International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT

IJIEMR Transactions, online available on 02th May 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 05

10.48047/IJIEMR/V12/ISSUE 05/01

Title SORTING VISUALIZATION OF ALGORITHM USING DIFFERENT TYPES OF SORTS.

Volume 12, ISSUE 05, Pages: 01-08

Paper Authors

Rushikesh Tangade, Gitanjali chavhan, Bhagyashri Dadhe, Tejal Chaudhari, Yash Tarachandi, Prof. Ritesh Deshmukh

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

SORTING VISUALIZATION OF ALGORITHM USING DIFFERENT TYPES OF SORTS.

Rushikesh Tangade[1], Gitanjali chavhan[2], Bhagyashri Dadhe[3], Tejal Chaudhari[4], Yash Tarachandi[5], Prof. Ritesh Deshmukh[6]

[123456] B.E Student of Computer Engineering, Jagadambha College of Engineering & Technology, Yavatmal, Maharashtra, India

[6]Assistant Professor at Jagadambha College of Engineering & Technology Yavatmal, Maharashtra, India

Sant Gadge Baba Amravati University, Amaravati, Maharashtra, India

---

ABSTRACT:

This paper outlines a study that dependable the benefits of animated sorting algorithms for teaching. To visualize four sorting algorithms, a web-based animation application was established. Visualization of data is implemented as a bar graph, after which a data sorting and algorithm may be adapted. The resulting animation is then performed either automatically or by the user, who then sets their own pace. This is research on the computer science subject's approach to learning algorithms. The experiment featured a presentation and a survey, both of which asked students questions that may illustrate improvements in algorithm conception. These findings and reactions are cataloged in this document and compared to earlier investigations Selection Sort, Merge Sort, Bubble Sort, Insertion Sort, Heap Sort.

Keywords: *Sorting, visualization, algorithm, Implementation, animation control*

INTRODUCTION:

How do you get something done? You don't have to be highly complex in solving the problem, for example, if your car's headlight is broken (although nowadays, manufacturers are trying the patience of the community with their increasingly abstract, space-age designs). The main argument is figuring out the best way to set about it. To locate step-by-step directions in your car's handbook, do you conduct research, or do you use instinct to find someone who knows how to do it? In short, my instinct tells me that I am a visual learner and hence more suited to acquire topics by watching them than by reading about them. In this case, I found that seeing the data move to its rightful

spot as the result of an algorithm is MUCH easier to follow than looking at the source code and trying to figure out where the data was supposed to drive. My project was born out of my curiosity about sorting algorithms, which inspired the idea for this paper, which details an online tool I built that explains how sorting algorithms transform and organize sets of data. It is possible to organize a list of people, for example, by their age in ascending order using different methods. To aid my visualization, I created a histogram of numerical data to represent four 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, and Merge Sort is the four sorting algorithms. 2 Let's imagine that you have printed each person's age on a separate index card. carry the youngest card to the anterior 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 boundary of the hill. 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. This technique is quite simple to explain to someone in conversation, but more advanced sorting algorithms, such as Quick Sort, which requires the data to be moved around a pivot point, are not easy to grasp using text alone. I wanted the animation to appeal to a wide spectrum of individuals utilizing various technology media, 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.

LETRATURE SURWAY:

This chapter reviews previous work done on sorting algorithms. The occasion of reviewing previous work is to guide through the previous work done by different people to sanction us guide through the research topic. The literature review reports the five sorting algorithms that have considered in this study. Bubble sort, Merge sort, insertion sort, selection sort and heap sort Each algorithm is defined and described with an example and their efficiencies and other properties are discussed. Review of related work on sorting state that sorting is graded as a fundamental problem in computer science.

After comparing Bubble, Heap, Insertion, Merge and selection sort algorithms have wine up that quality of a great sorting algorithm is not only the speed but other factors like length, code complexity, stability, performance consistency and data type handling should also be taken into consideration. have compared Insertion and Quick sort algorithms on both integer and character arrays concluded that performance of insertion short is better than quick sort. One and the other algorithms are for sorting portable number of items.

PROPOSED SYSTEM:

The purpose of our project was to develop an educational tool that would engage students in the learning process, helping them to acquire knowledge about well-known sorting algorithms.

Visualization of sorting algorithms (VISA) offers a full range of functionalities such as data set entry and animation control as well as the explanation and detailed algorithm analysis. Currently, data is the warm stock of the IT industry. All the large co-operations are vying to collect as much data as possible. While data collection is desired, the collection and storage of such data are time-consuming and costly. Yet, the benefits of having data collection outweigh them in the long run. Data are important to understand the perform-mance, accuracy, approval, etc. of the product and it helps businesses analyze them and improve their productivity and short comings. Examples of data collected by comp- nies can be information on the fitness tracker, internet browsing history, IP address, time, and frequency of using an app or device, etc. As important it is to collect and store data,it is also important to organize the data in a meaningful manner. Organization of data includes sorting of data, it provides proper structure to the data that makes it analyzable, comparable to another set of data, and accessible as fast as possible.

IMPLEMENTATION:

The use of HTML5 (Hypertext Markup Language 5), JavaScript, and CSS combine to form this project's implementation (Cascading Style Sheets). There is only one project file which is an HTML file thatcontains the code. The only additional piece of code added to the main HTML file is the .m4a sound files to support the sound animation functionality (which are saved as .m4a files). As of now, I only did extensive testing using the Mozilla Firefox browser, and it's the browser of choice in this context. However, tests done quickly revealed the possibility of Google Chrome and Safari integration. This software uses both object-oriented and functional programming paradigms in how

it organizes the code. Before the final phase of development, the design was almost completely functional, where only three objects were used: one to control the canvas that displayed the animation, another to represent a piece of data, or a "bar" object (blue rectangle with dynamically changing height and position), and a final one to represent the positions that each bar moved to, or "pos" objects. Although this incorporated several function calls, some instance variables and Boolean values were utilized to keep track of the algorithm picked and when to animate, but this led to a greatly integrated mass of code that was difficult to maintain. Several big refactoring's later, the code has now taken on the form of a Model-View-Controller Architecture. Although, because of its functional character, it possesses a multitude of individualized functions that alter the instance variables and Boolean values, which means it has a multitude of functions that directly alter the View and Controller. The major module in the HTML code between the tags is known as the global scope. Everything within the framework can access the aforementioned variables and methods.

1)The View: There are three items on the view: the sort area, the bar, and the position. These objects operate within the container defined by the tags in the html File. This function space is somethings referred to as the main function. The first function invoked in a program.

```c
#include <stdio.h>
int main() {
  int array[100], n, i, j, swap;
  printf("Enter number of elements n : ");
  scanf("%d", &n);
  printf("Enter %d Numbers: n :", n);
  for(i = 0; i < n; i++)
    scanf("%d", &array[i]);
    for(i = 0 ; i < n - 1; i++) {
      for(j = 0 ; j < n-i-1; j++) {
        if(array[j] > array[j+1])  {
          swap = array[j];
          array[j] = array[j+1];
          array[j+1] = swap;
        }
    }
}
printf("Sorted Array: n ");
  for(i = 0; i < n; i++)
    printf("%d ", array[i]);
return 0;
}
```

```
Enter number of elements n : 5
Enter 5 Numbers: n :7 5 3 2 4
Sorted Array: n 2 3 4 5 7
```

Bubble sort algorithm

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

```c
#include <stdio.h>
int main() {
  int array[100], size, i, j, position, temp;
  printf("Enter Number of Elements : ");
  scanf("%d", &size);
  printf("Enter %d Numbers : ", size);
  for (i = 0; i < size; i++)
    scanf("%d", &array[i]);
  for (i = 0; i < (size - 1); i++) {
    position = i;
    for (j = i + 1; j < size; j++) {
      if (array[position] > array[j])
        position = j;
    }
    if (position != i) {
      temp = array[i];
      array[i] = array[position];
      array[position] = temp;
    }
  }
  printf("Sorted list in ascending order:\n");
  for (i = 0; i < size; i++)
    printf("%d\t", array[i]);
  return 0;
}
```

```
Enter Number of Elements : 5
Enter 5 Numbers : 7 5 2 1 8
Sorted list in ascending order:
1       2       5       7       8
```

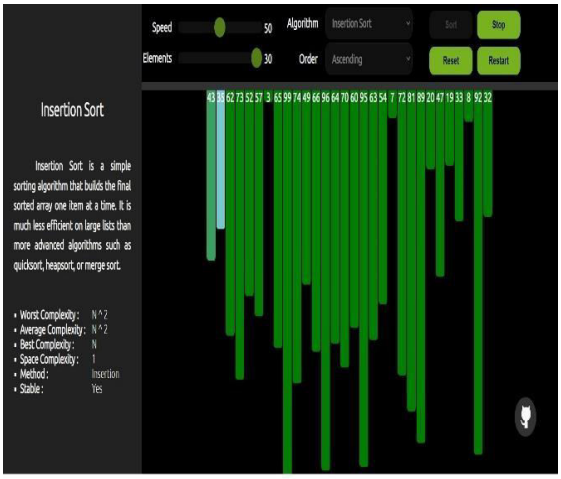
Selection sort algorithm

It is the sort area that keeps the bars up to date using a timer, while at the same time generating the bar graph. As a result, whenever "Step" is invoked, the bar values are updated depending on the steps array (discussed later). In the sort area, after every second iteration of the timer, the rectangles will be redrawn with varied heights that represent the new values. The bars change sixty times per second, so when the “Step” button is selected, the change is instantaneous. In the sort area, the bar object represents each piece of data. The statement encompasses all of the aspects of color, value, location, height, and sound. While having a distinct array named bars for the current bars in the bar graph helps preserve attributes such as the total number of bars (total value) independent of other characteristics, it is simple to update any or all of the attributes by iterating over the array as necessary. To update the bars to a fresh data configuration, the In Order, Reverse, and Random buttons iterate fast across the array. The bar object is related to the pos object (which is short for position). The region on the canvas that is updated when the Sort Area event fires is an x- y pixel coordinate grid. This item was made to make arranging the bars a little bit simpler (1- 32). Thus, if I wanted to move a bar, I would supply merely the number of the bar's location. For the bar to move, it will first determine the exact coordinates and then go to that location. Another way to say this is to say that, position one is defined by the two-dimensional coordinate pair (9, 135), which is the bottom left-hand corner of the bar. As long as each bar has a rectangular object that is associated with it, as well as a top left-hand corner point that defines the rectangle's height, the bar must be relocated to its right location.

B) The Model:

The model is made up of one item, known as the sorter. This object houses the algorithm's code divided into methods. The start method centralizes on an integer constant and uses it to order the

algorithm's possible algorithms. This object is directly controlled by the four sorting algorithms shown on the user interface as "Selection Sort, Bubble Sort, Insertion Sort, and Merge/Insertion Sort." The sort algorithm method is invoked as the user selects a sorting technique and clicks on one of the sort algorithm buttons. Then, when the algorithm sorts the data, a trace is created. The steps array, which contains all the movements in the animation, is a two-dimensional integer array that is available to methods on the web user interface. A typical back-end code interface is implemented using the steps array. The "Start," "Stop," and "Step" buttons function as controllers for which sub-array will be displayed on the canvas after the computational back-end has completed the tracing. If this loads before the user have selected "Start" or "Step", then this represents a user action. When you click the "Step" button, the next step on the canvas loads and animates. An animated sequence of a single bar moving across the others is produced because the timer continually redraws the bars. There's no "Start" button, only a "Step" button that is set to go off on a timer. Using a two-dimensional array gives you the option to view the sorting algorithm's stages within the View. The process of adding an algorithm is similar to writing down the trace of the new algorithm, which is then saved in the same location. To complete the algorithm's walkthrough, the View will cycle through the data and update the bars in the bar graph to show how the algorithm calculated the steps it took. It's important to note that if the algorithm generated a change in the position of a piece of data, the steps are merely recorded.



View of insertion

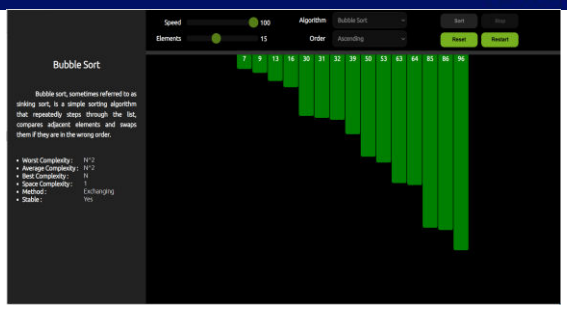
RESULT:

Start by organize the data, and then choosing the visualization algorithm to use. Algorithm buttons add sorting of data as it appears on the interface. 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 previously finished running the initialization with the former data set. There was considerable confusion caused

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

by the way the ordering buttons and algorithm buttons were shown in the UI after the surveys were completed. When beginning the sorting process, the student noted that she was having problems starting because she believed that she was hitting the buttons in the wrong order. This then led to her failing to execute the animation. The answers can be established in Appendix D. Overall, animation tool did not aid with the understanding of sorting algorithms. Among those who answered question 3, which questioned if their knowledge of a particular algorithm changed after using the tool, 5 of the 13 students (38%) stated that they had in some way updated their preceding knowledge of the algorithm. Many thought the tool was a good concept, while the other 7 did not find it useful at all.

Unsorted sort



Sorted sort

CONCLUSION:

This web-based effervescence for viewing the following sorting algorithms behavior in great part because of all the time and attempt that I invested into it.

In enmity of its memory aerial, the feedback given to it was mostly great from the students that performance with it.

This is consistent with my prior research, which revealed that there was no consequential difference in learning the content.

What I do assent with totally is the point of view that holds there is a great need to inquire into and produce animated demonstration to enhance education in the classroom.

Overall, I am not troubled that a large rework to a different language will be appropriate soon because JavaScript is still one of the most famous web languages.

REFERANCES:

T. Bingmann. "The Sound of Sorting - 'Audibilization' and Visualization of Sorting Algorithms." Panthemanet Weblog. Impressum, 22 May 2013. Web. 29 Mar. 2017.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, D. L.; STEIN, C. Introductionto algorithms. Second Edition. 2001. ISBN 0-262-03293-7.

V. Karavirta and C. A. Shaffer, Creating engaging online learning material with the JSAV javascript algorithm visualization library, IEEE Transactions on Learning Technologies 9(2) (2015), 171-183.

A. D. Mishra and D. Garg, Selection of best sorting algorithm, International Journal of Intelligent Information Processing 2(2) (2008), 363-368.

G. Tuparov, D. Tuparova and V. Jordanov, teaching sorting and searching algorithms through simulation-based learning objects in an introductory programming course, Procedia-Social and Behavioral Sciences 116 (2014), 2962-2966.