5/13/2023

Dijkstra Algorithm

[Visualization]

Meherab Hessen [20212005010]
Monowar Hossain [20212017010]
Md. Saif Mahmud Khan [21202055010]

Table of Contents

Dijkstra's Algorithm visualization user manual

1. Introduction

Dijkstra's algorithm is a pathfinding algorithm used to find the shortest path between two nodes in a weighted graph. It was developed by computer scientist Edsger W. Dijkstra in 1956. The algorithm uses a priority queue to keep track of the nodes to be visited and a distance array to keep track of the shortest distance from the starting node to each node in the graph.

To visualize Dijkstra's algorithm, we typically use a graphical representation of a weighted graph, where each node represents a location and the edges represent the connections between them. The weight on each edge represents the distance or cost to travel between the two connected nodes.

To start the visualization, we select a starting node and mark its distance as zero. Then, we add the starting node to the priority queue and begin to explore its neighboring nodes. For each neighbor, we calculate the distance from the starting node to that node by adding the cost of the edge connecting them to the starting node. If this distance is shorter than the current shortest distance for that node, we update the distance and add the node to the priority queue.

We continue to explore the neighbors of the nodes in the priority queue until we have visited all nodes or reached our destination node. As we visit each node, we update the visualization to show the shortest path found so far and the nodes that have been visited.

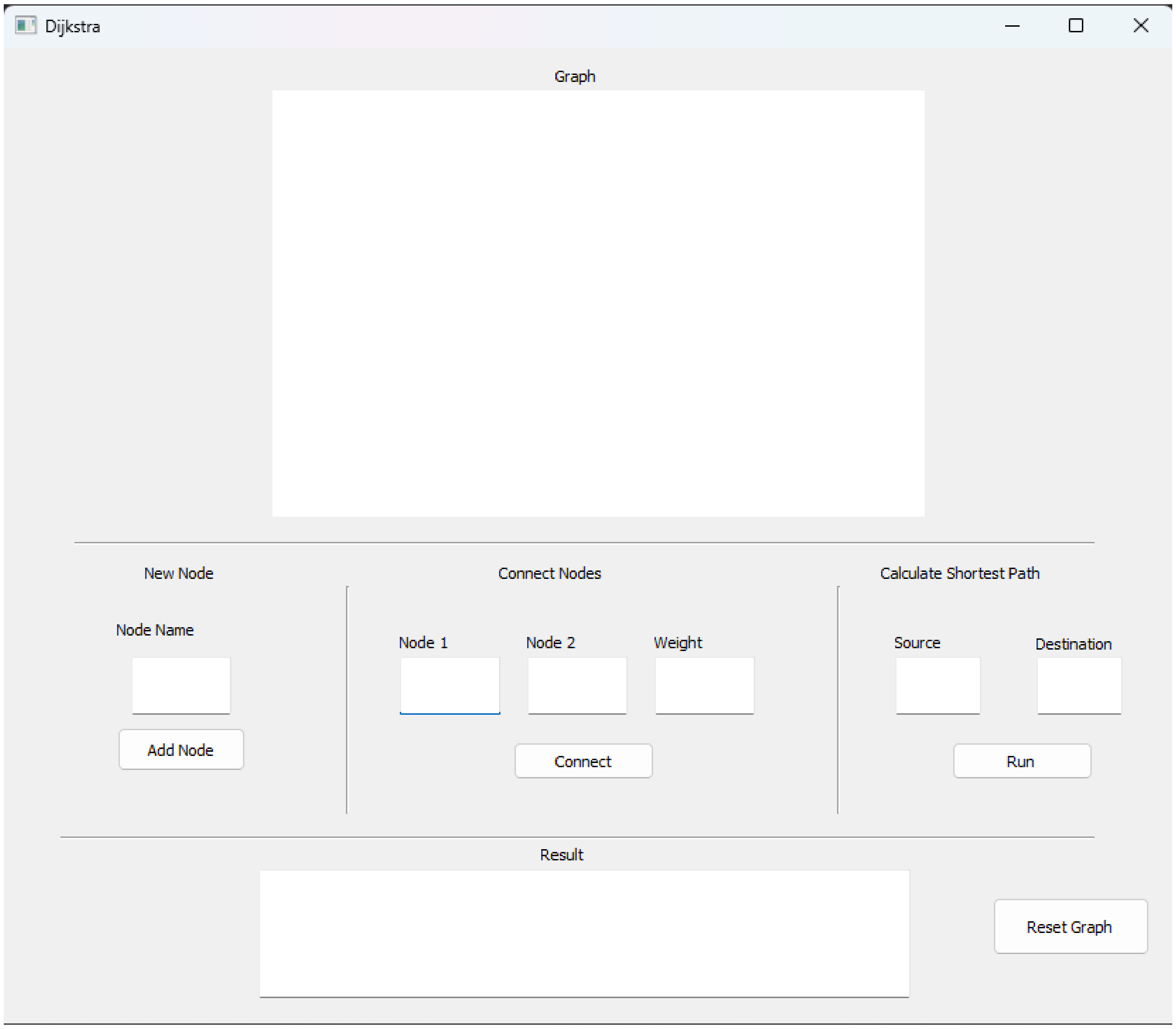
The result of the visualization is a clear representation of the shortest path between the starting node and the destination node, with the added benefit of visualizing the process that led to the discovery of that path.

2. Objectives

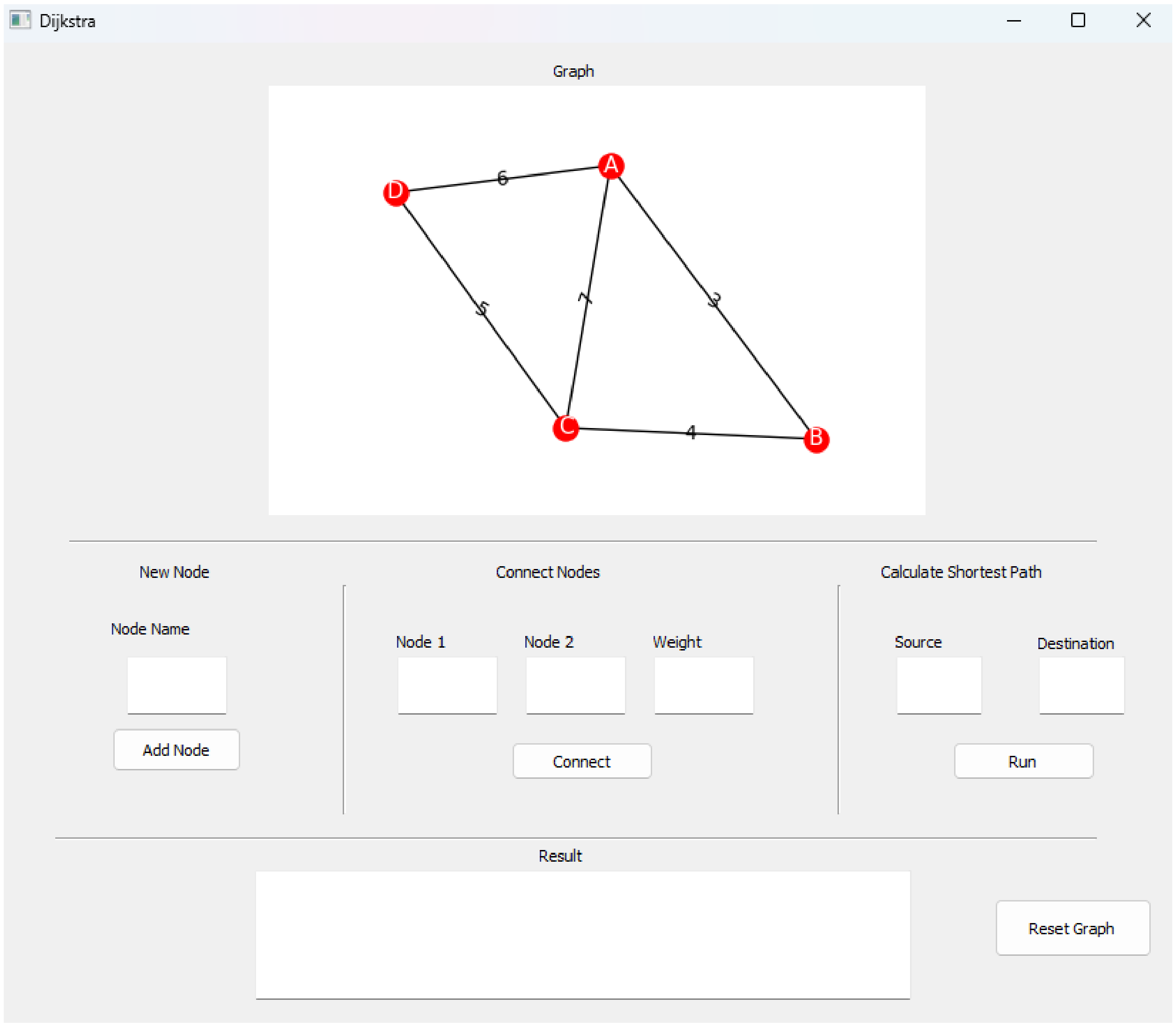
- Optimality: The algorithm guarantees that it will always find the shortest path between the starting node and the destination node.
- Completeness: The algorithm will find a solution if one exists, or report that there is no path if none exists.
- Accuracy: The algorithm produces accurate results because it takes into account the weights of the edges connecting the nodes in the graph.

3. Description

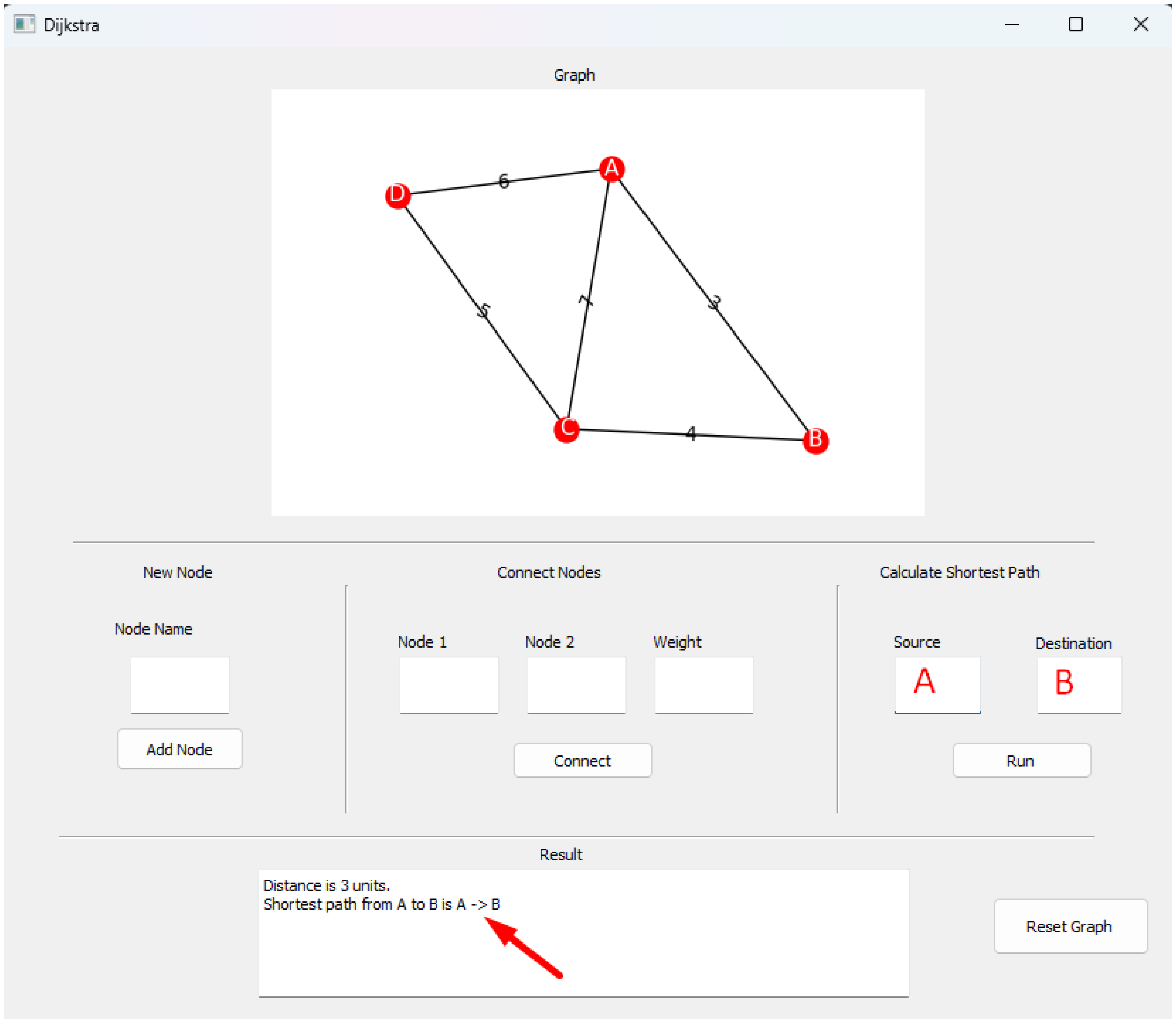
- This is the **Strat Page** of Dijkstra Algorithm



- We have to add some node with Edge's



- Then we have to add Source and Destination to calculate shortest path



Source: A

Destination: B

```
Distance is 3 units.
Shortest path from A to B is A -> B
```

Here we can see the shortest path between A and B.

4. Dependencies:

1. Install the virtualenv module.
```
pip install virtualenv
```
2. create a virtual environment.
```
virtualenv dijkstra_ws
```
3. activate the virtual environment.
```
cd dijkstra_ws
source bin/activate
```
4. Clone the repo and install dependencies.
```
pip install -r requirements.txt
```

After successfully installed, simply run: `python main.py`

Special Thanks to:
Md. Shymon Islam
Lecturer
Department Of CSE
North Western University
Khulna, Bangladesh

Developed By:
Md. Saif Mahmud Khan
Student ID: 20212055010

Monowar Hossain
Student ID: 20212017010

Meherab Hossen
Student ID: 20212005010