Mystery #1: mysterySort1 method
- While analyzing this method I noticed it is similar to Insertion Sort.

  - For the for loop, it can be said to be n because it runs through the list n times
  - the next line of code is constant thus 1.
  - In the while loop, we can conclude that we need at most n-2 to compare and n-2 to swap when dealing with the second to the last element and n-1 to compare and swap when dealing with the last element
  - Therefore, the algorithm is $n(n-1)$ which is quartartic and can also be simpleifed to n^2
  - Thus, the time complexity is $O(n^2)$

Mystery #2: mysterySort1 method