



NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)

Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC

☎: 08258 - 281039 - 281263, Fax: 08258 - 281265

Department of Computer Science and Engineering

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

Report on Mini Project

“Visualize Sorting Algorithms”

Course Code: 17CS703

Course Name: Computer Graphics and Multimedia

Semester: 7

Section : C

Submitted To,
Mr. Pawan Hegde

Submitted By:

Name:
Sahil Bhanastarkar
Rohit J
Sachin Choubey
Rishav Shekar

USN:
4NM17CS156
4NM17CS148
4NM17CS153
4NM17CS147

Date of submission: 26/12/20

Signature of Course Instructor

Visualize Sorting Algorithms

Abstract

- 1. Introduction**
- 2. Problem statement**
- 3. Objectives**
- 4. Related reading / Literature**
- 5. Solution approach / methodology**
- 6. Implementation details**
- 7. Results**
- 8. Observations / conclusion**
- 9. References**

1. INTRODUCTION

This project introduces a simpler way to animate and visualize three kinds of sorting algorithms. An OpenGL program was created to visualize three common sorting algorithms which we have chosen : Selection Sort, Bubble Sort, Insertion Sort.

This OpenGL program would represent data as a bar-graph and after selecting a data-ordering algorithm, the user can run an automated animation or step through it at their own pace. The user can also pause the animation at any given instance of time and switch between any of the three sorting algorithms as per their choice.

2. PROBLEM STATEMENT

To visualize working of an algorithm is always a challenge. People tend to grasp the concepts much effectively if they are given a visual representation of a problem.

It is always easier to pick up concepts by seeing it done, rather than reading about it. For example, when we were learning about sorting algorithms earlier, We found that seeing the data move to its correct position under the constraints of an algorithm was much easier to follow than tracing the code by hand. This led to the inspiration of this project, which describes a OpenGL program we created that animates how sorting algorithms modify and organize a set of data. We visualized three of the well-known sorting algorithm ones by representing numerical data as a histogram.

3. OBJECTIVES

- ❖ Represent the un-ordered array using a bar graph.
- ❖ Animate the sorting process for 3 sorting algorithms.
- ❖ Allow the user to switch between sorting algorithms.

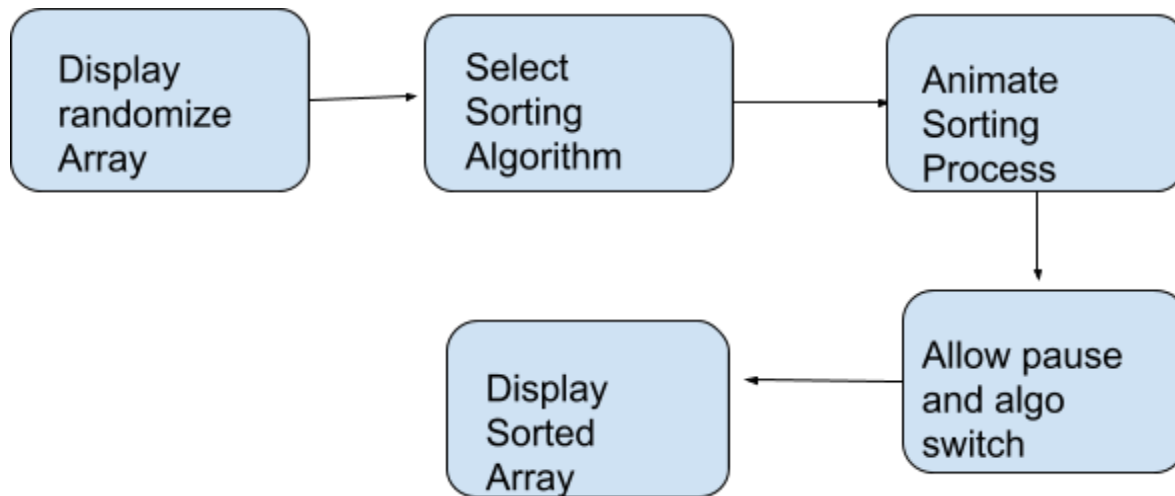
4. RELATED READING / LITERATURE

Over the years, there have been many studies and papers on the use of sorting algorithms as visual aids. These developments prove the continuing interest in creating animation tools.

For a direct analysis of how students respond to sorting animation, the paper “Do Algorithm Animations Assist Learning? An Empirical Study and Analysis” provides an in-depth view. A post-test study was used to gather information on comparing the results of students who only had textbook resources to those who had a textbook as well as an animation for assistance.

Another article we found is Stasko’s “Using Student-Built Algorithm Animations as Learning Aids.”. The results showed positive feed-back and overall better understanding of how the algorithms worked. Also, some students found that they had misconceptions about the material when implementing them in an animation.

5. SOLUTION APPROACH / METHODOLOGY



6. IMPLEMENTATION DETAILS

This project is implemented using OpenGL and using OpenGL Utility toolkit.

6.1 Creating and displaying randomize array

```
for(temp1=0;temp1<MAX;temp1++){
    a[temp1]=rand()%100+1;
    printf("%d ",a[temp1]);
}

for(ix=0;ix<MAX;ix++)
{
    glColor3f(1,0,0);
    glBegin(GL_LINE_LOOP);
        glVertex2f(10+(700/(MAX+1))*ix,50);
        glVertex2f(10+(700/(MAX+1))*(ix+1),50);
        glVertex2f(10+(700/(MAX+1))*(ix+1),50+a[ix]*4);
        glVertex2f(10+(700/(MAX+1))*ix,50+a[ix]*4);
    glEnd();

    int_str(a[ix],text);
    //printf("\n%s",text);
    glColor3f(0,0,0);
    bitmap_output(12+(700/(MAX+1))*ix, 35, text,GLUT_BITMAP_TIMES_ROMAN_10);
}
```

6.2 Selecting the sorting algorithm

```
// Keyboard Function
void keyboard (unsigned char key, int x, int y)
{
    if (sorting!=1)
    {
        switch (key)
        {
            case 27 : exit (0); // 27 is the ascii code for the ESC key
            case 's' : sorting = 1; break;
            case 'r' : init(); break;
            case 'c' : sort_count=(sort_count+1)%SORT_NO; break;
        }
    }
    if(sorting==1)
        if(key=='p')
            sorting=0;
}
```

6.3 Sorting Algorithm

```
//Bubble Sort
void bubblesort()
{
    int temp;
    while(notsorted())
    {
        while(j<MAX-1)
        {
            if(a[j]>a[j+1])
            {
                swapflag=1;
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;

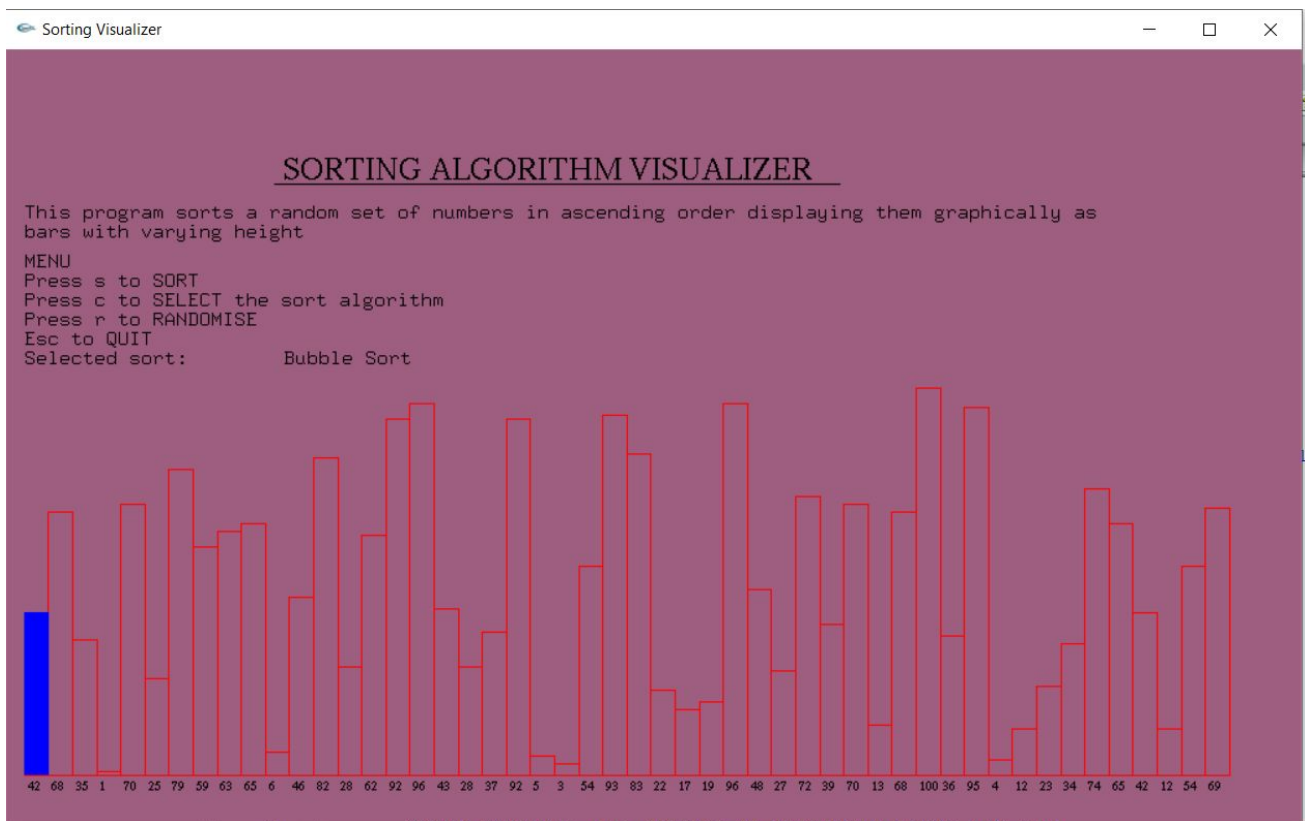
                goto A;
            }
            j++;
            if(j==MAX-1) j=0;
            printf("swap %d and %d\n",a[j],a[j+1]);
        }
        sorting=0;
    A: printf("");
    }
}
```

7. RESULTS

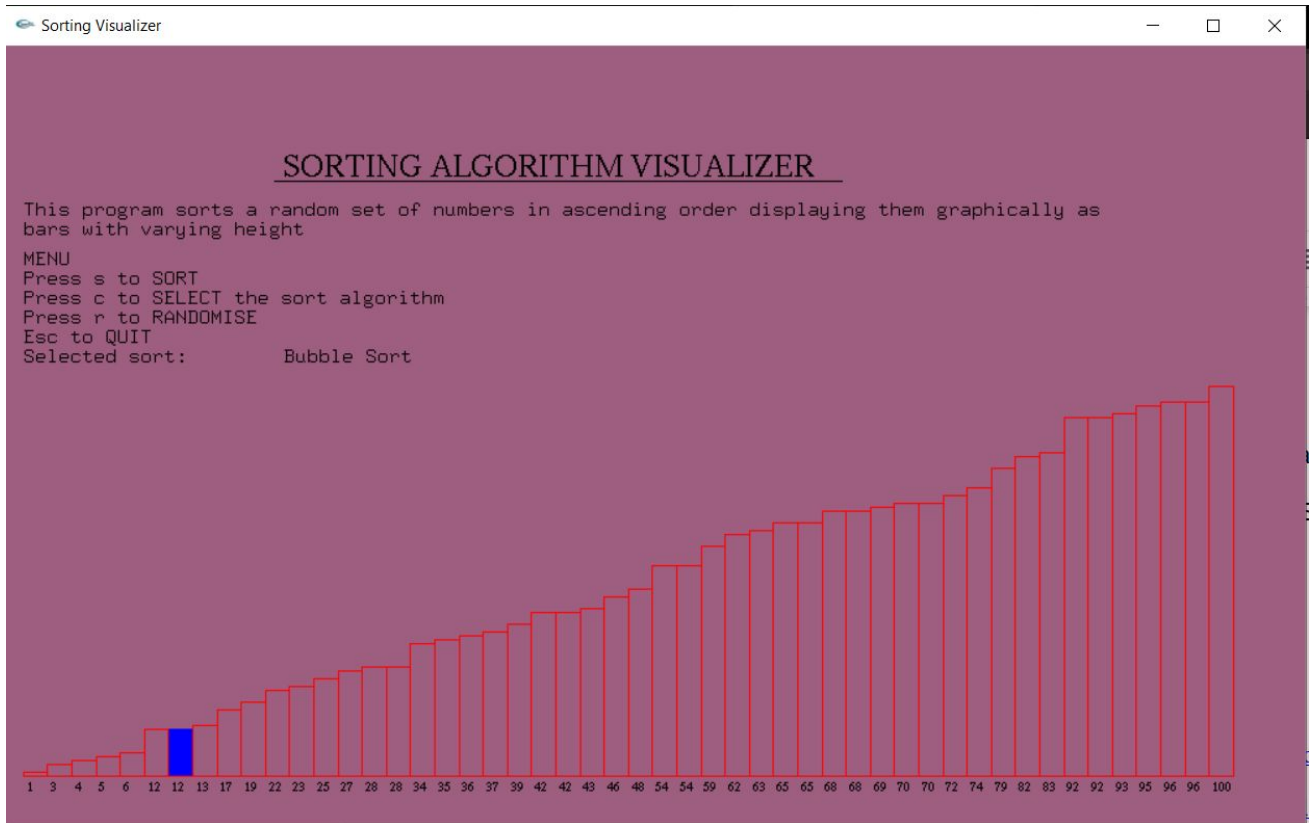
The best way to go about using the animation tool is to first select the ordering of the data and then select which algorithm to visualize. When any one of the algorithms are selected, it will sort the data as it appears on the interface.

8. OBSERVATIONS / CONCLUSION

Before Sorting:



After Sorting:



We have successfully created a working OpenGL animation tool for visualizing the following sorting algorithms: Selection Sort, Bubble Sort, and Insertion Sort. This helps us further understand how these sorting algorithms operate.

9. REFERENCES

- [1] . Visualizing Sorting Algorithms with OpenGL https://bi3mer.github.io/blog/post_6/index.html
- [2] . OpenGL - The Industry Standard for High Performance Graphics <https://www.opengl.org/>
- [3] . OpenGL - LearnOpenGL <https://learnopengl.com/Getting-started/OpenGL>