interface (GUI) provides a user-friendly experience, allowing users to input data, select sorting algorithms, and control the sorting animation.

- Real-time Visualization: The sorting process is visualized in real-time, with bars representing array elements and text labels displaying their values. This visualization helps users understand the behaviour of each sorting algorithm.