

EECS 118
Knowledge Engineering and Software Engineering
Fall 2018

Term Project – Part 2
A Graph Problem Solver
Assigned on 10/23

In this project, you will use Python to deliver a solver that solves a class of graph based combinatorial problems with TAs. Each one of you will be assigned one graph problem that is written in graph predicates. You have to understand the meaning of that graph problem with the definition of some graph predicates, and try to solve it. You also need to test someone else's solver with a set of test cases. This is a team project of two people; only one team member should submit the work. The due date is Thursday, December 5, 11:59 PM.

Deliverables

1. The source code of your problem solver:
 - a. A function file.
 - b. A main python file.
2. A report detailing your project that includes the problem solved by your solver, math behind your problem solver, peer test plans, peer test results, and discussion.

NetworkX

Take Python [NetworkX](https://networkx.github.io/documentation/stable/reference/algorithms/index.html) documentation as a reference (<https://networkx.github.io/documentation/stable/reference/algorithms/index.html>) for the given problem. Fully understand your portion in the document, and then include a description detailing the problem.

Graph Predicates

Your assign problem is a valid combination of standard comparisons operators $<$, $>$, $=$, $!=$, $>=$, $<=$ and the following predicates:

Let s to be a set of edges:

- $\text{is_path}(s, a, b)$: returns true if s is a path between node a and node b .
- $\text{total_weight}(s, n)$: returns true if n is the total weight of the edges in s .
- $\text{is_tree}(s)$: returns true if s is a tree.
- $\text{is_spanning_tree}(s)$: returns true if s is a spanning tree.
- $\text{no_nodes}(s, n)$: returns true if the number of nodes of s is n .
- $\text{height}(s, n)$: returns true if s is a tree and its height is n .
- $\text{max_degree}(s, n, G)$: returns true if the max degree of all the nodes in s on G (graph or set) is n .
- $\text{color}(s, k, n)$: returns true if number of nodes of color k in s is n .

Requirements

For the main python file:

1. Some problem might not have efficient algorithms and you may use exhaustive search.
2. Call the function file to get the result.
3. Your program should enable users to input a selected csv file that contains the edges of a graph. Use the following command prompt format:
`python <your_solver>.py <the_graph>.csv` (you can get the string of the command argument through `sys.argv[1]` in python)

For the function file:

1. To make your program structural, write different functions of your solver into a function file and provide an API based on TA's instructions.
- Input for the solver
A graph query assigned to you, with arbitrary instantiations of the constant parameters (expressed in upper case letters in your assignment; variables are expressed in lower case letters).
As for the input graph, we only deal with undirected graph, and require the user to specify all the edges with their two end points, weight, and color.
Format of input graph: (Node1, Node2, Weight, Color)
1, 2, 0.5, green
3, 4, 0.7, green
1, 4, 0.3, red
- Output for the solver
The .csv file result of your problem (e.g., a path, a tree), return NULL if the problem with its inputs cannot be solved. If you have to return multiple paths or trees, for each path please output a title (e.g. path_1, path_2, tree_1) in your .csv file first and output the edges below that title. Follow this format: (output endpoints only)
path_1
1, 2
3, 4
path_2
2, 3
3, 4

Example

To find paths with length smaller than C between node A and node B, we can define our graph problem as:

find s where `is_path(s,A,B)` and `total_weight(s,t)` and $t < C$

The user has to input A, B, C, and the graph. The output of the solver should be all the paths that satisfies the graph predicates requirement.

Test Plans

You need to develop a test plan for your peer problem solver and include the test results in your report. In addition, you need to develop a test plan for another problem solver that is assigned to you and report the test results in the report.