User’s guide

Table of Contents

Foreword

While scrolling through the internet, you might already have come across a video that explains, shows, or animates a sorting algorithm. It might have been a YouTube video or a TikTok clip. Still, the concept is the same: a visual representation of sorting algorithms works.

These videos often claim millions of views, and it’s easy to see why. They are not only informative but also visually appealing. Using colors, shapes, and animations captures the viewer’s attention and makes the learning process enjoyable.

Inspired by this approach, this project aims to give you the same engaging experience. Here, you'll find an interactive tool to explore, compare, and visualize sorting algorithms. Whether you're a beginner or just curious about how these algorithms work, this web application is designed to make the learning process fun and engaging.

If you are looking for a more technical explanation of the project, you can refer to the Technical Documentation.

What is this web application?

Objective

In short, this web application is an interactive tool that allows you to code and visualize your sorting algorithms in real-time in a web browser.

The objective of *Show My Sort* is to provide a platform for users to visualize and compare different sorting algorithms.

Main Features

If you open it for the first time *Show My Sort*, you will be greeted with a simple and user-friendly interface. The main features of the web application include:

1. Code your sorting algorithms.
2. Visualize the sorting process.
3. Compare the performance of different algorithms.

More details are provided in the section How do I use it?.

All of these features are presented to you in an all-in-one web application and in a user-friendly interface, which means you can access it from any device with a web browser. Simply lovely, right?

What It Is Not...

Despite all the features and the fun you can have with it, this web app is not a full-fledged IDE (Integrated Development Environment) for Python. It is not intended to replace professional development tools or environments. You will not find advanced features like terminal access, package management, or debugging tools. It is a **simplified** and **interactive** environment designed for educational purposes.

However, you will still receive some helpful feedback on your code.

Furthermore, be aware that the web application is not intended for code production. This software does not replace a professional IDE or development environment.

You must also know that all your written code is:

- Executed locally on your machine, see Environment safety
- Stored in the browser's local storage, see Storing your code
- Not sent to any server

This means that your code is not shared with anyone else, and you have full control over it. However, it also means that you need to be careful not to lose your code if you clear your browser's local storage or switch devices.

Target Audience

This web application is designed for anyone interested in learning and understanding sorting algorithms. It is suitable for students, educators, developers, hobbyists, and anyone curious about sorting algorithms.

This application still requires some basic knowledge of Python programming, as you will be writing your sorting algorithms. However, it is not necessary to be an expert in Python or sorting algorithms to use this web application. From a simple linear sorting algorithm to a more complex one, such as the merge sort, the code is your only limit!

License Information This project is licensed under the MIT License, please refer to the LICENSE for any further information.

How do I install it?

For simple usage, you do not need to install anything. You can simply access the web application by clicking on the following link: *Show My Sort*.

For the most advanced users, you can also run the web application locally on your machine. But this is not necessary for simple usage. Moreover, you must be connected to a web connection to use the web application, as it needs to download the required libraries.

In the following sections, you will find all the information you need to install the web application locally on your machine if you want to explore this option.

Where can I find the installation package?

First, you need to download the installation package. You can find it on the GitHub repository.

To install this package: 1. Open your terminal in a directory of your choice. - On macOS, you can use the Finder to navigate to the desired directory, then right-click and select "Open Terminal Here". - On Windows, you can use File Explorer to navigate to the desired directory, then right-click and select "Open PowerShell Here". 2. Make sure you have Git installed on your machine. You can check this by running the following command: `bash     git --version` If you don't have Git installed, you can download it from the official website. 3. Clone the repository using the following command: `bash     git clone https://github.com/FloSto-Bash/Bac3-Projet-Individuel`

You have now downloaded the installation package. Congratulations!

Prerequisites

To run the web application locally, you need to have the following software installed on your machine: - Node.js (and so npm): you can download it from the official website. - Python 3.x: you can download it from the official website.

Which software should I use to run the installation?

As this software is a web application, you can use any web browser to run it. However, it is recommended to avoid using Safari for performance reasons. I personally recommend using Arc on macOS, you can download it from the official website.

What modifications are required to the environment?

No modifications are required to the environment. The web application is designed to run in a standard web browser without any additional configuration.

How do I run the web application locally?

To run the web application locally, assuming you have already cloned the repository, follow these steps in your terminal: 1. Navigate to the cloned repository: `bash     cd Bac3-Projet-Individuel` 2. Install the required dependencies using npm: `bash     npm install` 3. Start the development server: `bash     npm run dev` 4. Open your web browser and navigate to `http://localhost:5173/`, or click on the link provided in the terminal. 5. You should see the web application running locally on your machine.

You can now start coding your sorting algorithms and visualizing the sorting process, locally!

How do I use it?

In this section, you will find helpful information concerning the usage of the web application.

By Features

Code your sorting algorithms To get started, you can write your sorting algorithms in Python, using the provided code editor. The editor supports syntax highlighting and basic code completion, making it easier to write and understand your code.

Getting started During your first visit, you will be invited to try out the predefined functions that you must use in your sorting algorithms to link your code to the visualization. These functions are the following:

1. Compare two elements in the array.

```
compare(arr, i, j) -> bool
    # Returns True if arr[i] <= arr[j], otherwise False.
```

2. Swap two elements in the array.

```
swap(arr, i, j)
    # Swaps the elements at indices i and j in arr.
```

Furthermore, for these two functions, you must also use a predefined variable `myList`, which is a list of integers that you will sort. This variable is linked to the visualization too. You might try to redefine it, but this action is not allowed and will cause an error.

Environment safety To ensure the safety of your code, *Show My Sort* runs your code in a secure environment on your machine. This means that your code is analyzed before execution, and any potentially dangerous operations are blocked. In other words, you are restricted to a limited set of operations, which are:

The following functions and keywords are allowed for use in your sorting algorithms:

Category	Functions/Keywords
Built-in Imports	`__import__`
Iteration	`range`, `reversed`
Length and Type	`len`, `int`, `float`, `str`, `list`, `bool`
Mathematical	`min`, `max`, `abs`, `sum`, `round`
Logical	`all`, `any`

The following modules are restricted and cannot be imported:

Restricted Modules
`os`
`sys`
`subprocess`
`shutil`
`socket`
`HTTP`
`ftplib`
`pickle`

These operations are carefully selected to ensure safety and functionality while coding your sorting algorithms. If you think a specific operation is missing, please let me know by using the feedback function, as described in the section What to Do in Case of Errors?, and I will do my best to add it in the next release.

Storing your code Your code will be stored every 2 seconds in the browser's local storage, so you can come back to it later. In addition to this feature, the local storage allows you to save multiple codes and linked statistics, which means you can easily switch between different algorithms and compare their performance. But you must be careful not to clear your browser's local storage or switch devices, as this will erase all your saved code.

Import/Export your code In addition, you can also import your Python code from your local machine by clicking on the import button in the top right corner of the web page.

Similarly, you can also export your code to your local machine by clicking on the export button in the top-right corner of the web page. This will download a `show-my-sort.py` file containing your code.

Visualize the sorting process Once you have written your sorting algorithm, you can visualize the sorting process in real-time. The web application provides a graphical representation of the sorting algorithm, showing how the elements are compared and swapped during the sorting process.

You can customize the visualization by: - Changing the array to sort, which is linked to the variable `myList` as previously mentioned Getting Started - Showing the comparisons - Showing the swaps - Changing the speed of the visualization (with the slider below the diagram)

Compare the performance of different algorithms Below the diagram, you will find a table that displays all the statistics of the algorithms you have run. This table is similar to this one: | Execution Count | Execution Time | Swap Count | Compare Count | Average Time | Standard Deviation | |:————:|:————:|:———-:|:————-:|:————:|:——————:| | xx | x.xxx | x | x | x.xxx | x.xxx |

Customize the web page style You can customize the editor style by switching between light and dark modes. This feature is available in the top right corner of the web page. The web application is designed to be responsive and works well on both desktop and mobile devices. You can use it on your laptop, tablet, or smartphone without any issues.

For those who prefer the dark side and want to *bring peace, freedom, justice, and security to their new sorting empire*, a dark mode is available. This option is linked to your system settings, so if you switch to dark mode on your system, the web application will automatically switch to dark mode too. May the Force be with you!

Stay updated on the software status At the bottom of the web page, you will find a status bar that indicates the current status of the web application. This status bar is updated during the execution and provides information about the current state of the web application.

If an error occurs during the execution, the status bar will display an error message. You can also use this status bar to check if your code is running correctly.

Start Here

To get started, you can simply click on the "play" button, which will run an example code, showing you how to use the `compare` and `swap` functions, with the predefined variable `myList`. This example only compares and swaps the first two elements of the array.

To go further, you can erase this code and write your sorting algorithm. If your code is correct, you will see the sorting process after clicking on the "play" button. If your code is incorrect, you will see an error message in the status bar at the bottom of the web page.

It is as simple as that! You can now start coding your sorting algorithms and visualizing the sorting process. Enjoy!

What to Do in Case of Errors?

If you encounter an error while running your code, the web application will display an error message in the status bar at the bottom of the web page. This message will provide information about the error. If you consider that the error is not due to your code, you can use the feedback functionality to report the error.

To do this, simply click on the feedback button in the bottom left corner of the web page. This will open your mail application with a pre-filled email. Please provide as much information as possible about the error, including the code you were trying to run and any error messages you received. This will help me to identify and fix the issue.

Before reporting an error, please make sure to check if the error is not due to your code or is not known. You can do this by checking the Frequently Asked Questions (FAQ) section or by searching for the error message in the documentation.

Frequently Asked Questions (FAQ)

1. What is the purpose of this web application? The purpose of this web application is to provide an interactive tool for coding and visualizing sorting algorithms. It allows users to write their sorting algorithms in Python, visualize the sorting process, and compare the performance of different algorithms.

2. Do I need to install anything to use this web application? No, you do not need to install anything to use this web application. You can simply access it by clicking on the following link: *Show My Sort*.

3. Can I run this web application locally on my machine? Yes, you can run this web application locally on your machine. Please refer to the section How do I install it? for more information.

4. What programming language is used in this web application? The web application uses Python as the programming language for coding sorting algorithms.

5. Can I use any Python libraries in my sorting algorithms? No, you cannot use any Python libraries in your sorting algorithms. The web application restricts the use of certain modules and functions to ensure safety and functionality. Please refer to the section Environment safety for more information.

6. Can I customize the web page style? Yes, you can customize the web page style by switching between light and dark mode. This feature is available in the top right corner of the web page. The web application is designed to be responsive and works well on both desktop and mobile devices.

7. How do I report an error or provide feedback? If you encounter an error while running your code, you can use the feedback functionality to report the error. Simply click on the feedback button in the bottom left corner of the web page. This will open your mail application with a pre-filled email. Please provide as much information as possible about the error, including the code you were trying to run and any error messages you received.

8. How do I save my code? Your code is automatically saved every 2 seconds in the browser's local storage. You can also import and export your code using the import and export buttons in the top-right corner of the web page.

9. Can I use this web application on my mobile device? Yes, the web application is designed to be responsive and works well on both desktop and mobile devices. You can use it on your laptop, tablet, or smartphone without any issues.

10. What is this error: `Something went wrong... please try again`, What should I do? This error message indicates that an error occurred while running your code. You are not responsible for this error. I will not enter the details, but this error often occurs when your sorting algorithm requires a long animation time.

Please excuse me for this inconvenience, the best you can do is to try again. If you encounter this error frequently, please let me know by using the feedback functionality.

Resources to Cover Prerequisites

- Python Official Tutorial

Appendices

If you have some trouble coding your sorting algorithms, you can use the following examples as a reference.

```
def linearSort(arr):
 n = len(arr)
    for i in range(n-1):
        for j in range(i, n):
            if not compare(arr, i, j):
 swap(arr, i, j)

linearSort(myList)
```

This code is a simple implementation of a linear sorting algorithm. It uses the `compare` and `swap` functions to sort the array `myList`. The algorithm iterates through the array and compares each element with the others, swapping them if they are not in the correct order.

Try it out and see how it works! After understanding it properly, you can try to implement your sorting algorithm.

I hope this guide has helped get you started with this web application. If you have any questions or feedback, please feel free to reach out. Happy coding!