19AIE203

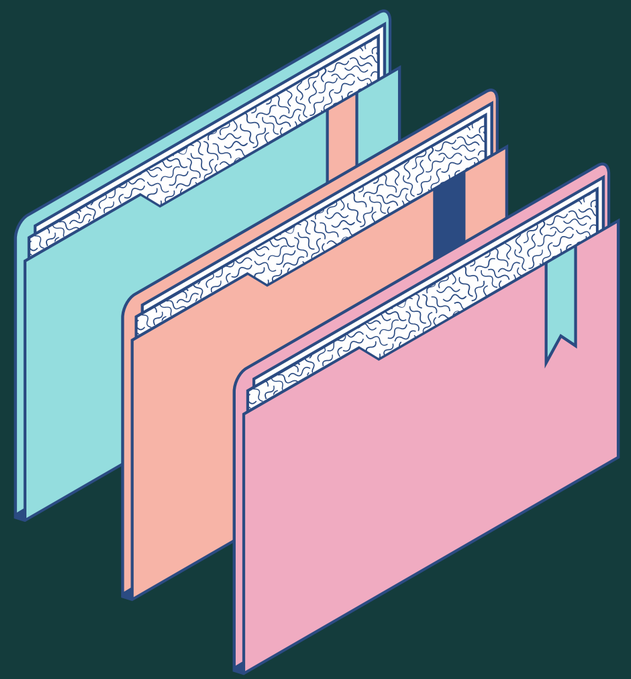
Data Structures & Algorithms 2

BUBBLE SORT

NATTE SAI BHARATH
RAJA PAVAN KARTHIK

Sorting

- Arrangement of data in a preferred order.
- Easier to search through it quickly.



Sorting Algorithms

- Rearrange a given array according to a comparison operator on the elements.
- Types of Sorting Algorithms
    - Ex: Merge Sort, Selection Sort, Bubble Sort

Bubble Sort

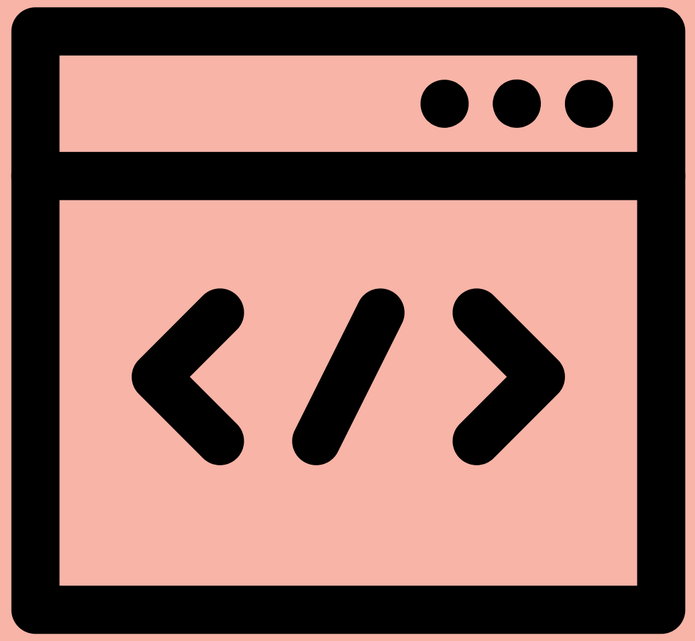
- Simplest Sorting Algorithm
- Compares adjacent elements, Comparison sort
- Swaps them if they are in the wrong order.
- The pass through the list is repeated until the list is sorted.

Implementation

Here are the methods used in this project

- Sort Algorithm (Bubble)
- Generator (The input Data)
- DrawData (Reprinting Data)
- StartAlgorithm (Main Class)

Bubble Sort Algorithm



```
procedure Bubble_Sort(A : list of sortable items)
    n := length(A)
    repeat
        swapped := false
        for i := 1 to n-1 inclusive do
            /* if this pair is out of order */
            if A[i-1] &gt; A[i] then
                /* swap them and remember something changed */
                swap(A[i-1], A[i])
                swapped := true
            end if
        end for
    until not swapped
end procedure
```

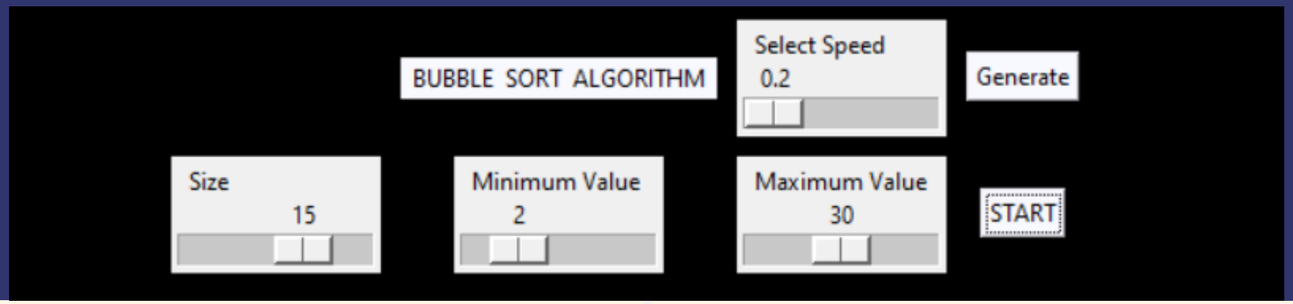
Working

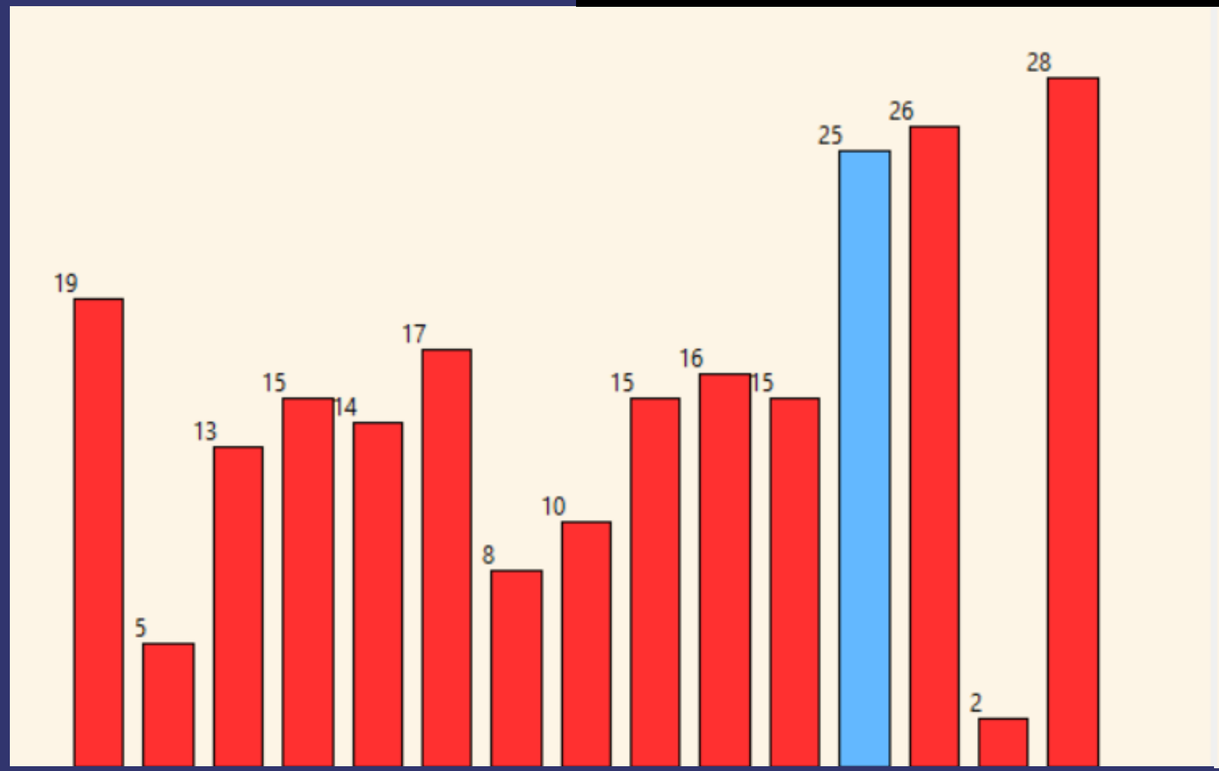
- IN AN UNSORTED ARRAY OF 5 ELEMENTS,
- START WITH FIRST TWO ELEMENTS AND SORT THEM IN ASCENDING ORDER.
- COMPARE THE SECOND AND THIRD ELEMENT TO CHECK WHICH ONE IS GREATER, AND SORT THEM IN ASCENDING ORDER.
- REPEAT STEPS UNTIL NO MORE SWAPS ARE REQUIRED.

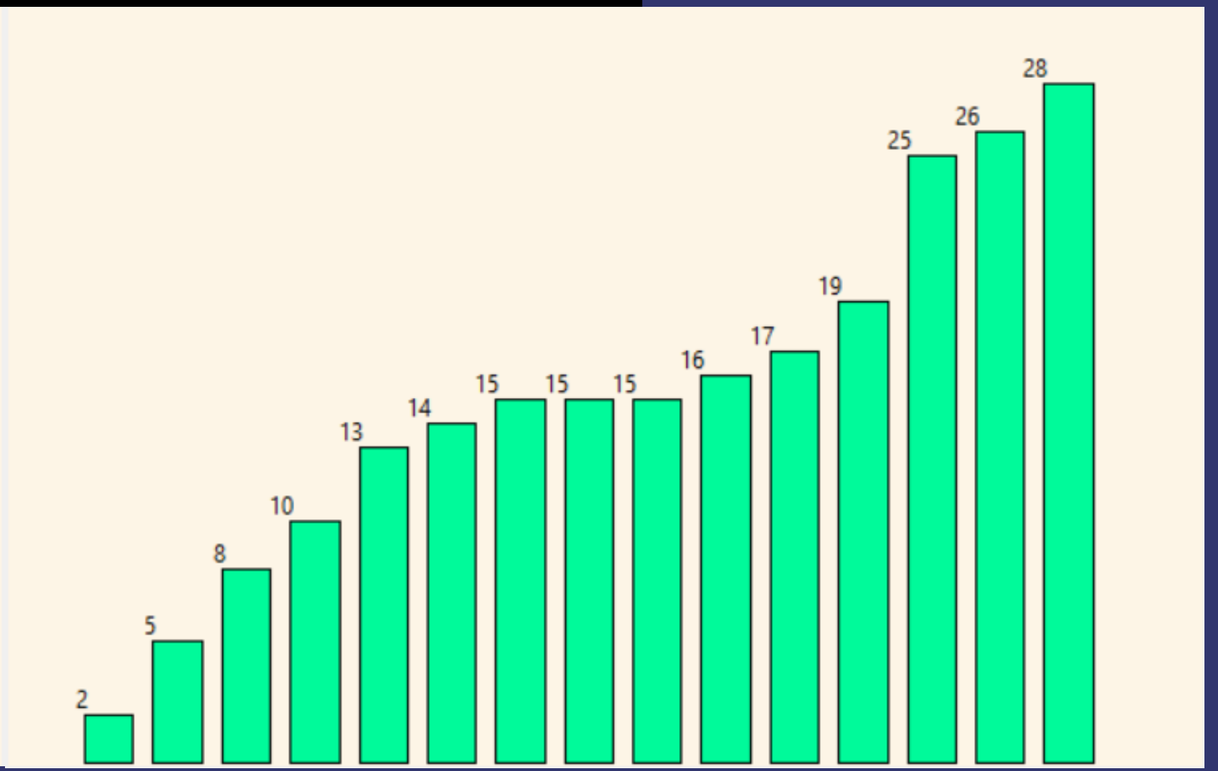
Methods

- **Generator** were the initiation of Dataset id done
- **DrawData** here the rearranging of the given data
- **Start_algorithm** its the main method were the Visualization occur with Bubble sort

Inputs & Outputs





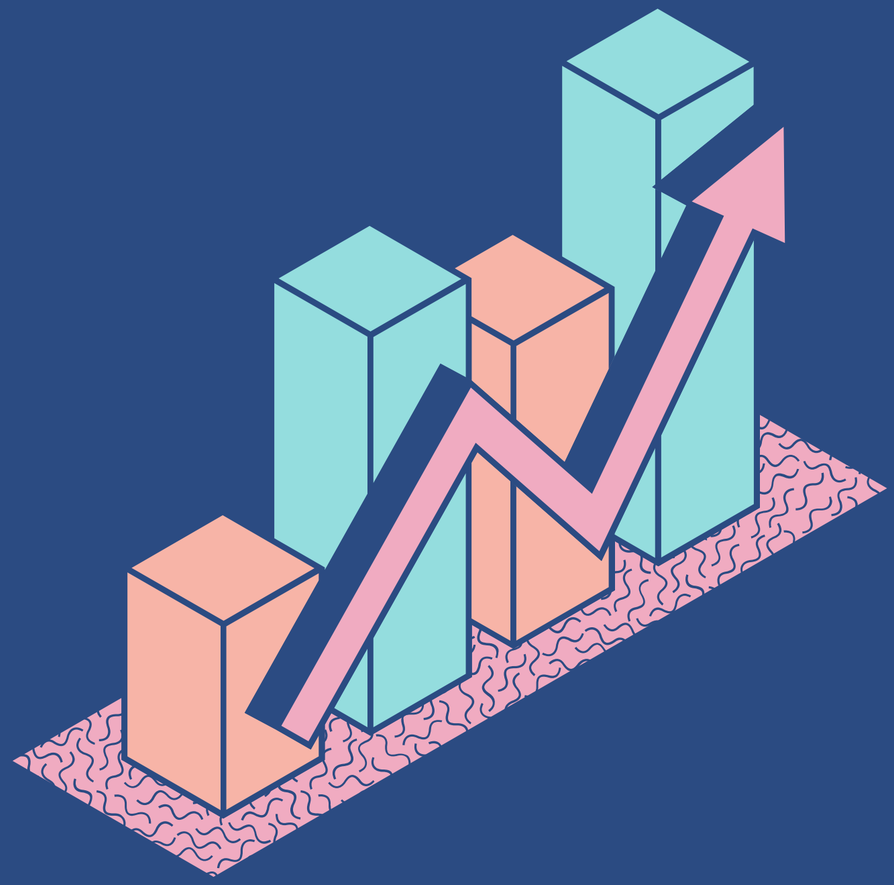


Performance

- Worst Case Complexity: O(n*n)
- Best Case Complexity: O(n)
- Average Case Complexity: O(n*n)
- n is the number of items being sorted.
- Bubble sort is not a practical sorting algorithm.
- Space complexity is O(1)

Advantages

- Simple to understand
- Stable sort algorithm
- Capability to detect small errors
- Occupies less memory
- Used in polygon filling algorithm



Disadvantages

- Highly inefficient for large data sets.
- Require n-squared processing steps for every n number of elements to be sorted.
- Suitable for academic teaching
- Not suitable for real-life applications.

FUTURE ENHANCEMENT

Optimized Bubble Sort Algorithm

- In the above algorithm, all the comparisons are made even if the array is already sorted.
- This increases the execution time.
- An extra variable swapped is introduced.
- The value of swapped is set true if there occurs swapping of elements. Otherwise, it is set false.
- After an iteration, if there is no swapping, the value of swapped will be false.
- This will reduce the execution time and helps to optimize the bubble sort.

Technology is an effective tool that can make education more meaningful and engaging for teachers and students alike.

Thank You :)