Alejandra Reyes

3/25/2021

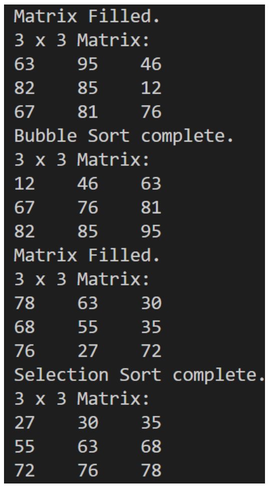
Homework 1 Question 3

Create a sort for 2d matrices. Do a bubble sort, insertion sort, and selection sort variation of this. Do NOT convert the matrix into a 1D array to sort. You must sort it as a 2D structure.

Program Description:

This program attempts to perform bubble sort, insertion sort and selection sort on a 2D structure. Our 2D structure is a Matrix object that has a 2D array to store the matrix, as well as protected variables to keep count of the number of rows and columns. Bubble sort works by first sorting each row, and comparing the first and last elements of two rows and swapping them. Each row is then sorted once again, until there are no more elements to swap. Selection sort finds the minimum element in the unsorted portion of the matrix, and places it into the sorted portion of the matrix. The function iterates until each element has been sorted. The matrices were filled with random numbers prior to sorting by using two for loops and inserting randomly generated numbers. Selection sort and bubble sort function as intended, but the insertion sort does not work properly.

Here is a sample output from the program:


```
Matrix Filled.
3 x 3 Matrix:
63    95    46
82    85    12
67    81    76
Bubble Sort complete.
3 x 3 Matrix:
12    46    63
67    76    81
82    85    95
Matrix Filled.
3 x 3 Matrix:
78    63    30
68    55    35
76    27    72
Selection Sort complete.
3 x 3 Matrix:
27    30    35
55    63    68
72    76    78
```

Problems Encountered:

After several attempts, I was unsuccessful at implementing insertion sort on the matrix. It was difficult to determine which element was next in the case where our next element was in a different row. For bubble sort, after sorting each row, a comparison between the last element of one row and the first element in the next row had to be made. However, after this comparison, each row had to be sorted once again. This workaround allows us to perform comparisons until each element in the matrix is sorted.

Conclusion:

Sorting a 2D structure requires sorting individual rows and comparing the last element of one row with the first element of the next when using bubble sort. This requires more comparisons and swaps than a 1D structure, however it allows us to sort matrices effectively. Insertion sort is more straightforward, as we are always swapping a minimum element with another element in the list to insert it into the proper position. Insertion sort on the matrix requires keeping track of each index individually so that we are able to traverse each element for each row in the matrix.