



Searching Algorithms Visualizer

Submitted by

| | Name | ID |
|----------|-------------------------------|-----------|
| 1 | Salma Ehab Mohamed | 18P6253 |
| 2 | Salma Osama Mohamed | 18P5097 |
| 3 | Sama Ayman Mokhtar Amin | 18p7873 |
| 4 | Yomna Yousry Muhammed Ibrahim | 18P7800 |

Home View

Search Methods Visualizer

A **Graph Visualizer**

B Enter the initial node

E Enter the goal node(s)

D Enter the nodes

F Enter the depth limit

C Breadth first

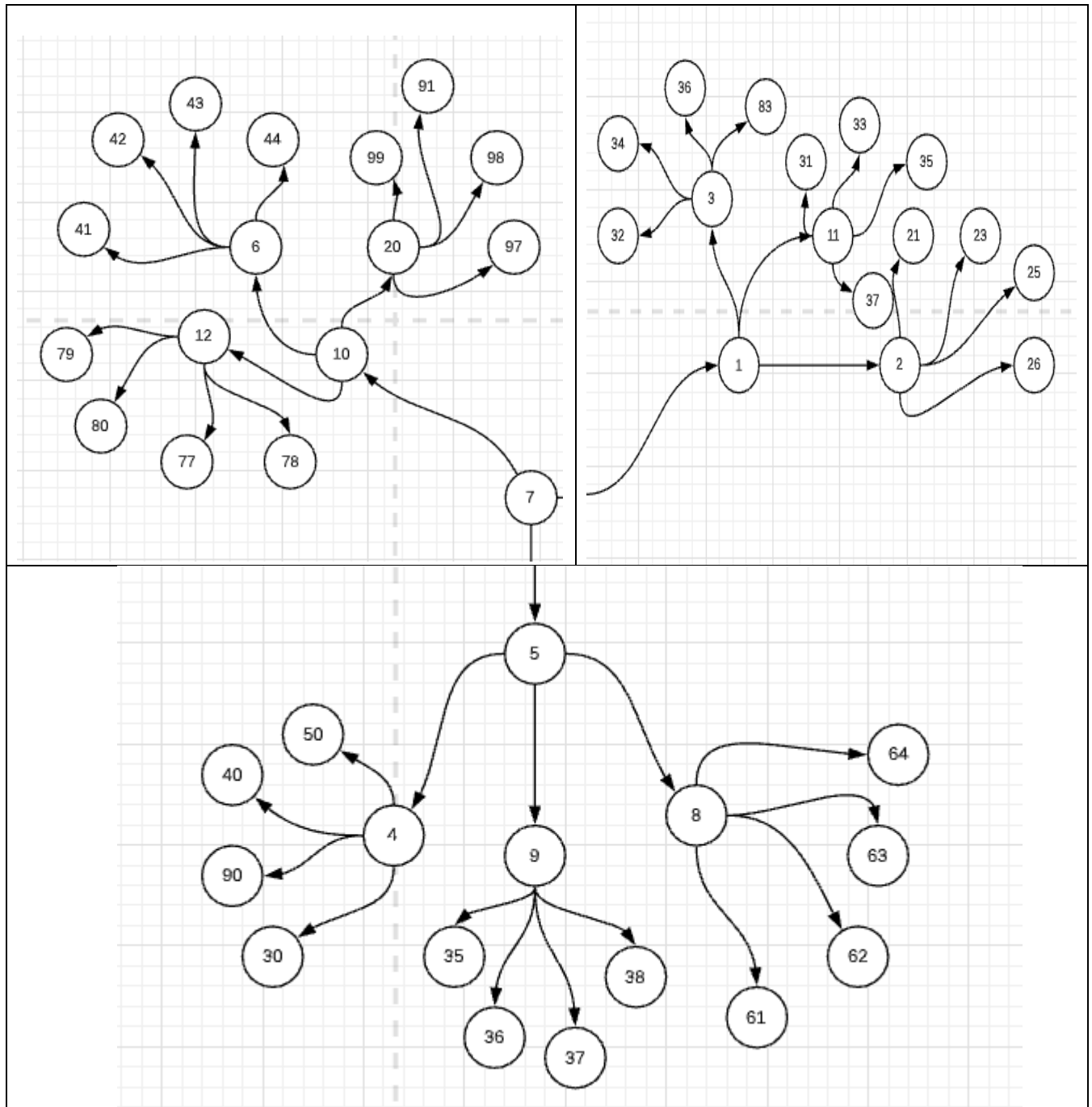
G Run

H Written Results

| Label | Brief Description |
|-------|---|
| A | The canvas where the graph is displayed. |
| B | The field where the user enters the initial node. |
| C | A combo box that allows the user to choose one of these searching algorithms: breadth first, depth first, uniform cost, depth limited, iterative deepening, greedy, and A*. |
| D | The user enters the graph either as adjacency list in case of breadth first, depth first, depth limited, iterative deepening or as (from, to, cost) structure in case of uniform cost, greedy, or A*. |
| E | The field where the user enters the goal node(s). |
| F | The field where the user enters the depth limit in case of iterative deepening. |
| G | The user clicks run to get the program started. |
| H | The result gets printed here. |

Non-weighted Graph Representation

In case of breadth first, depth first, limited depth, and iterative deepening an unweighted graph is used. If we want to enter a graph as shown below, the user writes it as an adjacency list. For example, node 7 has node 1,5, and 10 adjacent to it, so we write it as 7 1 5 10. The same applies for each node that have at least one adjacent node.

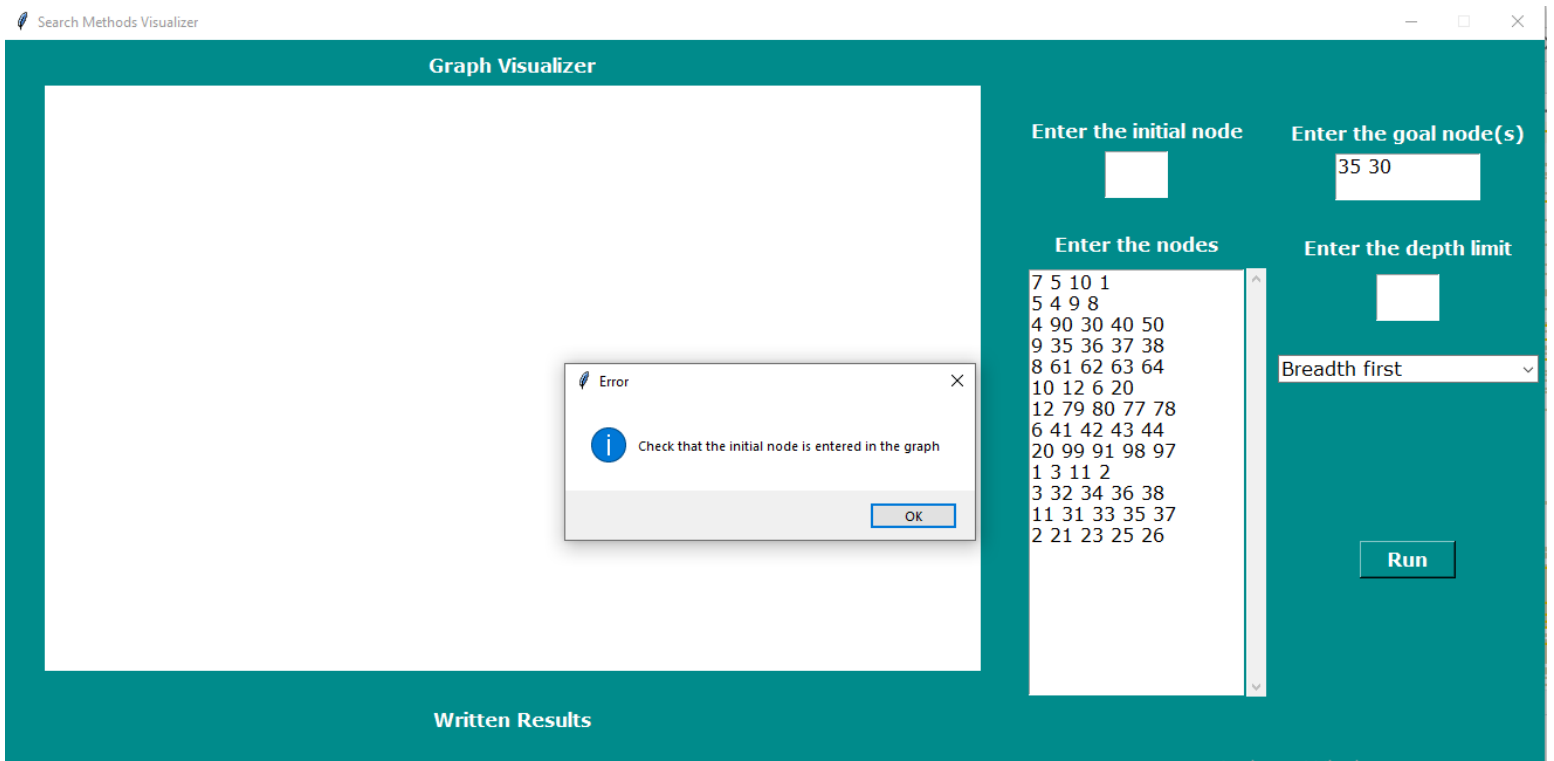


So the above graph's adjacency list is:

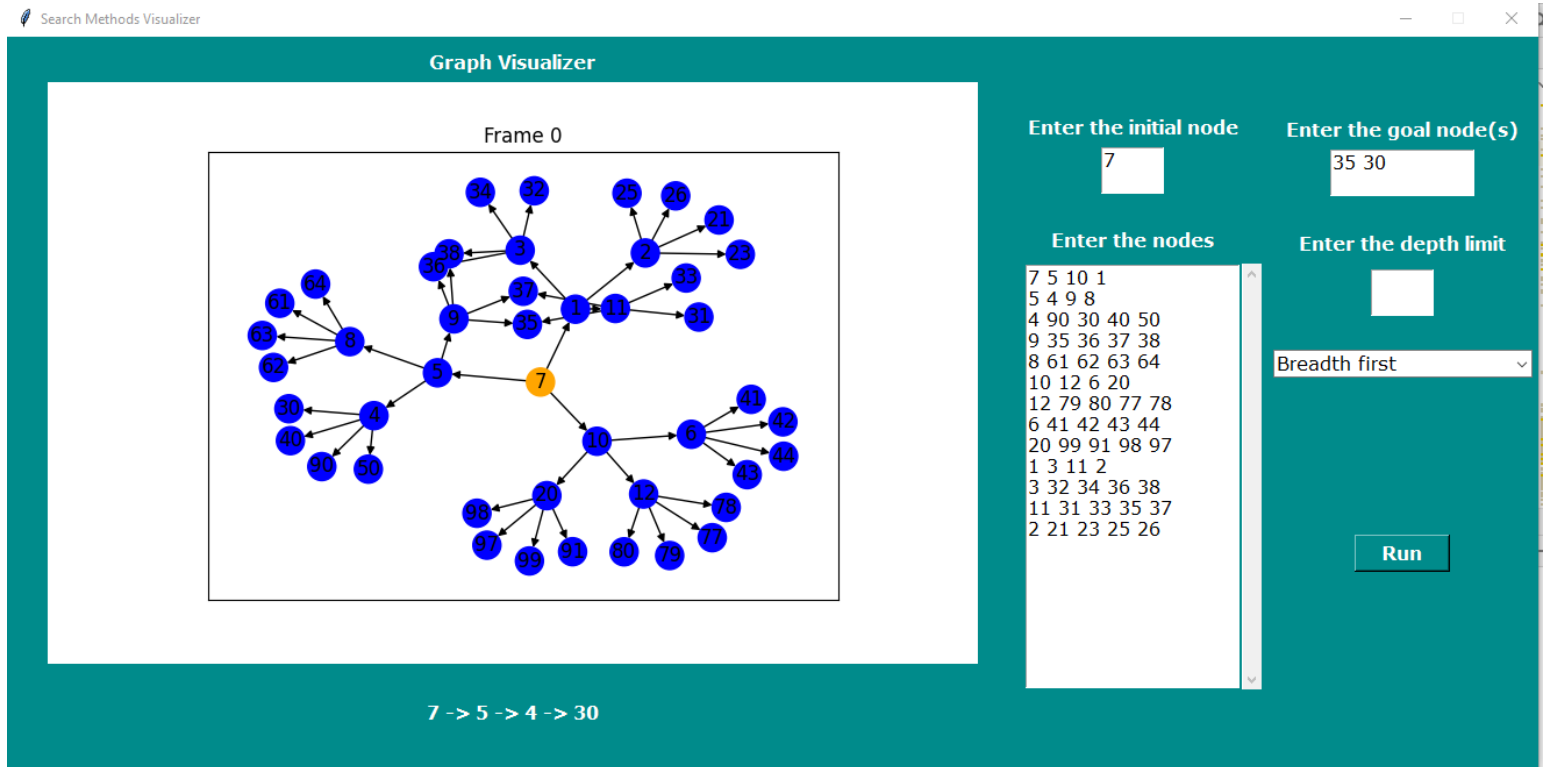
```
7 5 10 1
5 4 9 8
4 90 30 40 50
9 35 36 37 38
8 61 62 63 64
10 12 6 20
12 79 80 77 78
6 41 42 43 44
20 99 91 98 97
1 3 11 2
3 32 34 36 38
11 31 33 35 37
2 21 23 25 26
```

Breadth First Search

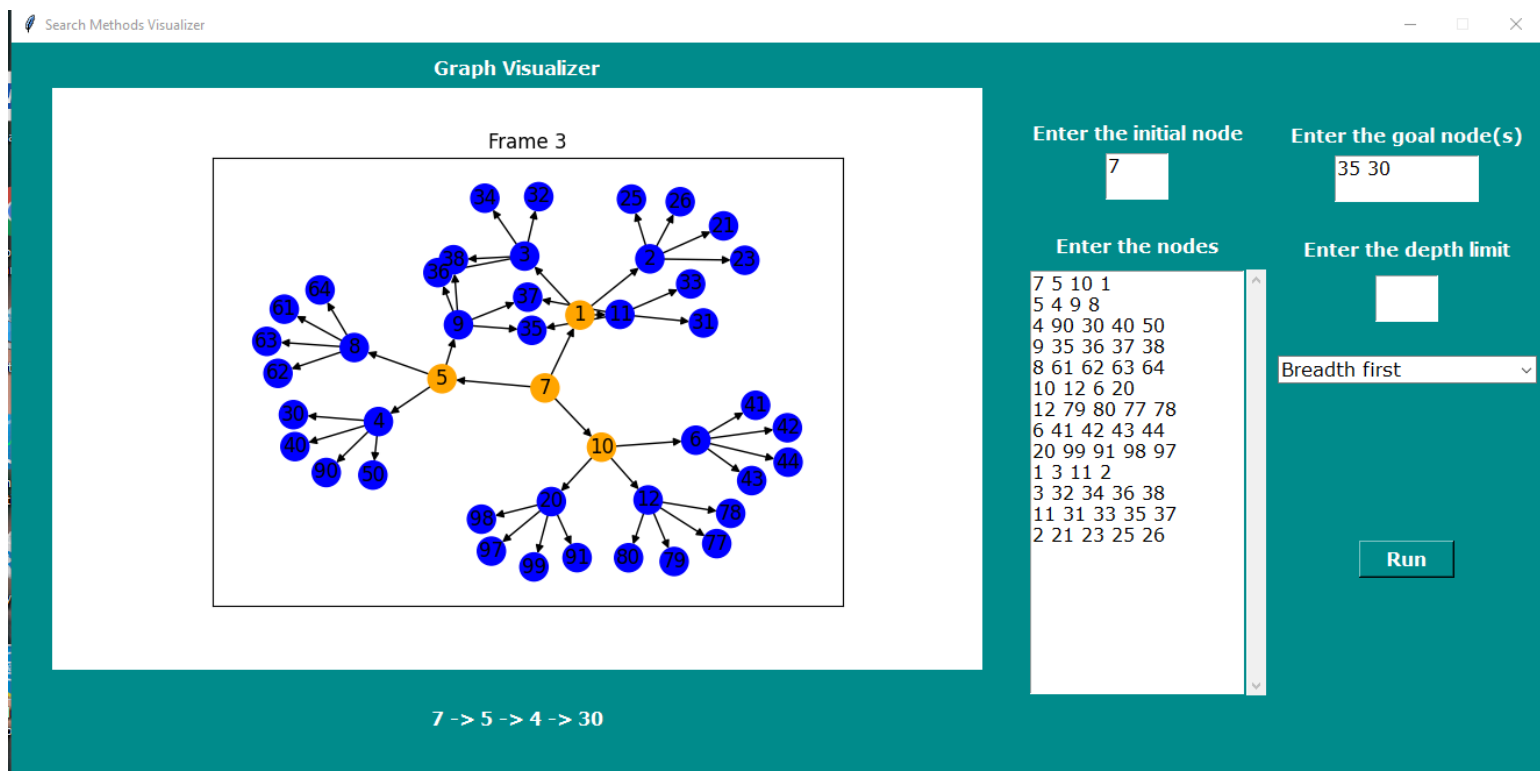
To try breadth first search the user entered a list of goal nodes, the graphs adjacency list and pressed run. However, since no initial node is specified, the program throws an error message to the user. This behavior applies to all of the searching algorithms.



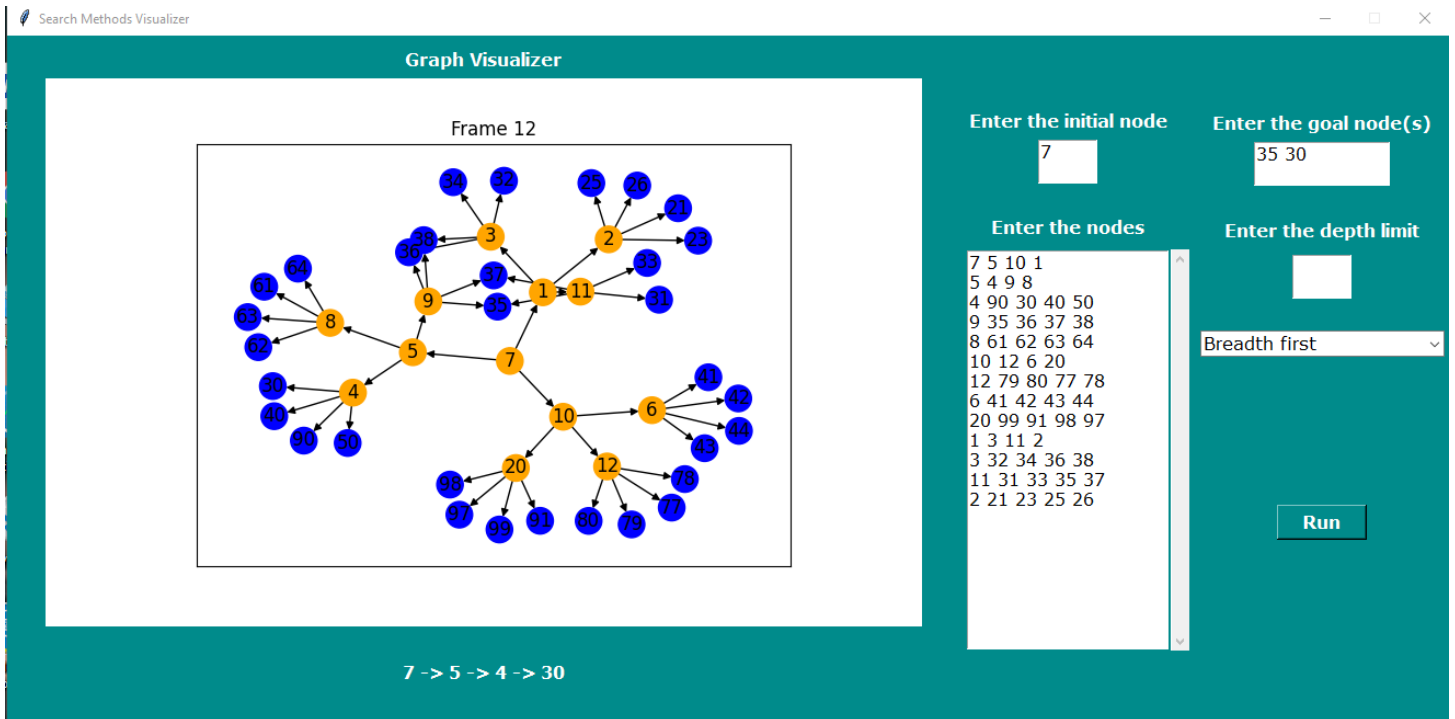
After the user enters the initial node and presses run. The graph is drawn. Lists added to fringe get colored in orange one at a time.



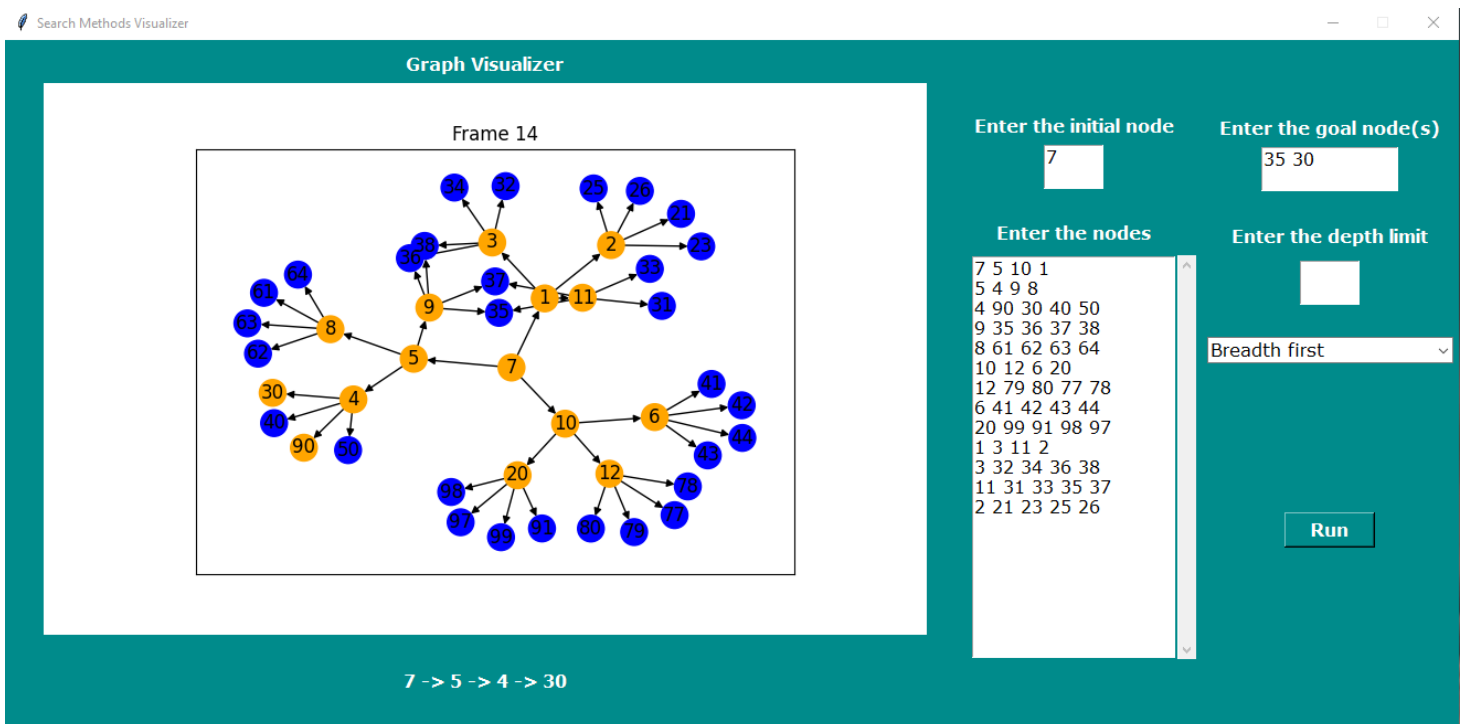
Two frames are skipped and below is a instance of the graph after another level of nodes is added to the fringe.



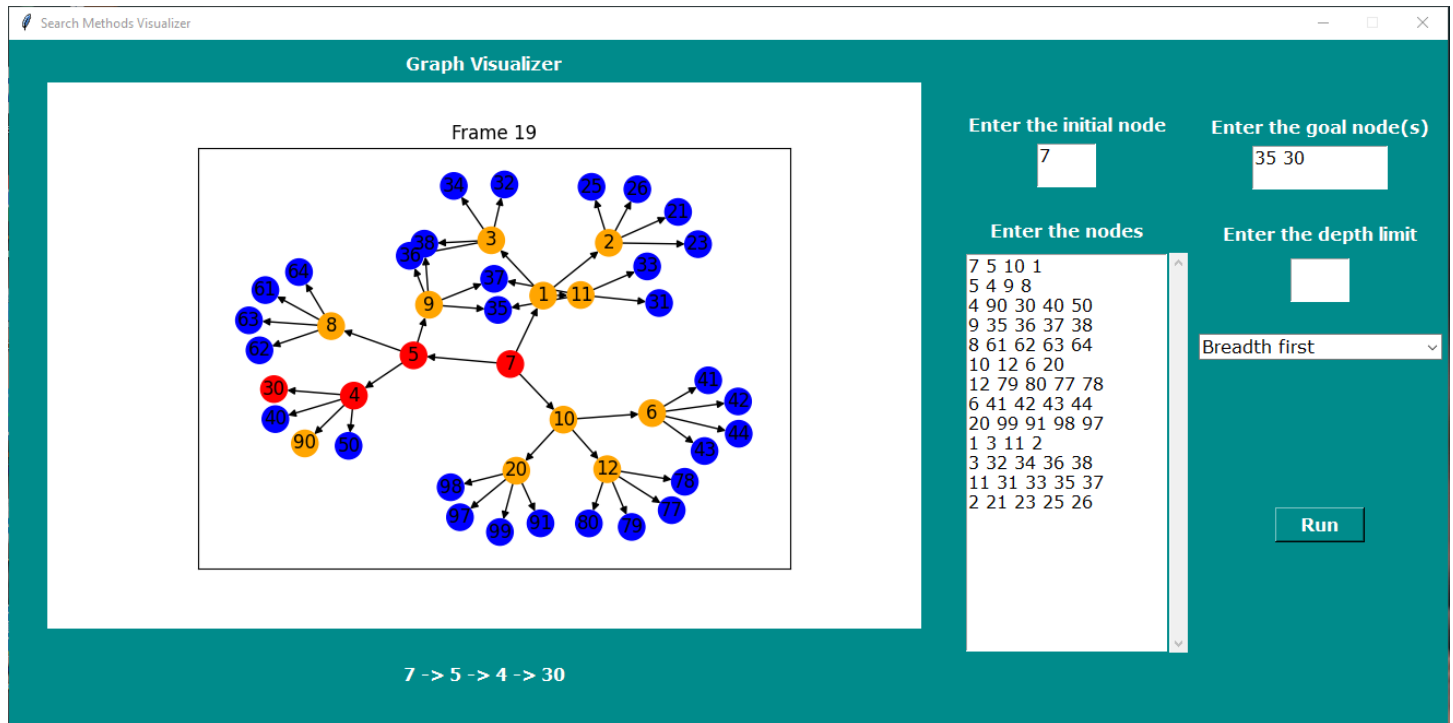
Few more frames are skipped and below is a instance of the graph after an extra level of nodes is added to the fringe.



Nodes keeps added to the fringe till the program reaches the first goal node: 30.

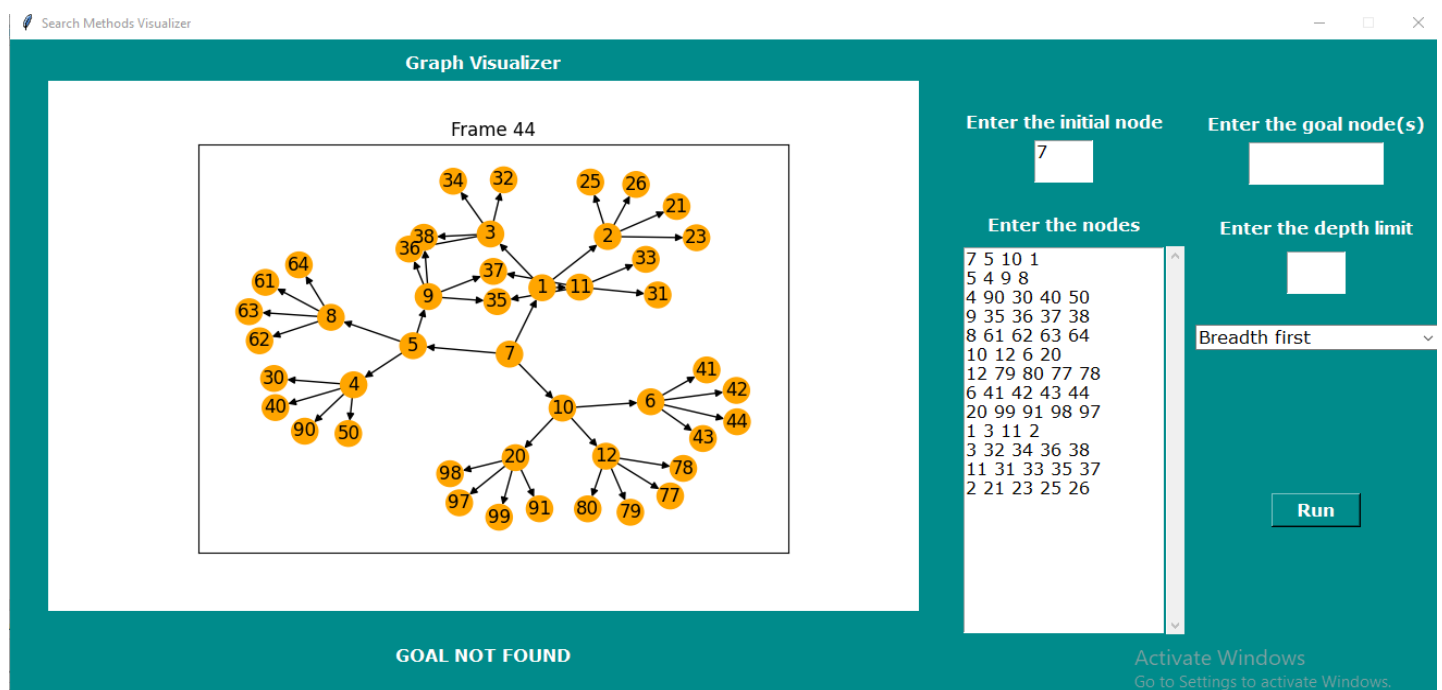


The visualizer colors the shortest resulting path in red. Besides, the path is printed in the bottom of the screen.



Exception(goal eliminated):

In the algorithms that deal with unweighted graphs, if the user does not enter a goal node the algorithms will search till the end of the graph and prints that the goal is not found.



Depth Limited

In depth limited search, the depth limit must be entered by the user or otherwise a text box will be displayed to ask the user to enter the limit.

Search Methods Visualizer

Graph Visualizer

Enter the initial node: 5

Enter the goal node(s): 63

Enter the nodes:

```
7 5 10 1
5 4 9 8
4 90 30 40 50
9 35 36 37 38
8 61 62 63 64
10 12 6 20
12 79 80 77 78
6 41 42 43 44
20 99 91 98 97
1 3 11 2
3 32 34 36 38
11 31 33 35 37
2 21 23 25 26
```

Enter the depth limit:

Depth limited

Run

Written Results

Activate Windows
Go to Settings to activate Windows.

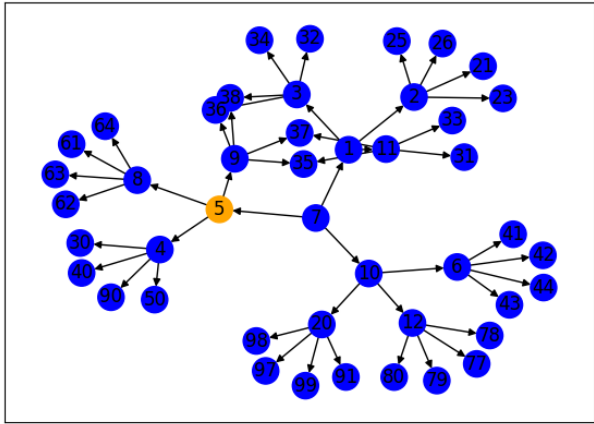
Error: Enter the depth limit

The search starts from the initial node that was entered by the user.

Search Methods Visualizer

Graph Visualizer

Frame 0



5 -> 8 -> 63

Enter the initial node: 5

Enter the goal node(s): 63

Enter the nodes:

```
7 5 10 1
5 4 9 8
4 90 30 40 50
9 35 36 37 38
8 61 62 63 64
10 12 6 20
12 79 80 77 78
6 41 42 43 44
20 99 91 98 97
1 3 11 2
3 32 34 36 38
11 31 33 35 37
2 21 23 25 26
```

Enter the depth limit:

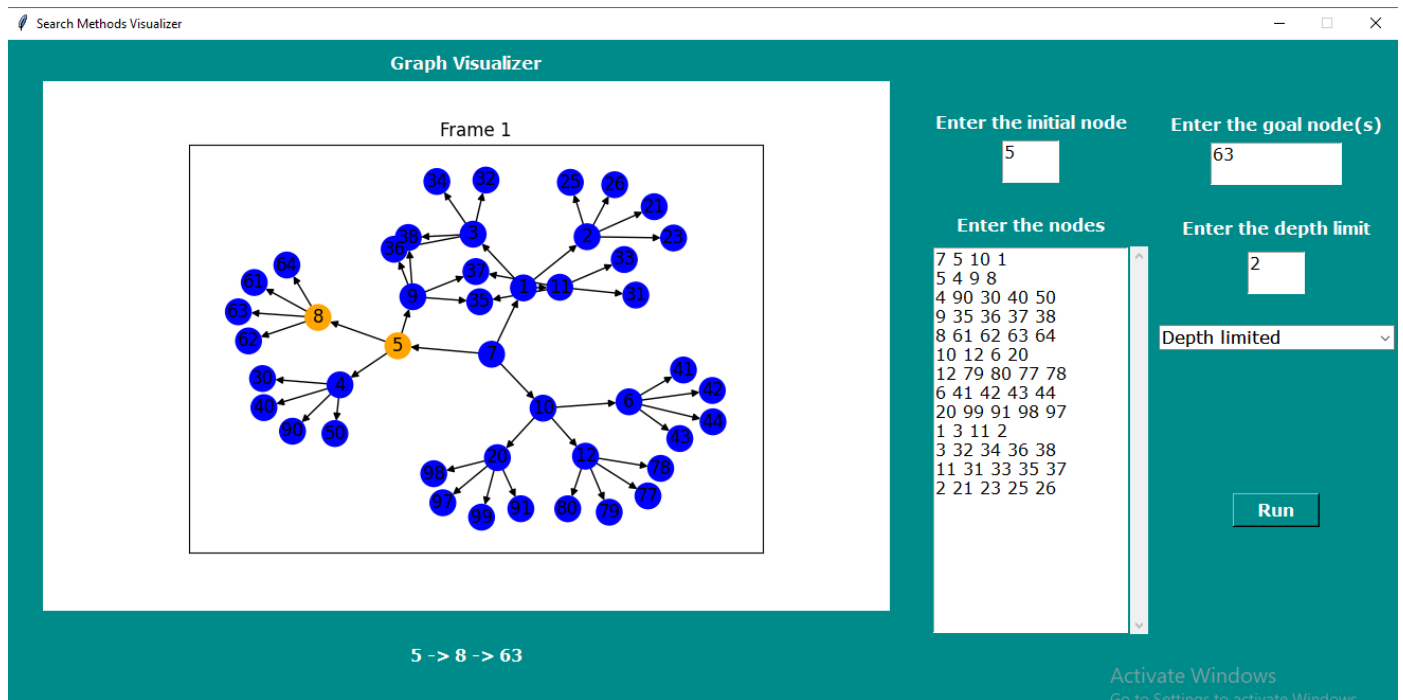
2

Depth limited

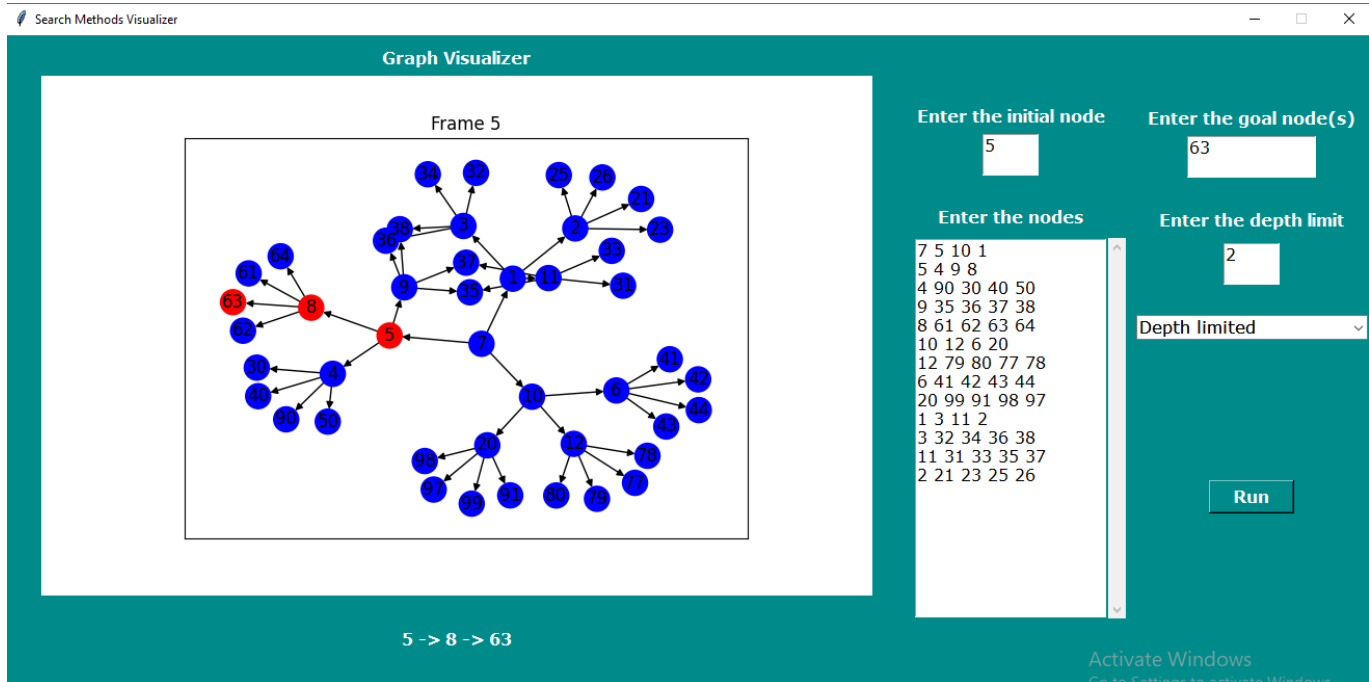
Run

Activate Windows
Go to Settings to activate Windows.

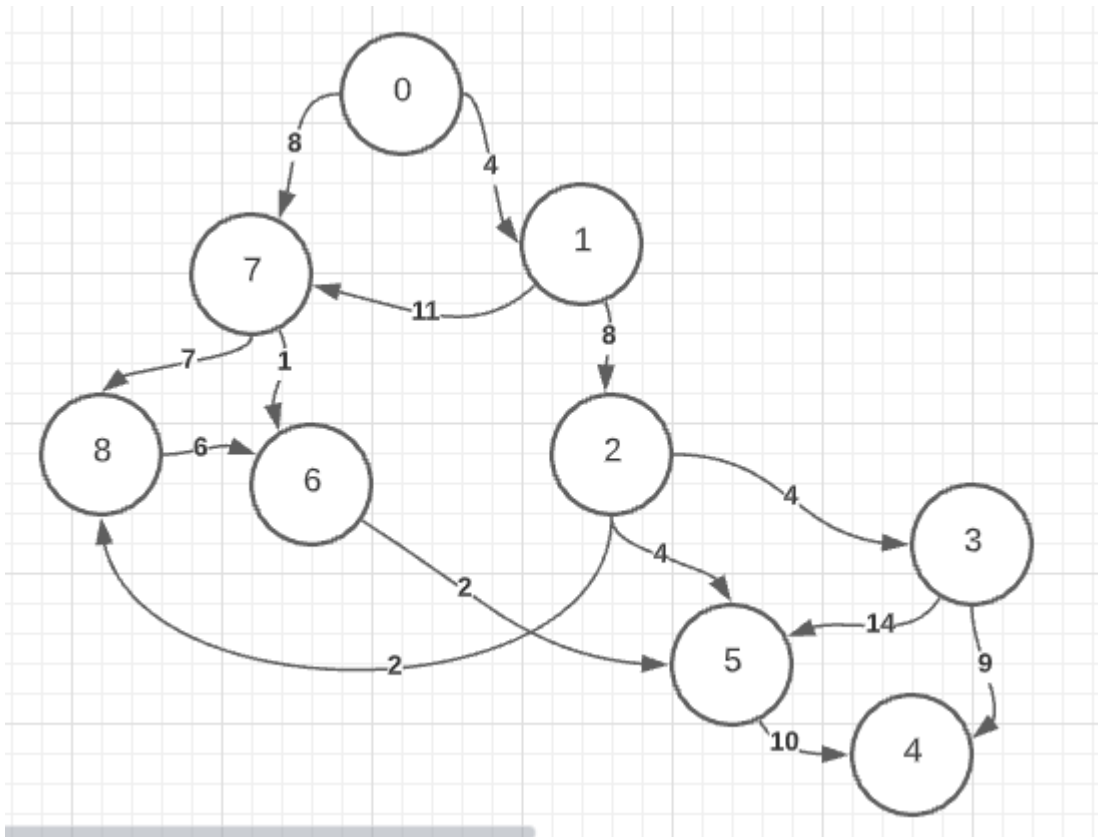
The depth limited algorithm will search in the first depth level since it has not yet reached its limit.



Since the goal was found in depth 2, the visualizer colors the shortest resulting path in red. Besides, the path is printed in the bottom of the screen.



Weighted Graph Representation

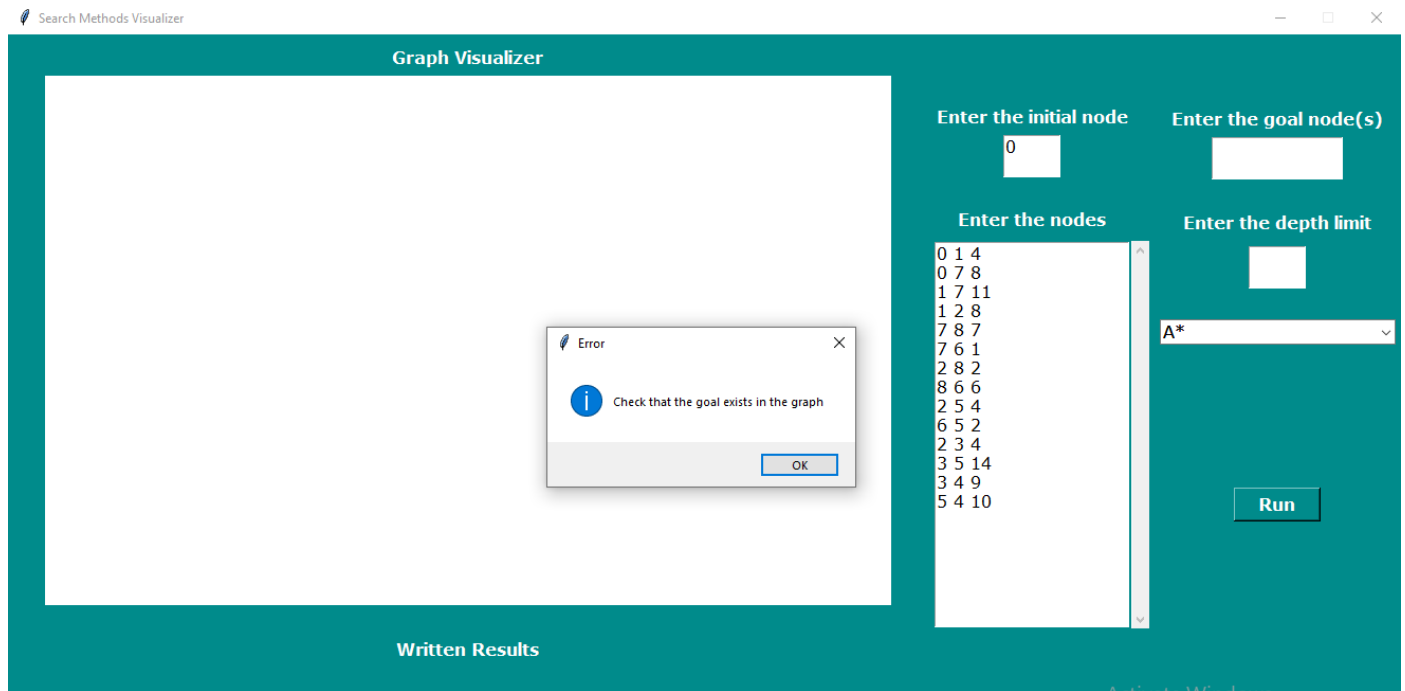


The weighted graph above is represented as follows:

```
0 1 4
0 7 8
1 7 11
1 2 8
7 8 7
7 6 1
2 8 2
8 6 6
2 5 4
6 5 2
2 3 4
3 5 14
3 4 9
5 4 10
```

Each row in the list represents the connection between two nodes where the first element is the first node, the second element is the second node, and the third element is the edge's weight.

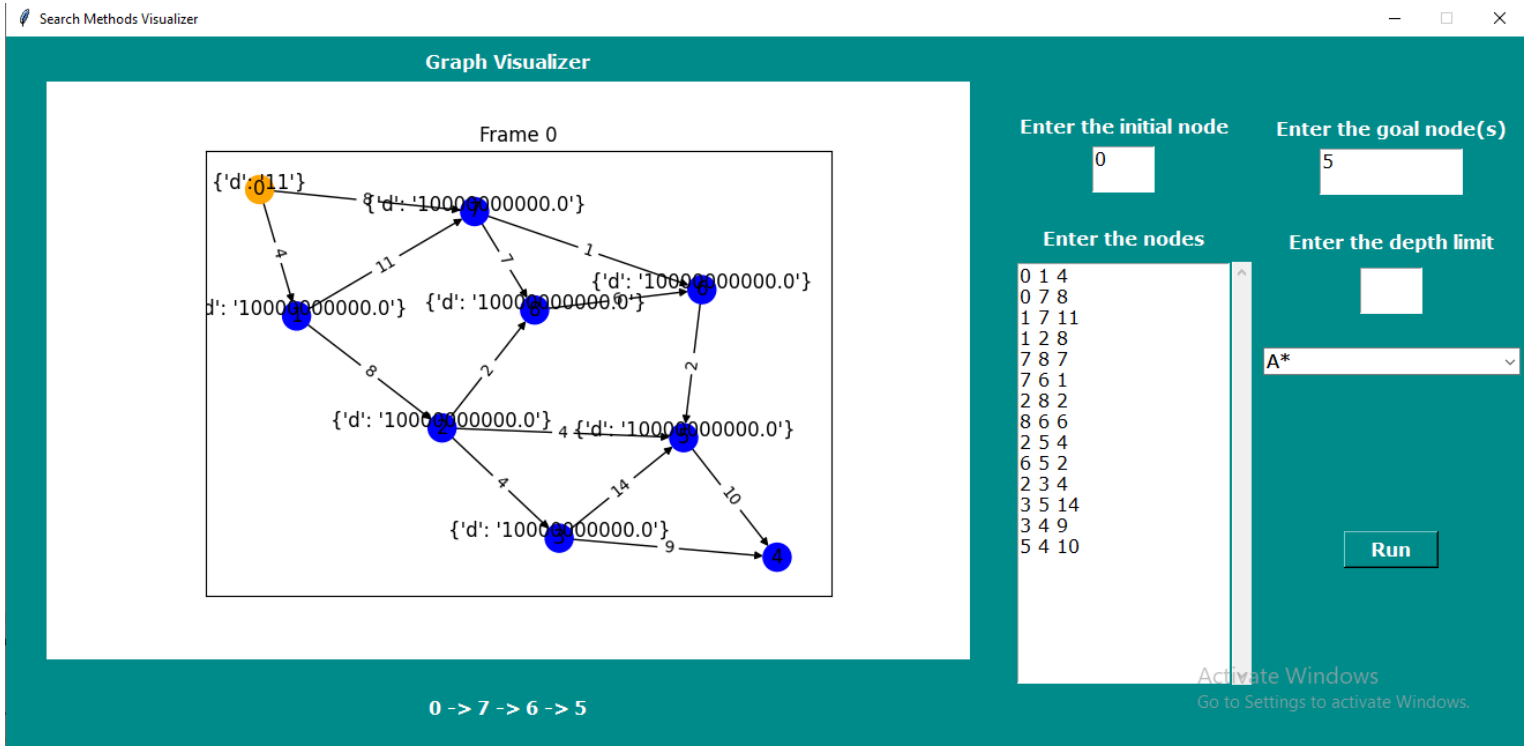
In every algorithm that deals with a weighted graph, if the goal is not entered a message will be displayed to ask the user to enter at least one goal that exists in the graph.



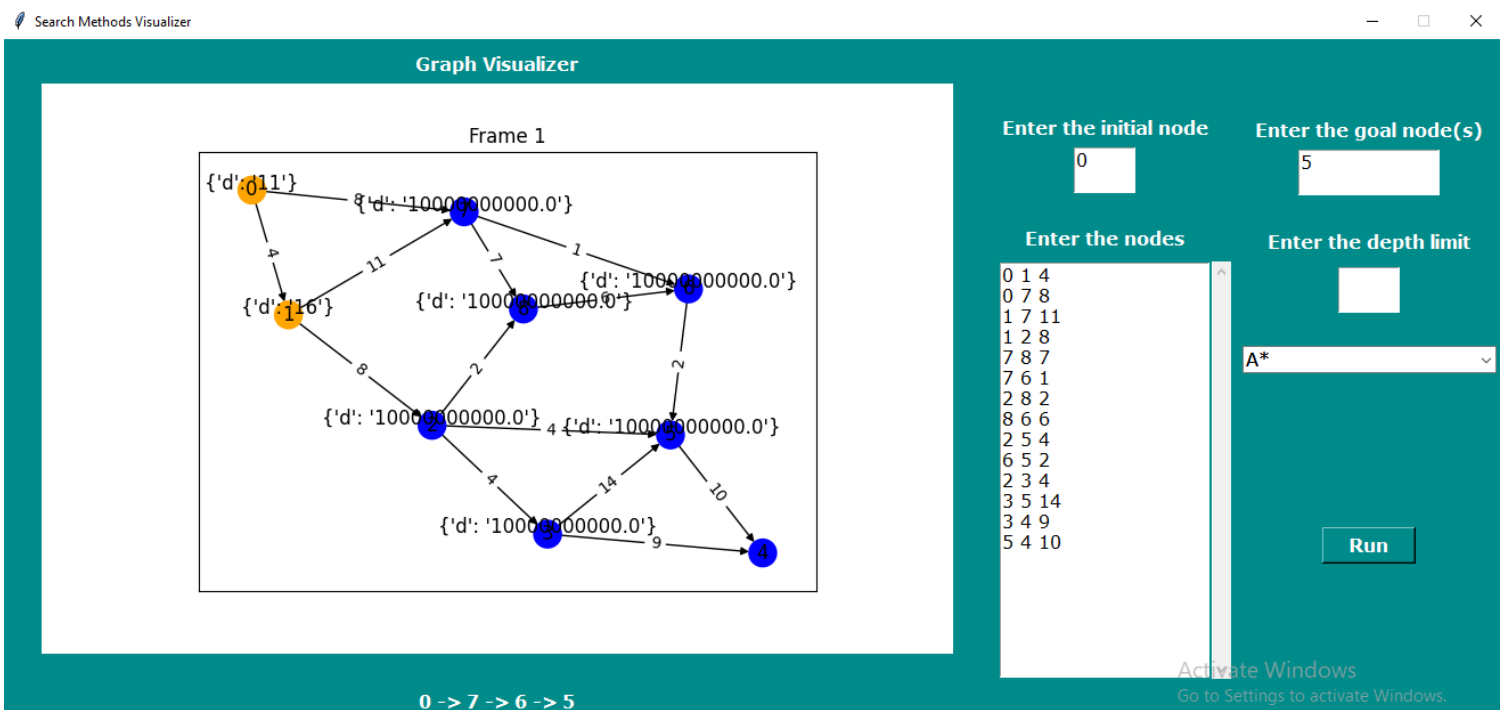
A* Informed Search

The algorithm starts from the initial node that was entered by the user. All nodes are initialized to $10e9$ where the values are later modified as their cost gets calculated. The cost of the node will be equal its heuristic added to its distance.

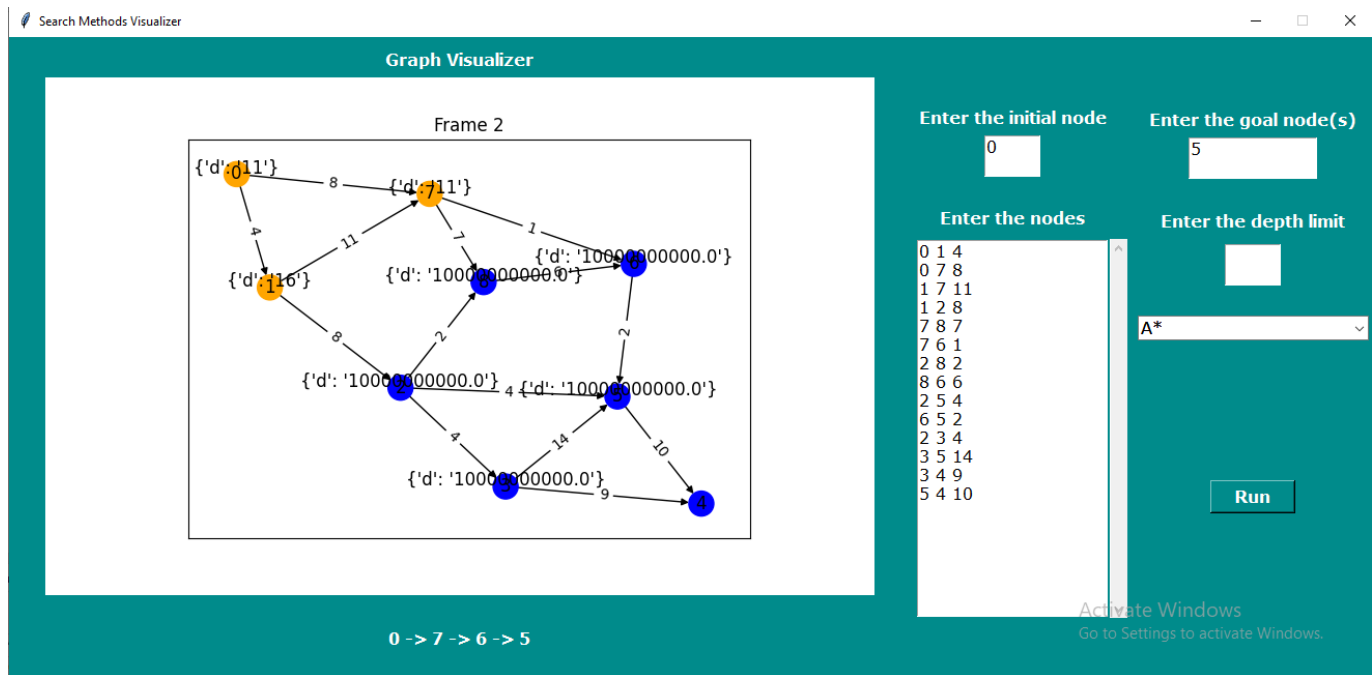
The heuristic function is calculated using the uniform cost algorithm to get the shortest distance between the goal and the nodes.



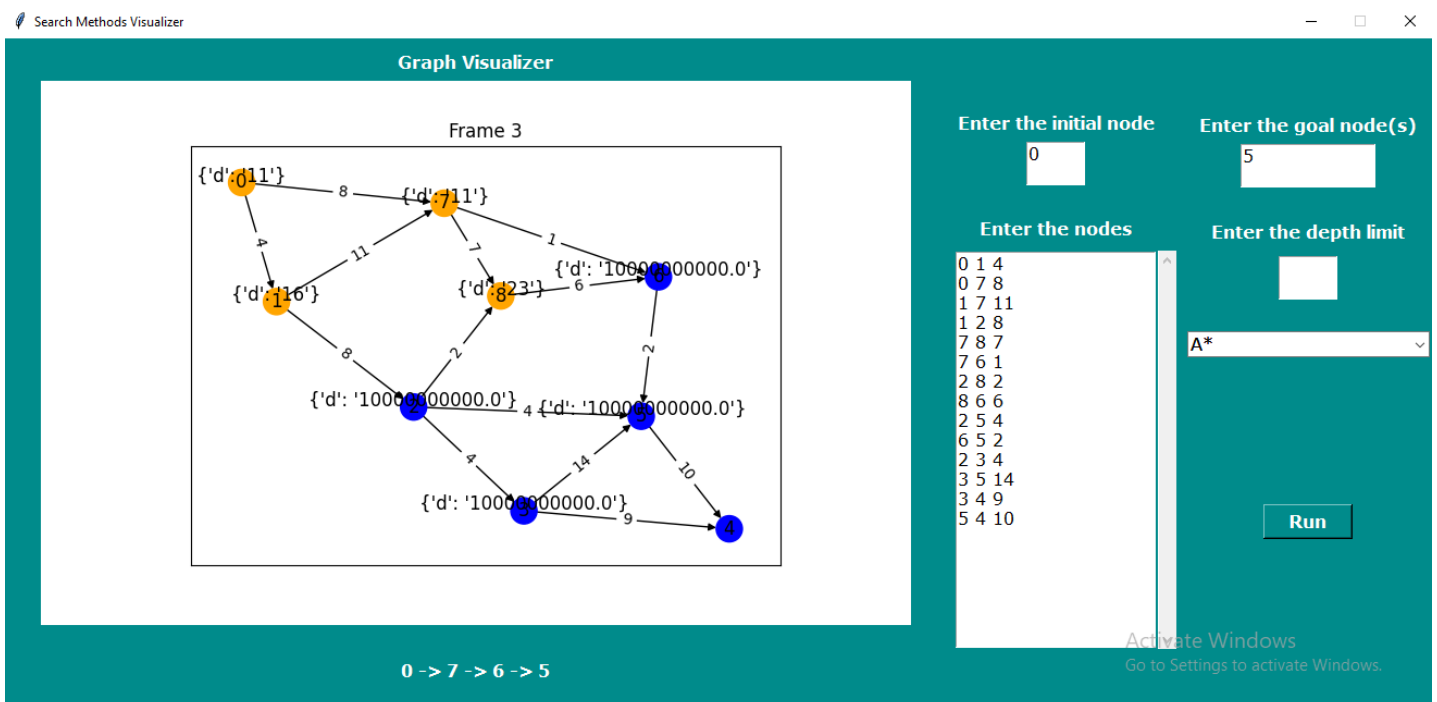
Node 1 is expanded.



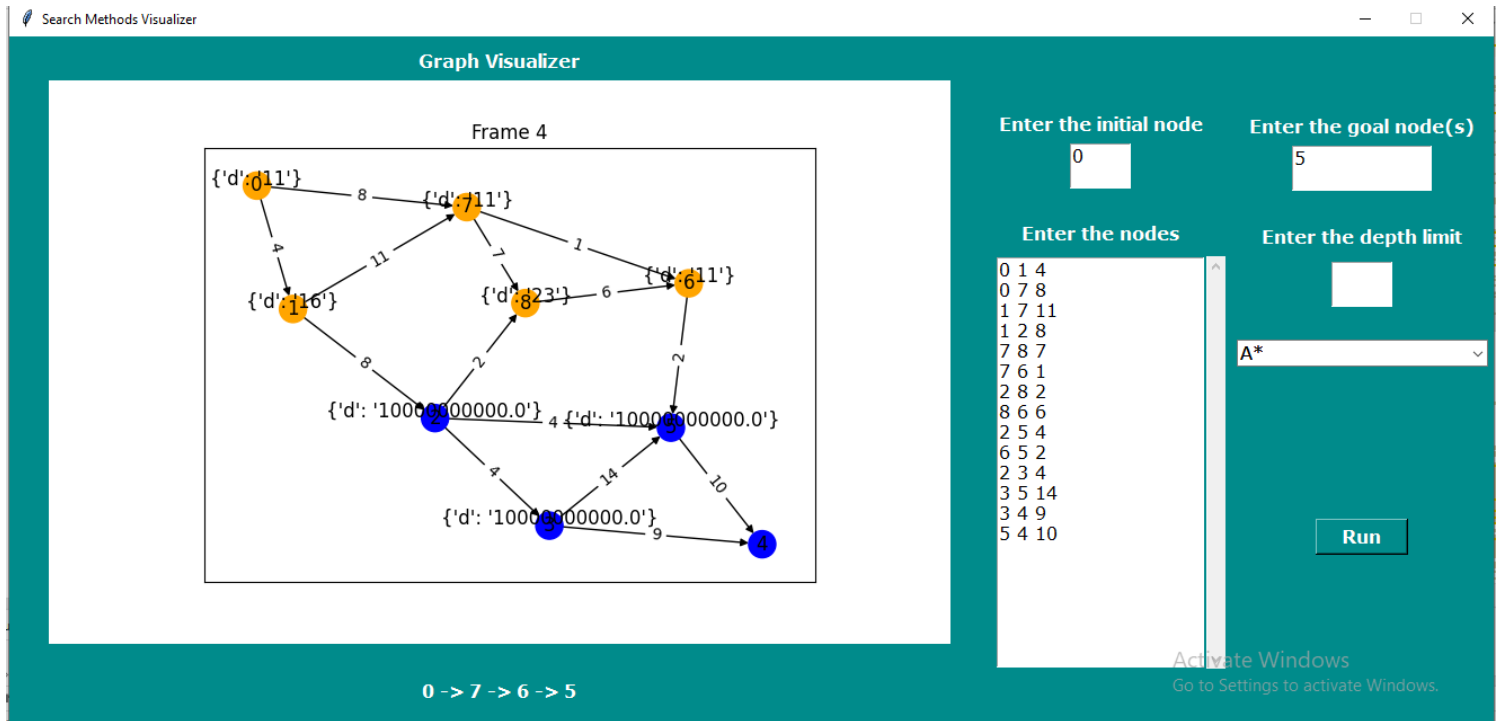
Node 7 is expanded and then visited.



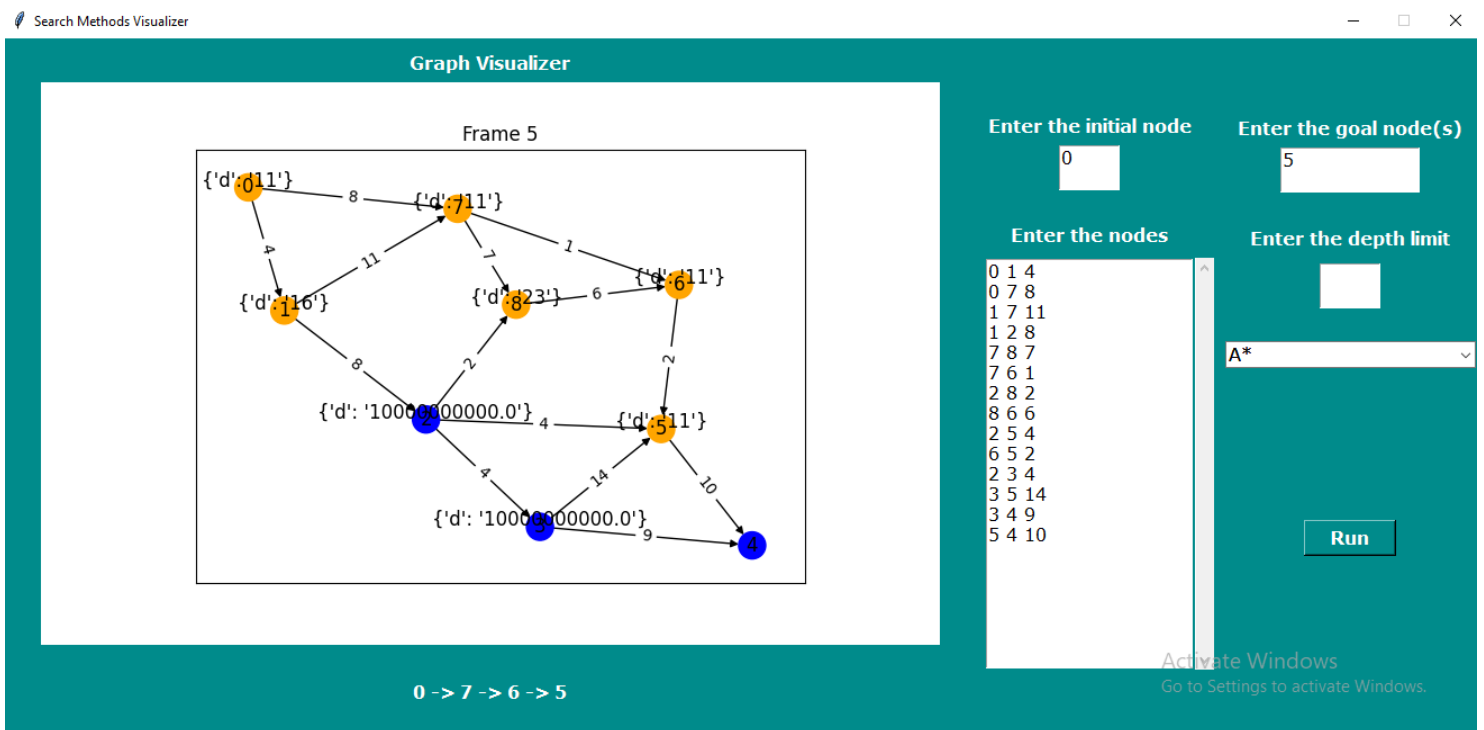
Node 8 is expanded.



Node 6 is expanded and then visited.



Node 5 is expanded.



Node 5 is visited. Since node 5 is the goal, the path will be colored in red and printed on the bottom of the screen.

