Caio Brighenti
COSC 302 - Analysis of Algorithms – Spring 2019
Assignment 1

1. Simply put, an algorithm is a way to solve a problem. In other words, it is a specific, well-defined step-by-step process to solve not just a single problem, but a category of similar problems. Algorithms must take some form of input, and must correctly use this input to solve its defined category of problems.

2. An example of a problem that cannot be solved is an algorithm that for any given input problem could output an efficient algorithmic solution to it. A similar scenario is NP-complete problems – neither an algorithm nor a human has determined whether there are efficient solutions to this class of problems. There are plenty of problems for which we do not know efficient solutions, such as the Halting problem. An example of a non-algorithmic solution an algorithmically solvable problem would be the tallest student problem. We could simply look at the class and see quickly which student is the tallest, without systematically going one by one and comparing to the currently tallest – we have an innate sense of things like this, and can process multiple pieces of information simultaneously, something a computer cannot do.

3. The book defines analyzing an algorithm as predicting the resources that the algorithm requires, such as memory space, communication bandwith, or disk storage. Most commonly, it refers to analyzing the rate of growth of the algorithm's runtime, also known as time complexity.

4. Each of the following are things that can be considered part of analyzing an algorithm

    - 1 - Showing correctness
    - 2 - Measuring time complexity
    - 3 - Finding the worst case
    - 4 - Finding the best case
    - 5 - Finding the average case
    - 6 - Measuring space complexity
    - 7 - Is it efficient? Can it be better?
    - 8 - Does it terminate?
    - 9 - What is the actual runtime, in real numbers?
    - 10 - What counts as a computer step?
    - 11 - Is the time complexity different for the best case?
    - 12 - Is the time complexity different for the worst case?
    - 13 - Does the algorithm depend on any other variables?
    - 14 - Is the order of growth determined a tight or not tight bound?