**Sorting Visualizer using Animation**

A Synopsis Submitted in Partial

Fulfilment of the Requirements

in the Course of

**Minor Project – II**

In

Third year – Sixth Semester of

**Bachelor of Technology** with specialization in

**Artificial Intelligence and Machine Learning**

Under guidance

Of

**Ms. Niharika Singh**

By

|  |  |  |
| --- | --- | --- |
| **500067776** | **R177218105** | **Yash Nabh Rana** |
| **500067111** | **R177218033** | **Chahat Bansal** |
| **500065751** | **R188218082** | **Shashank Panthri** |



DEPARTMENT OF INFORMATICS

SCHOOL OF COMPUTER SCIENCE

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES,

BIDHOLI, DEHRADUN, UTTRAKHAND, INDIA

**February, 2021**

# Index

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Topic** | **Page No.** |
| 1 | Abstract | 3 |
| 2 | Introduction | 4 |
| 3 | Literature Review | 5 |
| 4 | Motivation | 5 |
| 5 | Objectives | 6 |
| 6 | Methodology | 6 |
| 7 | Output | 7 |
| 8 | Result Analysis | 9 |
| 9 | Future Scope and Conclusion | 9 |
| 10 | References | 10 |

# Abstract

# 

Sorting is an important and widely studied issue, where the time complexity, execution time and the required resources for computation are of extreme importance, especially while dealing with real-time data processing. Vast use of sorting algorithms has allured a great deal of research and development to better refine sorting algorithms; because efficient sorting is important to optimize the use of other algorithms. Hence, it is important to study, compare and analyse thoroughly few notable sorting algorithms.

This project aims to build a sorting visualizer which can help in visualizing the internal working of algorithms and how a particular algorithm selects array elements and advances these elements to their sorted position using animation and the execution time is computed.

*Keywords: Sorting, Sorting algorithms, real-time data processing, visualization, animation.*

# Introduction

Algorithms are the most basic building blocks of an programmer. They are always required if you want to work in the programming industry. They can be used for problem solving, gaining better efficiency or better time management. One of the fields of these algorithms are Sorting algorithms that can be sort various data points with different methods that provide varying efficiency and complexity.

Sorting is an important and widely studied issue, where the Time Complexity, Execution Time and the required resources for computation are of extreme importance, especially while dealing with real-time data processing. There are many kind of sorting algorithms out there that have varied complexity and efficiency. Although the algorithms are not that difficult to understand, the human brain still understands visuals the best instead of some long written code.

Unfortunately, this task cannot be described by a static drawing; it requires a dynamic sequence. This can be done using static images but understanding the algorithm using this method will be difficult and not long lasting. Thus it would be useful to have animation sequences portraying the behaviour of programs constructed automatically as a by-product of their execution, and therefore guaranteed to portray this execution faithfully.

Animation is a compelling medium for the display of program behaviour. Since programs are inherently temporal, executing through time, they can be vividly represented by an animated display which portrays how they carry out their processing and how their essential state changes over time. Furthermore, many algorithms employ repetitive computations, whether expressed iteratively or recursively. These can be viewed efficiently when displayed as a motion picture. In so doing, we will better our understanding of the algorithms and have a deeper understanding of how they work and since it is an animation, it will last with the user for a long time.

Software visualization can therefore be a powerful tool for presenting computer science concepts and assisting students as they struggle to comprehend them.

# Literature Review

Over so many years of exponential growth and advancement in technology, there has been many successful attempts in visualising various sorting algorithms. While some animation works on the same topic have been focused on comparing various sorting algorithms, some are based only on analysing different algorithms independently.

Since a lot of work has been done on the topic, many projects have used ‘pygame’ for animation, library in python which is used in the course of development of the project.

We were taught sorting algorithms and its importance in our ongoing course of Bachelor in Technology in Computer Science and we took this chance to study, analyse and animate the same.

VisuAlgo was conceptualized in 2011 by Dr Steven Halim as a tool to help his students better understand data structures and algorithms, by allowing them to learn the basics on their own and at their own pace. VisuAlgo is an ongoing project and more complex visualizations are still being developed.

The paper ‘Algorithm Animation’ by A. Kerren and J. Stasko [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] has been very helpful and provided a clear understanding for our project.

Also, many projects that we found on internet has been helpful. Links for these projects like [https://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html] have been very helpful and are provided in the references.

**Motivation**

Throughout the course of our academic journey, we observed that pretty much all of the members of the group are visual learners i.e. we absorb information better when we see it. Naturally, a classroom is a very good place for visual learners like us. However, in a crucial subject like Algorithms, there was something left to be desired in the visualization department when we went through the course. Therefore, the desire to work on this project stems from that experience that a lot of us students face when learning about algorithms. We figured out it was about time to revisit algorithms and combine it with computer graphics and produce a much-needed visualization tool for us.

**Objectives**

# To implement a visualizer using python scripts which can help in visualizing several sorting algorithms and generate the animations for same via Matplotlib/Pygame.

1. Designing and creating a user friendly interface.
2. Taking input for algorithm, size, and the speed of animation user chose.
3. Processing each input and saving information.
4. Displaying animation as per options picked by the user.

# Methodology

An interactive sorting visualizer will be implemented in python programming language using pygame library for the visualization of different sorting algorithms. Input data will be taken from the user to perform real time data pre-processing.

1. **Creating Interface:**

We use pygame inbuilt library to create the interface and use basic shapes and colors to design the interface.

1. **Taking input:**

Input is to be taken in form of x, y – coordinates and checked for collisions.

1. **Saving Information:**

Information is saved in various variables for each valid input and is then used to set the parameter for the animation.

1. **Displaying Animation:**

A set of code will run in a loop with different arrangement of array every time, visualizing the internal working of the sorting algorithm.

Sorting of data elements in an application view in front of user helps in deepening the understanding of working of the algorithms in the process.

# Output:

For example, choosing

FIG 1.1

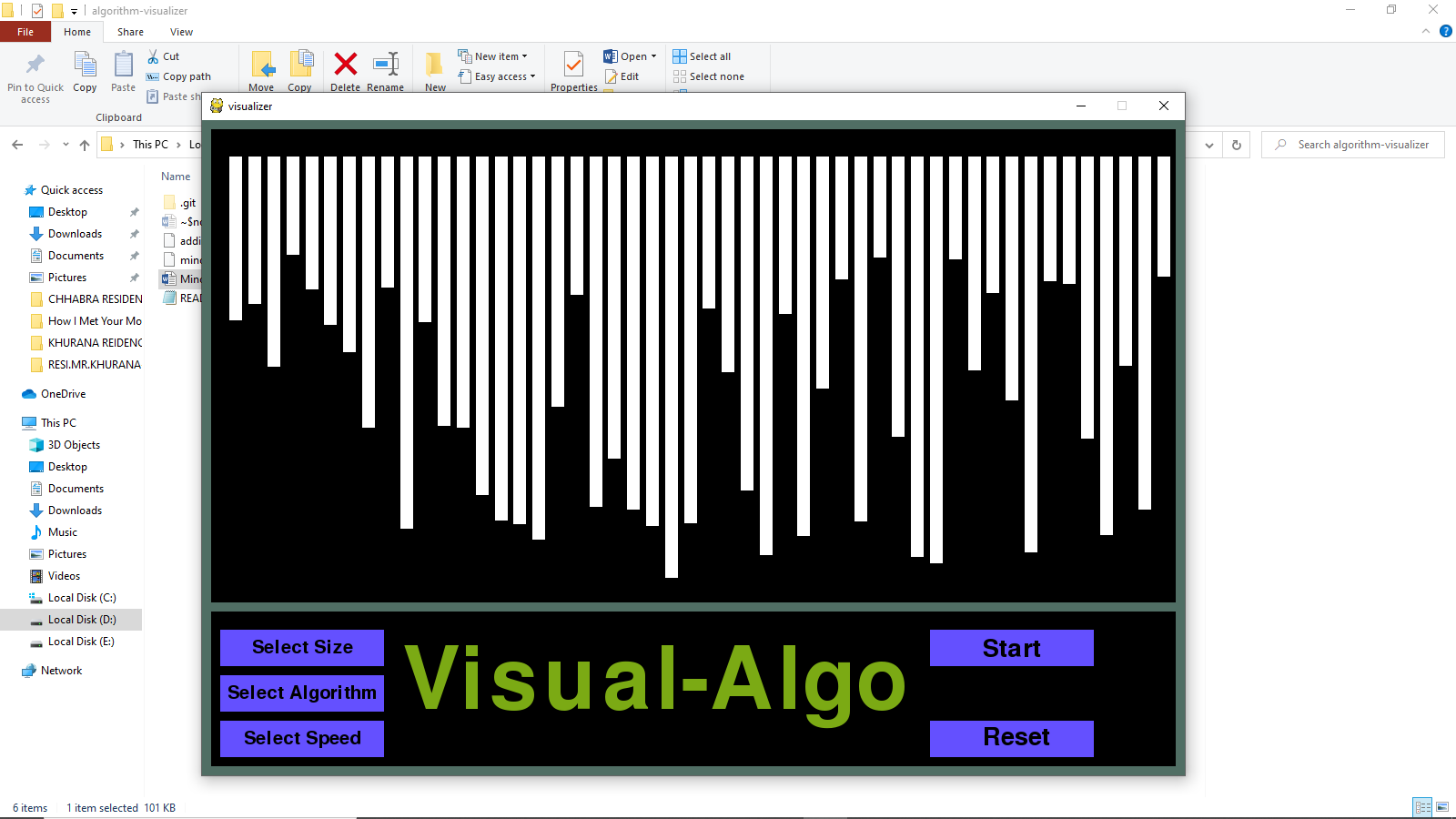


FIG 1.2

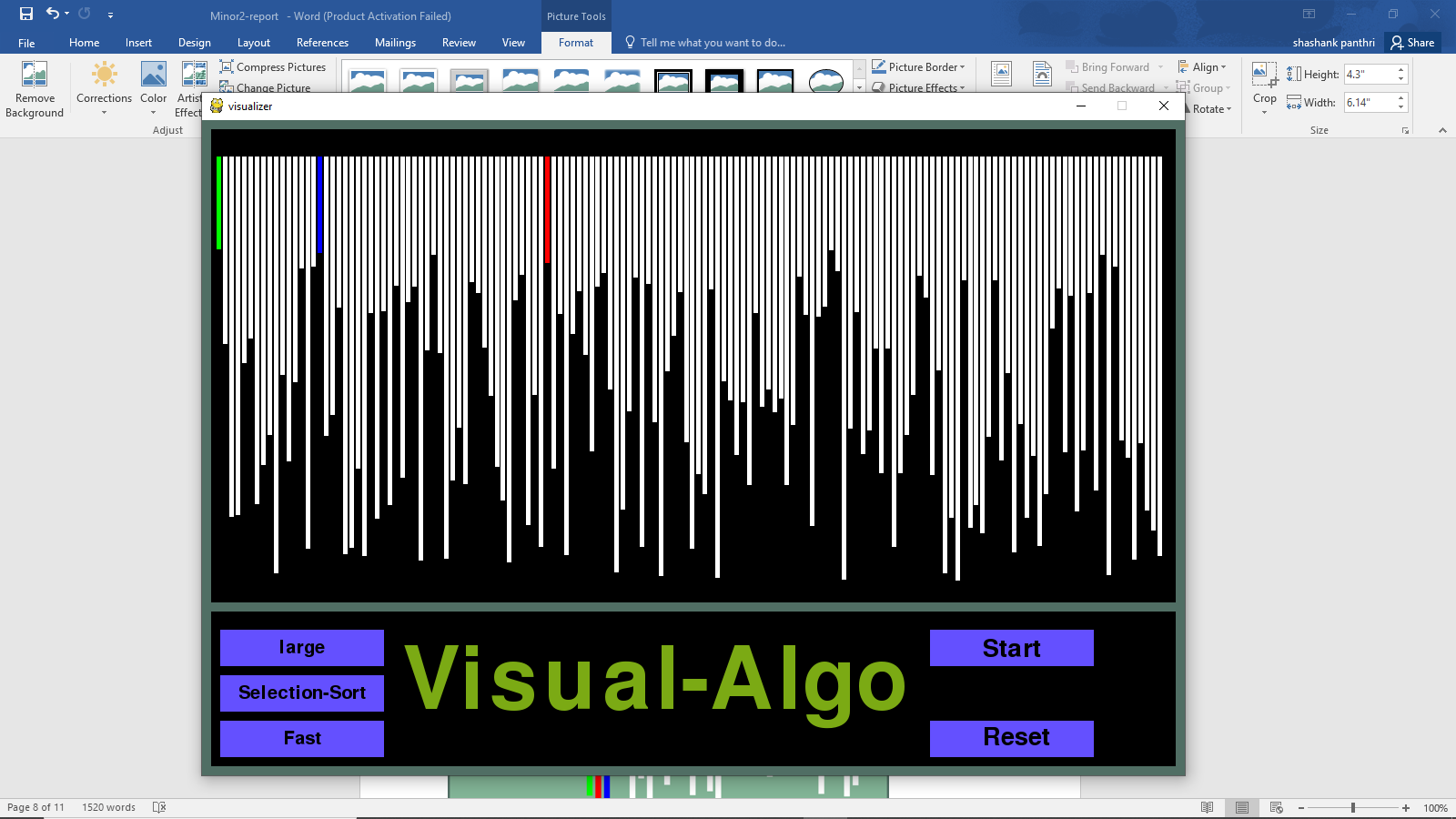


FIG 1.3

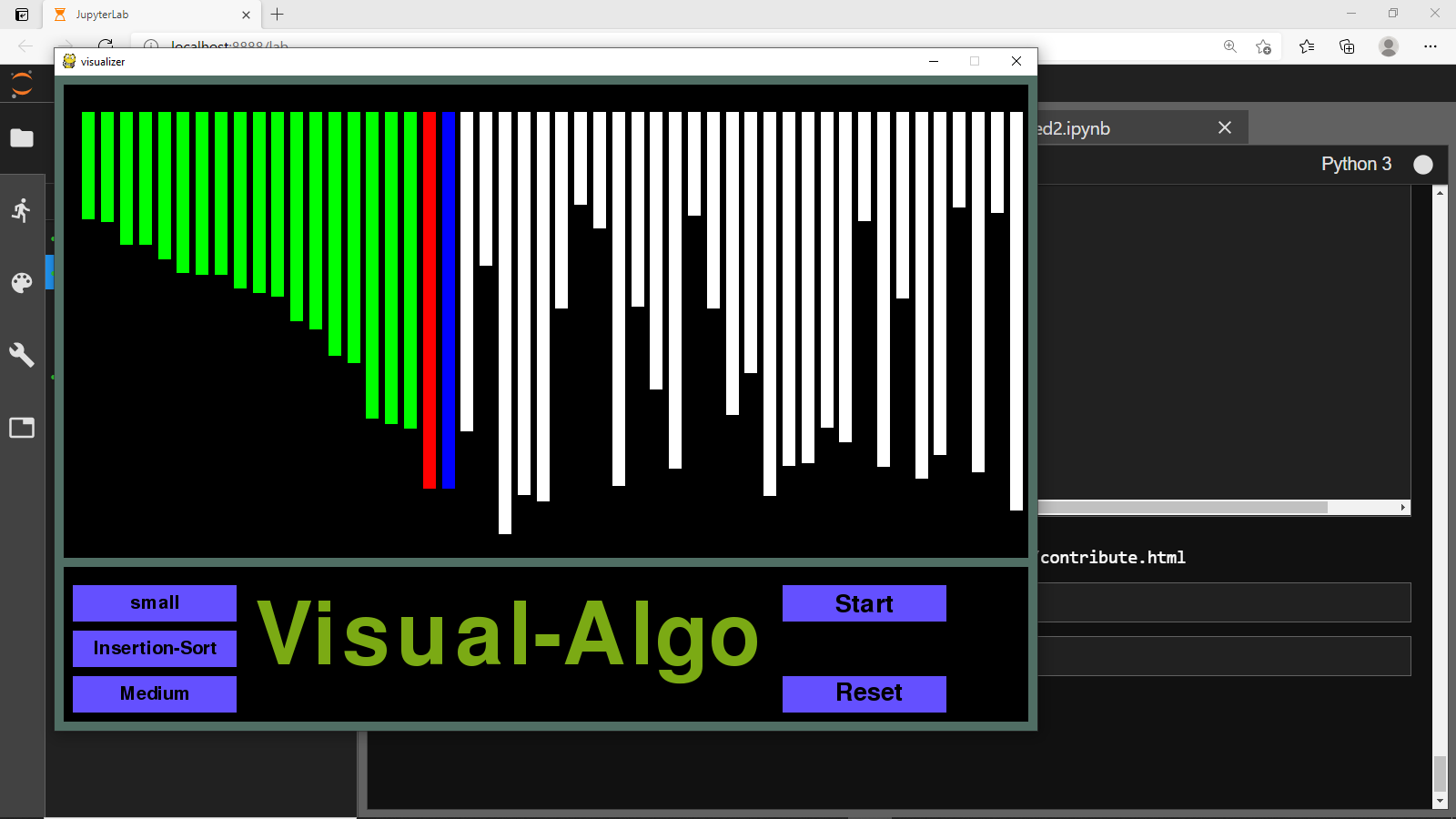


FIG 1.4



**Result Analysis**

The best way to go about using the tool is to first select the ordering of the data and then select which algorithm to visualize. When any one of the algorithm buttons are selected, it will sort the data as it appears on the interface. The ordering takes precedence, as selecting the ordering after the algorithm updates the interface momentarily, while the code has already run the initialization with the previous data set. After conducting the surveys, this sparked some confusion as the algorithm buttons are listed above the ordering buttons in the interface.

**Future Scope and Conclusion**

Through much time and effort, we have successfully created a working animation tool for visualizing the following sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, and Merge Sort. Analyzing the time it takes for the sorting animations to complete, is beneficial for comparative analysis. Even though the animation tool works locally, we are planning to give this project a form of web application, where a server can handle multiple requests to the web site by different users. This would also present some new challenges and would require more thought on how to optimize the code so that it can work with multiple people using it. We can use visualizations using animations in many other use cases like in path finding problems with a different user interface. I would need to analyze the time it takes for the sorting animations to complete, which would be beneficial for comparative analysis.

# References

1. A. Kerren and J. T. Stasko. (2002) Chapter 1 Algorithm Animation. In: Diehl S.(eds) Software Visualization. Lecture Notes in Computer Science, vol 2269.
2. https://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html
3. https://dev.to/kgprajwal/build-a-sorting-visualizer-in-python-2oej
4. T. Bingmann. “The Sound of Sorting - ‘Audibilization’ and Visualization of Sorting Algorithms.” Panthemanet Weblog. Impressum, 22 May 2013. Web. 29 Mar. 2017.
5. A. Moreno, E. Sutinen, R. Bednarik, and N. Myller. Conflictive animations as engaging learning tools. Proceedings of the Koli Calling ‘07 Proceedings of the Seventh Baltic Sea Conference on Computing Education Research - Volume 88, Koli ‘07 (Koli National Park, Finland), pages 203-206.
6. Segura, Clara, Pita, Isabel, del Vado Vírseda, Rafael, Saiz, Ana Isabel, & Soler, Pablo. (2008). Interactive Learning of Data Structures and Algorithmic Schemes Computational Science–ICCS 2008.