Sorting Algorithm Visualizer Documentation

Introduction

The **Sorting Algorithm Visualizer** is a graphical tool built using Python and Tkinter to demonstrate the working of various sorting algorithms. This application allows users to enter custom numerical input and visualize how different sorting algorithms operate in real-time.

Features

- **Interactive GUI** to input numbers and visualize sorting steps.
- **Sorting Algorithms Supported:**
  - Bubble Sort
  - Selection Sort
  - Insertion Sort
  - Heap Sort
- **Color-Coded Sorting Process** for better visualization.
- **Real-Time Updates** to observe sorting in action.
- **Error Handling** for invalid user input.

Technologies Used

Technology	**Description**
Python	Programming language
Tkinter	GUI Framework
time	Used to slow down visualization

Class & Function Documentation

`draw_data(highlight_indices=None)`

Description

Draws the bars representing the dataset, highlighting the indices currently being processed.

Parameters

- `highlight_indices (list)`: Optional list of indices to highlight during sorting.

`process_input_data()`

Description

Processes the user input, validates it, and converts it into a list of integers for sorting.

`clear_data()`

Description

Clears the input data and resets the canvas.

Sorting Algorithms

`bubble_sort()`

- Compares adjacent elements and swaps them if necessary.
- Time Complexity: **O(n^2)**

`selection_sort()`

- Finds the minimum element and swaps it with the first unsorted element.
- Time Complexity: **O(n^2)**

`insertion_sort()`

- Inserts each element into its correct position.
- Time Complexity: **O(n^2)**

`heap_sort()`

- Uses a binary heap to sort elements efficiently.
- Time Complexity: **O(n log n)**

GUI Layout

1. Input Section

- Entry field for entering comma-separated numbers.
- **Submit Input** button to process the numbers.

2. Sorting Controls

- Buttons to trigger different sorting algorithms.
- **Clear** button to reset the data.

3. Visualization Canvas

- Displays sorting operations with colored bars.

Example Usage

Running the Application

```
python sorting_visualizer.py
```

Using the Interface

1. Enter numbers separated by commas (e.g., `5, 2, 9, 1, 7`).
2. Click **Submit Input** to process the data.

3. Click on a sorting algorithm (Bubble, Selection, Insertion, or Heap Sort) to visualize it.
4. Observe the sorting process in real time.
5. Click **Clear** to reset and enter new numbers.

Notes

- Sorting is visualized step by step with a small delay for better understanding.
- The highlighted bars indicate active comparisons/swaps.
- Larger inputs may take longer to process.

Conclusion

The **Sorting Algorithm Visualizer** provides an **interactive way** to learn how sorting algorithms work. It is a great tool for students and educators to observe sorting behavior dynamically.