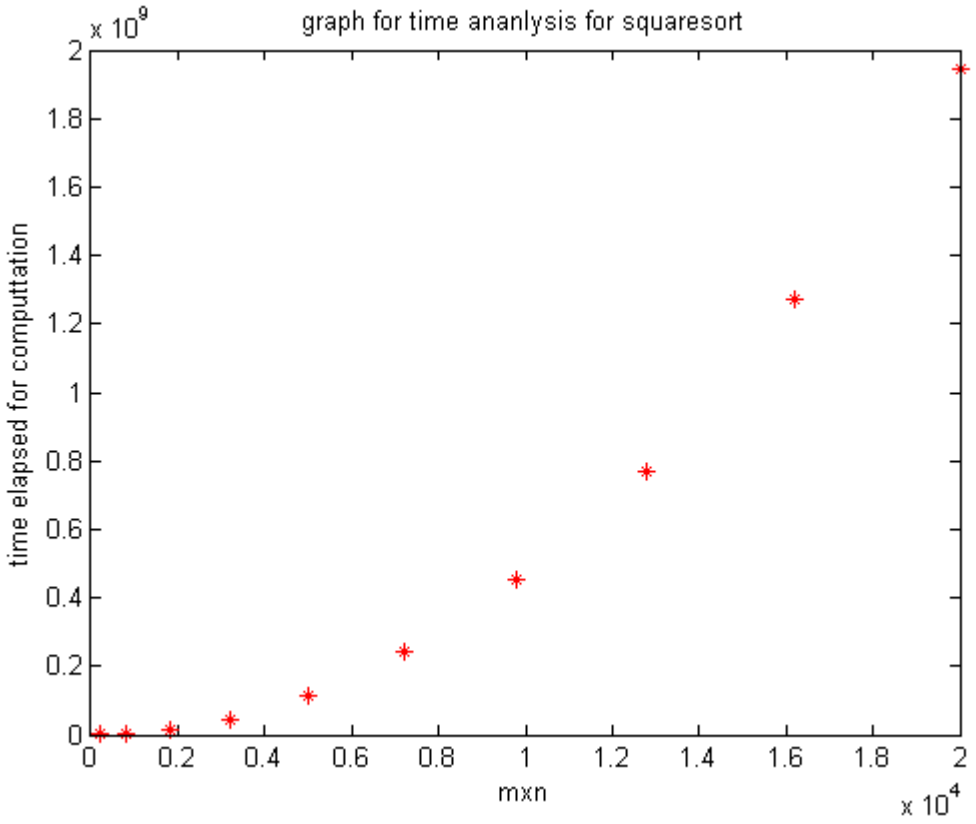
README: Assignment 2
Author: Anushree Singh
Course: CIT 594 (Spring 2015)

A) I estimated it would take 2 days to complete the assignment, and it did.

B)In my sorting algorthim, given a 2D array , the indices of this array are mapped to a 1D array(sub-indices are mapped to indices). This is done to make sure the linear sort technique is still in-place. After this mapping is done I implement the selection sort technique to the elements of the 2D array row-wise. Thus the smallest element is swapped in place of the first index and so on.
Since the array is of size mxn , the total number of elements that are mapped is m*n For mapping the indices back to indices I used the following formula:
for $a(i, j)$, $i = index/numberofcolumns$, and $j = index\%numberofcolumns$.

I used two loop invariants (at the end of two loops) for asserting this algorithm.
These are :
1) for all i bounded between the outer loop and inner loop, the array element at the "min" index is the one with minimum value.
2) for all j bounded between i and the outer loop, all elements are sorted and in correct place.
Both the for loops have exit conditions , to assert that the for loops have ended. And there is an invariant + exit condition at the end of the algorithm to see if all the elements in the array are sorted and in the right place.

C) The total number of elements in the 2D array is mxn and My squaresort algorithm uses selection sort at the core. Since selection sort operates over all the elements of the 2D array it takes $O(m^2 * n^2)$. I calculated timings for 10 set of 2D reactangular arrays. the timings for which were as follows:
Time for sort 1 3163502. array = 10x20
Time for sort 2 2905663. array = 20x40
Time for sort 3 14592850. array = 30x60
Time for sort 4 46412851. array = 40x80
Time for sort 5 116222595. array = 50x100
Time for sort 6 245253266. array = 60x120
Time for sort 7 452156080. array = 70x140
Time for sort 8 766602654. array = 80x160
Time for sort 9 1269067073. array = 90x180
Time for sort 10 1947067925. array = 100x200
As we can see from the graph the timing corresponds to $O(m^2 * n^2)$.


graph for time ananlysis for squaresort
x 10^9
time elapsed for computtation
mxn
x 10^4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2