

Rüya Buse Dinmezel
MMMFCH
Active (International Program)/ ERASMUS
ruyadinmezel@sabanciuniv.edu

Task of Semester Project

Project is about visualizing sorting algorithms using MATLAB. The Matlab program creates a random array of integer and sorts the array using choose algorithm among Bubble Sort, Insertion Sort, Selection Sort. While sorting the array the program visualizes the sorting algorithm with a figure of bar chart. It creates an animation of how the algorithm works .

Review of Theoretical Background

Sorting Algorithm is used to rearrange a given array according to results of a comparison operator on the elements. The comparison operator is used to decide the new order of element in the respective data structure.(Sorting Algorithms, 2022)

These algorithms are very useful in many areas of daily life and is being used in many computer programs and applications.

The most common example we experience every day is sorting clothes or other items on an e-commerce website either by lowest-price to highest, or list by popularity, or some other order.(Sorting Algorithms, n.d.)

Bubble Sort

Bubble Sort is a simple sorting algorithm which works by iterating through adjacent elements and swapping them if they are in the wrong order, second one is smaller then the first one.(Bubble Sort, 2022)

Insertion Sort

Insertion sort algorithm works similar to the sorting of playing cards in hands. It is assumed that the first card is already sorted in the card game, and then we select an unsorted card. If the selected unsorted card is greater than the first card, it will be placed at the right side; otherwise, it will be placed at the left side. Similarly, cards from the unsorted part are picked and placed at the correct position in the sorted part.

Insertion sort is similar to this approach. Insertion sort takes the first element, then iterates it through the sorted array(the first element is assumed to be a sorted array). (Insertion Sort,n.d.)

Selection Sort

The selection sort algorithm sorts an array by finding the minimum element repeatedly from unsorted part then placing that element at the beginning of the array. (Selection Sort,2022)

Development of computer algorithm

I have developed three separate m files for the three mentioned sorting algorithms. Also created an app that has 3 buttons for Bubble Sort, Insertion Sort, Selection Sort. When pressed on the respective button it show an example of sorting algorithm with randomly created array.

Description of implemented MATLAB code

Bubble sort

Bubble sort algorithm works by iterating through adjacent elements.

First loop iterates through elements $i = 1$ to $i=n$

Second loop iterates through elements $k = 1$ to $k=n-1$

If k th element is bigger than $k+1$ th element the elements swap values.

Insertion sort

The array is virtually split into a sorted and an unsorted part. Initially the first element is counted as sorted part.

Values from the unsorted part are picked and placed at the correct position in the sorted part.

Insertion sort algorithm iterates through the randomly created array $i=2$ to $i=n$

Iterate through the array starting from the second element

Assign i th element value to variable x

Assign $k=i-1$

If k th elements value is bigger then x and k is bigger or equal to 1, we assign k th elements value to $k+1$ th elements value and decrease k value with $k-1$

Selection sort

Selection sort algorithm sorts an array by continuously finding the minimum element from unsorted part and placing it to beginning.

Algorithm iterates through $i= 1$ to n

Min is first assumed to be the first element

After comparing with the rest of the array

If there is a smaller element than current min

Minimum is updated and the element is placed to its according place

Guide to programme usage

Either by running each file ('bubblesort.m', 'insertionsort.m', 'selectionsort.m') or running the application and clicking on the buttons.

From m files you can also change the size of randomly created array

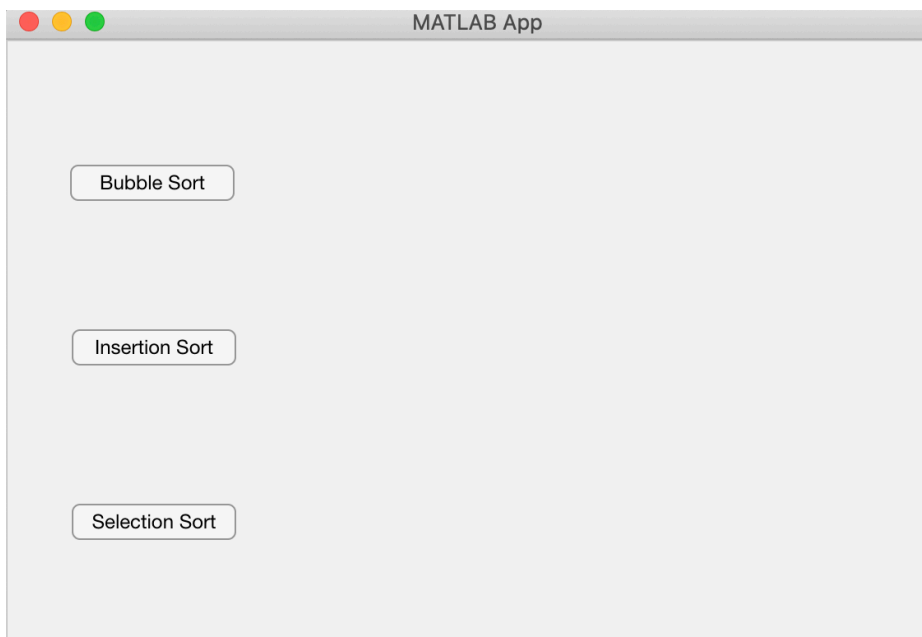
From this line

```
array= randi([1,100],1,40)
```

and the waiting time from

```
pause(0.2)
```

to observe the sorting better.



Validation

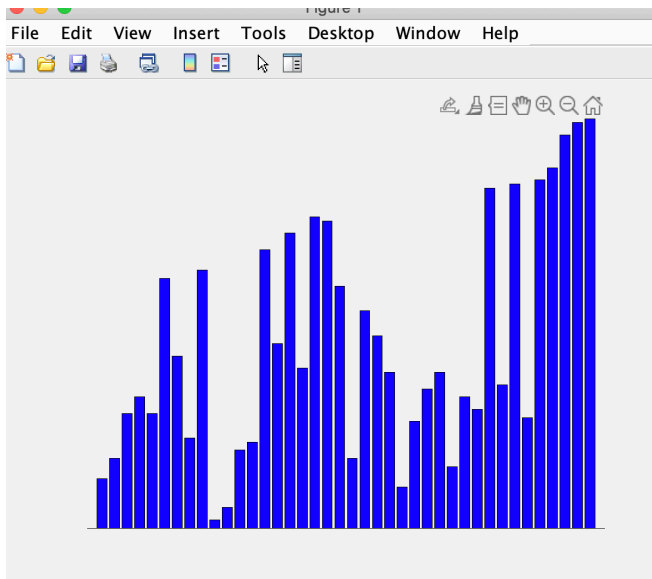
From the results we can see that the array is sorted and the program works correctly.

The aim of the project is to put the needed information in order by the program it can clearly seen that , the results are sorted and can be used according to needs.

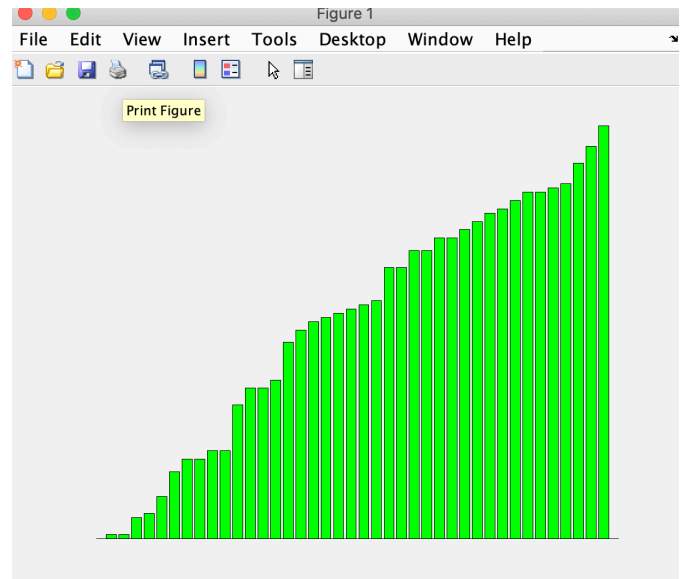
Results

Bubble Sort

Unsorted Array Bar Chart



Bubble Sorted Array Bar Chart



Insertion Sort

```
Command Window
>> insertionsort

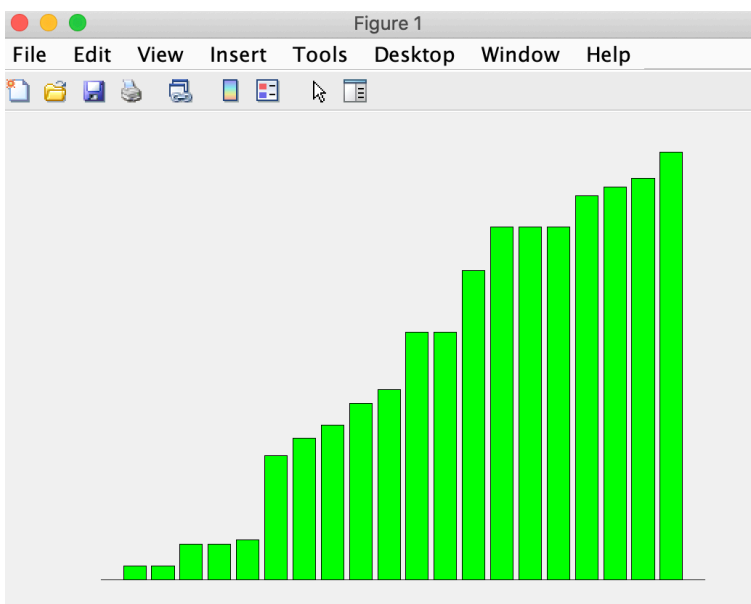
array =

    3    56    28    70    89     3    35    43     9     8     8    91    97    40    32    56    80    80    87    80

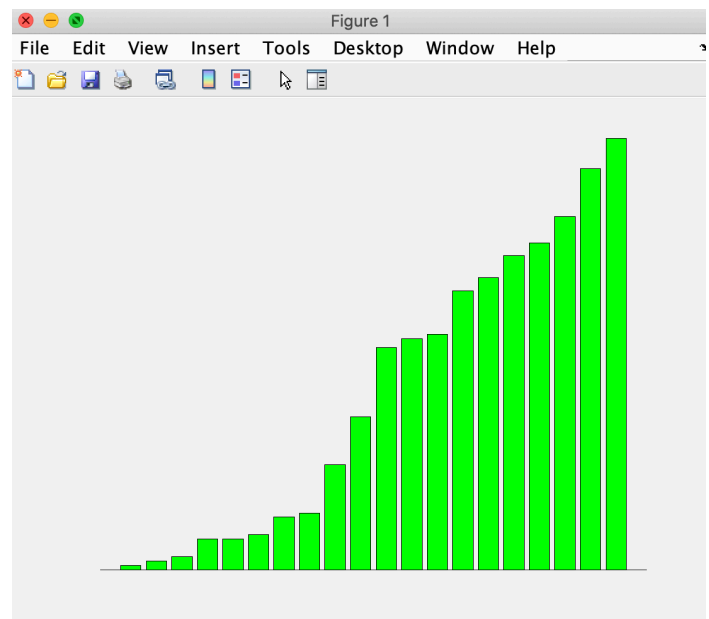
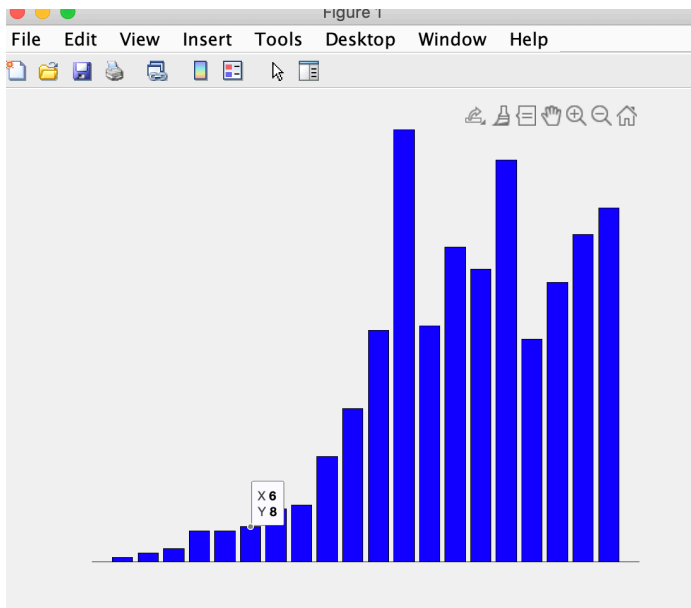
insertionsorted_array =

    3     3     8     8     9    28    32    35    40    43    56    56    70    80    80    80    87    89    91    97

fx >>
```



Selection Sort



```
>> selectionsort
array =
    1    92     2     3    72    54    53    51     7    24    12    99     8     7    67    35    13    64    75    81

selectionsorted_array =
    1     2     3     7     7     8    12    13    24    35    51    53    54    64    67    72    75    81    92    99

>>
```

Conclusions

By watching how the bars change places, we can understand how the sorting algorithms works better and see them in action.

References

Sorting Algorithms. (2022, April 22). <https://Www.Geeksforgeeks.Org/Sorting-Algorithms/>. Retrieved May 22, 2022, from <https://www.geeksforgeeks.org/sorting-algorithms/>

Sorting Algorithms. (n.d.). Interview Bit. Retrieved May 22, 2022, from <https://www.interviewbit.com/tutorial/sorting-algorithms/>

Bubble Sort. (2022, May 11). Geeks for Geeks. Retrieved May 22, 2022, from <https://www.geeksforgeeks.org/bubble-sort/>

Insertion Sort. (n.d.). Javatpoint. Retrieved May 22, 2022, from <https://www.javatpoint.com/insertion-sort>

Selection Sort. (2022, May 9). Geeks for Geeks. Retrieved May 22, 2022, from <https://www.geeksforgeeks.org/selection-sort/>