Lab 7: Algorithmic Complexity

Instructions:

This worksheet serves as a guide and set of instructions to complete the lab.

- You must use the starter file, found here, to get credit for the lab.
- Additionally, here is the workbook that you can read through for further context and additional (non-required) material.
- All material was sourced from the CS10 version of The Beauty and Joy of Computing course.

Submitting:

You will need to fill in the blocks under “Lab6 Starter (Algo)" and submit this to Gradescope.

- To receive full credit, you will need to complete the required blocks, and the required blocks must pass all tests from the autograder in Gradescope.
- For instructions on how to submit to labs to Gradescope, please see this document.

Please note, you must use the starter file, and you must NOT edit the name of any of the required blocks. Failing to do either for these will result in the autograder failing.

Objectives:

In this lab, we will explore **algorithmic complexity**, which refers to how the performance of an algorithm scales with the size of its input. Understanding time and space complexity is crucial for designing efficient algorithms. In this lab, you will apply your knowledge of Big-O notation to analyze and compare different algorithms:

- Use the Snap! timer to gather run time information
- Observe the difference between constant, linear, quadratic and logarithmic run times.
- Compare the efficiency of different algorithms based on their complexity.
- Understand the impact of algorithmic complexity on practical application performance.

Required Blocks:

- Block 1: add all numbers in (list) (Non Gauss)
- Block 2: add all numbers in (list) (Gauss)

Important Topics mentioned in the Workbook:

For better understanding of the lab please go through these workbook pages! Topics that are important but not required for this lab will be indicated with an asterisk**. These topics are best reviewed in order and as you complete the lab.

- A (Non-Video) Game
- Competing with Young Guass
- Time Is of the Essence
- Do You Have Time to Add?
- Constant-Time
- All the Numbers, All the Time
- Linear-Time
- Self Test: Searching Through Time**
- Timing Sum-thing's Up
- Constant versus Linear
- A Distinct Difference**
- Quadratic-Time
- Logarithmic-Time

Block 1: add all numbers in (list) (Non Gauss)

- Objective:
  - Creator a reporter that sums up the numbers in the numbers list the normal, "non-Gauss" way: walking through the list and adding the numbers one by one.
- Notes
  - **You must use either iteration or HOFs. The solution must run in O(N) time.**
- Inputs:
  - list = any list, of any length
    - We can assume that input is a list of numbers from 1 to N
- Output:
  - a number
  - Reports the sum of all the numbers from 1 to N
- Examples:
  - 

- 

Block 2: add all numbers in (list) (Gauss)

- Objective:
  - use the Gauss formula to find the sum of adding all numbers in the list
- Notes
  - **HOFs and Iteration are not allowed. You must use the formula provided. The time complexity must be in O(1).**
- Inputs:
  - list = any list, of any length
    - We can assume that input is a list of numbers from 1 to N
- Output:
  - a number
  - Reports the sum of all the numbers from 1 to N
- Examples:
  - 

  - 

Remember to submit on Gradescope!