## Introduction

In contemporary computer science curricula, the inclusion of a comprehensive module on data structures and algorithms has become imperative across numerous colleges and universities. This module encompasses various data structures, graph representations, and intricate logical problem-solving. Mastering algorithms not only fosters abstract thinking and problem decomposition skills but also instills proficiency in algorithm design and analysis.

Graph structures hold a pivotal role in computer science and its affiliated domains. These structures not only serve as powerful representations of relationships and connections in real-world problems but also offer profound insights and analytical abilities. Therefore, a profound understanding and mastery of graph algorithms are indispensable for computer science students.

Driven by our ardent interest in graph traversal algorithms and our fervent desire to facilitate a more intuitive and convenient learning experience for students, we have embarked on this project. Our goal is to develop two plug-ins for GrALoG, a visualisation software, that effectively and visually display the traversal results of the breadth-first search algorithm and the depth-first search algorithm. The plug-ins will serve as a valuable aid in algorithm classes and practical work, enhancing the teaching and learning of algorithms.

Moreover, our endeavor aspires to make significant contributions to research and education in graph theory, algorithmic analysis, and related fields. It is hoped that by providing accessible resources, this tool will help more researchers, students and enthusiasts gain insight into graph algorithms and their practical applications.

## Plugin Architecture

To integrate with GrALoG, the plug-ins take advantage of the software's extensibility. These plug-ins have been developed as extensions of GrALoG, enabling them to seamlessly fit into existing environments and user interfaces. This integration ensures a smooth user experience when working with visualised graph data.

The architecture of each plug-in builds on the GrALoG user interface with added functionality. The "Algorithms" drop-down in the navigation bar allows the user to choose to execute the breadth-first search algorithm or the depth-first search algorithm, allowing the user to easily interact with the visualisation data. Users can also freely select the starting vertex and observe the difference between the two algorithms for traversing paths on different graphs and starting points. At the same time, the two plug-ins provide two different undirected graphs as templates for each algorithm, so that users can execute the algorithms on the templates and view the results.

In terms of algorithms, the plug-ins use the breadth-first algorithm and the depth-first search algorithm optimised for the GrALoG software; the breadth-first algorithm ensures efficient and systematic traversal of the graph, exploring vertices hierarchically from the starting vertex outwards. Meanwhile, the depth-first search algorithm uses a depth-first exploration approach, traversing along each branch as far as possible before backtracking. The plug-ins also employ appropriate data structures, such as queues for the breadth-first algorithm and stacks for the depth-first search algorithm, to manage the order of vertices to be explored.

## System Requirements and Installation

You can download Gralog from https://github.com/gralog/gralog. Two plugins for visualising the results of traversing the depth-first search algorithm and the breadth-first search algorithms can be downloaded from https://github.com/feliciawq/gralog-BFS-DFS.

First, make sure that Java version at least 17 is installed on your machine.

On Linux and MacOS you can install Gralog by

```
cd install

./install-nix.sh
```

Call

```
./install-nix.sh -h
```

to get help on the installer options. If the installation script didn’t work for some reason, read below how to install Gralog manually.

Install the plug-ins:

1. Place the packages for the two plug-ins ("gralog-Depthfirst-Search" and "gralog-Breadthfirst-Search") in the same folder as "gralog-core".

2. Replace the two configuration files in the gralog folder: build.gradle and settings.gradle

3. Replace the config.xml file under the path gralog-fx/src/main/java/gralog/gralogfx

4. Open the Gralog package in a popular compiler such as eclipse or IntelliJ IDEA.

### Manual installation

On Windows or if install-nix.sh didn’t work, it should be not difficult to install Gralog manually. The binaries can be found in build/dist. If you want to compile Gralog yourself, please, download the full version with sources (gralog-all.zip, not just (gralog-bin.zip) and run

```
./gradlew
```

on a *nix system and

```
./gradlew.bat
```

on Windows.

### Start Up

In the terminal:

1. Enter the command

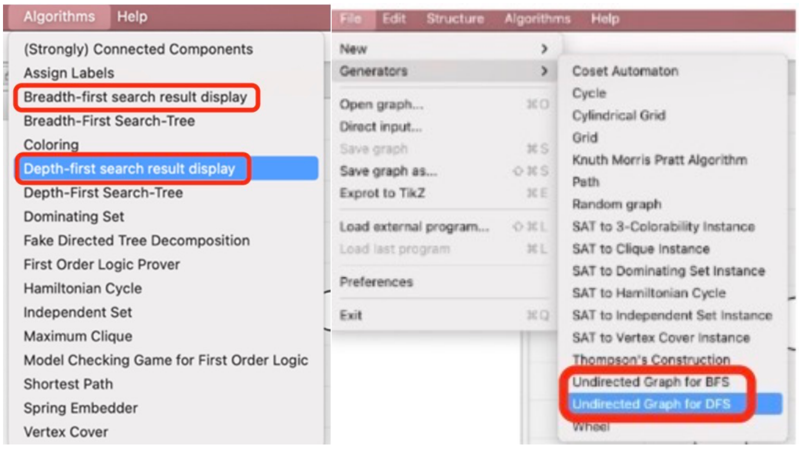
```
cd build/dist
```

and go to the dist folder.

2, Execute

```
java − −add − opens java.base/java.util
                 = ALL − UNNAMED  − −add − opens java.base/sun.net.util
                 = ALL − UNNAMED  − jar gralog − fx.jar
```

If the options "Depth-first search results display" and "Breadth-first search results display" are displayed in the "Algorithms" drop-down menu, and the options "Unidirectional map for DFS" and "Unidirectional map for BFS" are displayed under "Generator" in the "File" menu, the installation of both plug-ins is confirmed to be successful.



## Usage and Functionality

1. Draw a vertex

Left click on the screen to draw a vertex.

2. Draw an edge

Select a vertex on the screen and right-click to another point to draw an edge between the two vertices. The edges drawn on the default graph canvas and the directed graph canvas are directed edges by default. Edges drawn on an undirected graph canvas are undirected by default.

3. Select a template

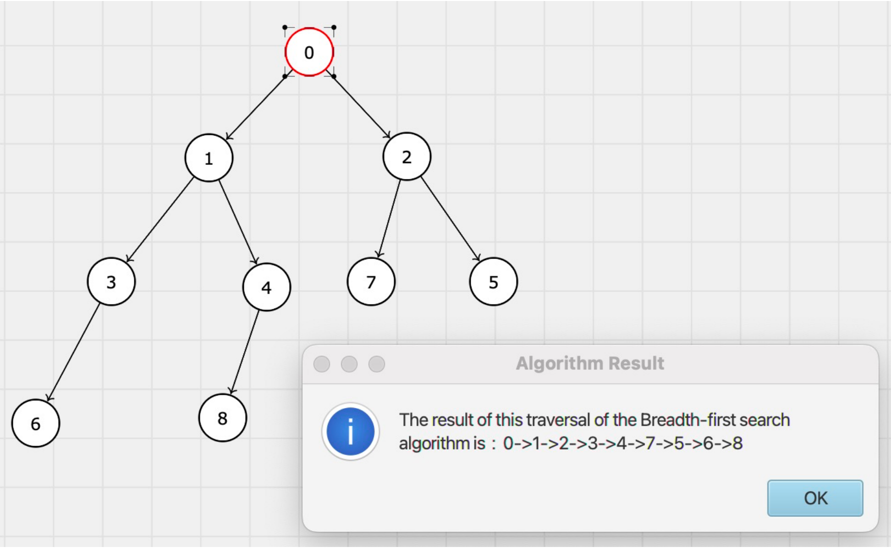
If you want to view the traversal paths and results of the two algorithms through the templates, you can select "Undirected Graph for DFS" or "Undirected Graph for BFS" from the "Gerenator" in the "File" in the navigation bar to view the template graphs for the depth-first search algorithm and the breadth-first search algorithm.

4. Execute the algorithm

After selecting a vertex, simply click on "Depth-first search result display" or "Breadth-first search result display" in the navigation bar under "Algorithms" to execute the depth-first search algorithm or the breadth-first search algorithm.

5. View of results

After executing the algorithm, a JavaFx pop-up box will appear on the current page (as shown below). You can view the traversal path and results.



The result of this traversal of the Breadth-first search algorithm is： 0->1->2->3->4->7->5->6->8