



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SORTING ALGORITHM VISUALIZER

Syed Shahaba, Student, Department of AI&DS, Rizvi College of Engineering

Razeen, Student, Department of AI&DS, Rizvi College of Engineering

Ansari Zubia, Student, Department of AI&DS, Rizvi College of Engineering

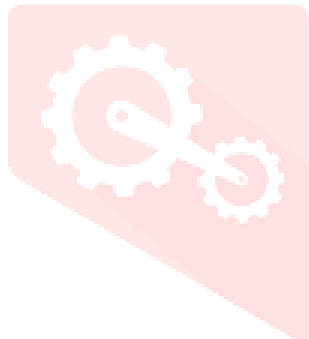
Prof. Tushar Surwadkar, Assistant Professor, AI&DS Department, Rizvi College of Engineering

Dr Varsha Shah, Principal, Rizvi College of Engineering

Prof. Nargis Shaikh, HOD, AI&DS Department, Rizvi College of Engineering

Abstract: With the recent surge in interest in computational thought, an important question has arisen: what are the best methods for teaching students basic computing concepts. Visualization is considered as one way to support student learning. This paper outlines a study that tested the benefits of animated sorting algorithms for teaching. To visualize three sorting algorithms, a web-based animation application was constructed. A visualization of data is implemented as a bar graph, after which a data sorting and algorithm may be applied. The resulting animation is then performed either automatically or by the user, who then sets their own pace. This is a research on the computer science curriculum's approach to learning algorithms. These findings and reactions are catalogued in this document and compared to earlier investigations.

Keywords: Sorting Algorithms, React Visualizer, Selection Sort, Bubble Sort, Insertion Sort.



INTRODUCTION

Visualization is proposed as one of the ways for supporting student learning. Me and my partners are visual learners, and we are keener to picking up concepts by seeing it get done, rather than reading about it. For example, when we were learning about sorting algorithms during the second semester Computer Science degree, we found it hard to grasp the working of a sorting algorithm just by seeing it explained on board or reading it as there is loads of recursion that takes place, we thought that maybe if these algorithms were taught visually and professors show how the data moves to its position, we would understand the concept better. Visualization can be incorporated into presentation software such as PowerPoint, which is a widely used slide ware, but critics argue that they are mostly used for decoration rather than just in innovative ways to promote learning. Visual representation has a lot of potential to improve learning and teaching at all stages, from pedagogical practice research to scholarship, connecting research and teaching, planning and curriculum development, and presentation and assessment, to name a few. To aid my visualization, I created a histogram of numerical data to represent three of the well-known examples. Each number is depicted as a bar, and the height of each bar represents the value of that number. It is being shifted by the algorithm from its original, unordered location to its final ordered place, making it distinct from the rest of the data. Selection Sort, Bubble Sort, Insertion Sort, are the three sorting algorithms. Let's imagine that you have printed each person's age on a separate index card. Bring the youngest card to the front and then sort the cards by age. To discover the next smallest item, identify the age that has already been ordered and position it behind the already ordered age. Index cards full of ages will be at the end of the pile. Selection Sort works in the same way as this. In this case, to sort a set of data, you select the smallest first, and then the next smallest, and so on until you've sorted all of the data. I wanted the animation to appeal to a wide spectrum of individuals utilizing various technology media, and so I had it made in a web-based format. Instead of requiring the user to install extra software or attempt to organize setups to use the tool, this helps to remove this source of anxiety. It uses HTML5 (Hypertext Markup Text Language) JavaScript, and CSS for the website's layout (Cascading Style Sheets).

Chapter 2 Review of Literature

There have been a lot of studies and animation tools on visualizing sorting algorithms. Some are based on how to make such an application and others on different technologies which were aimed for increasing the understanding of the concept.

The work that we looked up and which also helped us throughout this project was Clement Mihailescu Algorithm Visualizer. Other visualization tools include ViSA, Selection sorting Algorithm Visualization Using Flash to name a few. These applications had a few drawbacks therefore, the approach we focused on was making the visualizer more interactive and fix the drawbacks that the other visualizing tools had.

The paper "Algorithm Animation" by A. Kerren et al [1], is a bit-by-bit guide to show the means and necessary coding methods to use sorting visualization. The paper

mentioned about a web application-based tool which is outdated now, and the rest of the tools had us wanting to use other software's or applications.

The paper "SELECTION SORTING ALGORITHM VISUALIZATION USING FLASH" by Hadi Sutopo, The International Journal of Multimedia & Its Applications (IJMA) Vol.3, No.1, February 2011[2], discussed about only Selection Sort algorithm and visualizing it. The project was built using Adobe Flash and ActionScript programming tools and as of December 2020 Adobe Flash has been discontinued due to the rise security risk of using it.

In the paper "ViSA: Visualization of Sorting Algorithms" by I. Reif and T. Orehovacki, 2012[3] they provide an in-depth view of how sorting visually takes place on the data, but the limitation was that the data should be entered manually by the user which isn't effective if the user wants to enter large numbers of data.

In Clement Mihailescu's work, 2016 [4] not all the commonly used algorithms were implemented, the project was only half done as the user interface was not interactive and there wasn't any description about the sorting algorithms.

In the paper "Visualizing Sorting Algorithms" by Brian Faria, 2017 [5] though the work only had four commonly used sorting algorithms the implementation by him on using different techniques like using sound to show the swapping of data was different from others, but there wasn't any way for the user to control the speed at which the visualization takes place nor the user was able to control the size of the array which in my opinion was needed for better understanding of the algorithms when the data size is huge.

In 2019, paper "Open Interactive Algorithm Visualization" presents a work-in-progress project form developing an open interactive algorithm visualization website.

In the paper "Increasing the Engagement Level in Algorithms and Data Structures Course by Driving Algorithm Visualizations" by Slavomir Simonak, 2020[6], here he uses Algomaster platform, and the project is developed by C programming language, thus the user-interface isn't as interactive as a one uses HTML, CSS and JavaScript because these tools are more into web application development thus, making the UI more interactive.

In 2021, paper "Algorithm Visualizer" aims to simplify and deepen the understanding of algorithms operation. Within the paper we talk about the possibility of improving the standard methods of teaching algorithms, with the algorithm visualizations.

We have seen quite a few sorting visualizers online in YouTube. There were quite fascinating ways of visually showing how the data was sorted, the algorithm visualization showed in these videos were via dance. These videos are on posted on the YouTube channel known as AlgoRhythmic.

Chapter 3

Report on the Present Investigation

AIM

The primary aim of this proposed system is to help students who are naturally visual learners and tend to grasp concepts and remember it if they see it visually that is where SVA (Sorting Algorithm Visualizer) comes which helps people understand basic sorting algorithms by seeing them visually take place. This is not only for new learners but also for people who want to brushup on their knowledge of sorting algorithms helping them remember the topic faster rather than reading through it all over again.

SCOPE

Since SVA is a web application a large number of people can use it with ease without having the need of downloading any external software or application and this can be very helpful for students.

VISUAL LEARNING

People have natural tendencies to communicate visually, and visuals play a huge part in how the information is processed. Primitive cave drawings show an original desire to use pictures in an attempt to convey thoughts and meaning. Even today, most toddlers seek to make meaningful marks, drawing rudimentary pictures to show what's going on in their heads. Generative Art is one of the ways to represent computational processes. Transforming the data generated by an autonomous system into the features of an artwork can lead to unexpected results. SAV was born to create visual representations of sorting algorithms with the hope of finding visual patterns. It turned out that the visual footprints of algorithms are unique and differ from each other and helps people who are new to the concept of sorting understand these algorithms quicker and the way they work.

Chapter 3.1

OBJECTIVE AND

ALGORITHM PROPOSED

SYSTEM

The design and structure of the user interface components has remained unchanged even if the underlying back-end code was refactored midway through the construction. Each component has its own feature. The main objective of Sorting Visualizer project is to learn and visualise the sorting algorithms how these algorithms works.

We can see the steps of swapping or any divide and conquer part of these bars. Here 3 different types of sorting algorithms is used to visualise. A visualization of data is implemented as a bar graph, after which a data sorting and algorithm may be applied.

The resulting animation is then performed either automatically or by the user, who then sets their own pace.

REQUIREMENT SPECIFICATION

This proposed software runs effectively on a computing system that has the minimum requirements. Undertaking all the equipment necessities are not satisfied but rather exist in their systems administration between the user's machines already. So, the main need is to introduce appropriate equipment for the product. The requirements are split into two categories, namely:

SOFTWARE REQUIREMENTS

The basic software requirements to run the program are:

1. Windows 10
2. Visual Studio Code
3. HTML, CSS, JavaScript
4. Browser example Mozilla Firefox, Safari, chrome, etc.

HARDWARE REQUIREMENTS

The basic hardware required to run the program are:

1. Hard disk of 5 GB
2. System memory (RAM) of 512 MB
3. Processor-based computer or higher.

OBJECTIVE

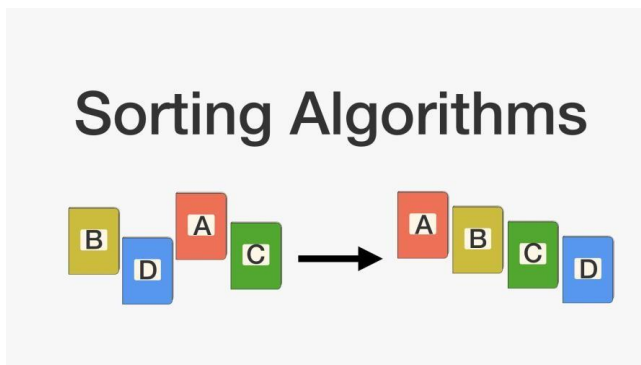
The main objective of Sorting Visualizer project is to learn and visualise the sorting algorithms how these algorithms works.

Design a web-based sorting algorithm visualizer, enabling users to interactively compare and learn sorting techniques with real-time animations and user-controlled execution.

Implement responsive design for mobile compatibility and provide comprehensive algorithmic insights for a technical learning experience

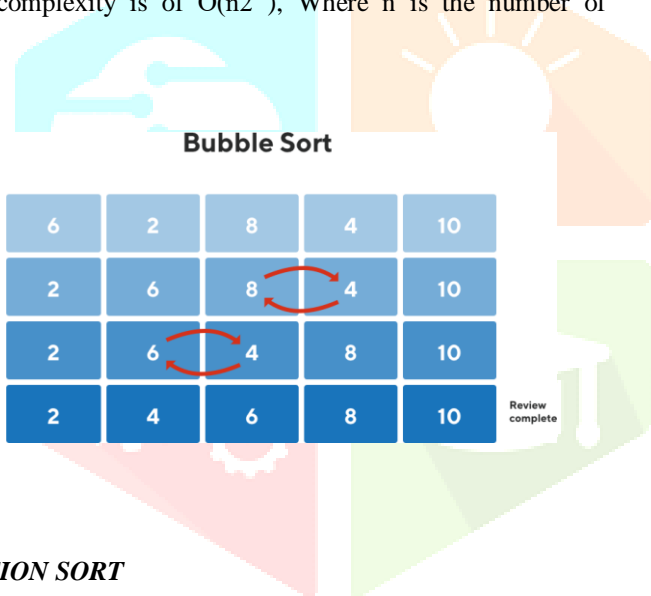
SORTING ALGORITHMS

A Sorting Algorithms main purpose is to reposition a given array or list of elements in ascending or descending order. Now talking about the algorithms itself we are using 3 sorting algorithms in this visualizer namely Bubble Sort, Selection Sort, Insertion Sort These algorithms are the commonly used algorithms for sorting.



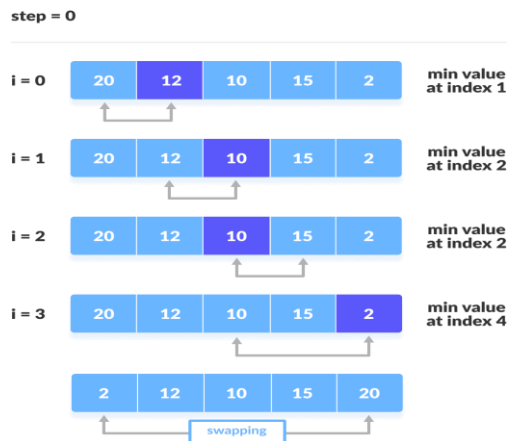
BUBBLE SORT

Bubble sort also known as sinking sort from time to time, 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. But the algorithm is not suitable for amounts of data as its average complexity is of $O(n^2)$, Where n is the number of elements.



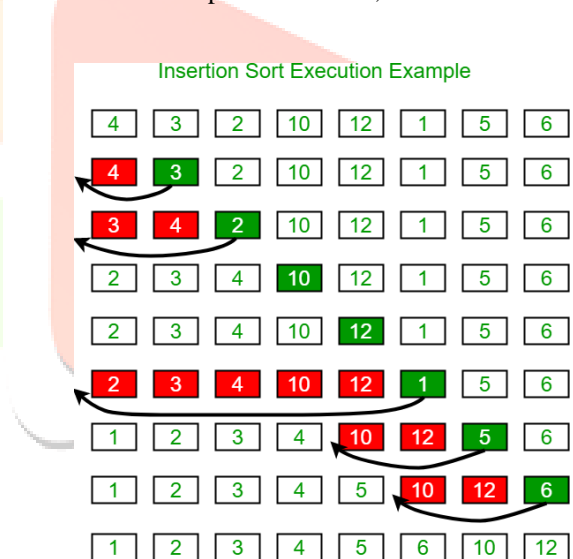
SELECTION SORT

Selection sort is an in-place comparison sorting algorithm. It has an $O(n^2)$ time complexity, which makes it unsuitable for an array size of large numbers, and in general executes worse than the similar insertion sort. Selection sort is well known for its simplicity and has performance advantages over more intricate algorithms in certain circumstances, particularly where additional memory is limited. The algorithm works by pushing the least element to the start of the array by swapping it with the first element of the array and the pointer pointing to the first element is moved to the next element and this process goes on till the last element of the array and eventually the array will be sorted.



INSERTION SORT

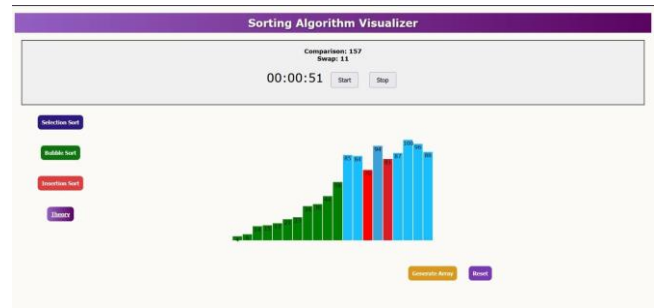
Insertion Sort is an in-place comparison sorting algorithm. In this algorithm, a sub array is kept which is always sorted. An element which needs to be inserted in this sub array, has to find its right position where it should be and then it will be inserted in that position. Hence, it's called Insertion Sort.



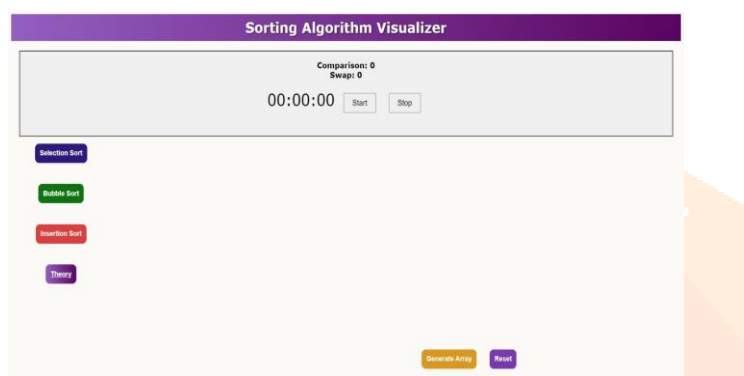
Chapter 4

RESULTS AND DISCUSSIONS

The main goal was to keep the web page and user interface as simple as possible so that it's easy for the user to navigate through the page. All the similar elements are coupled together. In this system the user interface build is professional. It is the front end of the project or it can also be termed as user interface. Here the user gets three options to execute or access their tasks per need.



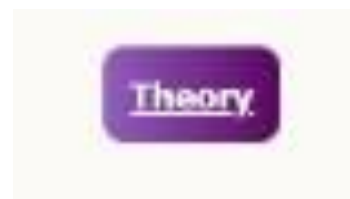
By clicking the generate array buttons the process will start. We have click selection sort button and the process as been started .It has been showing the numbers of comparison and number of swapping that has been take place uptill now. Also if we want to set the timer onn we can do that. The timer will tell the time that how much the time a sorting is taking.



These id the front page of the website. On the centre the title of the website will be written and below that the numbers if comparison that will be made will be showing. Along with numbers of comparison the numbers of swapping of the order will also displayed..There is a timer to count the time taken by each ofthe sort.

There are altogether 8 buttons and each buttons has its own functionality.The first button is Selection sort and below that bubble sort button is there and below that will be insertion sort button. At the down centre there will be two button. One for generating the array and second one isfor reset the graph.Then there is a timer button that is starting the timer and stoping the timer. First we have to click on the generate array button .By doing this a graph will be display with numbers .These number will be the provided automatically. You can also reset the graph if you want.

This is the final output of the selection sort .The total number of comparison and the total number of swapping is also been shown.Plus the total time taken by the selection sort is also been shown. Asking to specify the ordering of elements takes precedence because when the algorithm has completed running the initialization process, the interface is now showing a new ordering, while the code has already completed running the initialization with the prior data set.



There is a theory button

By clicking on the insertion sort bar ,information about selection sort is been shown.

On clicking the theory button a page will open in which information about sorting algorithm is shown. There are three bar in which information about the three sorting algorithm that is selection sort , bubble sort and insertion sort is been shown respectively.

Chapter 5 Conclusions

Relationship between concepts and their theory can be more easily and quickly understood by students using diagrams and visualization rather than in textual form. The result presented here doesn't indicate that it outperforms black board learning but shows a different way of engaging students to learn not just by reading and writing but also by engaging them visually.

It introduces the proposed approach for sorting visualization, the implementation details, detailed description of the website, and project directory. The results on sorting visualisation and also time complexities of selection sort, bubble sort, insertion sort in all the cases (i.e., best case, average case and worst case) are shown.

The available evidence of its effectiveness is decisively mixed. To summarize, although some successes in both research and education have been reported in the literature, they are not as impressive as one might expect.

By clicking on the selection sort bar ,information about selection sort is been shown.

By clicking on the bubble sort bar ,information about selection sort is been shown.

Chapter 6

References

- [1] CORMEN, T. H.; LEISERSON, C. E.; RIVEST, D. L.; STEIN, C. Introduction to algorithms. Second Edition. 2001. ISBN 0-262-03293-7.
- [2] KNUTH, D. The Art of Computer Programming: Sorting and Searching.
- [3] SEDGEWIK, R. Algorithms in C : Fundamentals, data structures, sorting, searching. Third Edition. 2007. ISBN 0-201-31452-5.
- [3] GeeksforGeeks. Available from:<https://www.geeksforgeeks.org/i>.
- [4] Stackoverflow. Available from:<https://stackoverflow.com/i>.
- [5] Bingmann. "The Sound of Sorting - 'Audibilization' and Visualization of Sorting Algorithms." Panthemanet Weblog. Impressum, 22 May 2013. Web. 29 Mar. 2017.
- [6] A. Kerren and J. T. Stasko. (2002) Chapter 1 Algorithm Animation. In: Diehl S.(eds) Software Visualization. Lecture Notes in Computer Science, vol 2269. Springer, Berlin, Heidelberg.
- [7] J. Stasko. Using Student-built Algorithm Animations As Learning Aids. Proceedings of the Twentyeighth SIGCSE Technical Symposium on Computer Science Education. SIGCSE '97 (San Jose, California), pages 25-29. <http://doi.acm.org/10.1145/268084.268091>

