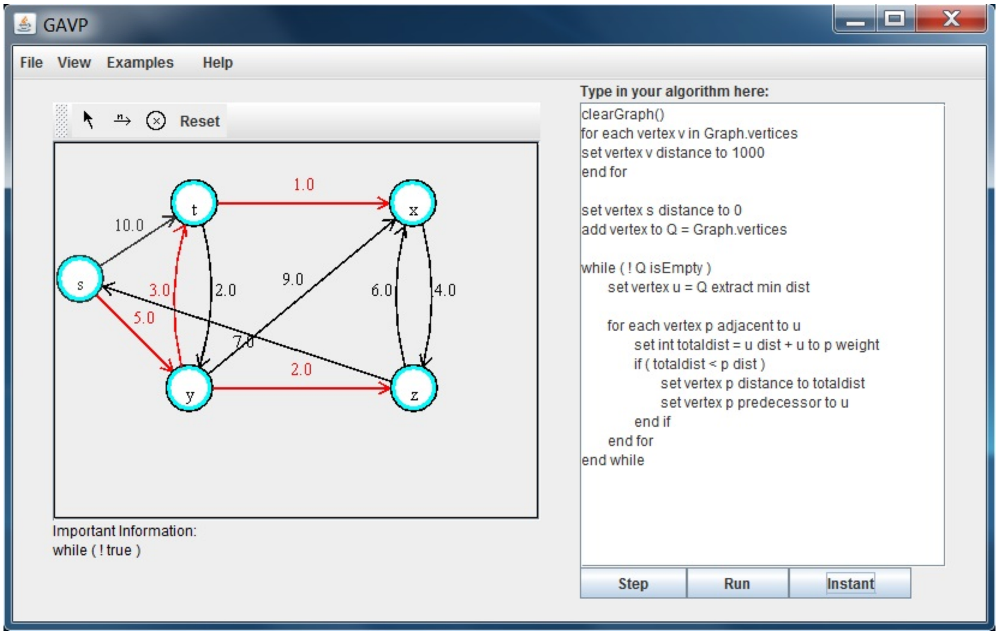
Graph Algorithm Visualization Program

David Pagels
Department of Computer Science
University of Minnesota, Morris
Morris, MN 56267
pagel093@morris.umn.edu

Abstract

Algorithms in the field of graph theory can be difficult to conceptualize through current teaching methods. To address this problem I created a tool which allows users to construct graphs and enter algorithms so that they may watch the algorithm act on the graph. This demonstration shows how to use this software. The three main components of this software allow a user to 1) construct a graph, 2) enter algorithms, and 3) watch the step-by-step effect of the algorithm acting on the graph.



Type in your algorithm here:

```
clearGraph()
for each vertex v in Graph.vertices
set vertex v distance to 1000
end for

set vertex s distance to 0
add vertex to Q = Graph.vertices

while ( ! Q isEmpty )
    set vertex u = Q extract min dist

    for each vertex p adjacent to u
        set int totaldist = u dist + u to p weight
        if ( totaldist < p dist )
            set vertex p distance to totaldist
            set vertex p predecessor to u
        end if
    end for
end while
```

Important Information:
while (! true)

Figure 1: Dijkstra's Algorithm example.

1 Introduction

Graph theory is a rapidly growing field in the computer science industry. This is because the abstract nature of graph theory allows it to have many practical applications. However, due to this abstract nature, graph theory concepts can be fairly difficult to grasp. The current visual methods of teaching graph theory typically involve sequences of figures depicting graphs shown near an associated algorithm. This leaves most of the interpretation of the figures to the students, counting on them to associate the correct statement of the algorithm with the correct figure. If these figures were turned into a single graph that changes as an algorithm is iterated over visually, it would be much easier to understand the effects of the algorithms on the graph.

2 Implementation

To solve this problem, I developed a Graph Algorithm Visualization Program (GAVP) for a wide range of uses. This program allows computer scientists learning graph theory to watch complex algorithms unfold. It also provides experienced computer scientists a tool to help develop novel graph algorithms. The main components of this program include: a graphical user interface with tools in which a user may construct a graph and enter their algorithm, and the simple language used to enter the algorithms (Figure 1). Once the graph is constructed and the algorithm is entered, the program shows the effect and progress of the algorithm on the graph step-by-step. It indicates progress by changing the colors of relevant parts of the graph, changing values associated with graph components, highlighting the line being executed, and by displaying helpful dialog under the graph.

3 Demonstration

This demonstration will show the construction of a graph and entering of a fundamental algorithm to demonstrate the visual tools and effectiveness of the program. This program was developed solely in Java, with the graphics handled by Java Swing.