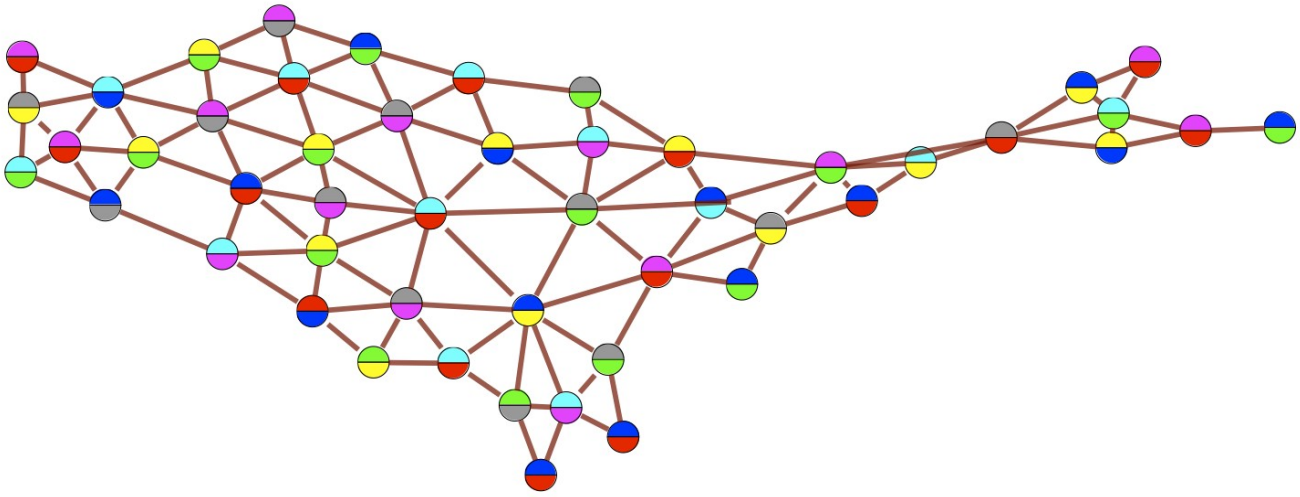
Algorithms Design and Analysis

Assignment 2

by
Ximei Liu & Yue Li



Preface

In Assignment 2, we mainly practised different algorithms on Digraph and Wgraph classes. This assignment helped us to analyse and formalize real world problems into possible graph problems in our future study and careers.

Features

Product Design

As Assignment 2 has two classes: Digraph and Wgraph. Both classes are relatively independent to each other. In order to reduce the complexity of our work, we applied Singleton Design Pattern on this assignment.

The benefit of Singleton are:

- Allow team members to work on different classes without creating conflicts.
- Loose coupling condition enables the later project integration with limited change of code.
- Centralize information and improve the OO concept of data encapsulation.

The Singleton class in our project is AssgnmentTwoFrame. The implementation simply adds both Digraph and Wgraph classes into a tabbed pane. The user can switch between two panes without restarting the program.

We also added a JWindow as the splash image of our product.

Animation

In Digraph, most commands are presented in drawing to indicate the results.

In Wgraph, the Dijkstra and Minimal Spanning Tree implementation is presented in drawing different colours to differentiate the significant path from ordinary paths.

In AssignmentTwoFrame menu, the user can alter the frequency of drawing in miliseconds.

Project Structure

This project structure is derived from both Digraph and Wgraph. As for some utility classes, such as PriorityQueue, should be implemented as a stand-alone structure. Reduced length of code increases the readability of our code.

Compilation and Run

You can either run our code from terminal by doing following steps:

- cd the assignment2 path
- javac assignment2/*.java
- java assignment2.AssignmentTwoEntry

Or port the source code into an IDE project.

Commands Explanation

We changed some default settings of this program, changes are specified below:

When user runs “load” command, the user doen't need to enter the full name of the file. For an instance, under “assignment_2/testcases”, when load q2.graph, the user only needs to enter

“load q2”

however, when the user tries to import different test scenarios, he/she must apply the “.graph” suffix as the file name, and the file must be under “assignment_2/testcases” path.

We also added an MenuItem into our menu showing all the commands in brief.

Usage:

Question 1
1. dfs

command: "dfs nodeNumber", such as "dfs 0"

effect: DFS parenthesis form and highlight the DFS exploration in the graph

2. bfs

command: "bfs nodeNumber",such as "bfs 0"

effect: BFS parenthesis form and highlight the BFS exploration in the graph

3.isCycle

command: "cycle nodeNumber",such as "cycle 0"

effect:

if there is a cycle, highlight the cycle in the graph

otherwise, print "The graph is a DAG"

4.linearize

command: "linearize nodeNumber",such as "linearize 0"

effect:

if there is a cycle,highlight the circle, and print "The Graph has a cycle. It cannot be linearized!"

otherwise, print the linearization list.

5.scc

command: "scc nodeNumber",such as "scc 0"

effect:

print all the SCCs, and also highlight in the graph, the the vertexes and edges belonging to

the same SCC use the same colour to highlight.

6.distance

command: "scc node1 node2",such as "distance 0 4"

effect:

if there is a path from node1 to node2, print the distance from node1 to node2, also highlight the shortest path from node1 to node2

otherwise, print "The distance from node 0 to node 4 is infinity."

7.isEulerian

command: "eulerian", such as "eulerian"

effect:

If the graph is eulerian, print the eulerian trail,and also highlight all the edges in the eulerian trail

otherwise,print "Eurian Trail does not exist!"

8.isBipartite

command: "bipartite", such as "bipartite"

effect:

the element in the two sets will be distinguished with red and blue colour. If there's an

edge connecting the same colour,print "The digraph is not bipartite!,otherwise, print

"Yes, the digraph is bipartite!".

Question 2

1.dijkstra

command: dijkstra node1 node2

effect: plot the shortest path from node1 to node2

2.mst

comamnd: mst

effect: plot the minimal spanning tree by using kraskal's algorithm

3.bellmanford

command: print bellmanford 0 2

effect: generate the shortest paths between 2 nodes in a table

4.floydwarshall

command: print floydwarshall

generate all nodes shortest paths.

Test Cases

These are the candidates for loading graphs:

sample

test

q1

q1_eulerian_trial

scc

scc2

q2

q2-1

q2-2

bellman