Instructions:

1. Analyze the performance of your method sortOfSort. You are expected to clearly explain / justify your answer to this question. Your answer should address the best / worst / and average cases for the performance of your method. You will describe in a word document called **YourLastName-YourFirstName-Week7.docx** the performance of your sortOfSort method in terms of n, the length of the input array.

The algorithm of my method sortOfSort starts with 4 step instructions, to initialize variables and assign values.

Next, a while loop that will execute n number of times n being the length of the array used. Inside of the while loop there is an if statement with its else property that will execute n times as well.

Inside of the if and else statement there is 1 instruction to set a value, followed by a for loop and 5 instructions to swap the elements of the array. The for loop that will execute 3 times n number of times plus 4 “one time” instructions to compare the elements of the array and find and assign the maximum value to the current index. 2 of this “one time steps” are located in the for loop declaration and the other 2 inside the for loop. One is an if statement and the other is a variable assignation once the maximum value is found.

Finally there is an if statement that will execute n/4 times since the counter will only reset when the for loop form the if statement and the for loop of the else statement finish executing.

Visual counter of steps:

|  |  |
| --- | --- |
| **Instructions** | **times** |
| Initialization of variables | 4 |
| First while loop executions | n |
| If and else statement execution | n |
| Instruction to set value | 1 |
| For loop execution | 3n |
| One time instructions inside the for loop | 4 |
| Instructions to swap current index with the index with the greatest value | 5 |
| If statement to reset the counter | n/4 |
|  |  |

The formula of the performance of my sortOfSort algorithm:

4 + (n(n(1+4 + 3n + 5) + n/4)

The Big 0 performance: O(n^3)

Justification:

4 + n^2( 1 + 4 + 3n+ 5) + n(n/2)

4 + 10n^2 + 3n^3 + (n^2)/2

For this method the worst case scenario, the best case scenario and the average case scenario:

Worst case scenario: O(n^3) or 4 + 10n^2 + 3n^3 + (n^2)/2 ( every element needs to be arranged)

Best case scenario: : O(n^3) or 4 + 3n^3 + (n^2)/2 ( all elements are sorted)

Average case scenario: : O(n^3)

Since the method have to compare all the elements of the array no matter If the array is already sorted accordingly. The counter is initialized as 0 and will execute the for loops. The difference between already sorted algorithm for the best case scenario are the one time instructions inside the for loops but this steps are not part of the O(n^3).