

1. O-notation emerged as a popular technique for algorithm analysis because as the size and complexity of problems grew, such as the growth in the size in the amount of data that needs to be enacted on, programmers realized that the specific constants such as the amount of operations done in a single loop or the amount of operations done outside of a loop matters less, as performing more than a million of those operations vs. performing just 10 matters a lot more when looking at how good an algorithm is.
2. If you analyze two algorithms and they have the same closest fit O-notation, you can compare the amount of data that needs to be used for each algorithm. For example, if one algorithm needs an entirely new list to be created as a temporary storage space, you might favor an algorithm that can just enact on the given list, as it saves  $n$  storage space.