

AERE 361: Lab 13

Sean Hellar

Due 12 MAY 2022

1 Report Questions

Complexity Analysis

- Overall complexity of code will be based on the bash script, C program and the Gauss-jordan solver. The bash script is linear, thus has notation $O(n)$. The C program is of notation $O(n^2)$, with n relative to the number of resistors in the circuit. Finally, the Gauss-Jordan Solver has complexity $O(n^2)$. The final complexity will be $O(2n^2) + O(n)$.

Loop Analysis

- Since the bash script is linear, the majority of the complexity comes from the C program and Gauss-jordan solver. The major loops contributing to the complexity are the variable and line counter function within the C program. The invariants of both loops will be the magnitudes of the variables and lines. I'm pretty sure the loop to print the answer within the Gauss-jordan solver will not add to the complexity because we will only pass it once at the end of the program.

Final Comments

- As you'll find out by running my bash/c the program is not complete. I ran into the problem of figuring out how to implement an array list from bash into c so that I could then use my `gauss_jordan` solver from lab 11. My analysis is based on how my current code behaves and what I would have done, if I had the necessary intelligence. Sorry for the letdown, but I hope you (the grader) have a fulfilling summer, as well as the other TA's ☺.

2 Sources

Majority of bash scripting

- temporallogic.org/courses/AERE361/

Splitting the string to just read the coefficients

- <https://www.educative.io/edpresso/splitting-a-string-using-strtok-in-c>
- <https://www.w3resource.com/c-programming-exercises/string/c-string-exercise-3.php>

Creating array list

- https://en.wikipedia.org/wiki/Linked_list#Singly-.2C_doubly-.2C_and_multiply-linked_lists