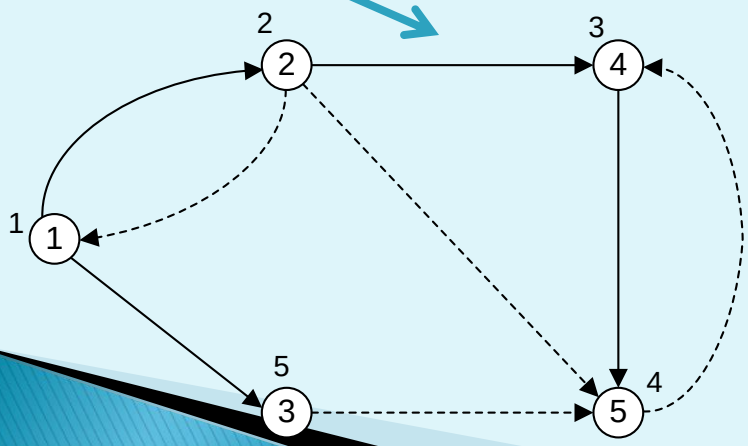
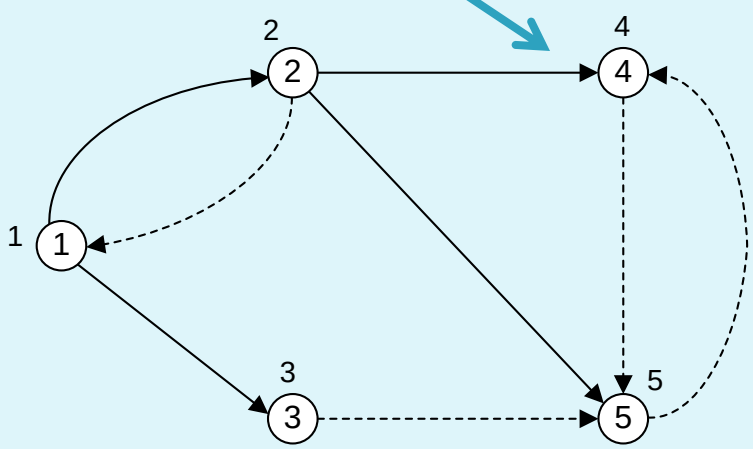
Static and Dynamic Visualization of Network Algorithms

Dr. Baloukas Thanasis

Graph and Network Problems

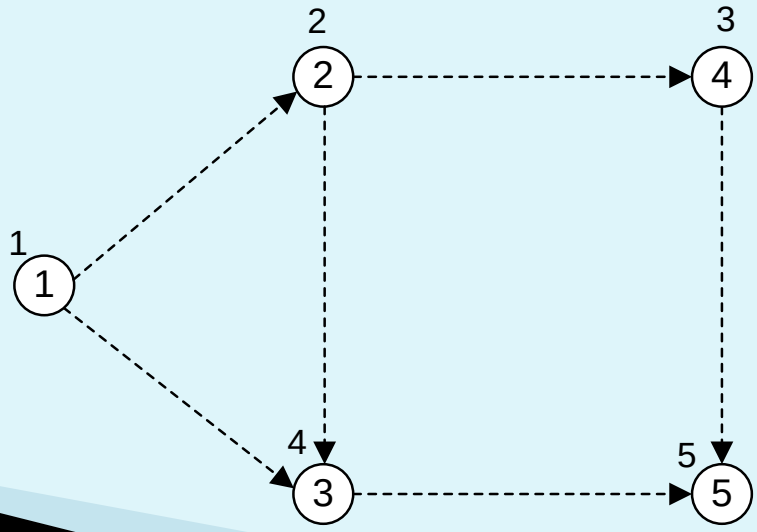
- **Graph Traversal**. How to visit all nodes in a graph (directed or undirected) in a particular manner.
- Known algorithms, Depth First Search (DFS), Breadth First Search (BFS).





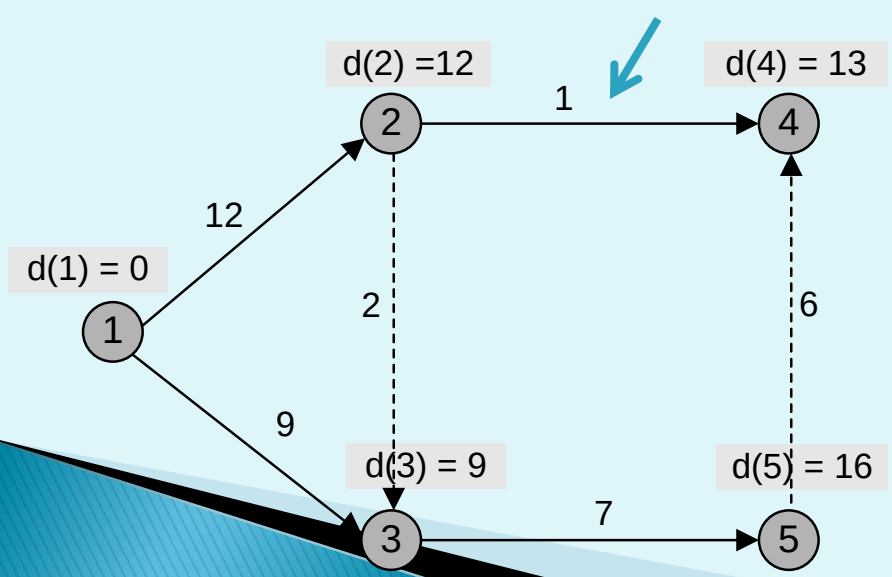
Graph and Network Problems

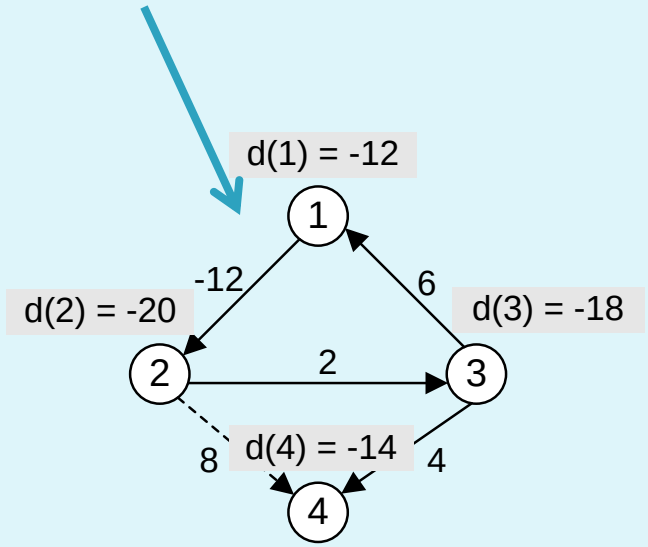
- **Topological Sorting**, of a directed acyclic graph (DAG) is a linear ordering of its nodes, in which each node comes before all nodes to which it has outbound edges.



Graph and Network Problems

- **Shortest Path Problem**. How to find a path between two nodes such that the sum of the weights of its constituent edges is minimized.
- Algorithms: Dijkstra, Bellman-Ford.



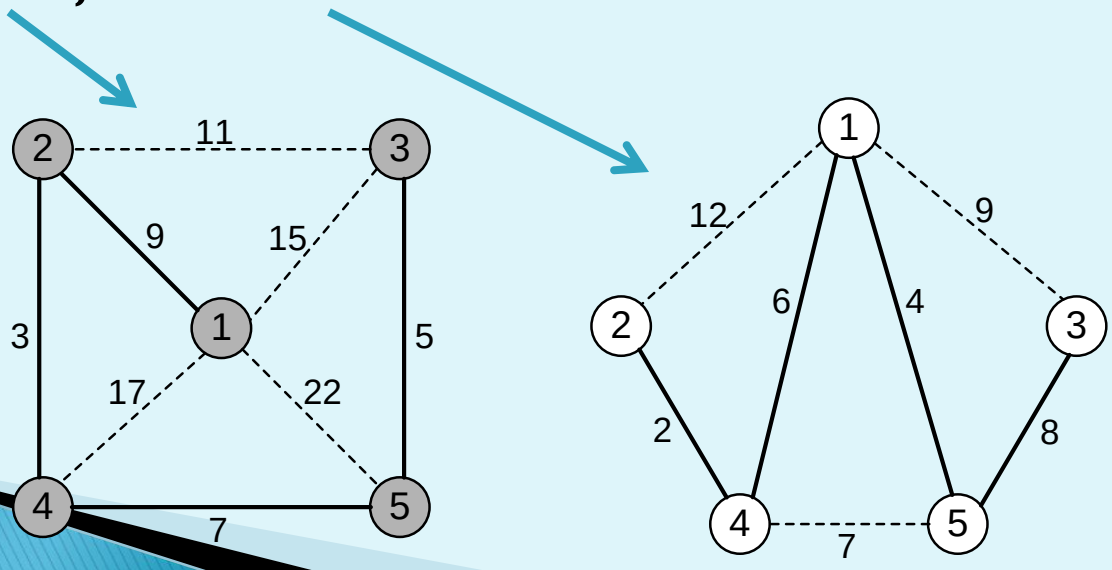


Graph and Network Problems

- **Minimum Spanning Tree** problem.
- Given a connected, undirected graph, a spanning tree of that graph is a subgraph which is a tree and connects all the vertices together. We can assign a weight to each edge.
- The weight of a spanning tree is the sum of the weights of its edges.

Graph and Network Problems

- A **minimum spanning tree (MST)** is a spanning tree with weight less than or equal to the weight of every other spanning tree. Known algorithms: Prim, Kruskal.

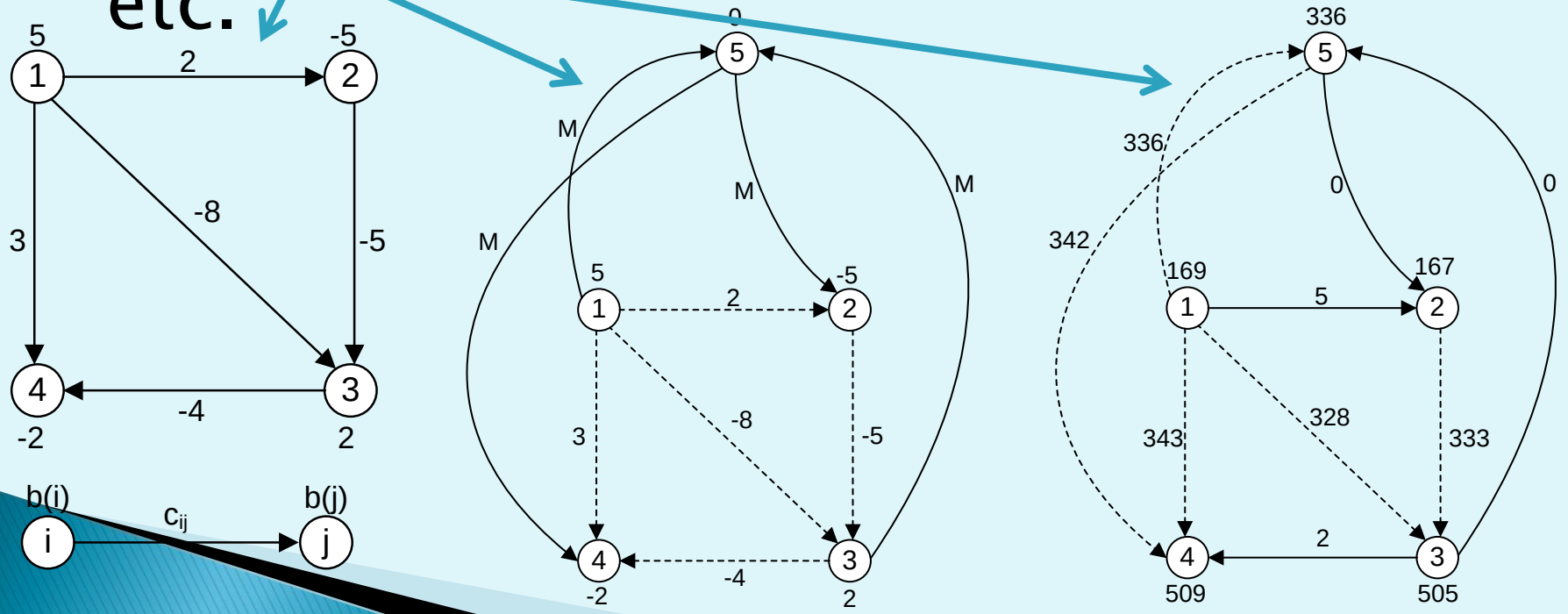


Graph and Network Problems

- **Minimum Cost NetWork Flow Problem.** Let $G = (N, A)$ be a network with n nodes and m arcs . Each node can be a supply node (supplies units of some commodity) , a demand node (demands units) or transshipment node (neither supplies nor demands).
- There is a cost to ship commodity units along arcs.

Graph and Network Problems

- **Minimum Cost NetWork Flow Problem.**
- Well-known algorithms: Network Primal Simplex, Network Dual Simplex etc.



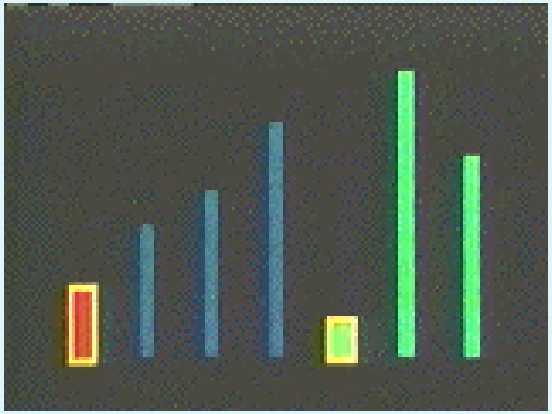
Algorithm Visualization

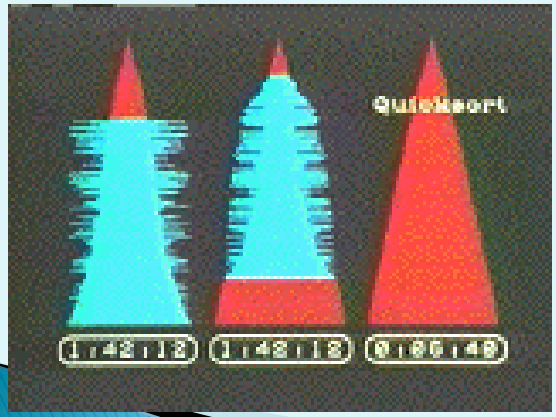
- Algorithms and Data Structures courses include difficult notions, so instructors struggled to find ways to make them easier to students.
- They use pictures or diagrams, in order to communicate complicated concepts of algorithms.
- However, the execution of algorithms is a dynamic process as it changes over time.

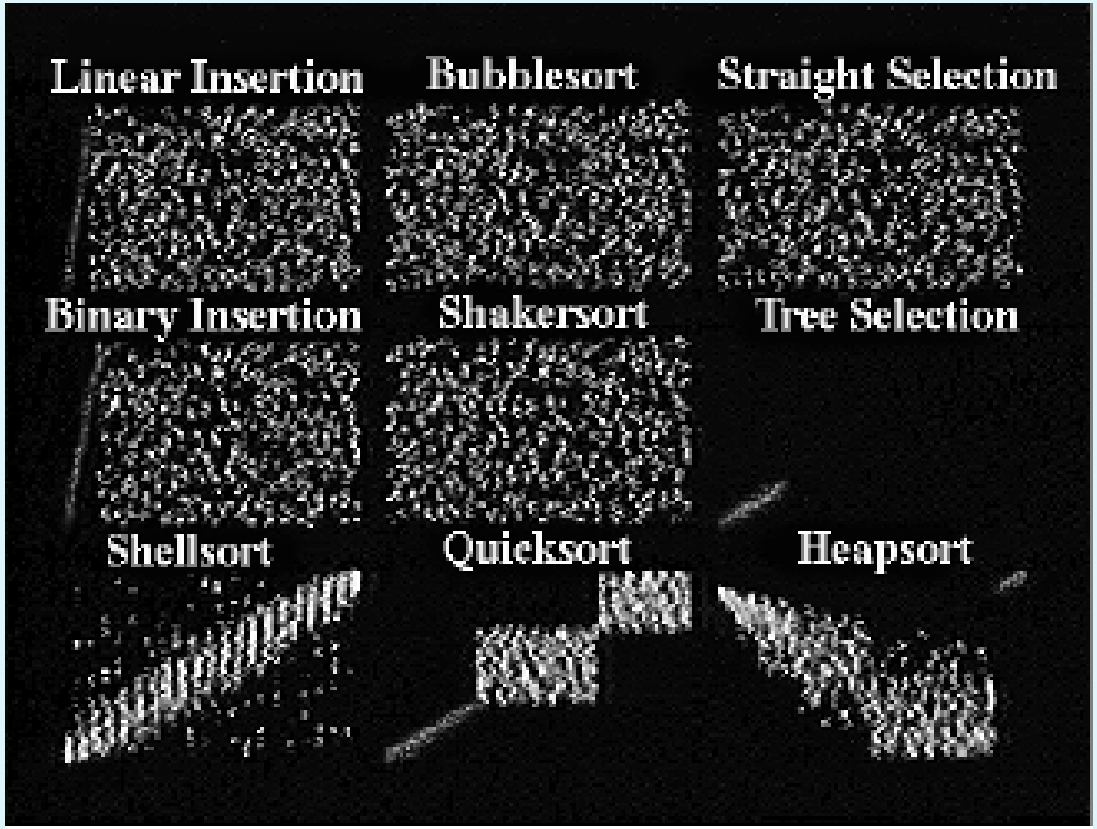
Algorithm Visualization

- Instructors realized that it would be useful to explain algorithms through animation.
- The video “Sorting out Sorting”, was a 30-minute movie, displaying animations of nine (9) sorting algorithms
- It was the first attempt to portray the execution of algorithms with moving pictures.

Video “Sorting out Sorting”







Algorithm Visualization (AV)

- Since then, many AV systems have been developed.
- Algorithm Visualization systems aim to simulate the execution of algorithms using graphics, text, sound, and animation.
- AV's aim is pedagogical.

Algorithm Visualization (AV)

- AV that uses still (motionless)

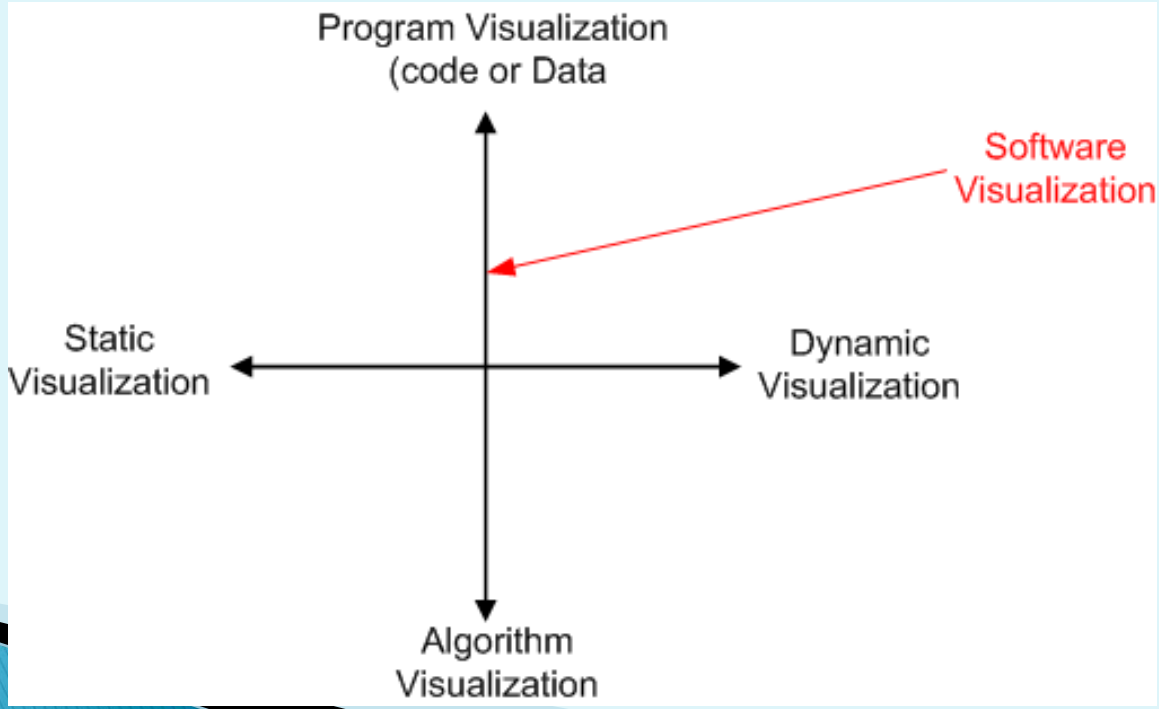
graphics is known as *static AV*.

Program Visualization (PV)

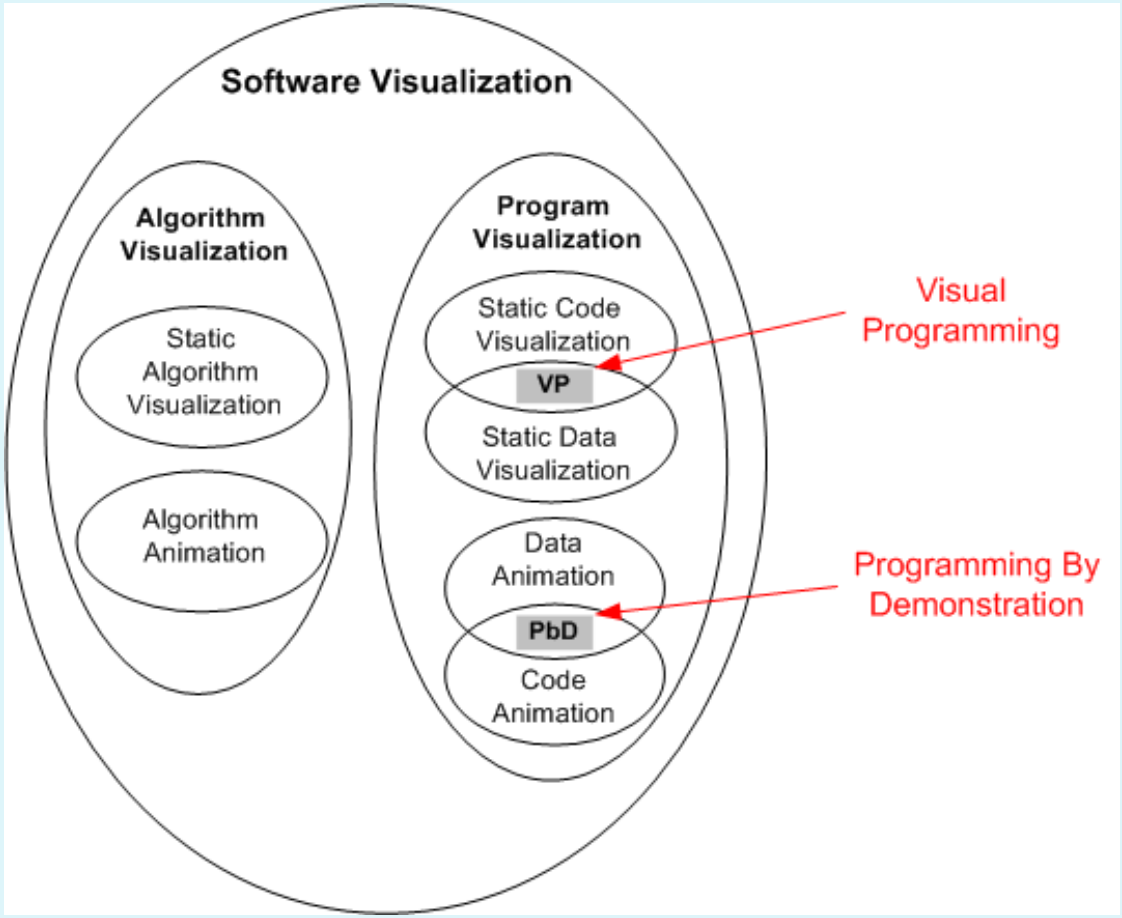
- *Program visualization(PV)* is concerned with the visualization of program execution, of either program data or code.
- It can be used in:
  - Software Engineering
  - Debugging
  - Program analysis

Software Visualization (SV)

- *Software visualization* (SV) is a scientific area, which encompasses research in both algorithm and program visualization.



Software Visualization (SV)



Software Visualization (SV)

- **SV Definitions.**
- The use of the crafts of typography, graphic design, animation, and cinematography with modern human-computer interaction and computer graphics technology to facilitate both the human understanding and effective use of computer software.

Software Visualization (SV)

- **SV Definitions.**
- A more recent definition of SV is that "SV is the visualization of artifacts related to software and its development process".

AV Systems characteristics

- The mainstream of AV systems are implemented in Java.
- Some AV systems display a number of ready-made visualizations implemented by their authors.
- Other AV systems incorporate some scripting language, allowing users to create AV's by themselves, by writing programs in this scripting language.

AV Systems characteristics

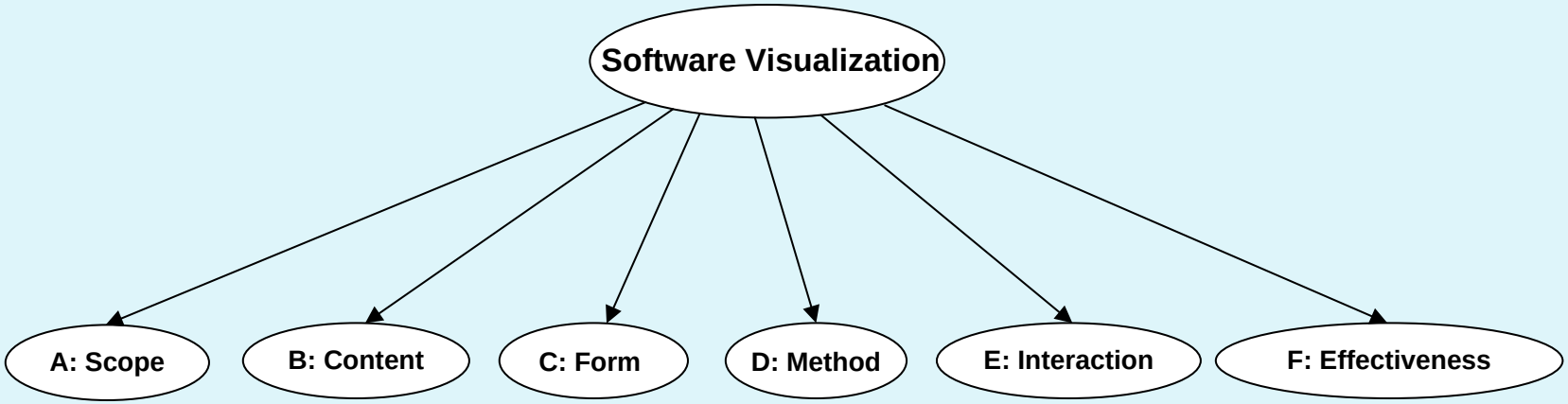
- In several AV tools, users are passive viewers of visualizations whereas other systems let users interact with them.
- Some recent AV systems exhibit assessment and grading features, as they can evaluate and score students' knowledge about algorithms and data structures, in relevant exercises generated by the systems.

SV Systems taxonomy

- To date, one classification regarding SV systems is widely known and it was contributed by Price et al.
- It is a multi-level hierarchical classification consisting of many categories and sub-categories, based on the characteristics of twelve (12) SV systems.

SV Systems taxonomy

- The first level of this classification includes the following six categories: *scope*, *content*, *form*, *method*, *interaction*, and *effectiveness*.



AV Systems taxonomy

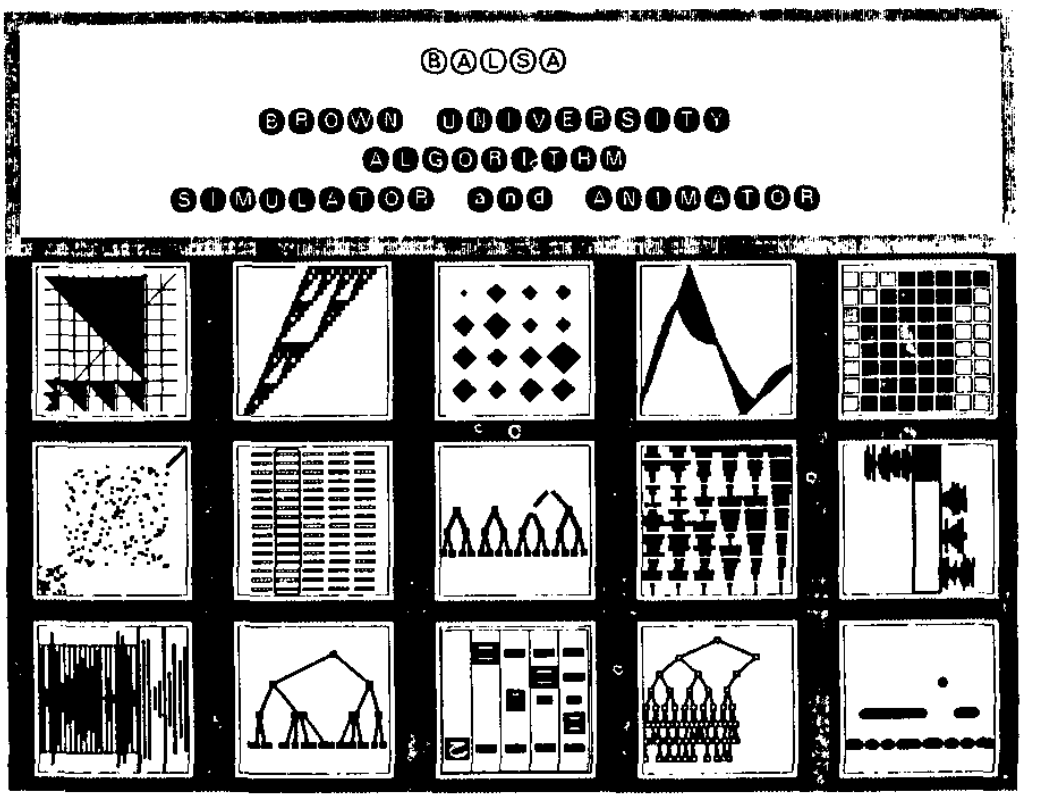
- A further classification has been suggested and it is concerned with classifying AA scripting languages.
- According to this classification, the top-level categories are *animation*, *interaction*, *positioning*, *style*, *utilities*, and *vocabulary*.

AV Systems- BALSA

- BALSA (1984) is an early algorithm animation software.
- Views, in BALSA terminology, display images that illustrate different aspects of the algorithm being animated.
- The system supports multiple *dynamic* views. That is, their content is updated dynamically at the same time with the execution of an algorithm.

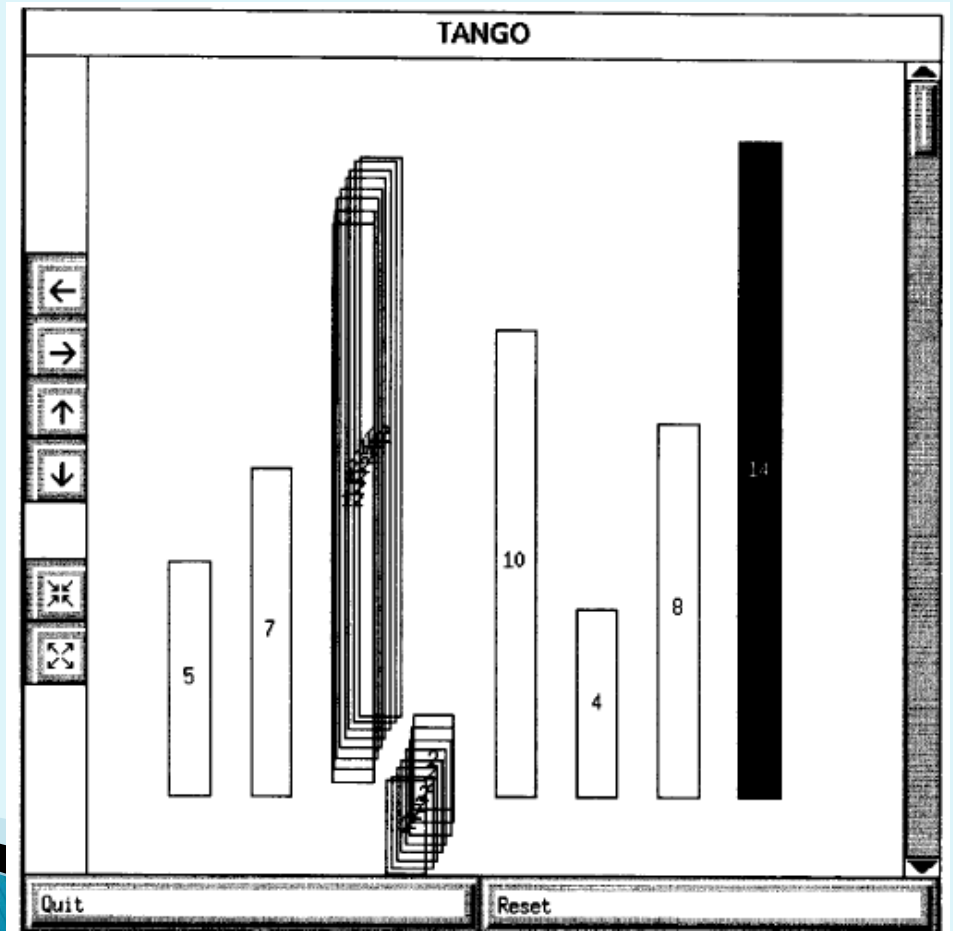
AV Systems- BALSA

- Users of the system can control the execution of algorithms and adjust the presentation of views.



AV Systems- TANGO

- TANGO (1990) is a system quite similar to BALSA.

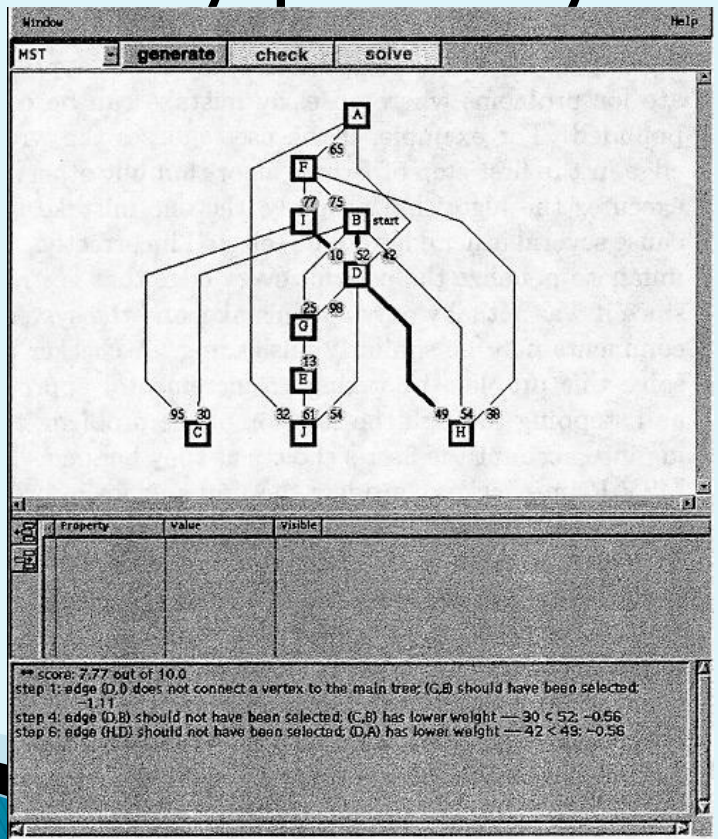


AV Systems- PILOT

- PILOT (2000) is an AV tool, which can be used to assess students' knowledge about common graph algorithms.
- The tool can generate random graph problems allowing users to solve them on-line.
- After submitting their answer to the PILOT, users can see their scores as well as the correct solution.

AV Systems- PILOT

- The tool supports partial grading. Students obtain a grade even if their solution is only partially correct.

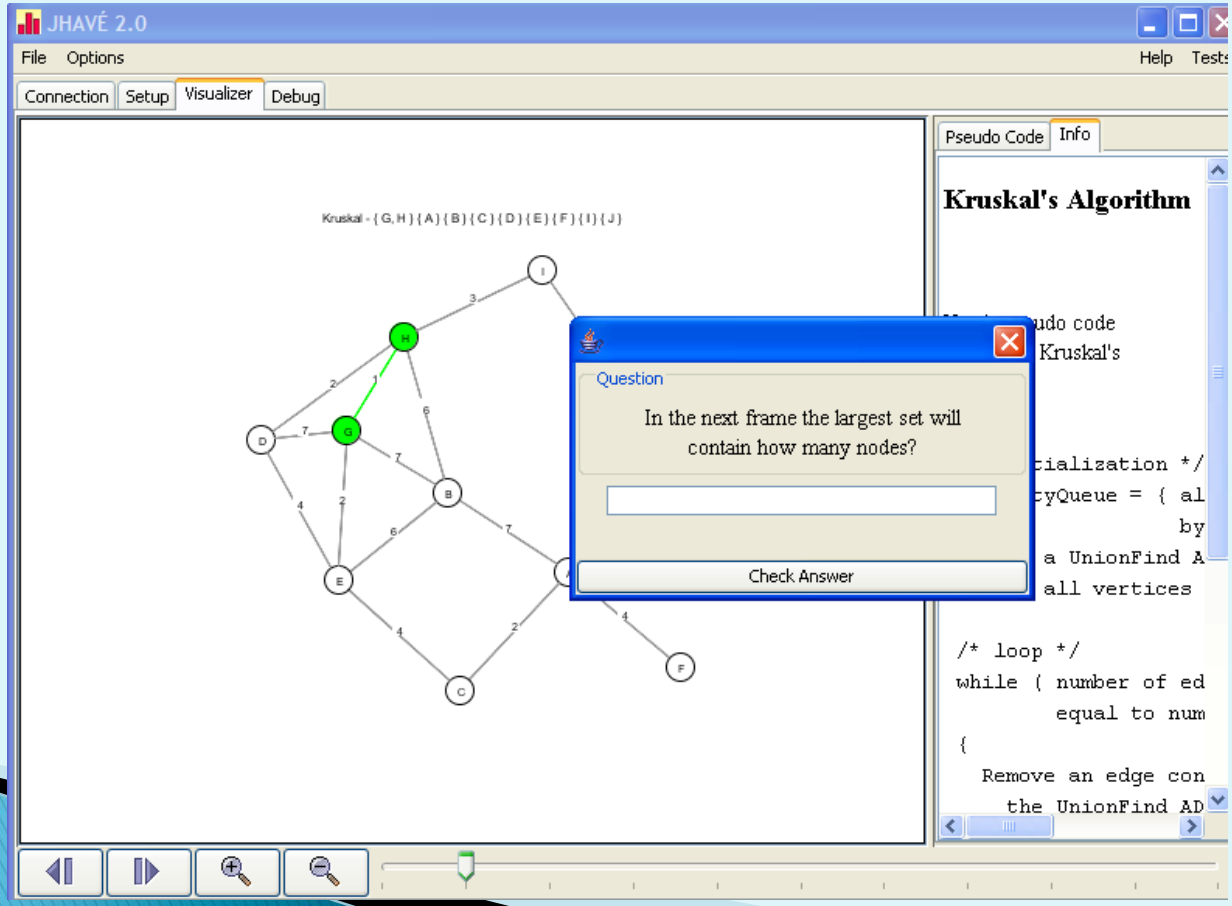


AV Systems- JHAVE

- JHAVE (2000) is another AV tool.
- While executing some AV, the tool highlights the relevant line of the algorithm's pseudo-code in a separate window.
- One strong feature of the system is that it can generate predictive questions, which pauses the visualization, forcing students to respond to them.

AV Systems- JHAVE

- Students are not passive viewers of AV's, but they are actively engaged.

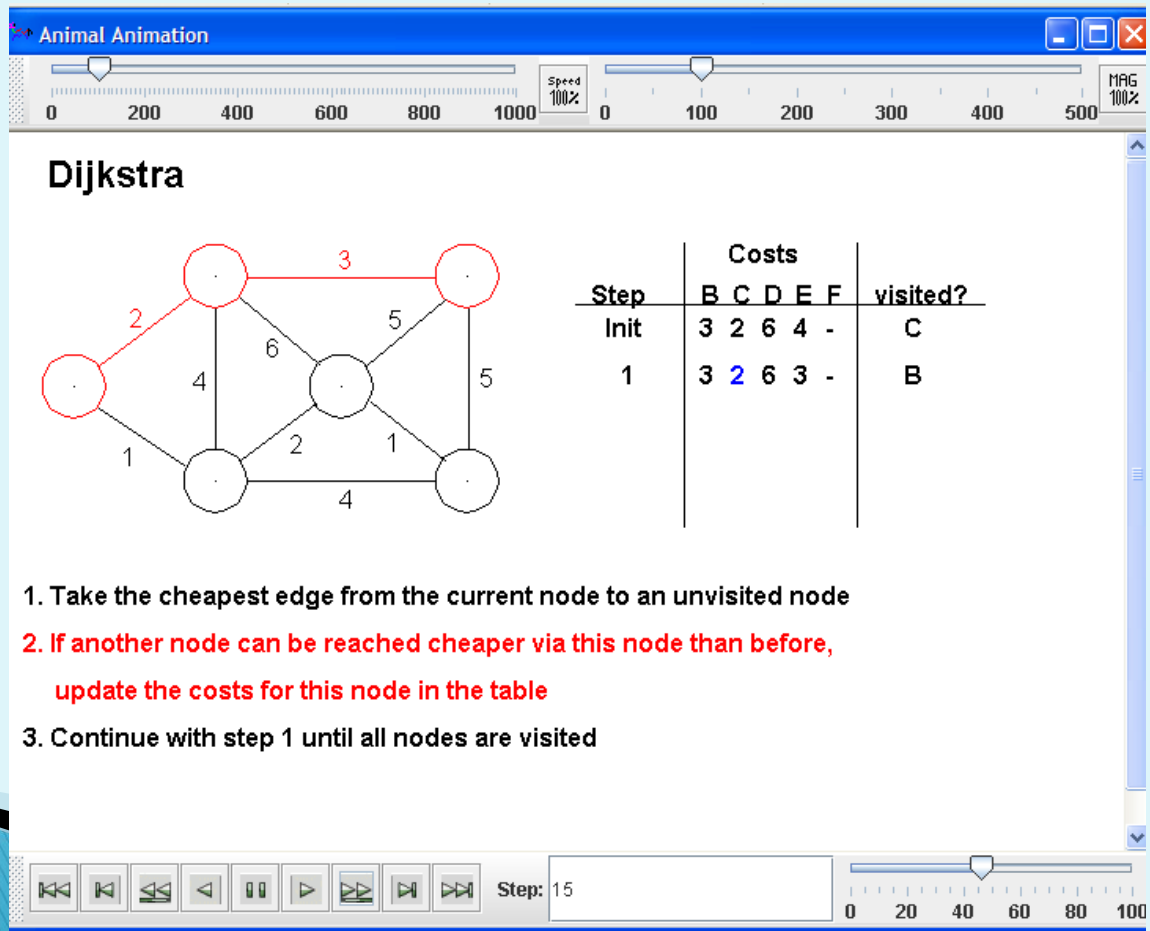


AV Systems- ANIMAL

- ANIMAL (2002) is a further AV software demonstrating animations for various algorithms and data structures.
- The originality of this tool lies in the fact that it supports three user roles: users, visualizers, and developers.

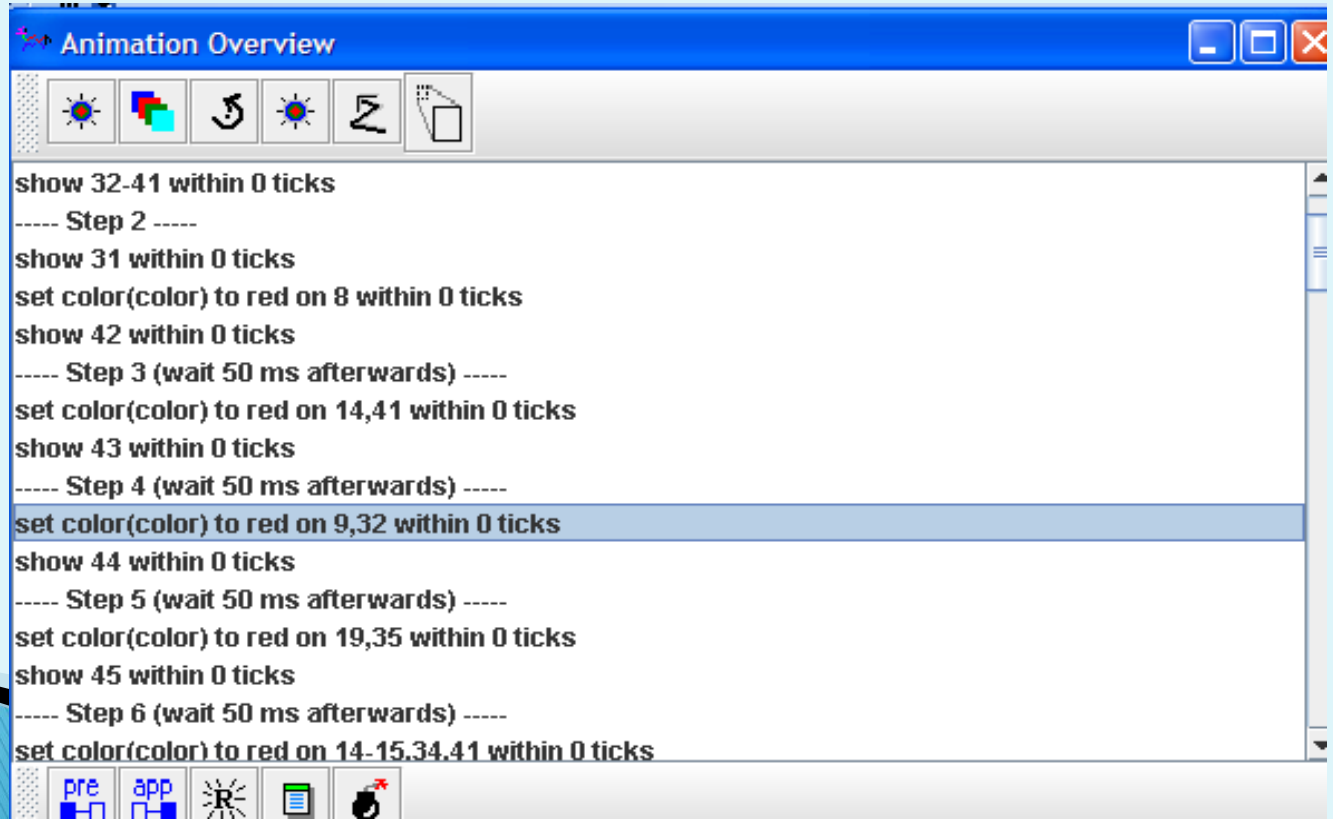
AV Systems- ANIMAL

- Users can adjust the presentation and the execution of visualizations.



AV Systems- ANIMAL

- Visualizers can produce animations through a graphic interface using a scripting language.



AV Systems- ANIMAL

- Developers can easily extend the system without modifying the system's code. They only have to know the very basics of Java.

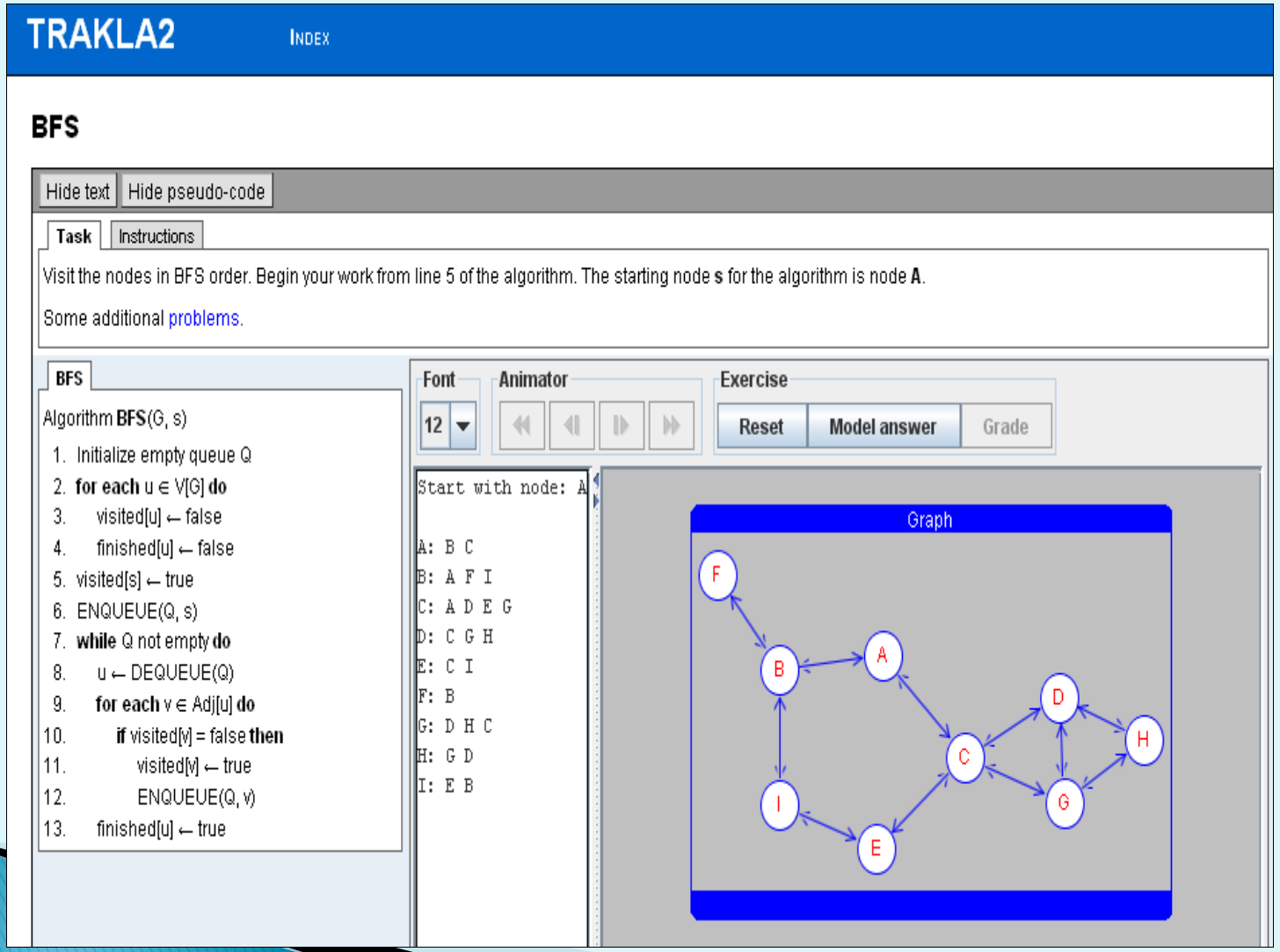
AV Systems- TRAKLA2

- TRAKLA2 (2004) is an assessment tool for algorithmic assignments.
- It generates random algorithmic assignments and asks students to solve them, by graphically simulating operations of algorithms.
- While students solve assignments, the system records the sequence of graphic operations performed by them.

AV Systems- TRAKLA2

- Then, students submit their solutions to the system for evaluation and they receive immediate feedback about the number of correct steps that they have followed to solve the problem.
- The tool can model solutions of exercises by presenting students with relevant algorithm animations.

AV Systems- TRAKLA2



Algorithm Visualization Wiki

- In a recent study (2007), researchers collected and catalogued over 350 visualizations, thus creating an AV Wiki (http://algoviz.org/).
- They concluded that the majority of existing AV tools visualize simple algorithms and data structures.
- In contrast, more difficult and complex algorithms have not yet been visualized.

Algorithm Visualization Wiki

AlgoViz.org
The Algorithm Visualization Portal

Search

About AlgoViz.org | Participate | Your account | User list | Log out
Hello, thanasis!

Home | AV Catalog | Annotated Bibliography | Field Reports | For Developers | Forums | Getting Started

AlgoViz.org is a gathering place for users and developers of algorithm visualizations and animations (AVs). It is a gateway to AV-related services, collections, and resources.

Browse the Catalog

Algorithmic Techniques
Compression Algorithms
Computational Geometry
Graph Algorithms
Linear Structures
Misc. Data Structures
Misc. Sorts
Misc. Topics
N log N sorts
NP Completeness
Numerical Algorithm
Quadratic Sorts
Search Algorithms
Search Structures
Spatial Search Structures
Systems and Languages

AlgoViz Notifications via Twitter and RSS

Submitted by shaffer on 4 November 2010 - 4:02pm

Want to know when there is a new AlgoViz news story? Or want to know if there is a new field report here, but don't want to need to keep checking just in case? You can receive notifications about activity at AlgoViz.org in three ways: email, Twitter, or RSS feed. We now have five Twitter and RSS feeds: (1) News articles and field reports; (2) Forum posts; (3) New catalog entries; (4) New bibliography entries; and (5) The full set of all AlgoViz notifications.

Add new comment Read more 48 reads

Nominations are now open for the 2011 AlgoViz Awards

STAY CONNECTED

RSS feed Twitter facebook YouTube

AlgoViz.org Awards

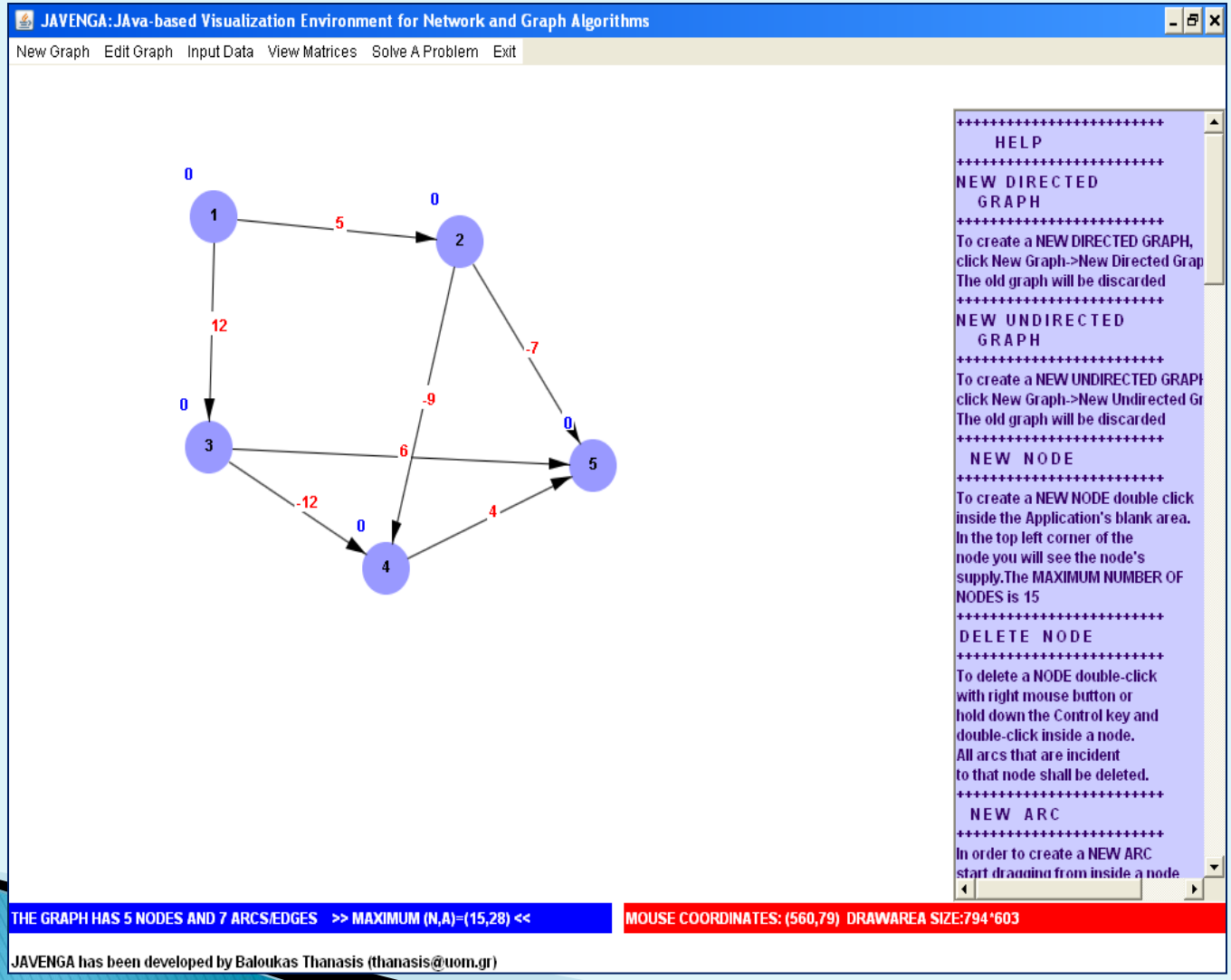
- AlgoViz.org Awards
  - Nominees List [+] Feedback

Latest Field Report

JAVENGA

- **JAVENGA** (**JA**va_based **V**isualization **E**nvironment for **N**etwork and **G**raph **A**lgorithms) comes to fill this gap, as it features a visualization of a quite complex and important algorithm, the Network Simplex Algorithm.

JAVENGA


JAVENGA:JAva-based Visualization Environment for Network and Graph Algorithms
New Graph
Edit Graph
Input Data
View Matrices
Solve A Problem
Exit
0
1
5
0
2
12
-7
-9
0
3
6
5
-12
0
4
4
HELP
NEW DIRECTED GRAPH
To create a NEW DIRECTED GRAPH,
click New Graph->New Directed Grap
The old graph will be discarded
NEW UNDIRECTED GRAPH
To create a NEW UNDIRECTED GRAPH
click New Graph->New Undirected Gr
The old graph will be discarded
NEW NODE
To create a NEW NODE double click
inside the Application's blank area.
In the top left corner of the
node you will see the node's
supply.The MAXIMUM NUMBER OF
NODES is 15
DELETE NODE
To delete a NODE double-click
with right mouse button or
hold down the Control key and
double-click inside a node.
All arcs that are incident
to that node shall be deleted.
NEW ARC
In order to create a NEW ARC
THE GRAPH HAS 5 NODES AND 7 ARCS/EDGES >> MAXIMUM (N,A)=(15,28) <<
MOUSE COORDINATES: (560,79) DRAWAREA SIZE:794*603
JAVENGA has been developed by Baloukas Thanasis (thanasis@uom.gr)

JAVENGA – Features

- JAVENGA exhibits many novel features, the combination of which cannot be found in any similar purpose AV tool.
- Essential features of the tool are:
  - Integration of a graph editor for drawing graphs.
  - Adaptability of the tool to users' preferences.
  - Display of the execution history of AVs.

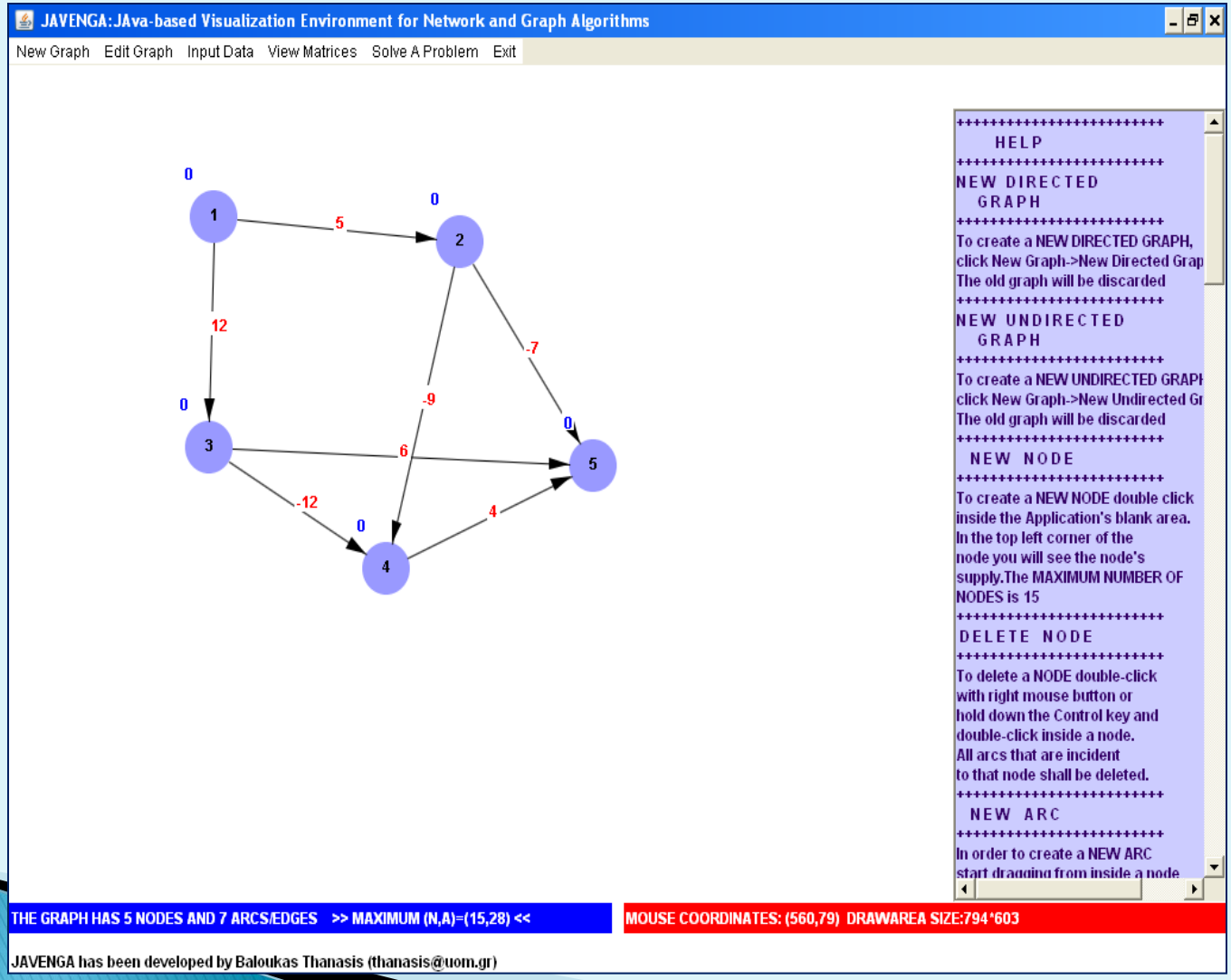
JAVENGA – Features

- Essential features of the tool are:
  - Possibility to try out different input data for AVs.
  - High degree of interactivity.
  - The flexibility of the tool's execution either as Java applet or Java application.

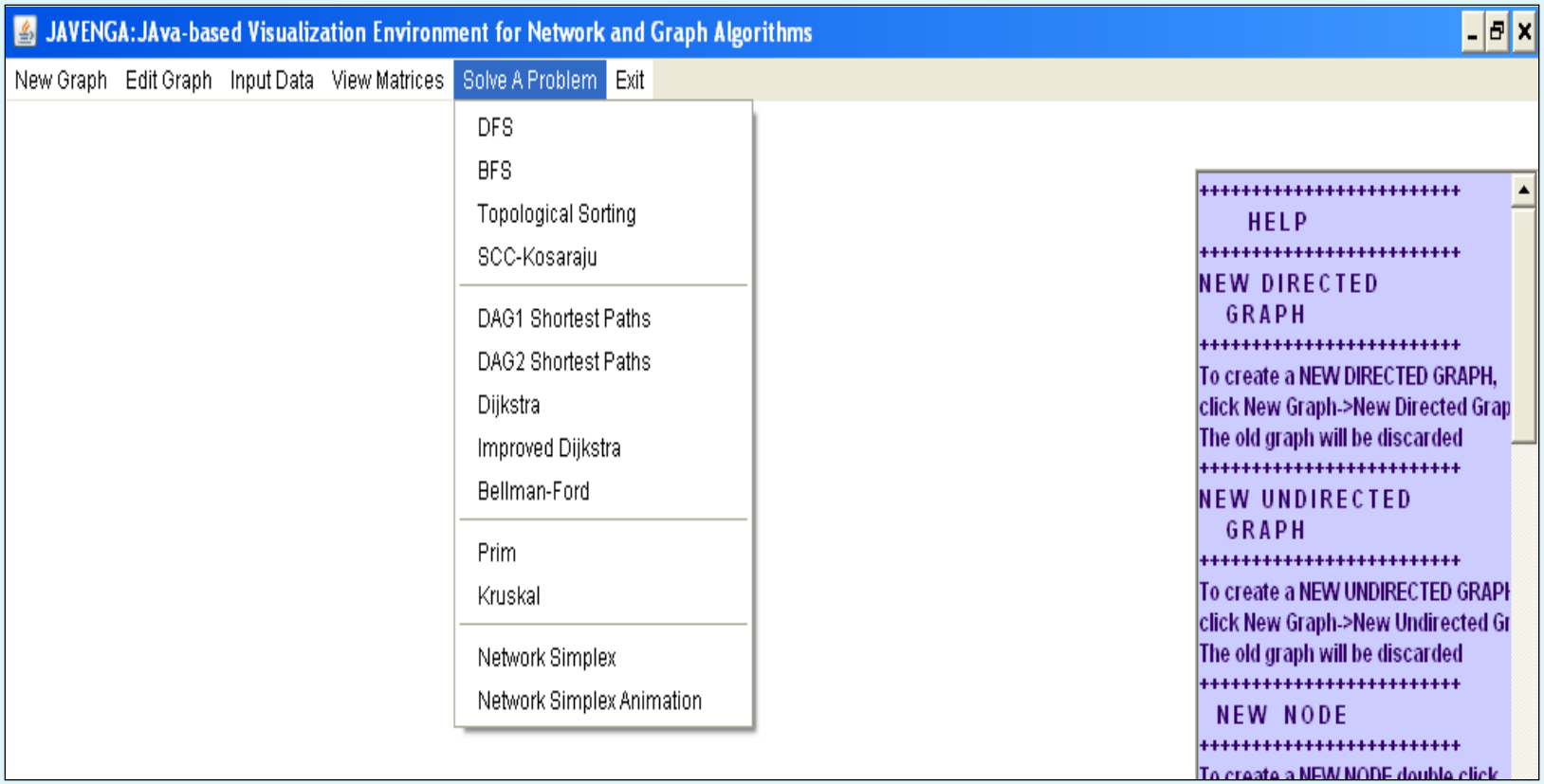
JAVENGA – GUI

- The GUI consists of four parts:
  1. The *application menu bar*.
  2. The *main application window* , that can be used :
     - As a graph editor, allowing users to edit directed or undirected graphs.
     - To display visualizations of algorithms.

JAVENGA – GUI


JAVENGA:JAva-based Visualization Environment for Network and Graph Algorithms
New Graph
Edit Graph
Input Data
View Matrices
Solve A Problem
Exit
0
1
5
0
2
12
-7
0
-9
0
3
6
5
-12
0
4
4
HELP
NEW DIRECTED GRAPH
To create a NEW DIRECTED GRAPH,
click New Graph->New Directed Grap
The old graph will be discarded
NEW UNDIRECTED GRAPH
To create a NEW UNDIRECTED GRAPH
click New Graph->New Undirected Gr
The old graph will be discarded
NEW NODE
To create a NEW NODE double click
inside the Application's blank area.
In the top left corner of the
node you will see the node's
supply.The MAXIMUM NUMBER OF
NODES is 15
DELETE NODE
To delete a NODE double-click
with right mouse button or
hold down the Control key and
double-click inside a node.
All arcs that are incident
to that node shall be deleted.
NEW ARC
In order to create a NEW ARC
start dragging from inside a node
THE GRAPH HAS 5 NODES AND 7 ARCS/EDGES >> MAXIMUM (N,A)=(15,28) <<
MOUSE COORDINATES: (560,79) DRAWAREA SIZE:794*603
JAVENGA has been developed by Baloukas Thanasis (thanasis@uom.gr)

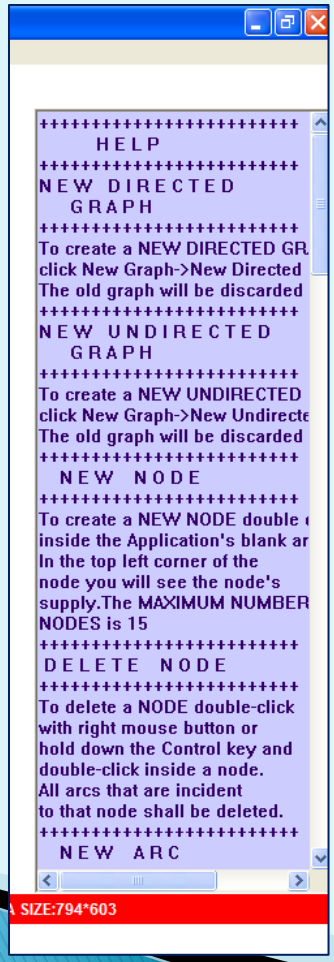
JAVENGA – Menu Bar

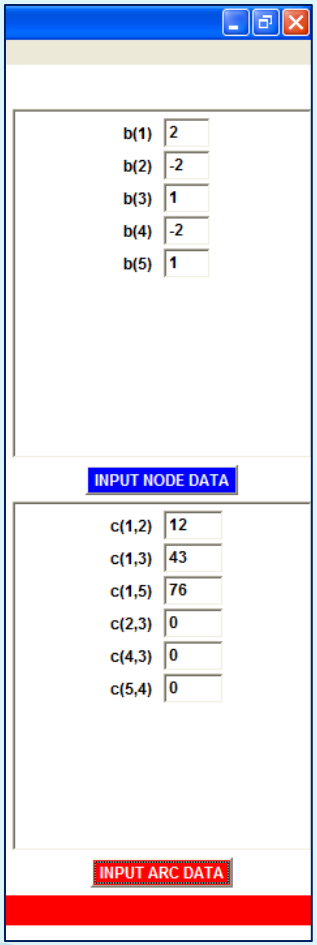


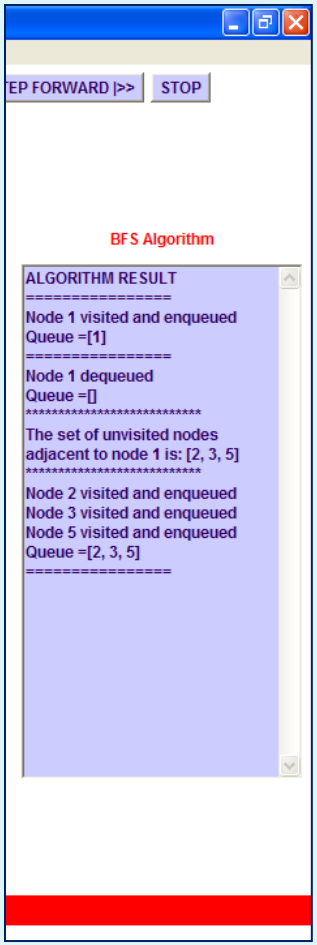
JAVENGA – GUI

- The GUI consists of four parts:
  3. The *vertical window* next to the graph editor can be used:
     - To display the software help at the same time that a user draws a graph.
     - To give input data for a weighted graph (arc costs and / or node supplies / demands).
     - To display the values of variables during execution of algorithms' visualization.

JAVENGA – GUI

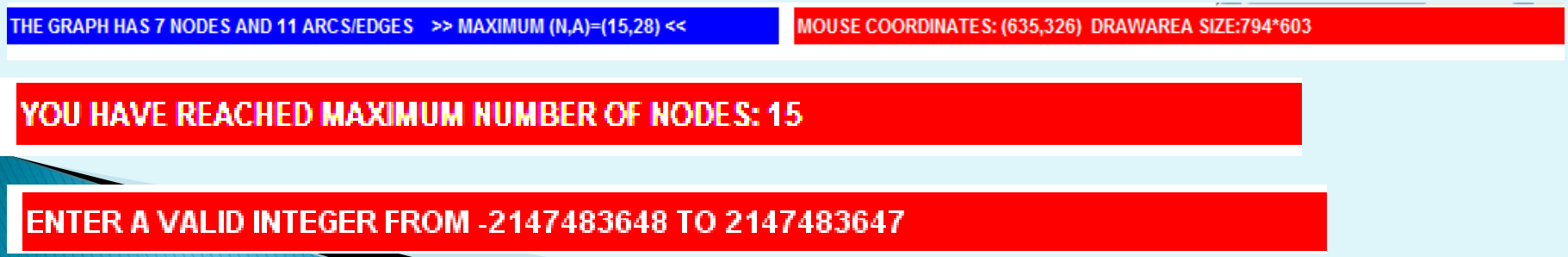

HELP
NEW DIRECTED GRAPH
To create a NEW DIRECTED GR
click New Graph->New Directed
The old graph will be discarded
NEW UNDIRECTED GRAPH
To create a NEW UNDIRECTED
click New Graph->New Undirecte
The old graph will be discarded
NEW NODE
To create a NEW NODE double
inside the Application's blank ar
In the top left corner of the
node you will see the node's
supply.The MAXIMUM NUMBER
NODES is 15
DELETE NODE
To delete a NODE double-click
with right mouse button or
hold down the Control key and
double-click inside a node.
All arcs that are incident
to that node shall be deleted.
NEW ARC
SIZE:794*603


b(1) 2
b(2) -2
b(3) 1
b(4) -2
b(5) 1
INPUT NODE DATA
c(1,2) 12
c(1,3) 43
c(1,5) 76
c(2,3) 0
c(4,3) 0
c(5,4) 0
INPUT ARC DATA


EP FORWARD |>>
STOP
BFS Algorithm
ALGORITHM RESULT
Node 1 visited and enqueued
Queue =[1]
Node 1 dequeued
Queue =[]
The set of unvisited nodes
adjacent to node 1 is: [2, 3, 5]
Node 2 visited and enqueued
Node 3 visited and enqueued
Node 5 visited and enqueued
Queue =[2, 3, 5]

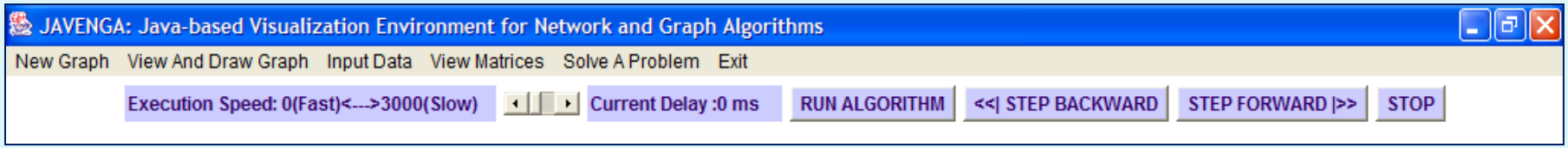
JAVENGA – GUI

- The GUI consists of four parts:
  4. The *horizontal bar* at the bottom which consists of two labels. Inside these labels, informative text is displayed, as well as warning messages in response to users' erroneous actions.



JAVENGA – GUI

- There are two modes in which an AV can be executed:
  - Step by step (forward or backwards) or
  - At once (non-stop), providing users with the option of specifying the speed of visualization execution.

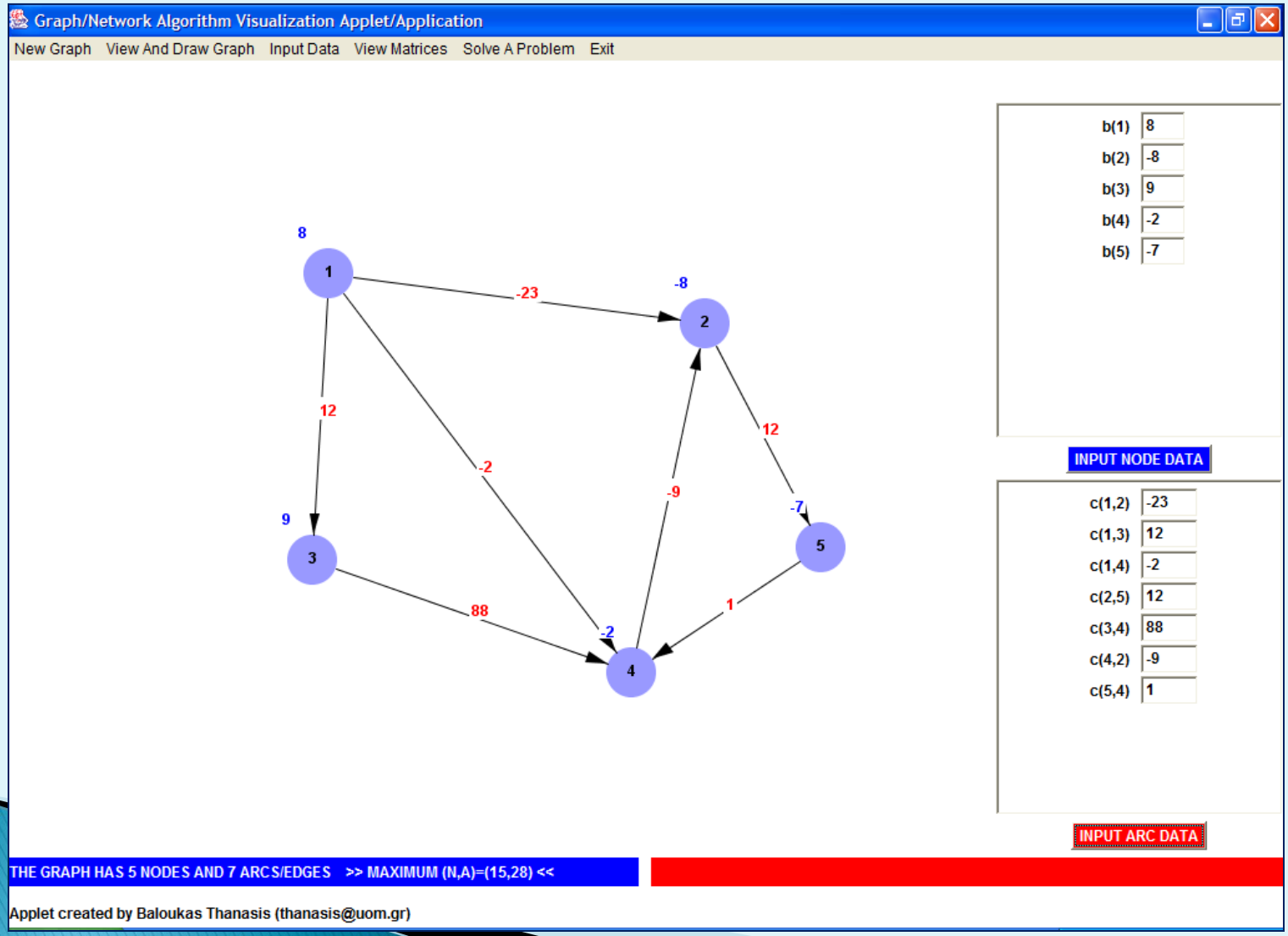


JAVENGA – AV Example

- The animation of Primal Simplex Network algorithm on a step-by-step basis, is executed as follows:

1. The user has to create a new directed graph using the menu “New Graph” →“New Directed Graph”.
2. He enters the graph editor, where he constructs the graph.
3. He chooses “Input Data” from the main menu bar, to input values for nodes/arcs.

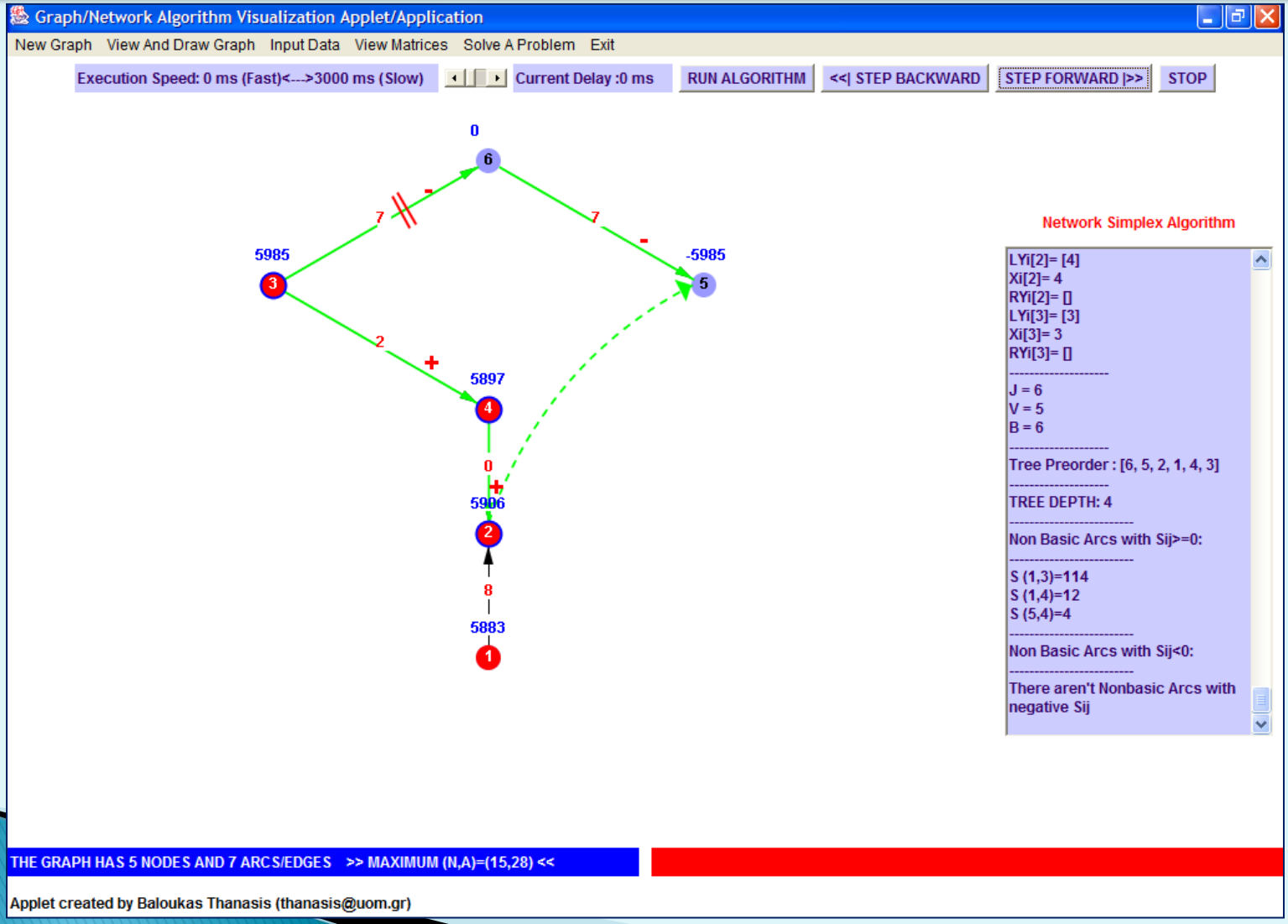
JAVENGA – AV Example



JAVENGA – AV Example

4. The user clicks *Solve A Problem → Network Simplex Animation*.
5. Next, a toolbar containing four buttons, one slider, and two labels is shown.
6. Using this toolbar, one can opt to see either a nonstop animation (button "RUN ALGORITHM") or a stepwise animation.

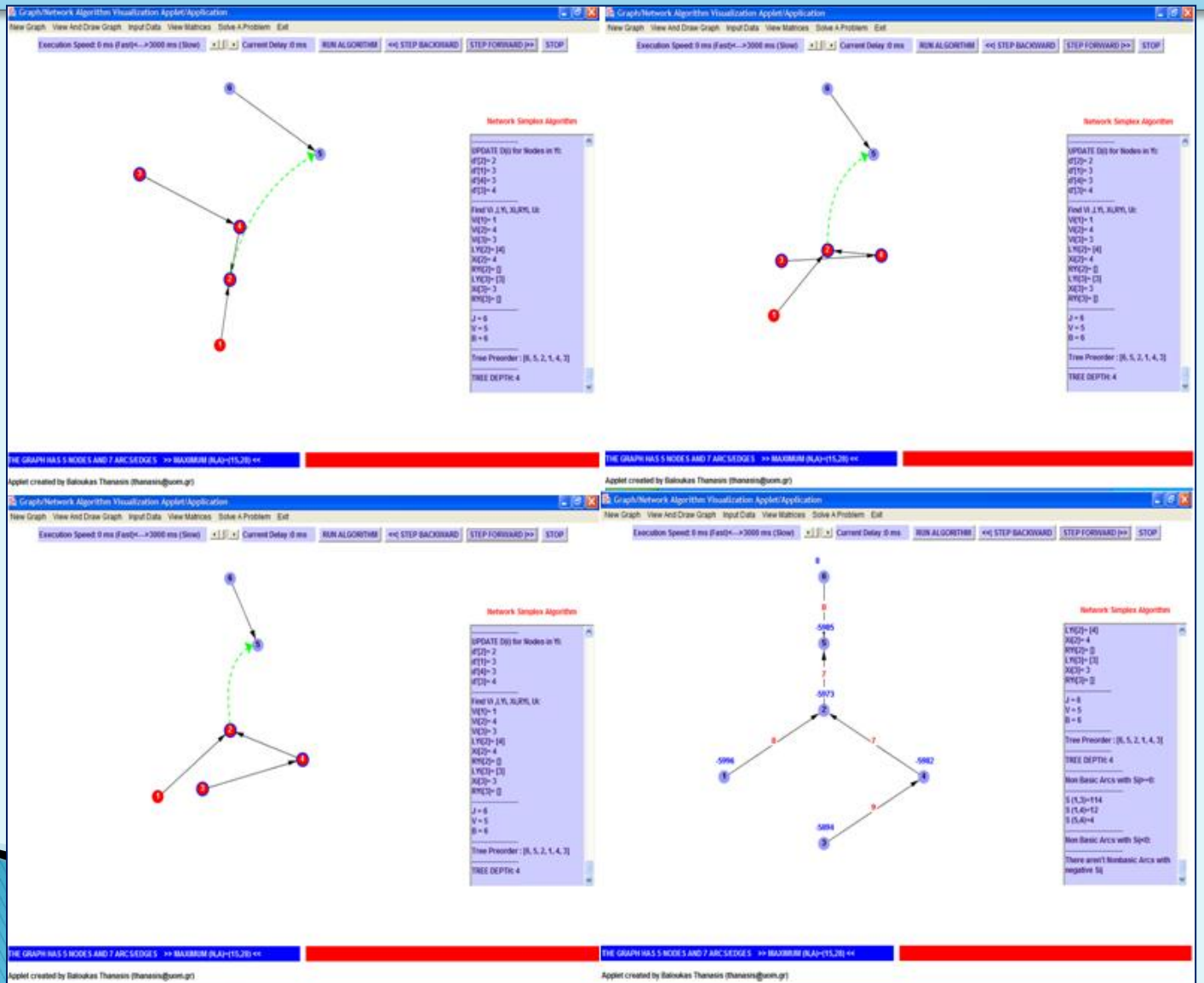
JAVENGA – AV Example



JAVENGA – AV Example

- The stepwise animation can be run either step-by-step forwards by pressing the button labelled “STEP FORWARD |>>” or step-by-step backwards by pressing the button labeled “<<| STEP BACKWARD.”
- In the following figure we display graphical snapshots of the animated transition of the Network Simplex algorithm.

JAVENGA – AV Example



JAVENGA – AV Example

- During the visualization process, values of algorithm variables are displayed inside the blue vertical window that lies beside the main visualization window.
- Values of variables are shown for all iterations from the beginning of the algorithm execution.

JAVENGA – Possible uses

- Potential uses of the tool are:
- Computer Science or Engineering students can use the tool in an independent study approach as they attempt to comprehend graph and network algorithms.

JAVENGA – Possible uses

- Computer Science or Engineering instructors can use the tool as auxiliary teaching material in the context of a course such as graph theory, combinatorial optimization, or network flows.
- Besides, they can make use of the tool in order to grade students' exams or assignments.

JAVENGA – Possible uses

- Researchers in AV can try out the software, obtaining new ideas as they struggle to develop a new AV system.
- Researchers in Computer Science Education can use the tool to conduct educational experiments in order to assess the educational usefulness of AV tools.

Educational Evaluation

- AV designers conduct empirical studies to find out to what degree AV tools contribute to a better comprehension of algorithms.
- During experiments, which consist of both training and examination sessions, students are divided into two or more groups.

Educational Evaluation

- Every group displays a different degree of interaction with some AV tool:
  - Members of one group only watch prepared AV's.
  - Members of another group watch and interact with AV's.
  - Members of another group answer predictive questions, and so on.

Educational Evaluation

- In the final sessions of experiments, post-tests are employed to evaluate students' understanding of algorithms.
- Test questions are usually designed in line with Bloom's taxonomy of understanding.
- Researchers reach conclusions about AV's didactic usefulness by comparing groups' performances in these post-tests.

Educational Evaluation Results

- Results are somewhat mixed. Some studies have shown little pedagogical benefits but other ones do show benefits.
- Potential for pedagogical aid exists, but we can't just throw an AV and expect it to help.

Educational Evaluation Results

- Interaction is a key factor.
- Blindly watching AV is not really helpful according to several studies.
- Student must interact with animation and be engaged.
- Some studies show that animations help students' motivation and they can make algorithms easier to understand.

JAVENGA Experiment

- We carried out an experiment to measure students' understanding on Dijkstra's algorithm.
- Dijkstra's algorithm was used for first time to assess AV's didactic value.
- Participants were fifteen (15) volunteer students.
- Subjects were randomly divided into two groups.

JAVENGA Experiment

- In the first twenty(20) minutes we gave a lecture to all students, explaining the algorithm.
- Then, we allowed students to study a printed copy of the lecture slides, for fifteen (15) minutes.
- Afterward, we let students in the **first only group** to interact with the tool.

JAVENGA Experiment

- Next, students were asked to answer ten (10) questions in a post-test.
- The first seven (7) were true/false and could be characterized as **easy** questions.
- The last three (3) were free response questions which required a **deeper understanding** of algorithm concepts.

JAVENGA Experiment

- We **hypothesized**, that students in first group (which interacted with the tool), would score better **both** in easy and difficult questions.
- Post-test results, showed that the group which did not use the AV tool scored higher on the easy questions (**hypothesis rejection**).

JAVENGA Experiment

- However, results **verified** our hypothesis, that the group which used the AV tool would score higher on the **difficult** questions.

Future Research

- We plan to display the pseudo-code of an algorithm as it is being visualized.
- The visualization of more algorithms is another plan.
- The incorporation of smooth animations to the existing static visualizations would be a valuable enrichment.

Future Research

- It is our intention to include questions, asking students to guess an algorithm's subsequent behavior.
- We want our tool to have the capacity to assess students' knowledge about graph and network algorithms.
- Finally, more empirical studies need to be done using JAVENGA.