Sienna Ballot

Hayden Yu

EECS 118

5 December 2019

Graph Solver Project Report

We were assigned problem number 11, which is find s where is\_path(s, A, B) and color(s, Color, t) and t>C. A, B, C, and Color are user inputs, and t is number of edges that has the color k. For our solver, we’re finding all sets of edges that has a path from A to B, and the paths must satisfy the condition t>C.

In main.py, it asks for a csv file in the command argument, and then, asks user inputs. The input nodes must exist in the graph to output a path. Once the program obtains all the inputs, it creates a graph from network module that contains all nodes between every edge, their edge weight, and their edge color. Then, it is passed to find\_paths(), which will output all possible paths into a output.txt file.

In find\_paths(), it is a recursive function that starts with node A, and, using networkx’s neighbor(), it will find every node that is connected to A. When it finds B, it will check is\_path() and color() and determine whether the path will be outputted to output.txt. Any dead end or paths that don’t satisfy the conditions will not be outputted. Figure 1 shows when the user’s inputs are A=1, B=6, C=1, k=green, and output.txt. Figure 2 shows the graph.csv file which the program runs from.

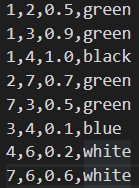
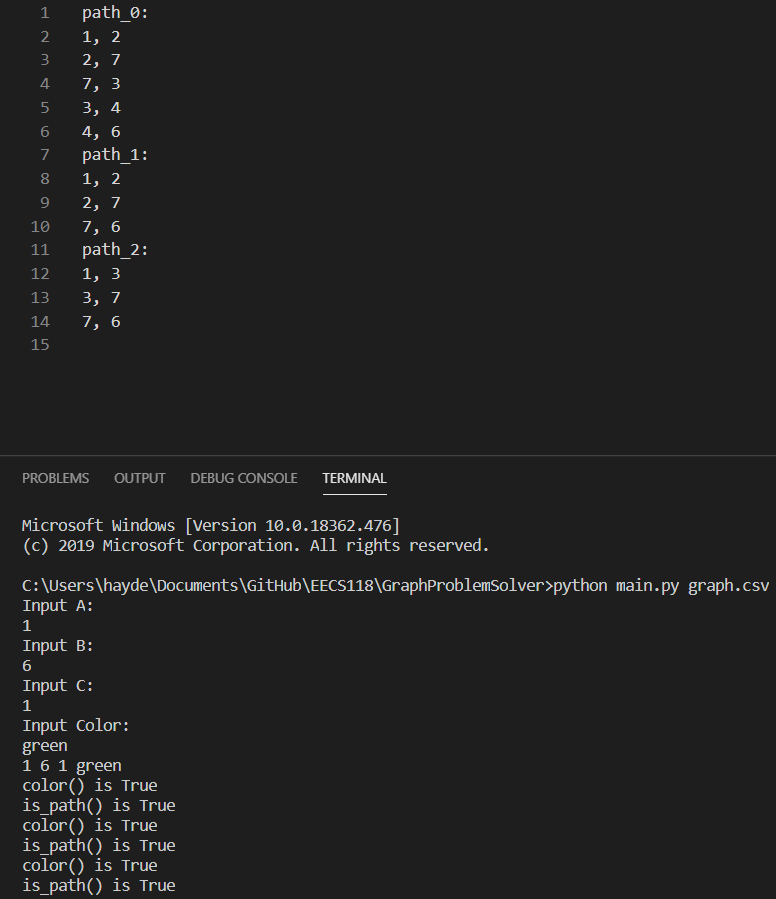
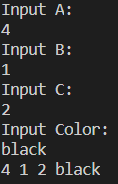


Figure 1 Figure 2

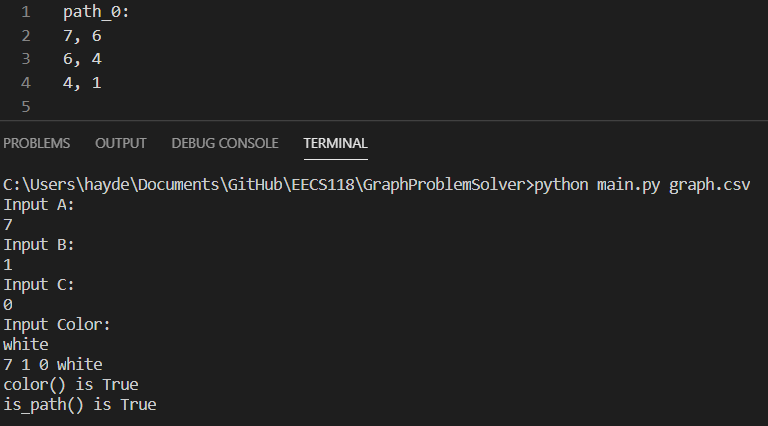
Above is one of the test cases. Rest of the test cases are shown below:

Test case 2 using the same graph from Figure2:



From the inputs, the program didn’t output anything because there isn’t a path that satisfy t>C, where t=1 and C=2.

Test case 3 using the same graph from Figure 2:



From the inputs, there exist only one path that satisfies the condition from node 7 to 1.

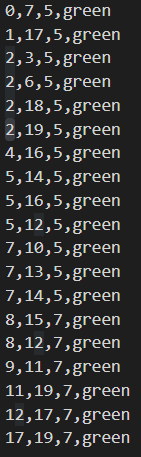
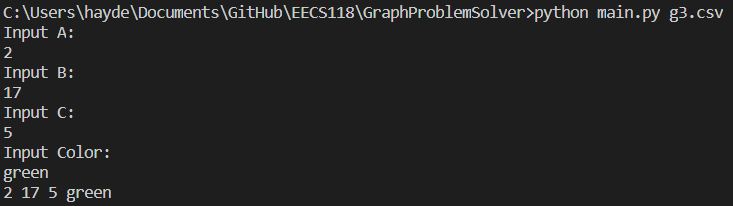


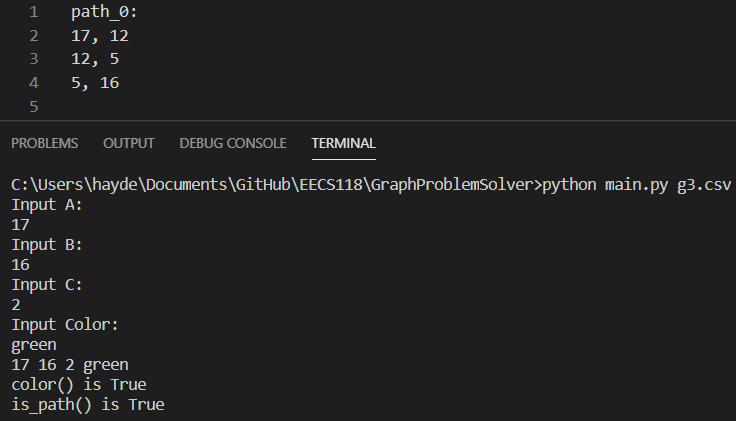
Figure 3

Test case 4 using graph from Figure 3:



The program didn’t output anything because there no such path.

Test case 5 using graph from Figure 3:



The program found a path that satisfies the conditions.