

19AIE203 “Data Structures and Algorithms”

Project Report

BUBBLE SORT

**Bachelor of Technology in
Artificial Intelligence & Engineering**

Roll No:

Name of Student:

AM.EN.U4AIE20052

NATTE SAI BHARATH

AM.EN.U4AIE20060

RAJA PAVAN KARTHIK

Subject Teacher

Dr. Sethuraman Rao



School of Computer Science & Engineering

Kollam, Kerala, India – 690 525.

3rd Semester 2021

Introduction

Ordering a list of items is one of the fundamental issues in computer science. Now there are a huge amount of sorting algorithms, In the research field, sorting problem has a good attraction. Effective sorting is key to helping the use of other algorithms. These Sorting algorithms have been studied substantially for decades. These are having many uses in applications like operating systems, real-time systems, In some of the cases, the application efficiency does depend upon the use of this sorting algorithm. Recently, the operation of graphic cards for general-purpose computing has again redefined sorting algorithms.

Overview

The main aim of this project is to take a random array or it can be user input and the process is to convert this unsorted or random array to an ascending order sort array. We selected this topic mainly because it is comparably simpler to understand than some other sorting algorithms, if we check the number of sorting algorithms on the internet it will be more than 30. We see this type of sort in daily life such as programming TV to sort the channels based on the audience viewing time. The most significant advantage of this sort over most other implementations, even quicksort, is that sort has the ability to detect that the list is sorted efficiently and is built into the algorithm itself. When the array is already sorted (best-case), the complexity of the bubble sort will be $O(n)$. Comparing this algorithm to others, this is efficient and it has the best time complexity.

Bubble Sort

Bubble sort can be explained as just like the movement of air bubbles in the water that rise up to the surface, each element of the array moving to the end in each iteration. The algorithm can be explained from the starting from the first index, it compares the first and the second elements. If the first element is greater than the second element, they will be swapped accordingly. Next, it compares the second and the third elements and swaps them if they are wrong in order. This process will be repeated up to the last element. This process will be the same for the remaining iterations too. After every iteration in the array, the largest element among all the elements will be moved or swapped to the end. In every iteration, the comparison will take place up to the final unsorted element in an array. The array will be sorted when all the unsorted elements in an array are placed at their correct positions which will be the ascending order. The array is sorted if all the elements in an array are swapped correctly in the rightful order.

Visualization

We have implemented this bubble sort visualizer using with help of code for python and Tkinter here we have imported the time module and created a function to implement bubble sort for the following parameters such as data, draw data, timer. Here data is used to pass the set of unsorted data values and drawdata is used to generate the data bars. The timer is used for the speed range. Coming to the display part while swapping is Here if the number gets swapped then the color becomes Green else it remains/stays as Red and if they sorted, the elements will be generated with Green color. We need to initialize the root class for Tkinter with some maximum window size and create a function which generates the data values by accepting a given range. In this method, we have used some variables as minval, which is the minimum value of the range, The maxval, the maximum value of the range, and the sizeval, for the number of data values/bars to be generated.

Firstly we create a blank data list which will be further filled with random data values with in the entered range and drawdata function is used to create the data bars by creating a canvas in Tkinter we normalize the data for rescaling real-valued numeric data within the given range. These data bars are generated as Red-colored vertical rectangles. Now, this is the main function to initiate the sorting process by calling the extension code by creating the main user interface frame and basic layout by creating a frame and creating a user interface area in the grid manner and an algorithm menu for showing the name of the sorting algorithm and a Start Button to start the sorting visualization process even we have creating Speed Bar using the scale in Tkinter these are the first row components in GUI. second-row components are sizeEntry to scale or select the size/number of data bars and minEntry to scale the select the minimum value of data bars and the maxEntry for to scale the select the maximum value of data bars and finally a generate button to display the elements how they sort.

Results & Conclusion

We have learned how the sorting algorithms work and their uses. In our opinion, the bubble sort algorithm works well with large datasets where the things are nearly sorted since it takes only one recapitulation to notice whether the list is sorted or not. But if the list is unsorted to a huge amount then this algorithm holds good for small datasets or lists. It can be decided that bubble sort is an effortless method of sorting the weather of an associative array, so having more time complexity. It is a stable and in-place algorithm that is most used for introducing the thought of sorting algorithms.