Analyzation of sortOfSort method

In this report, I will clearly explain as to what the best / worst / and average performance case time complexity will be for the method.

**Best case:**

The best case for the method will be O(1).

The reason the best case for the method will be O(1) is that it will find the largest number on the first attempt

**Worst case:**

The worst case for the method will be O(n).

The reason for this case, is that if it does not find the largest number between the two , it would of have gone through them both, so the for loop would of have gone through one complete pass.

**Average case:**

The average case for the method will be O(n).

The reason for it being average on O(n) is that the algorithm only has two numbers to look at in reality, so whether the number finds it at first or not, it would still take on average of O(n).