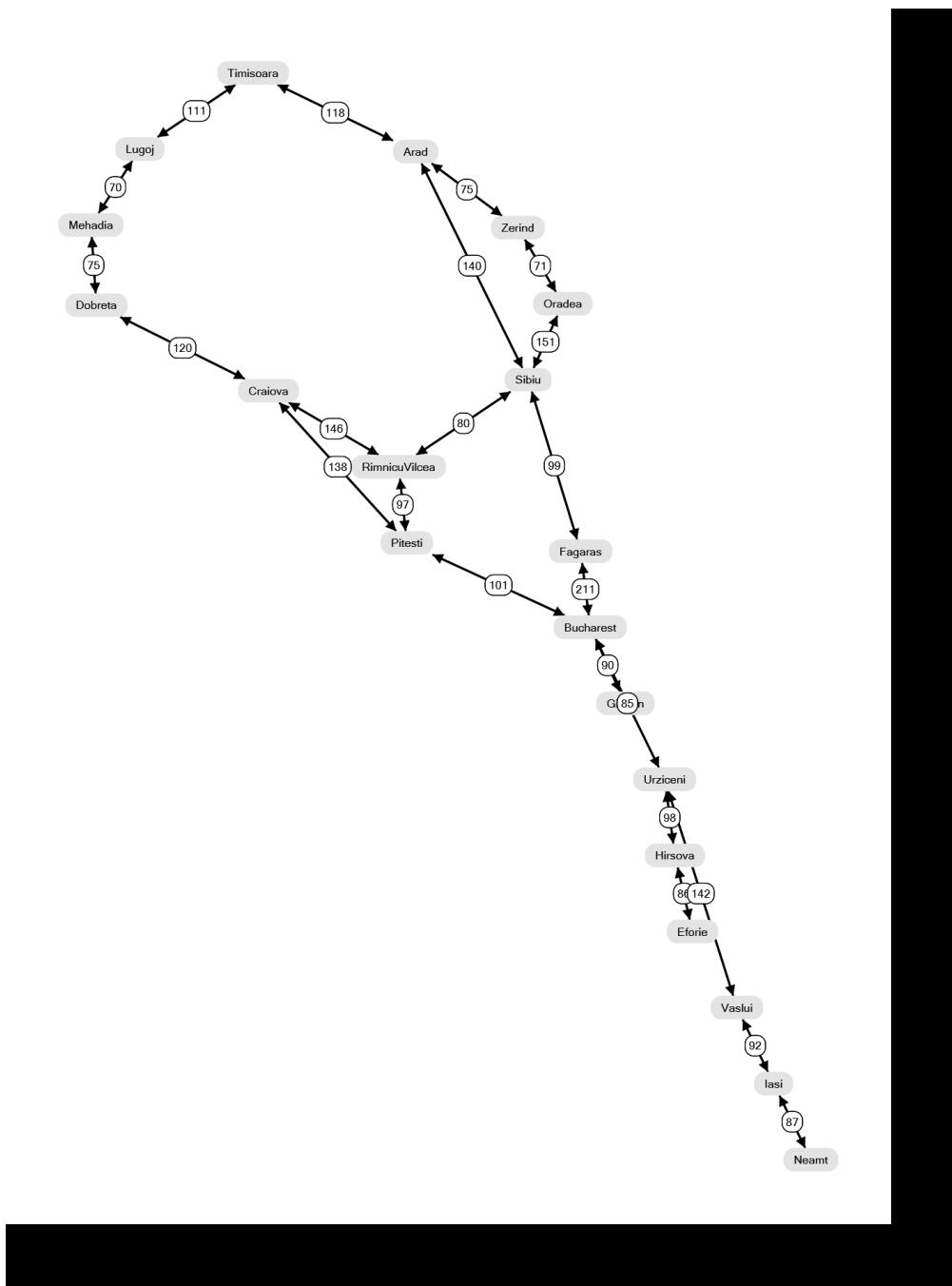


GraphSEA

Graph Search Algorithms

Benchmark Report

The Graph



We will perform a search for Bucharest starting from Arad node.

We will use various search algorithms such as (BFS, DFS, UCS, IDLS, Greedy, A*, IDA*).

This report will include Algorithms benchmarks and resolution steps, by illustrating each step performed by the search algorithms.

Results Summary

Algorithm	Elapsed time
Breadth-first search	00:00:00.0026484
Depth-first search	00:00:00.0009414
Uniform Cost Search	00:00:00.0003160
Iterative Depth limited search	00:00:00.0012527
A*	00:00:00.0026923
Greedy Best-first search	00:00:00.0021669
Iterative deepening A*	00:00:00.0032667

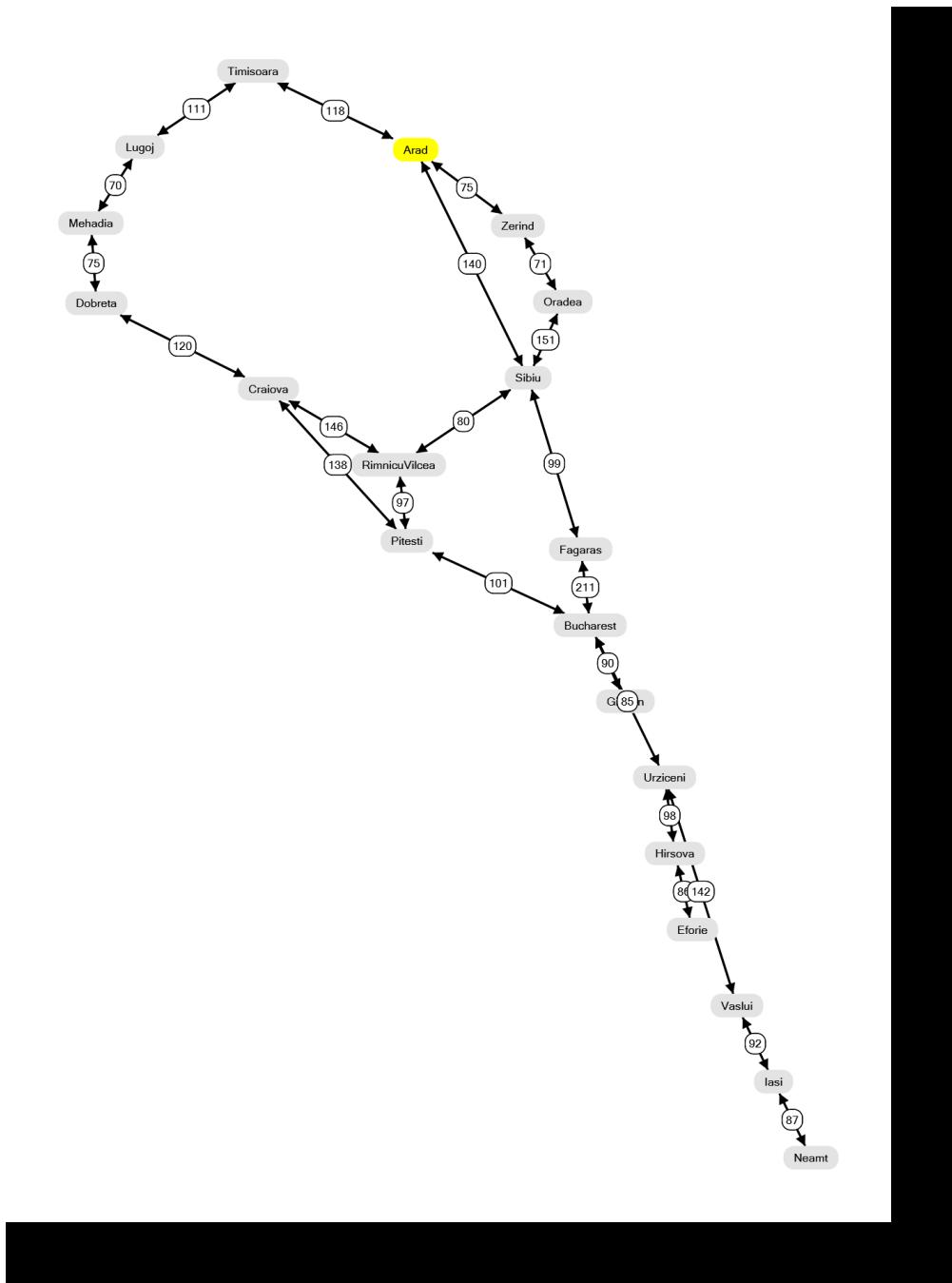
Breadth-first search

Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a 'search key') and explores the neighbor nodes first, before moving to the next level neighbours.

The algorithm took 00:00:00.0026484 to find the goal.

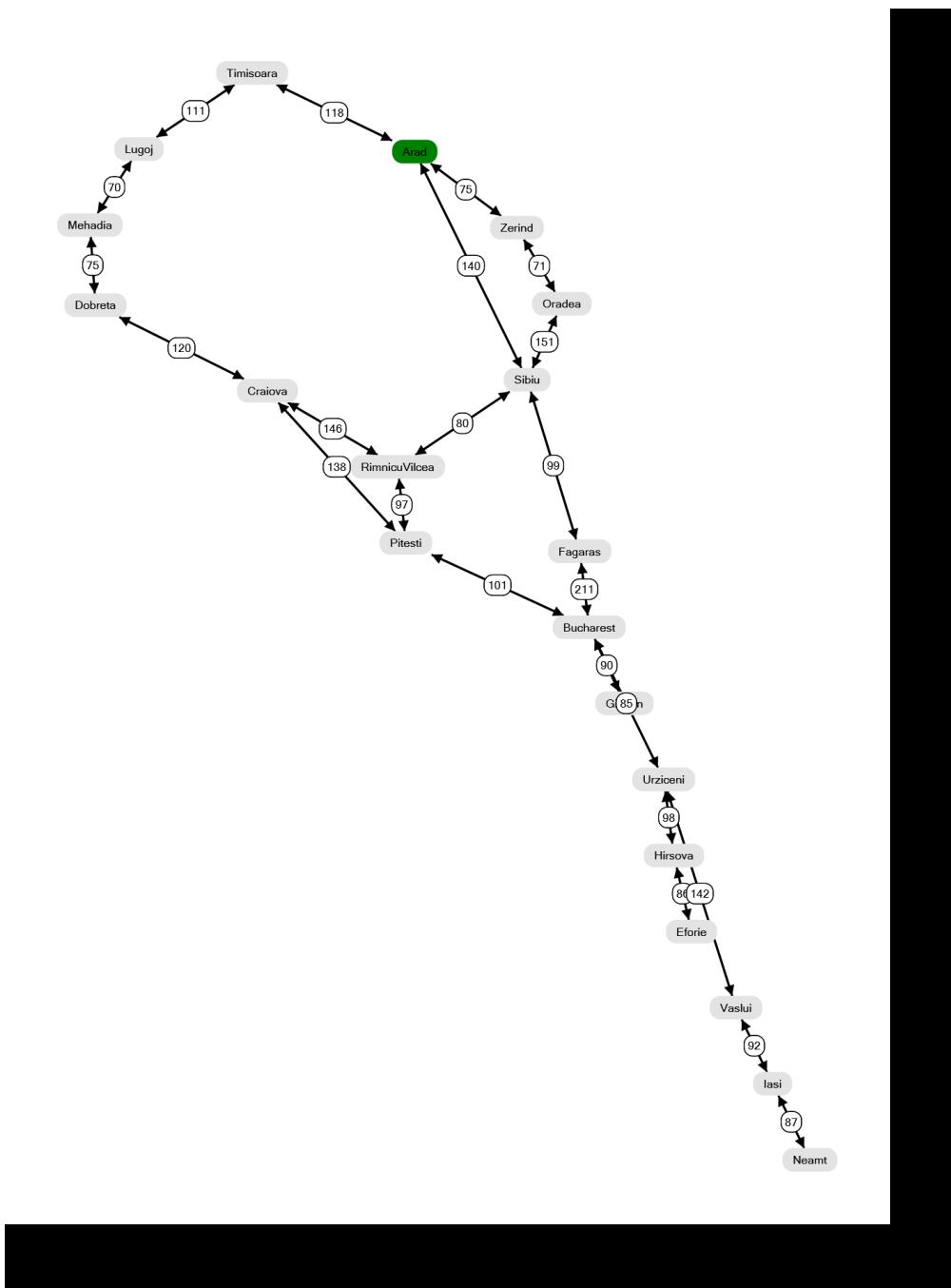
Pre-Visit

The algorithm has opened the node Arad



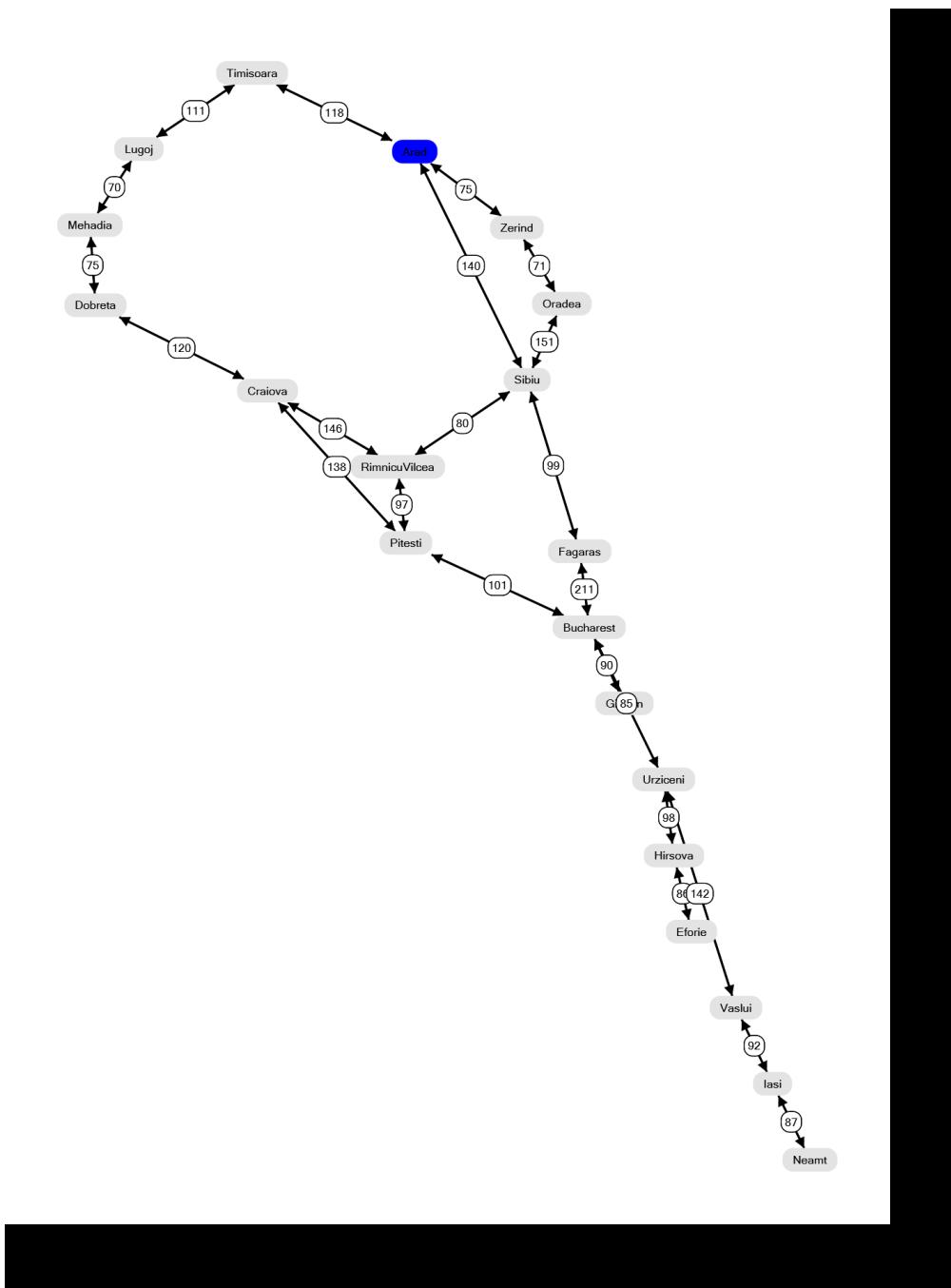
Visit

The algorithm has marked the node Arad as visited



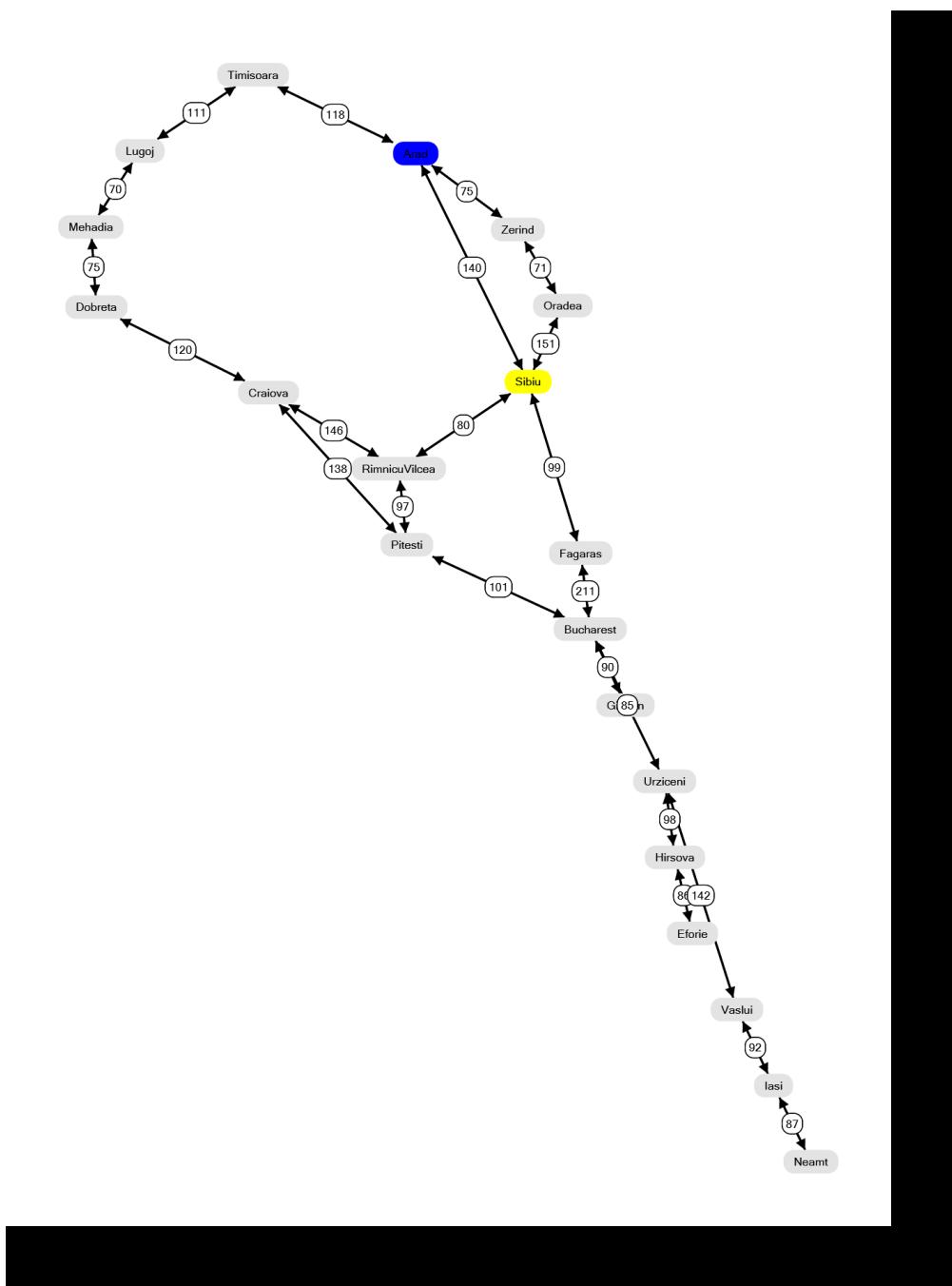
Post-Visit

The algorithm has closed the node Arad



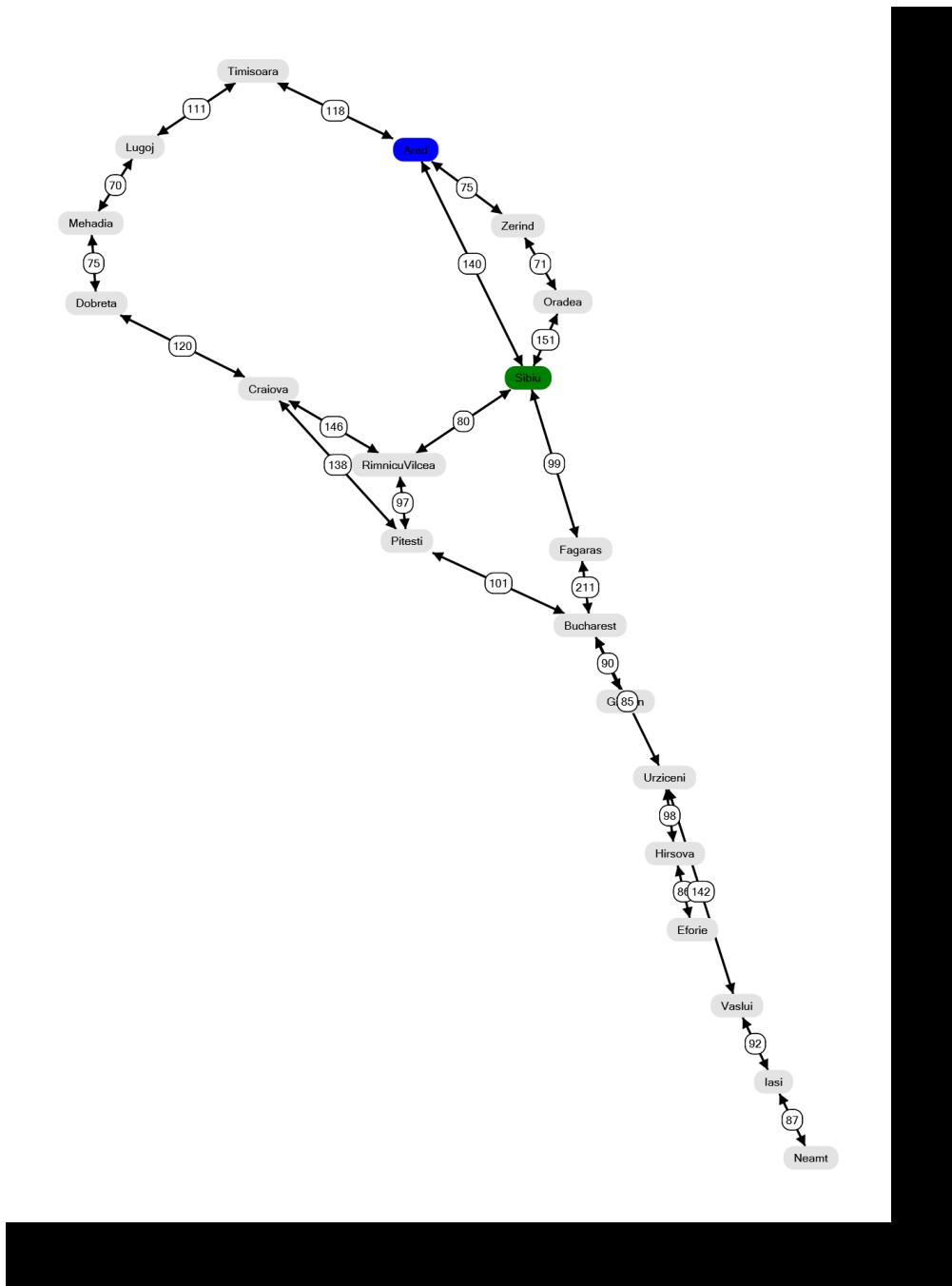
Pre-Visit

The algorithm has opened the node Sibiu



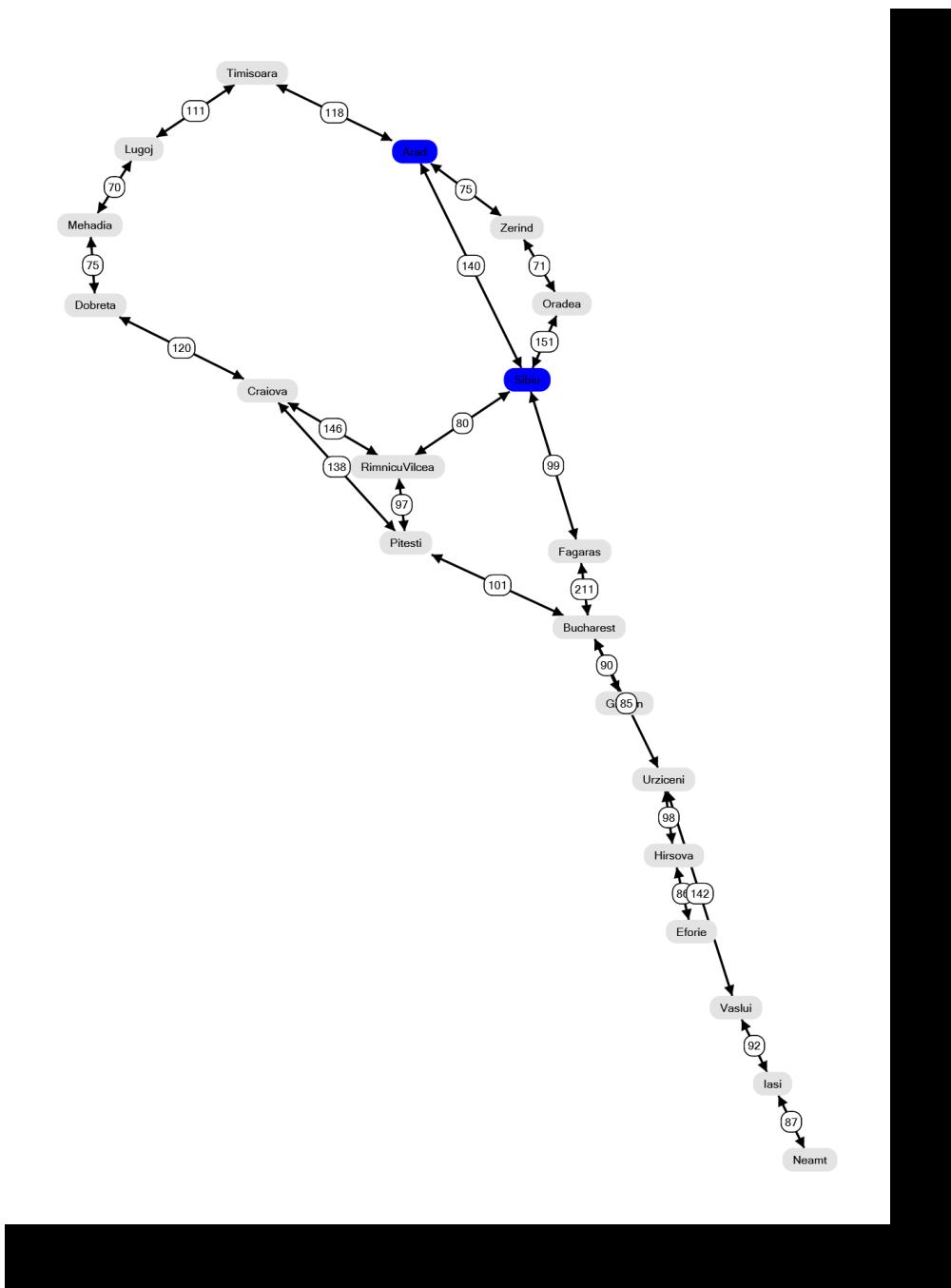
Visit

The algorithm has marked the node Sibiu as visited



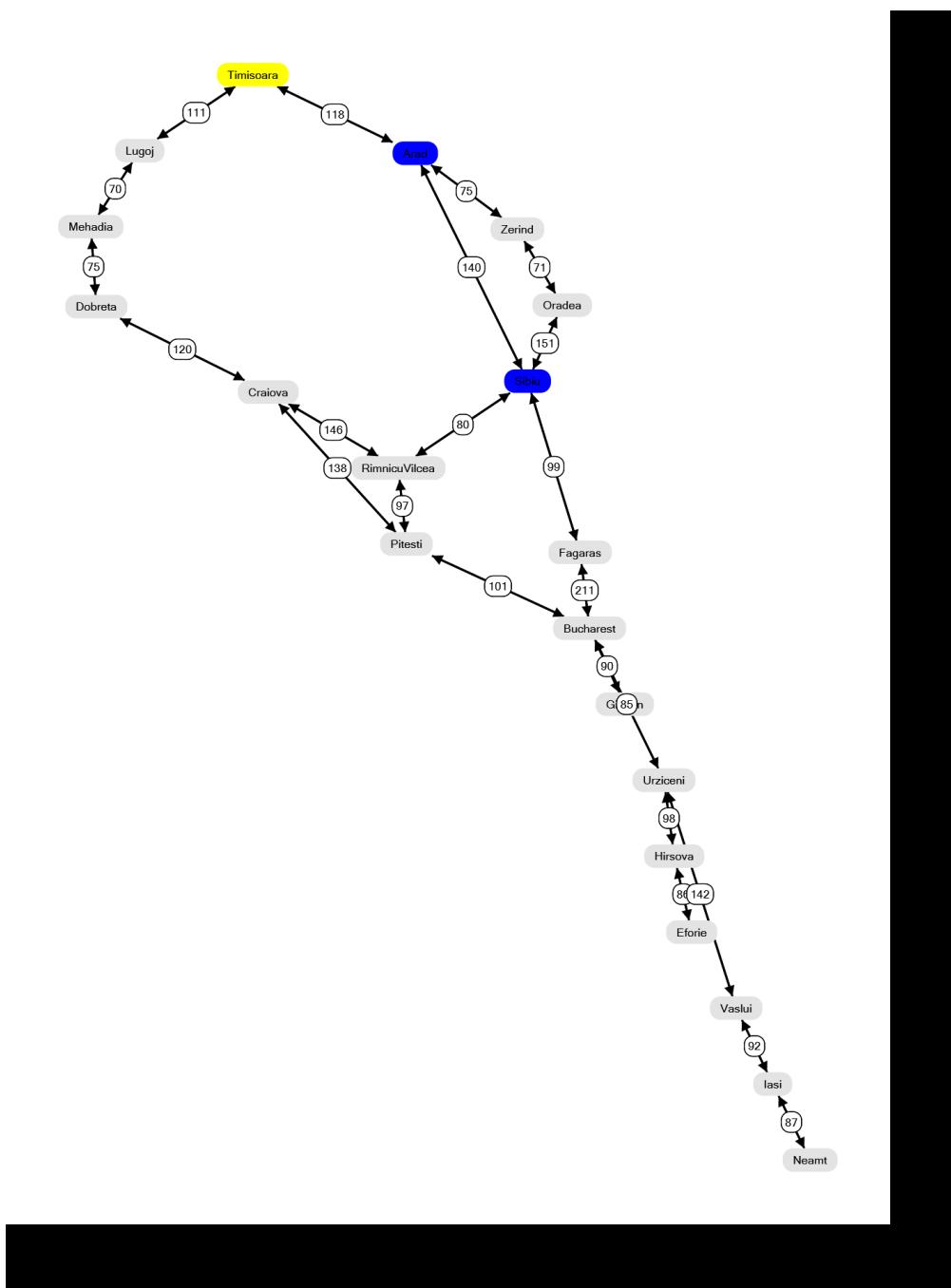
Post-Visit

The algorithm has closed the node Sibiu



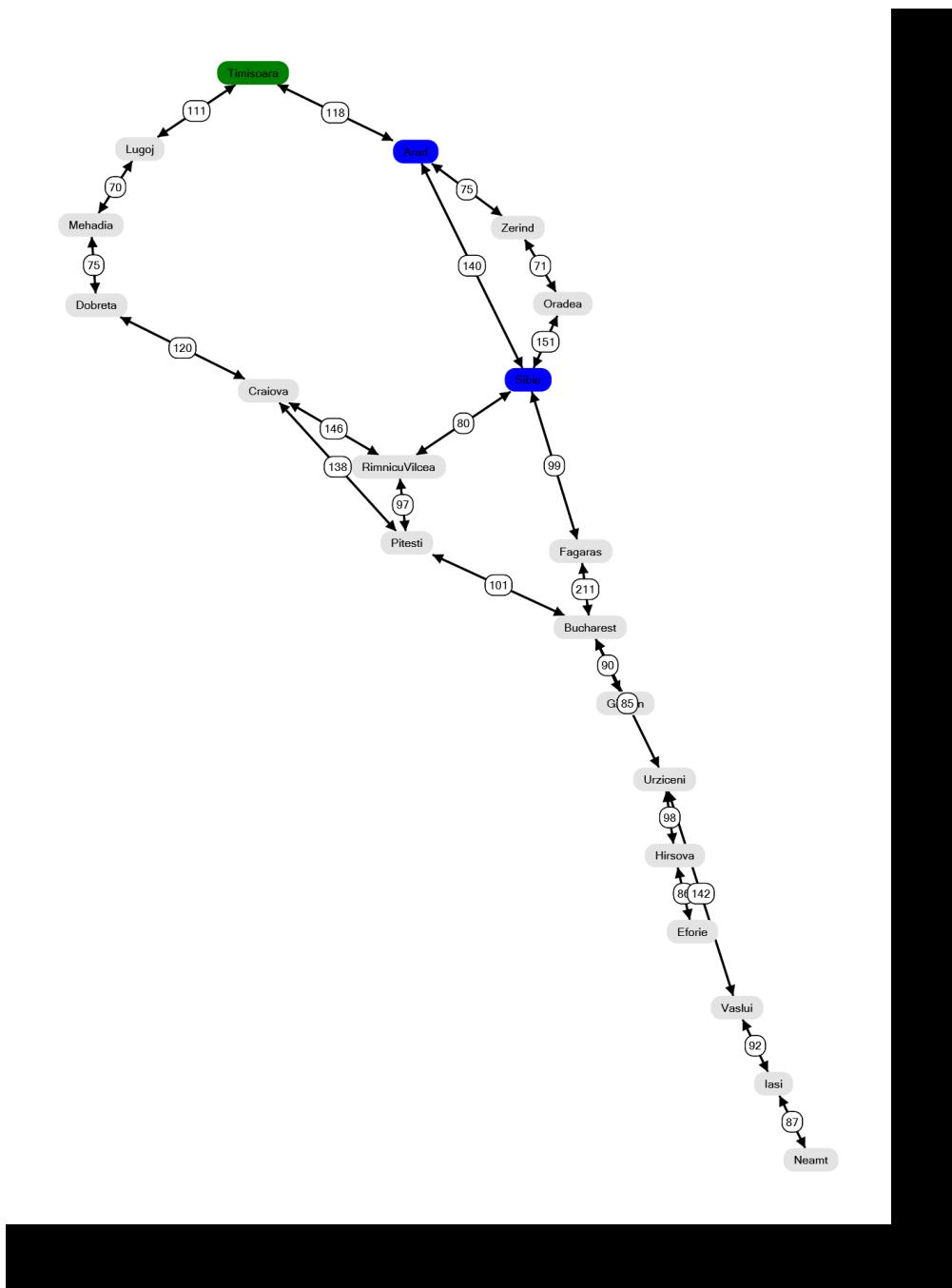
Pre-Visit

The algorithm has opened the node Timisoara



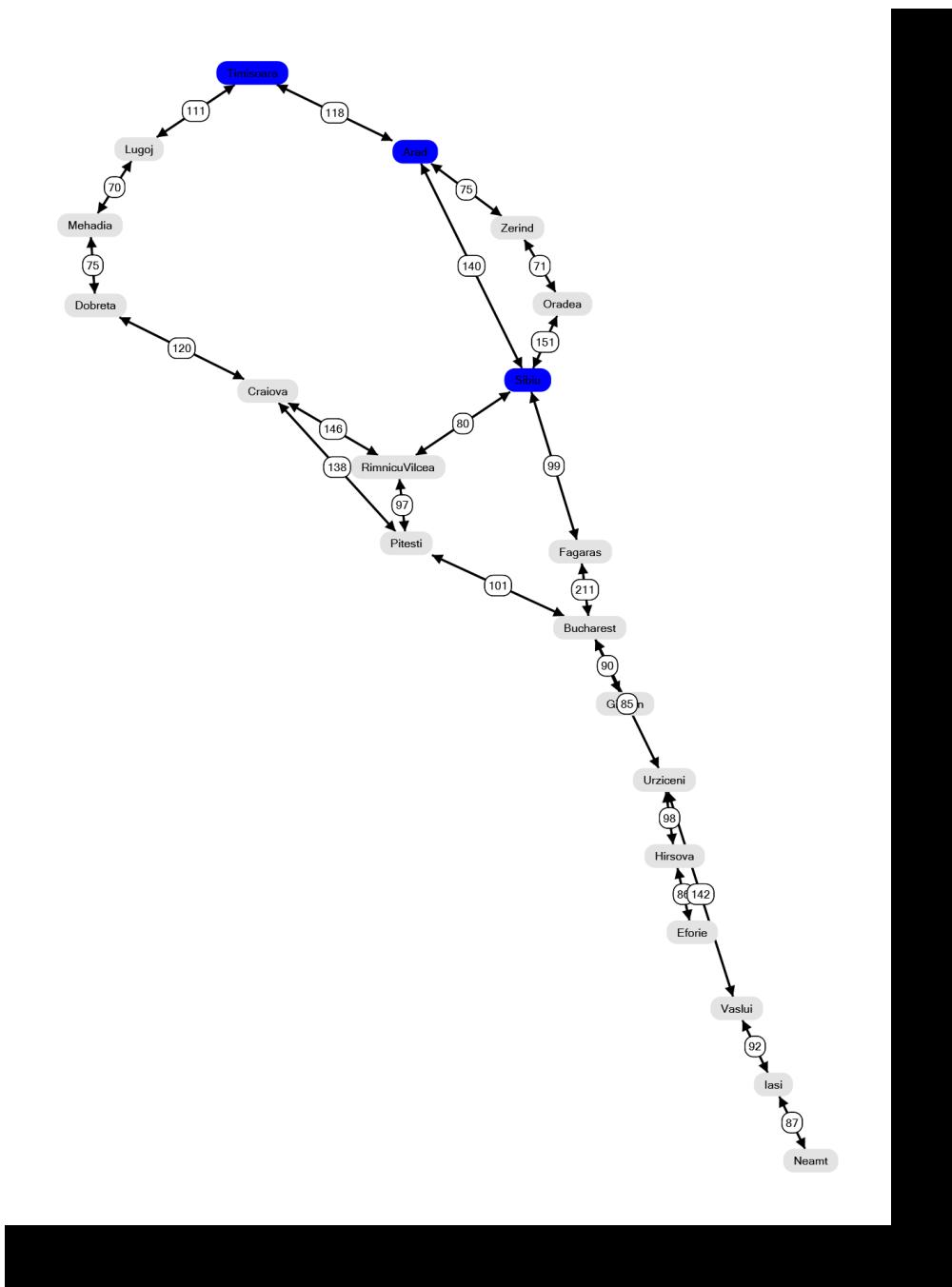
Visit

The algorithm has marked the node Timisoara as visited



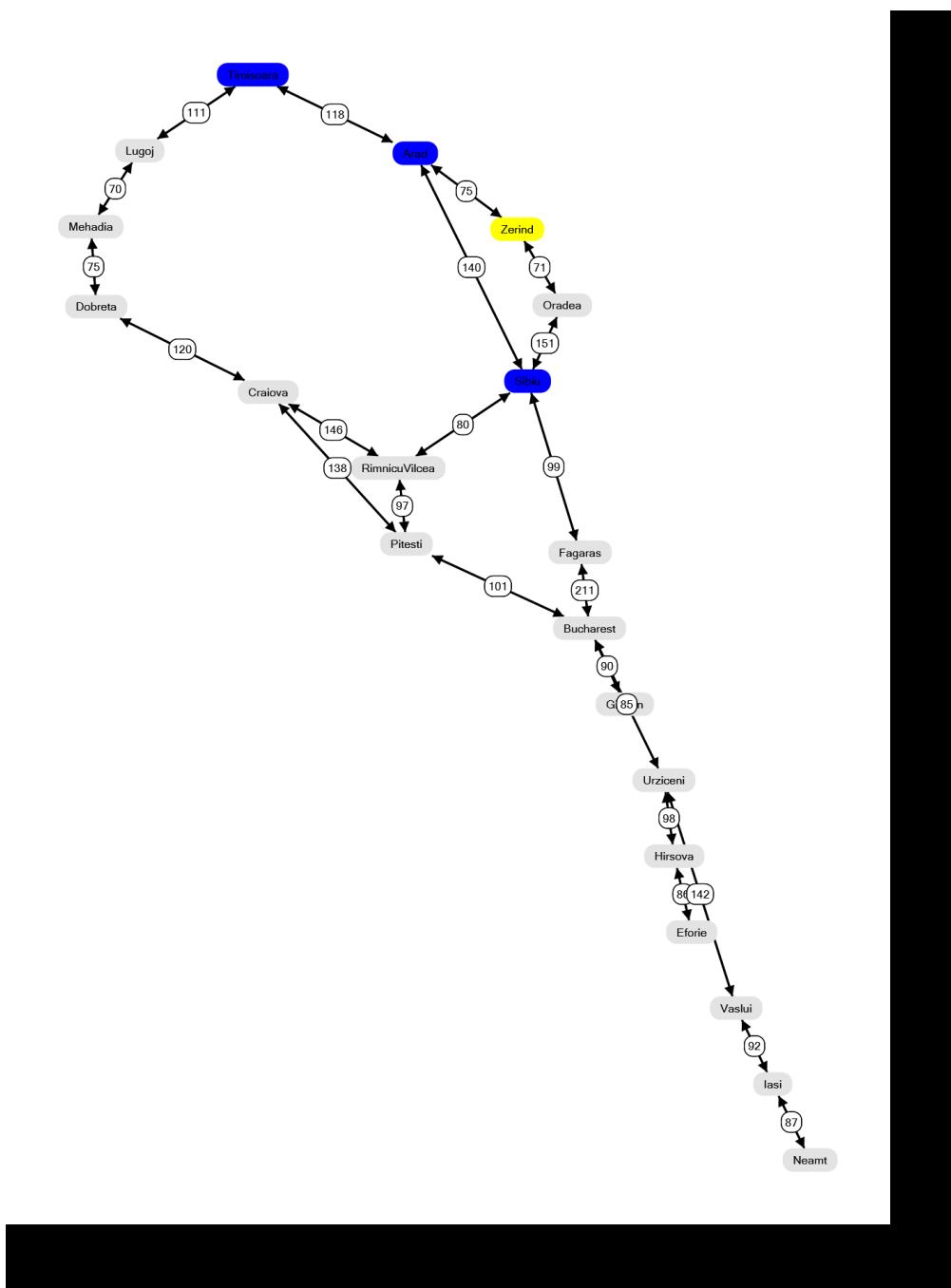
Post-Visit

The algorithm has closed the node Timisoara



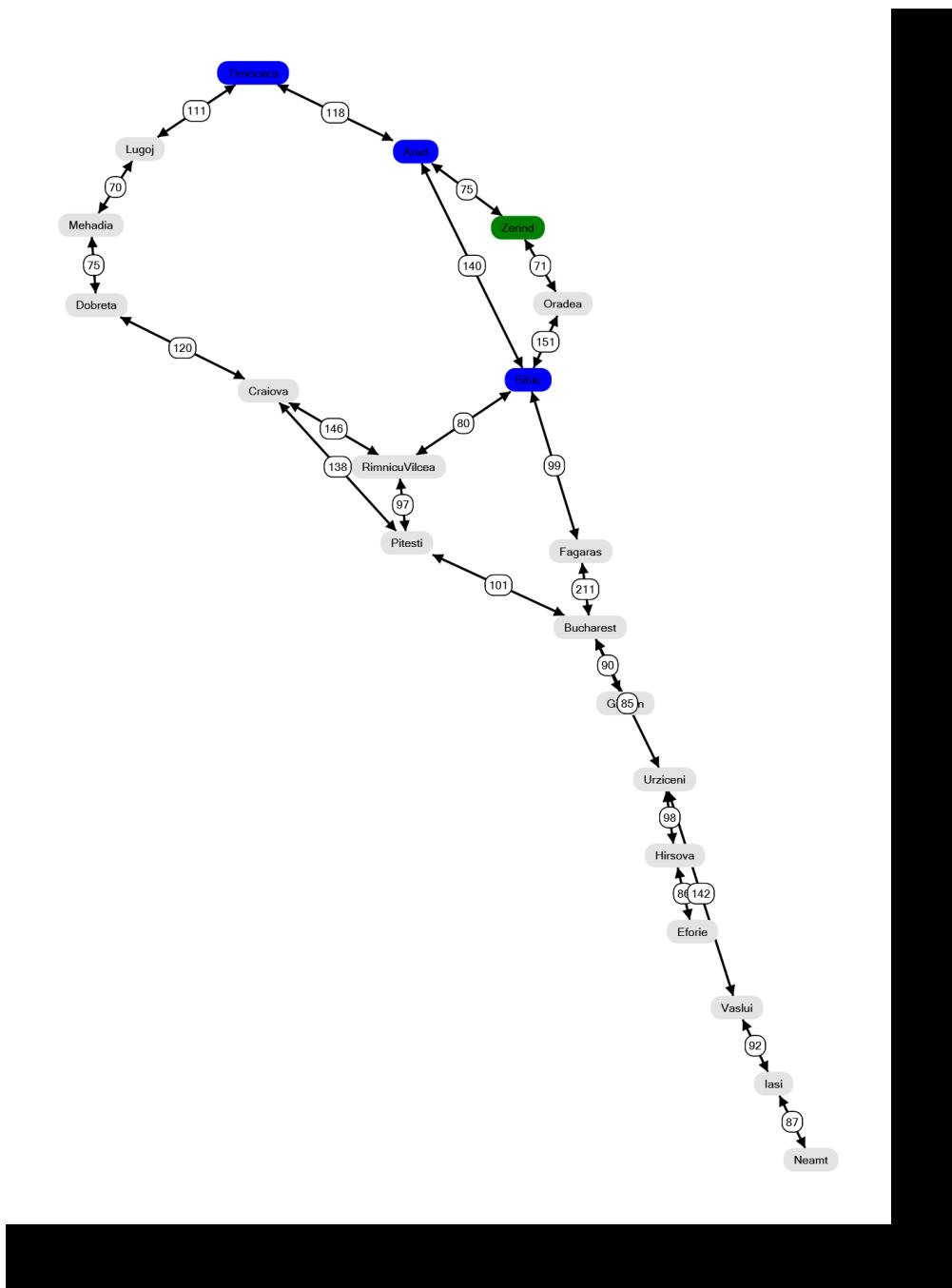
Pre-Visit

The algorithm has opened the node Zerind



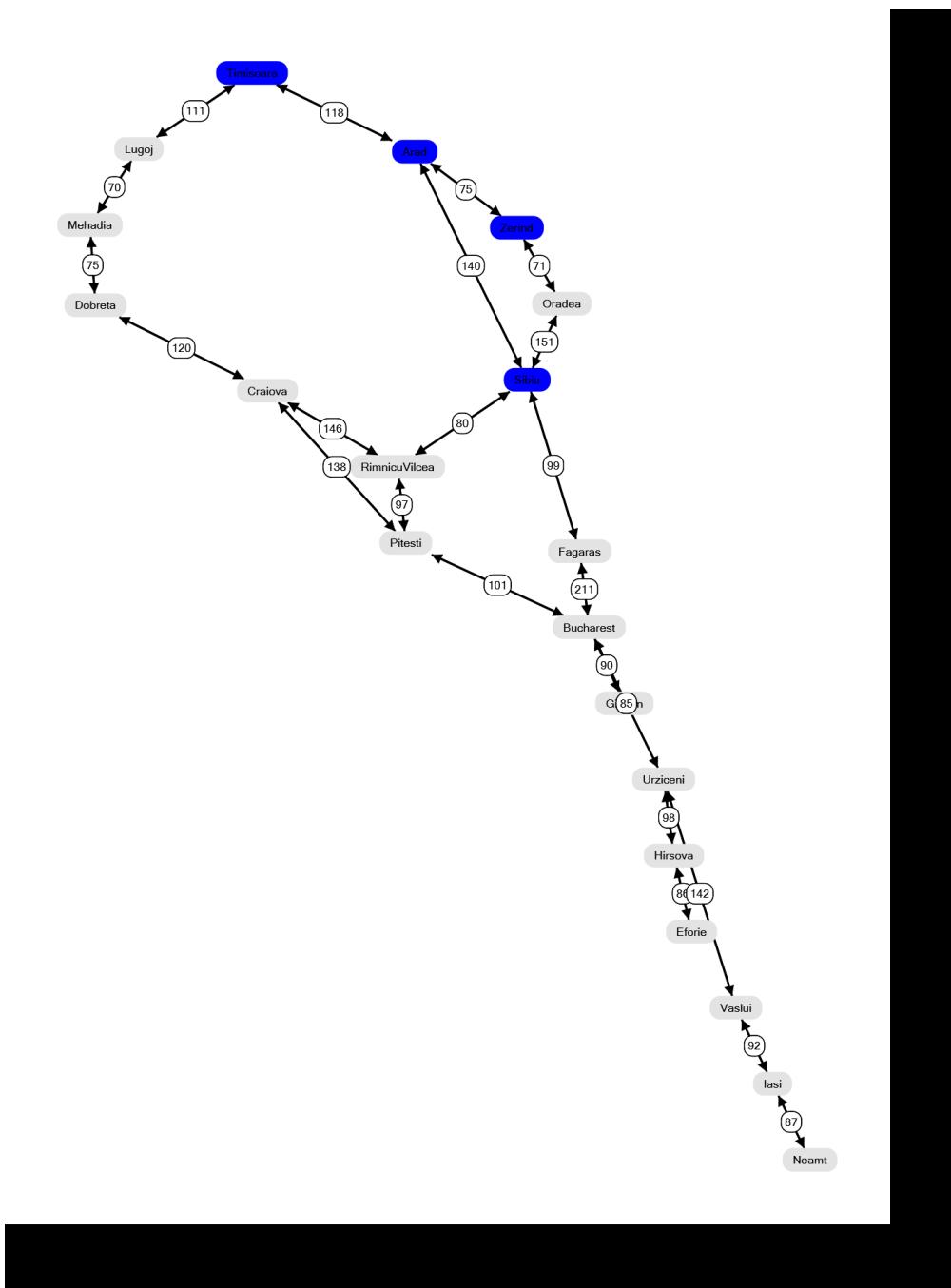
Visit

The algorithm has marked the node Zerind as visited



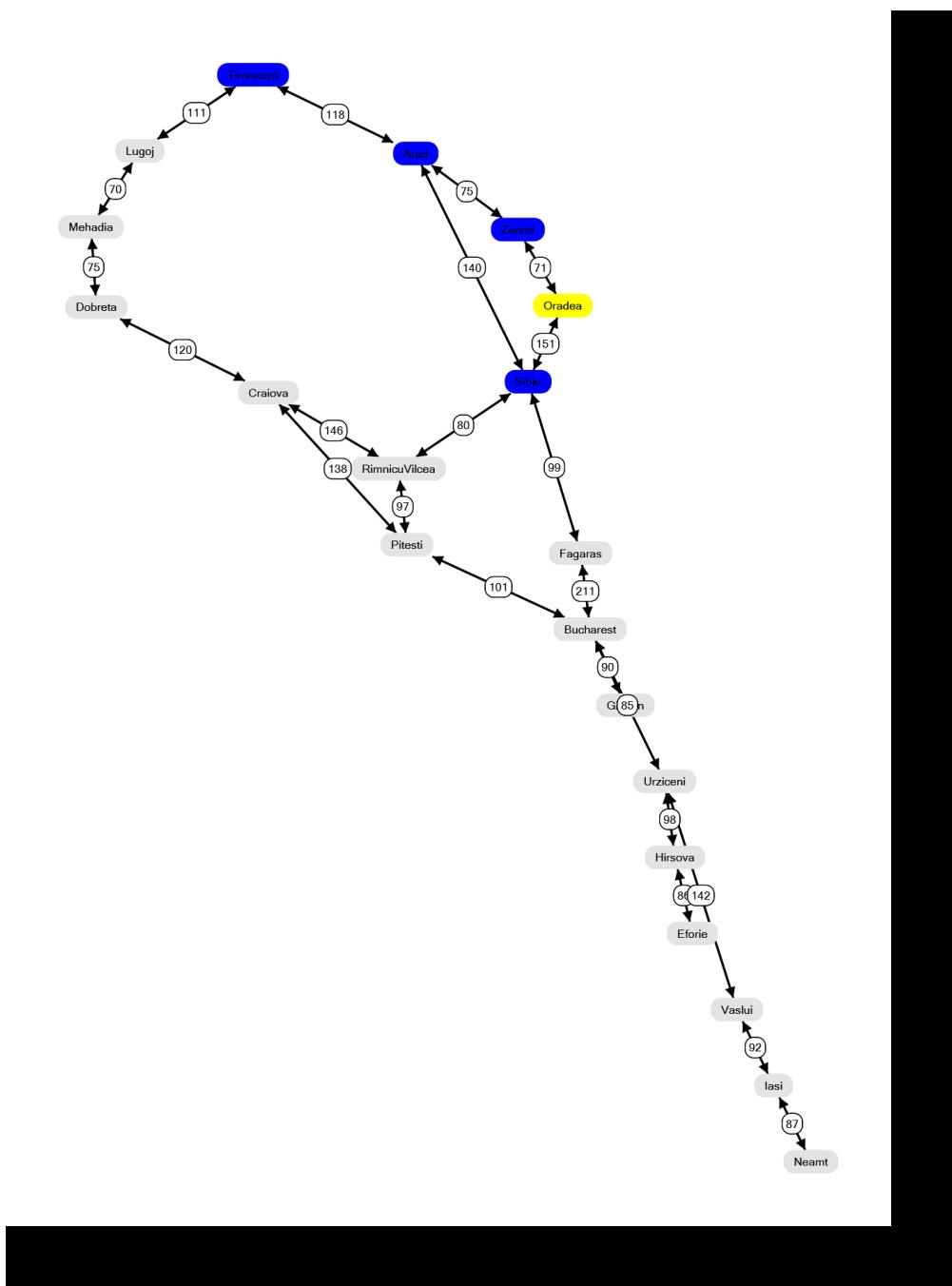
Post-Visit

The algorithm has closed the node Zerind



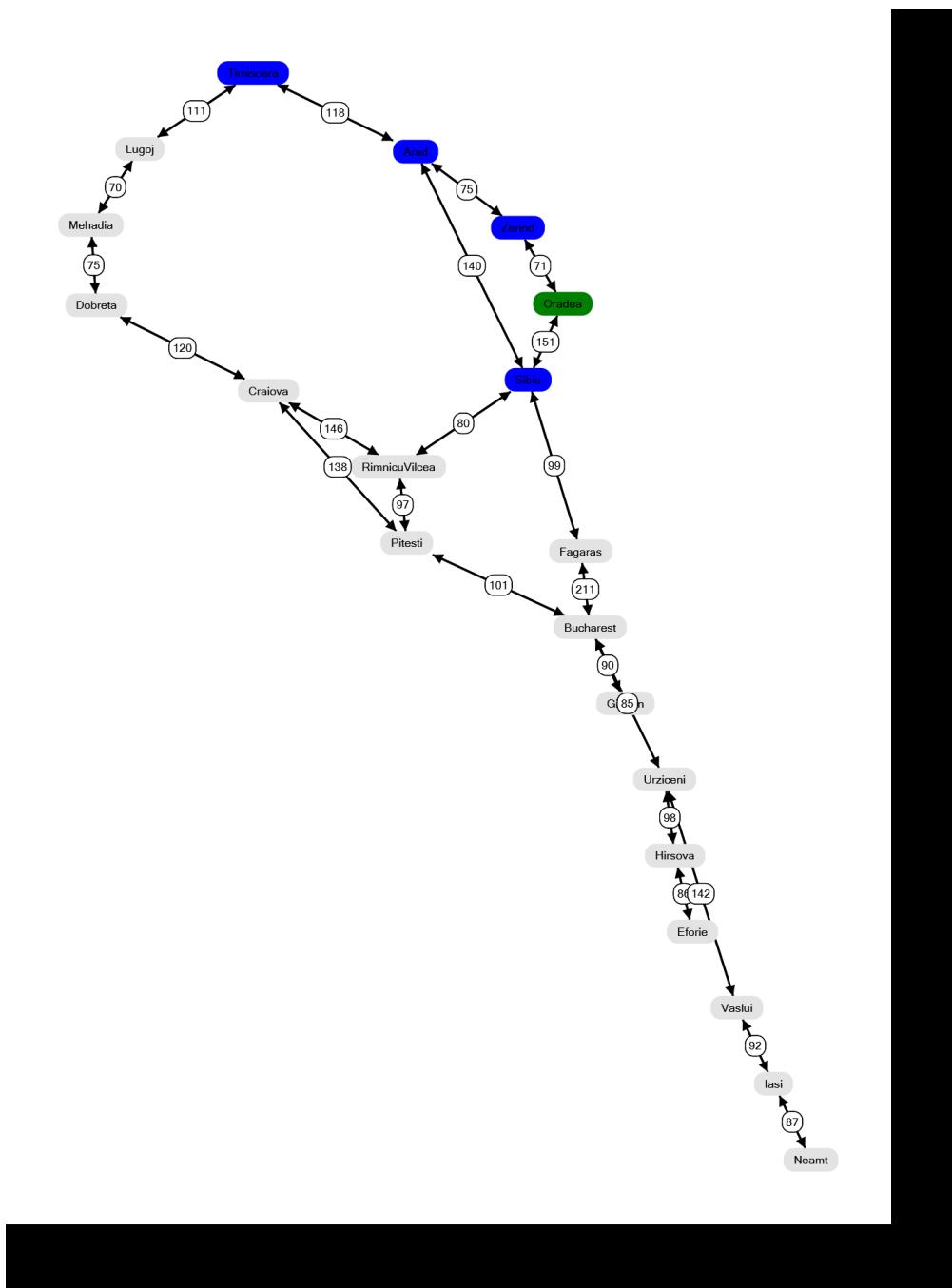
Pre-Visit

The algorithm has opened the node Oradea



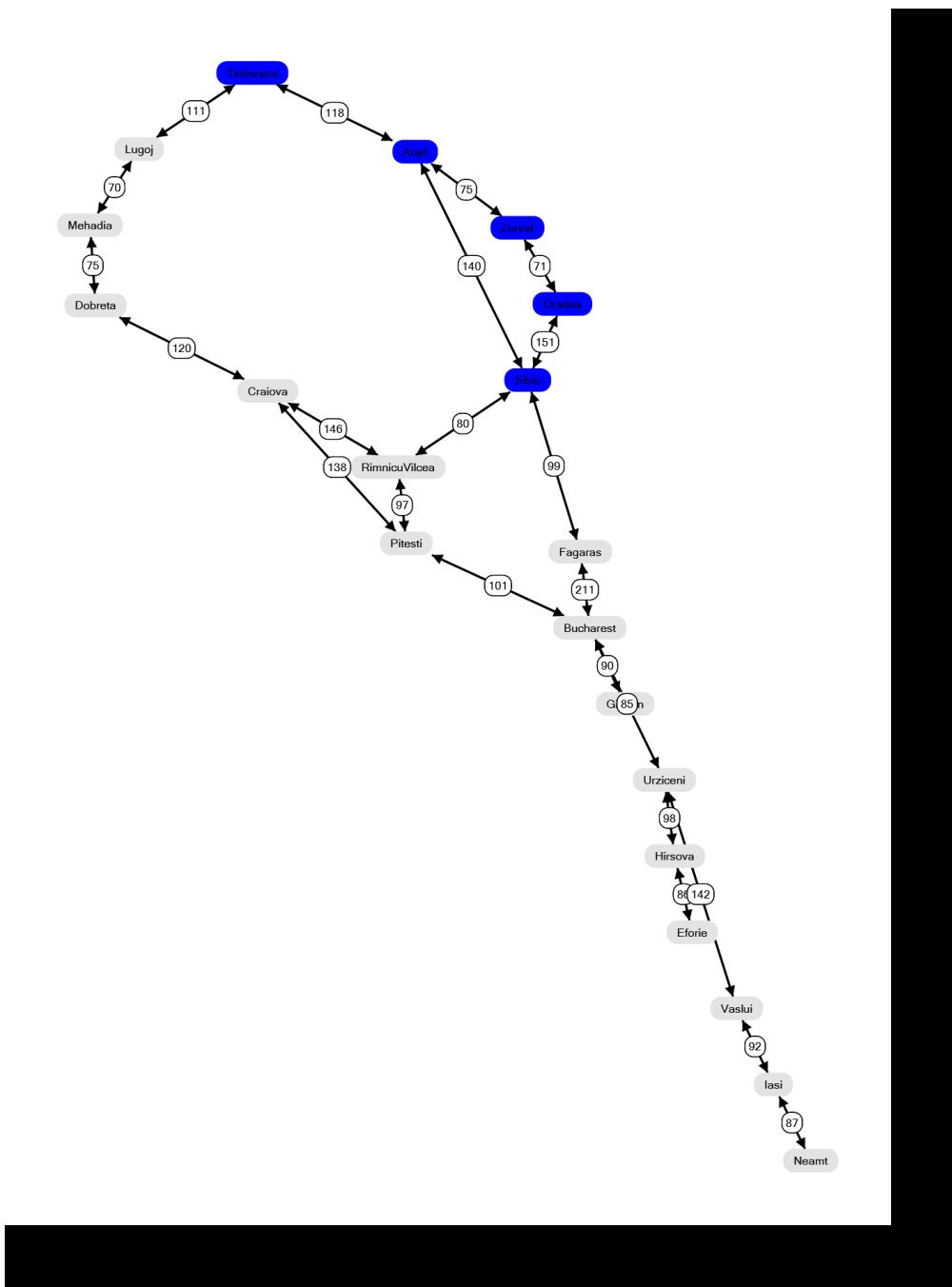
Visit

The algorithm has marked the node Oradea as visited



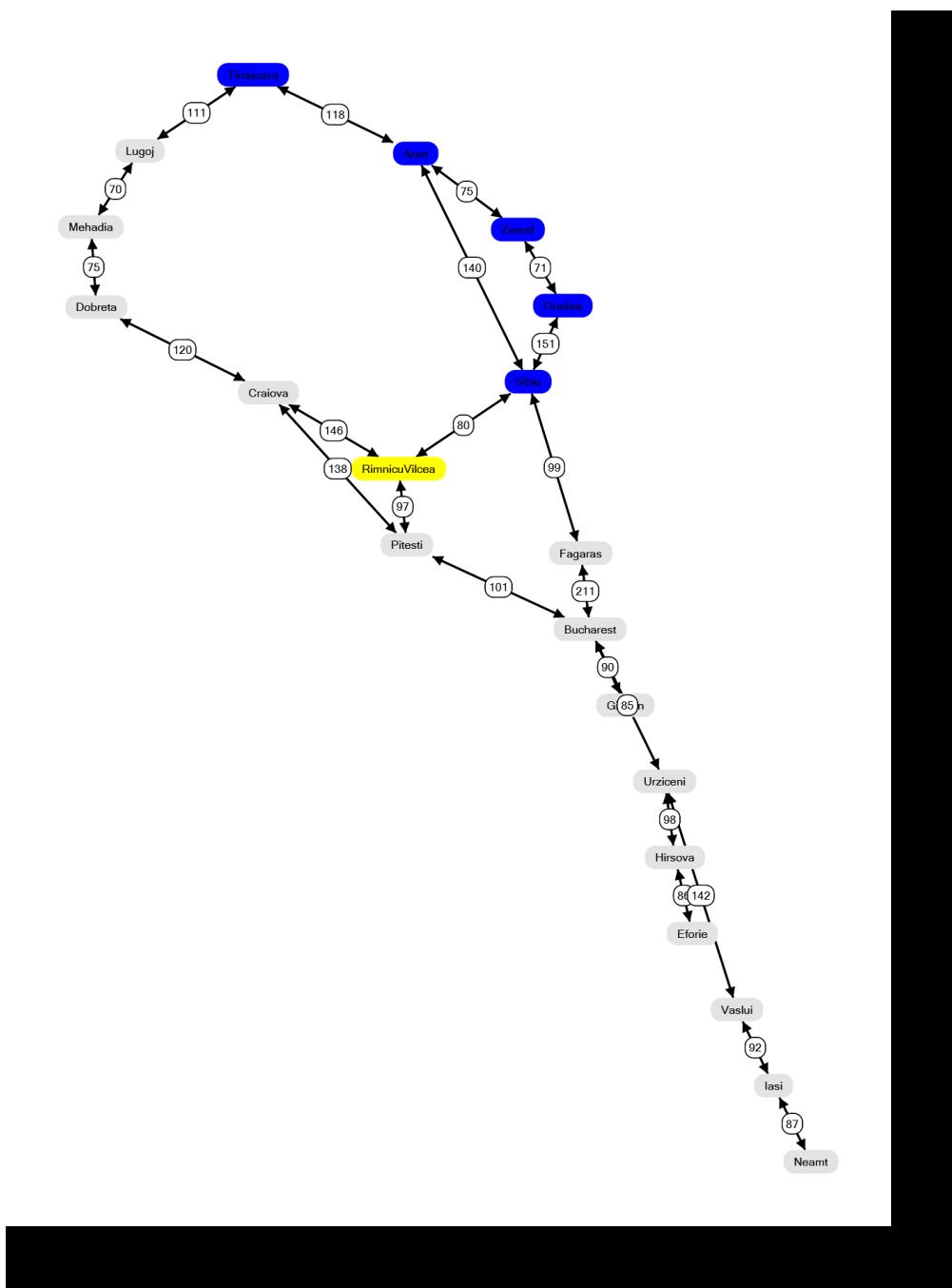
Post-Visit

The algorithm has closed the node Oradea



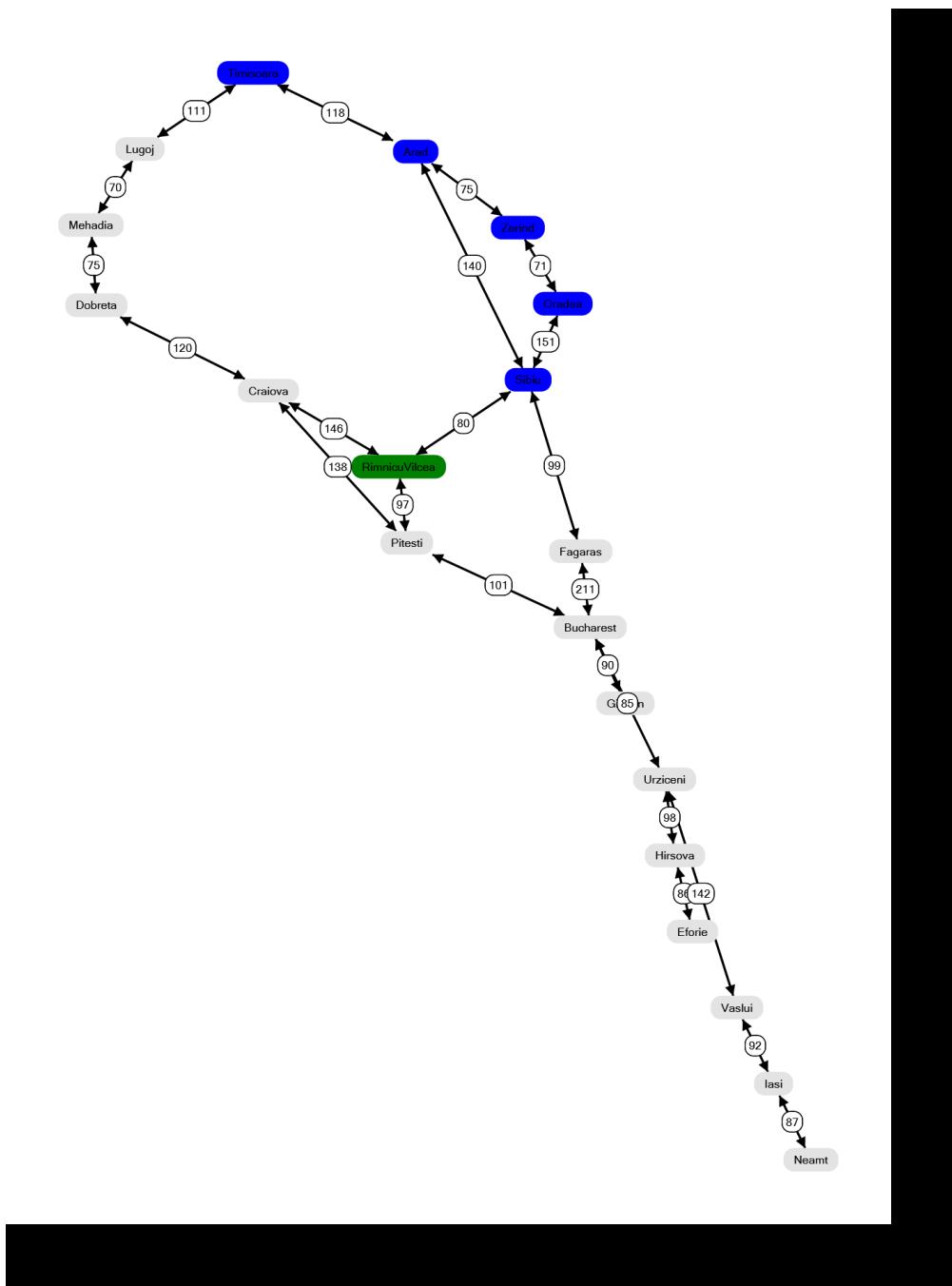
Pre-Visit

The algorithm has opened the node Rimnicu Vilcea



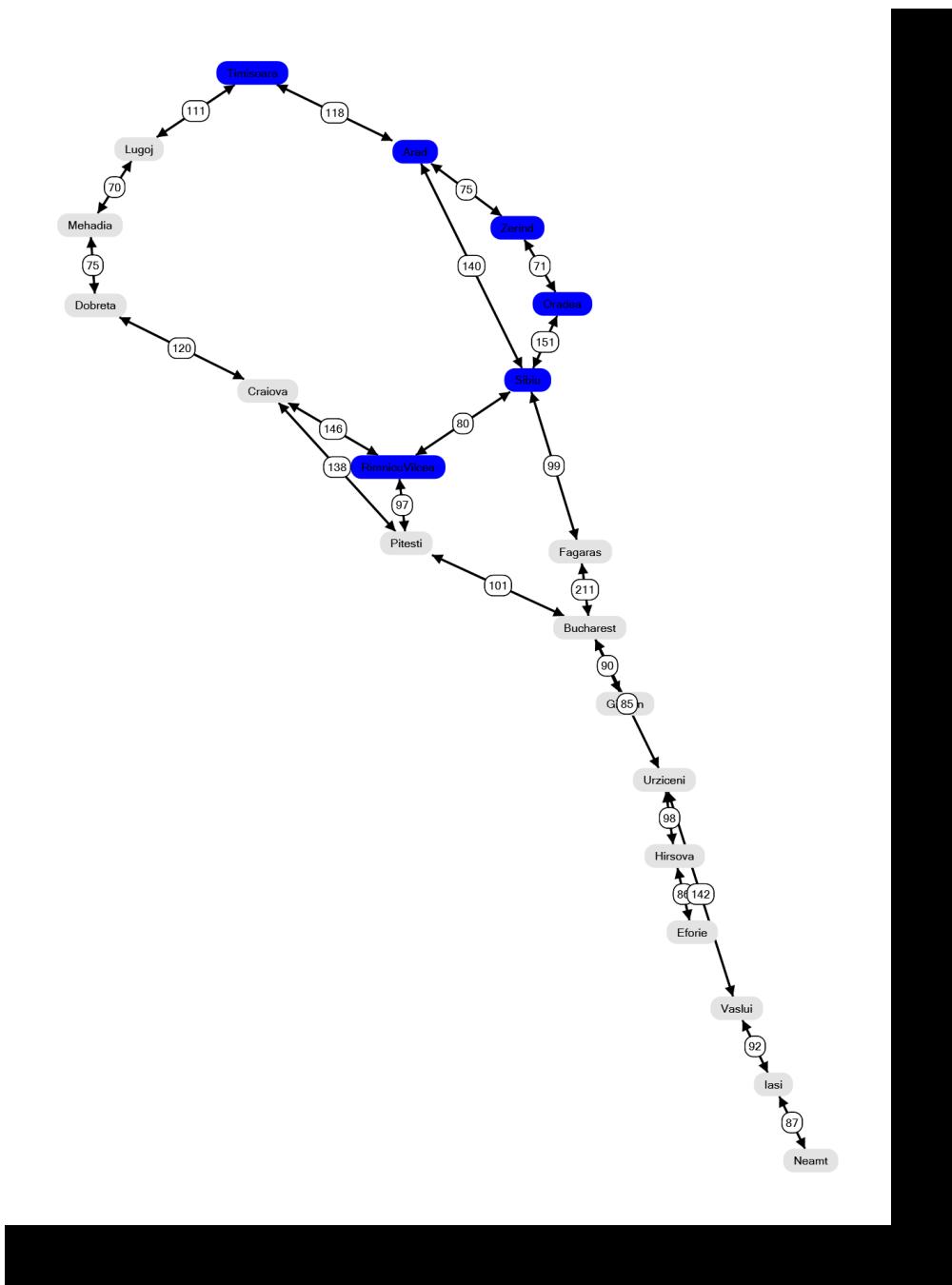
Visit

The algorithm has marked the node Rimnicu Vilcea as visited



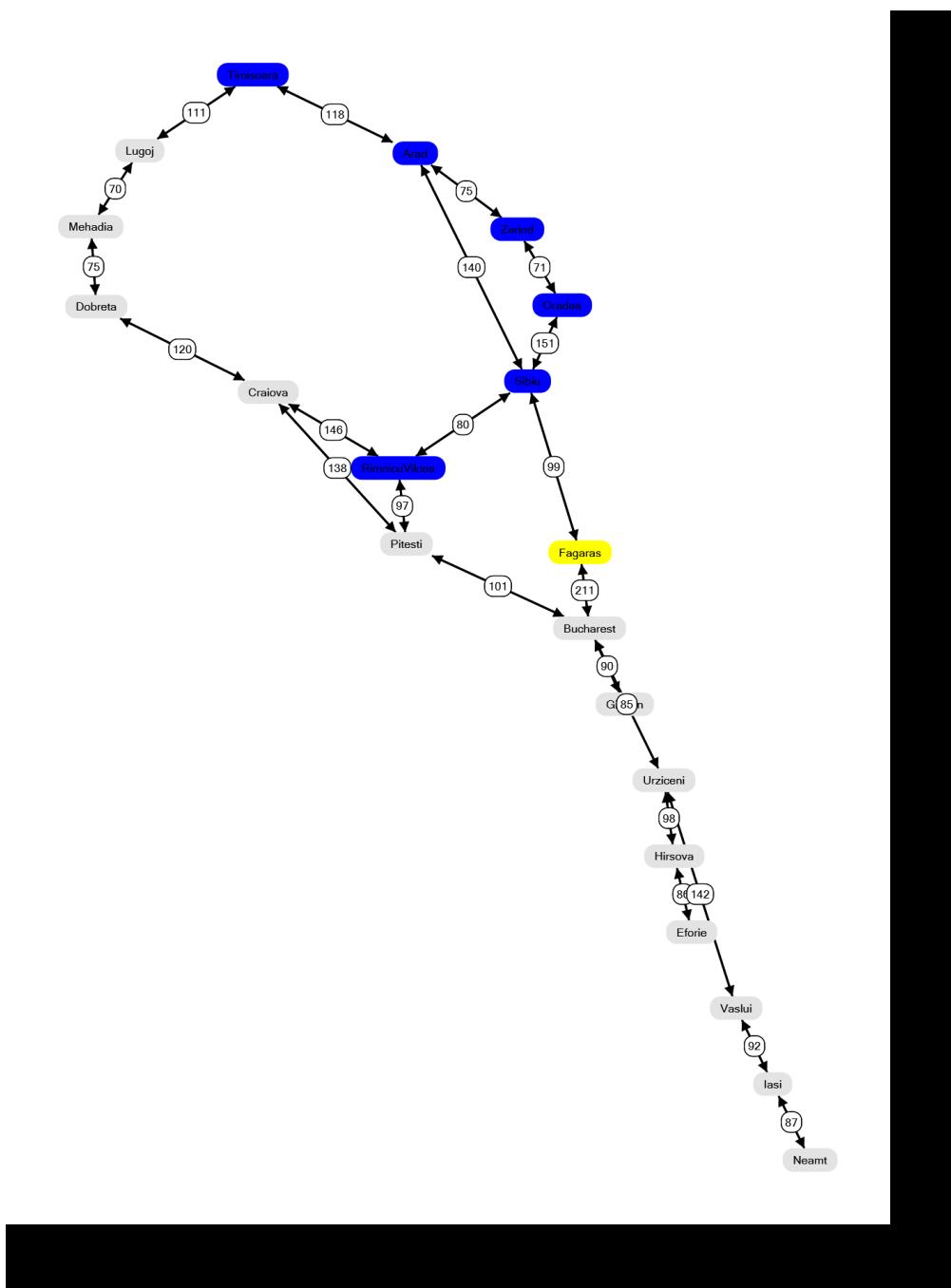
Post-Visit

The algorithm has closed the node Rimnicu Vilcea



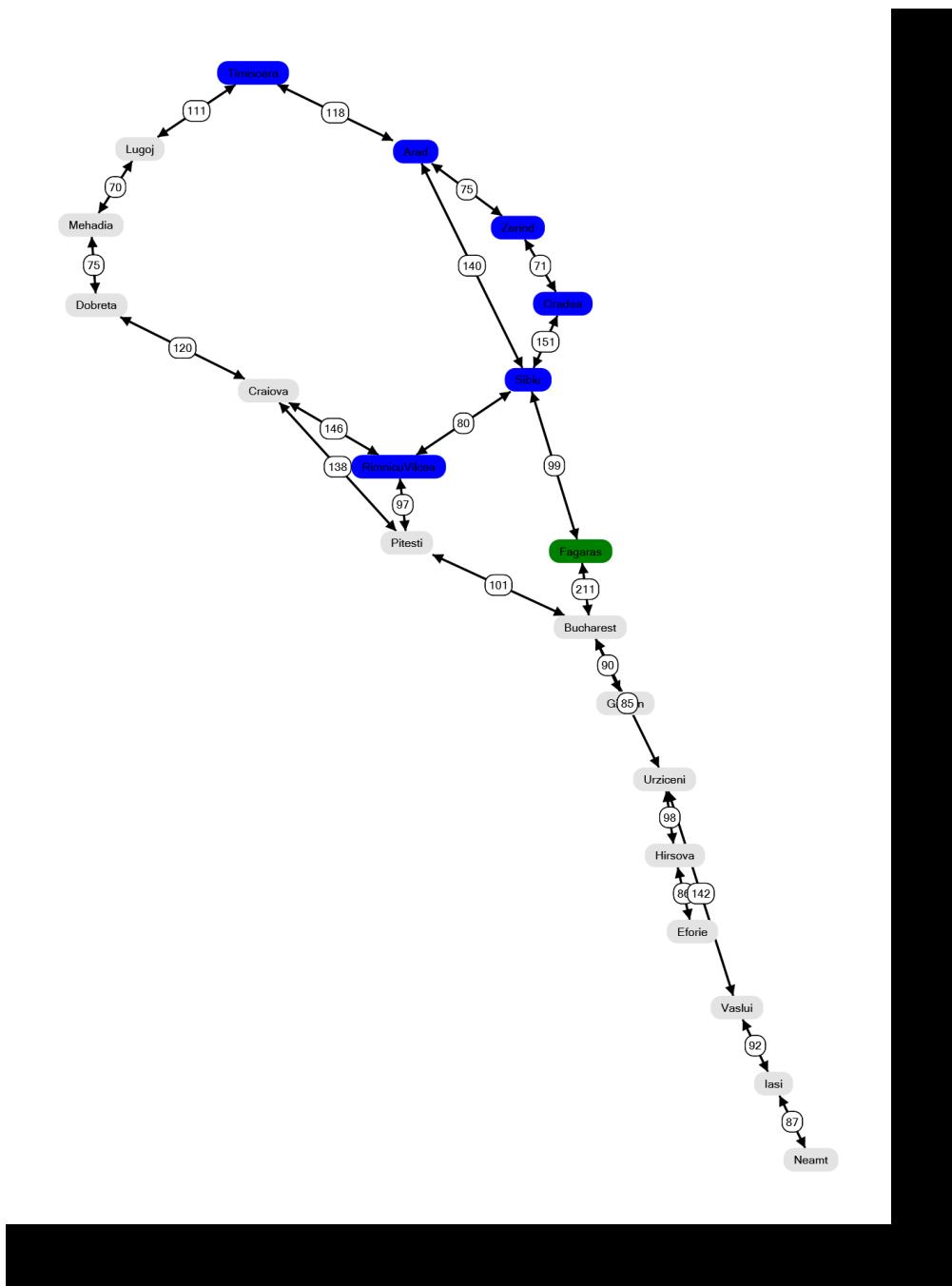
Pre-Visit

The algorithm has opened the node Fagaras



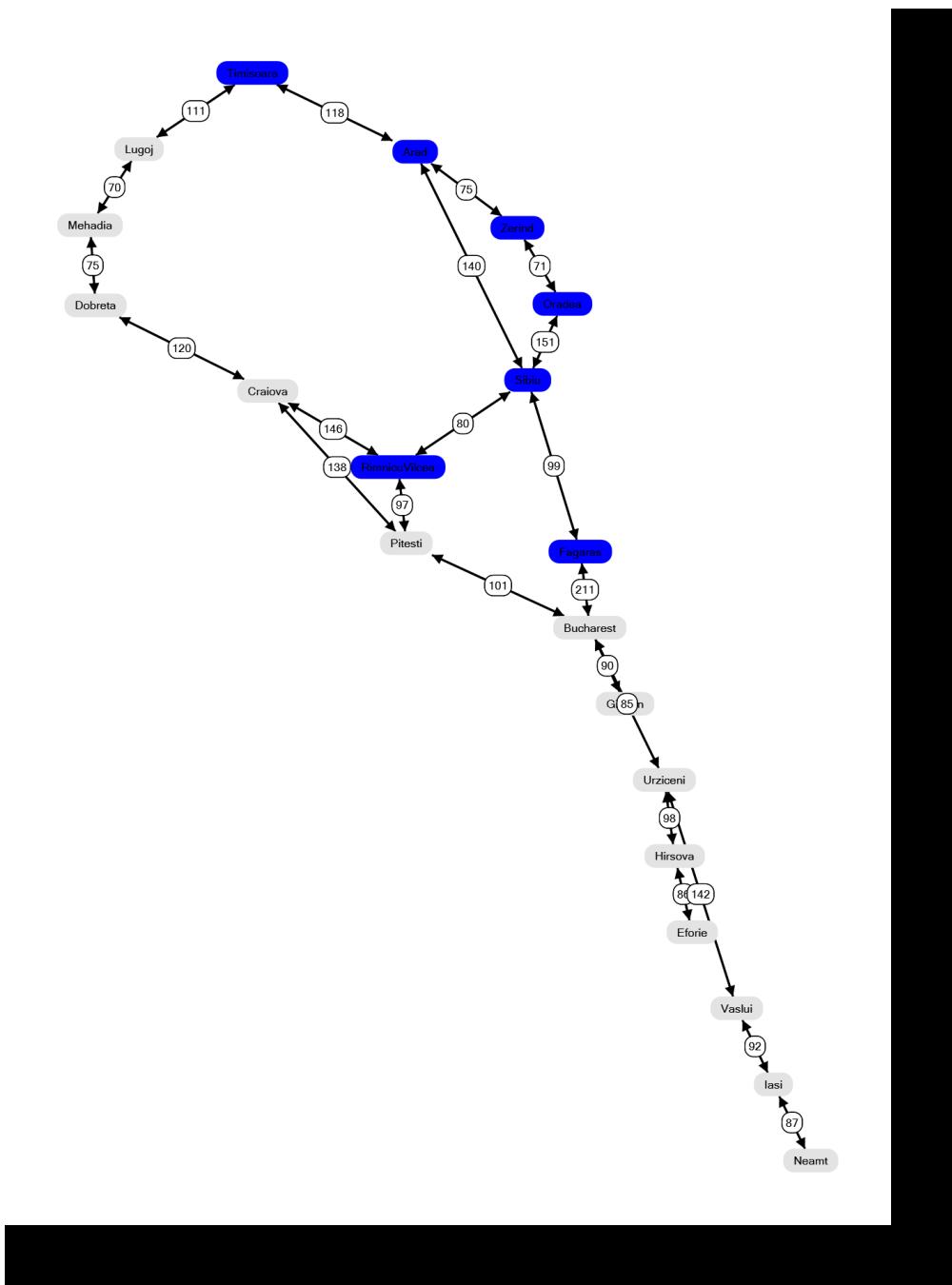
Visit

The algorithm has marked the node Fagaras as visited



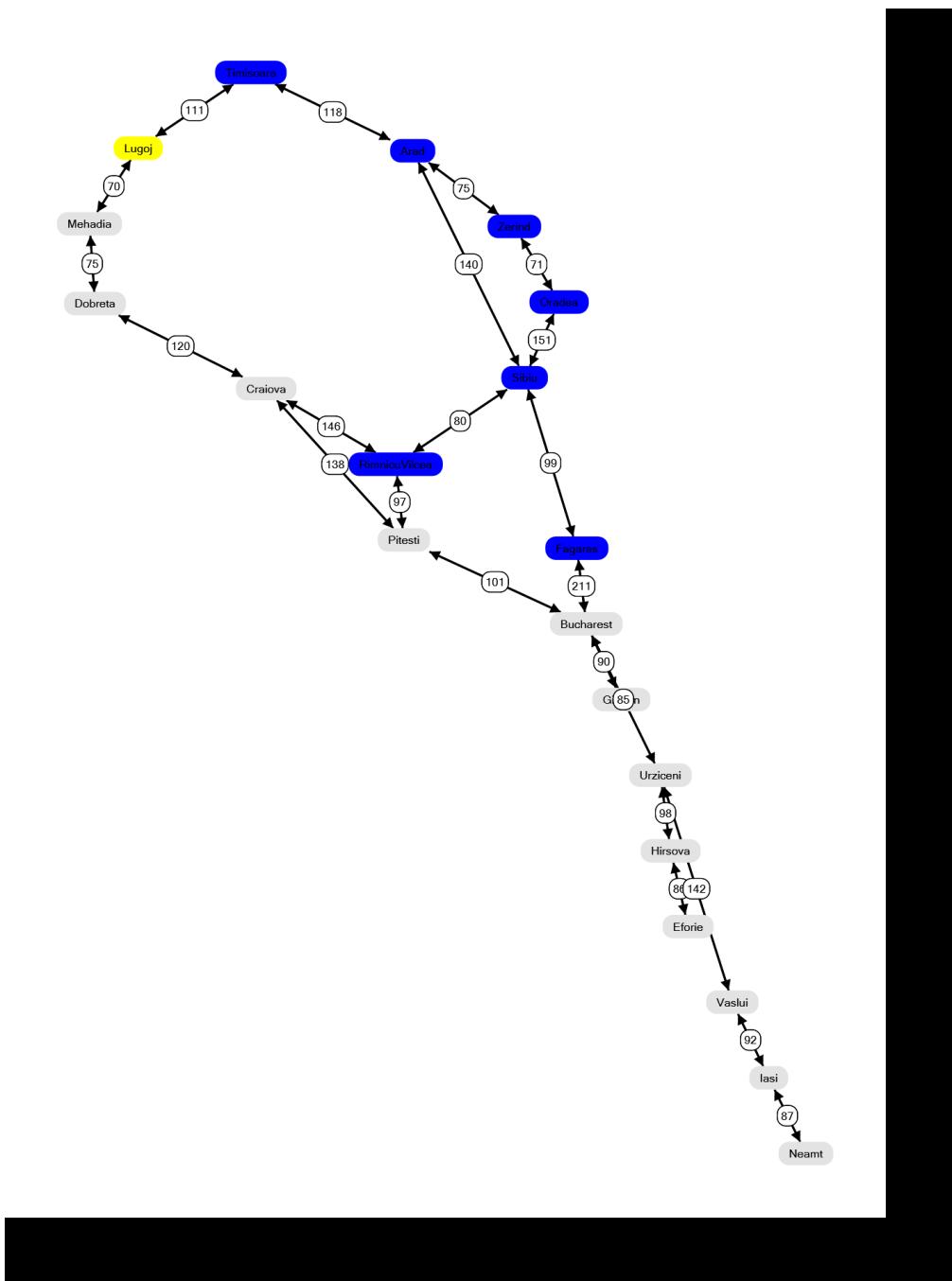
Post-Visit

The algorithm has closed the node Fagaras



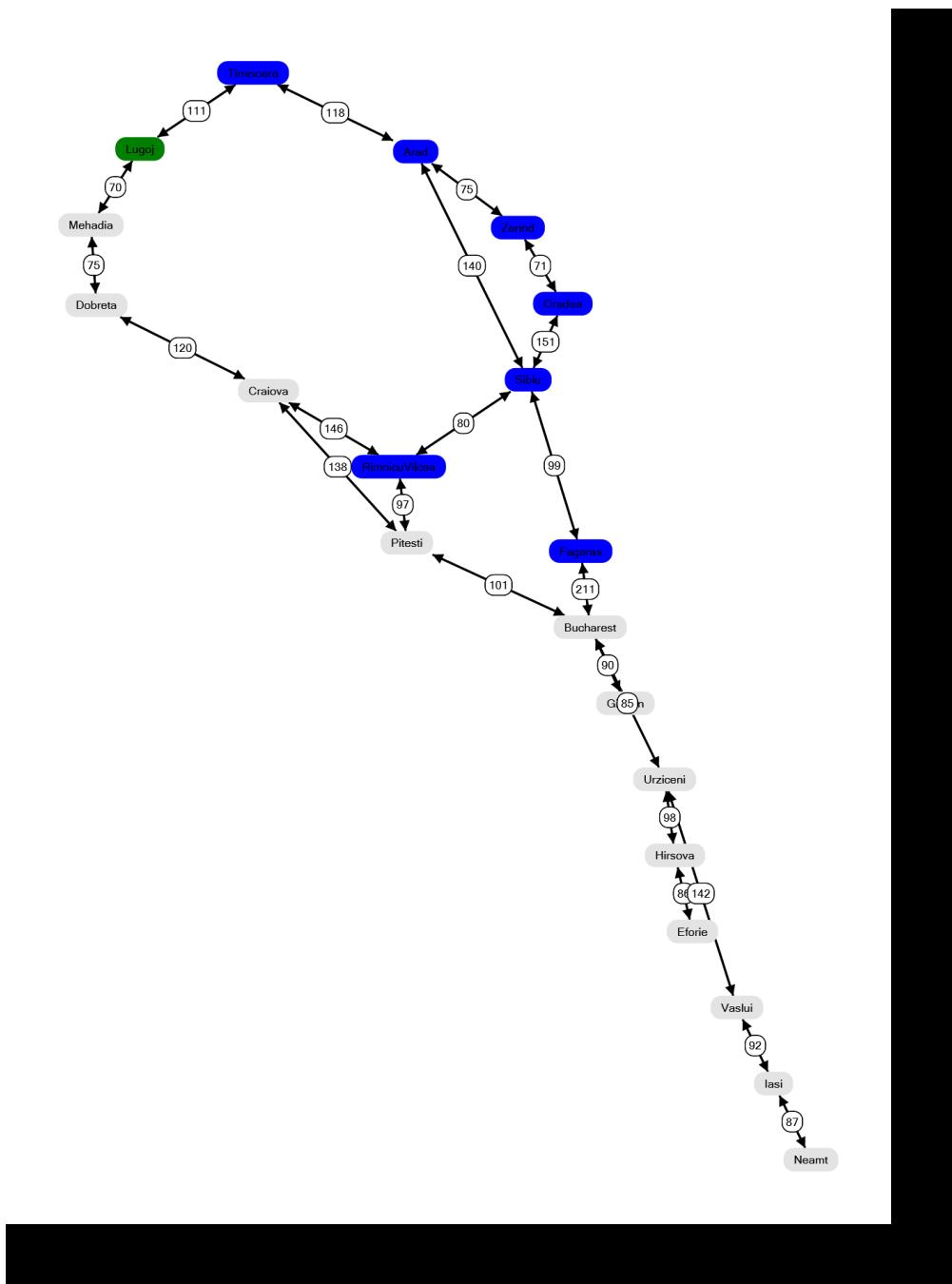
Pre-Visit

The algorithm has opened the node Lugoj



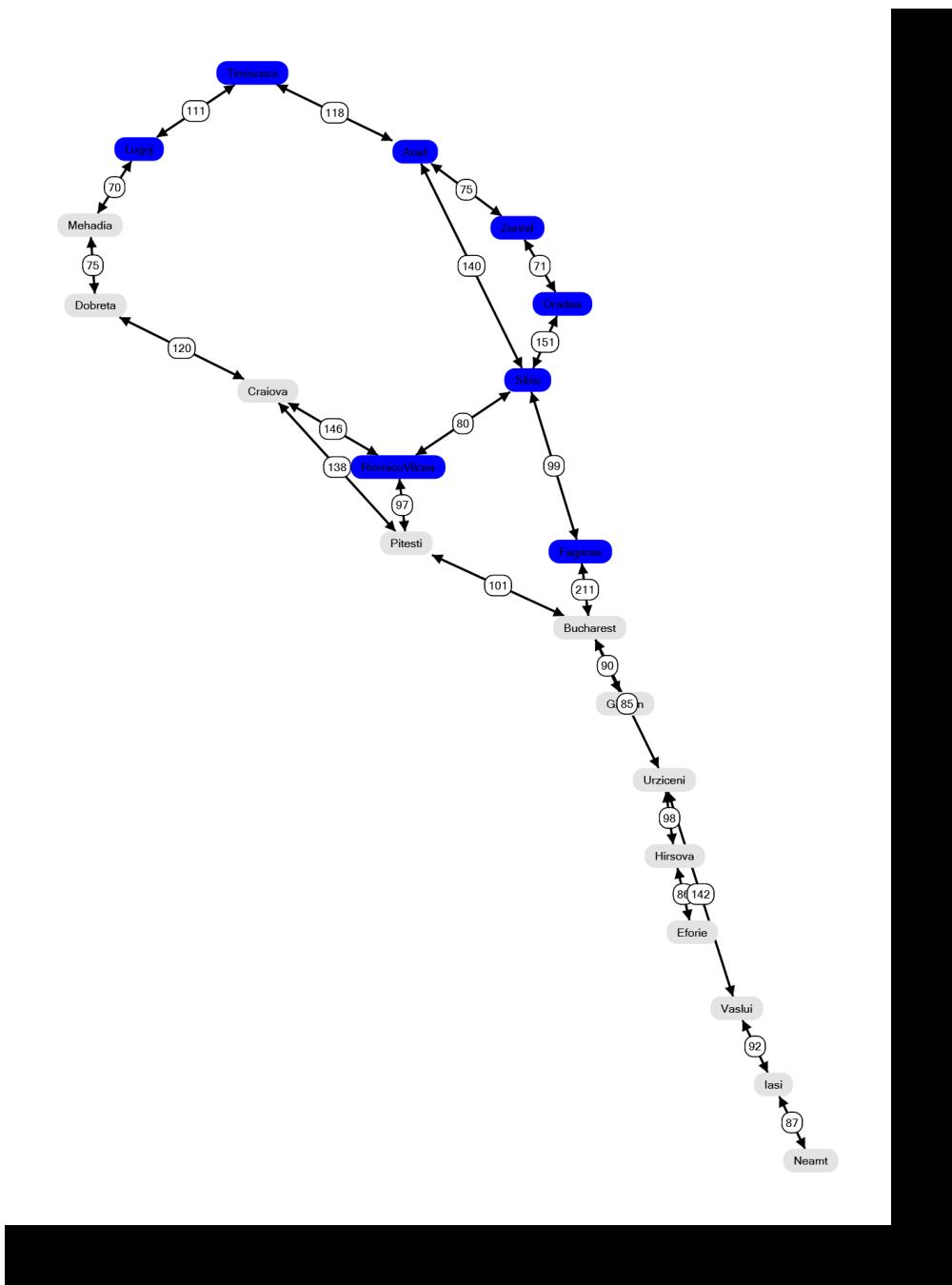
Visit

The algorithm has marked the node Lugoj as visited



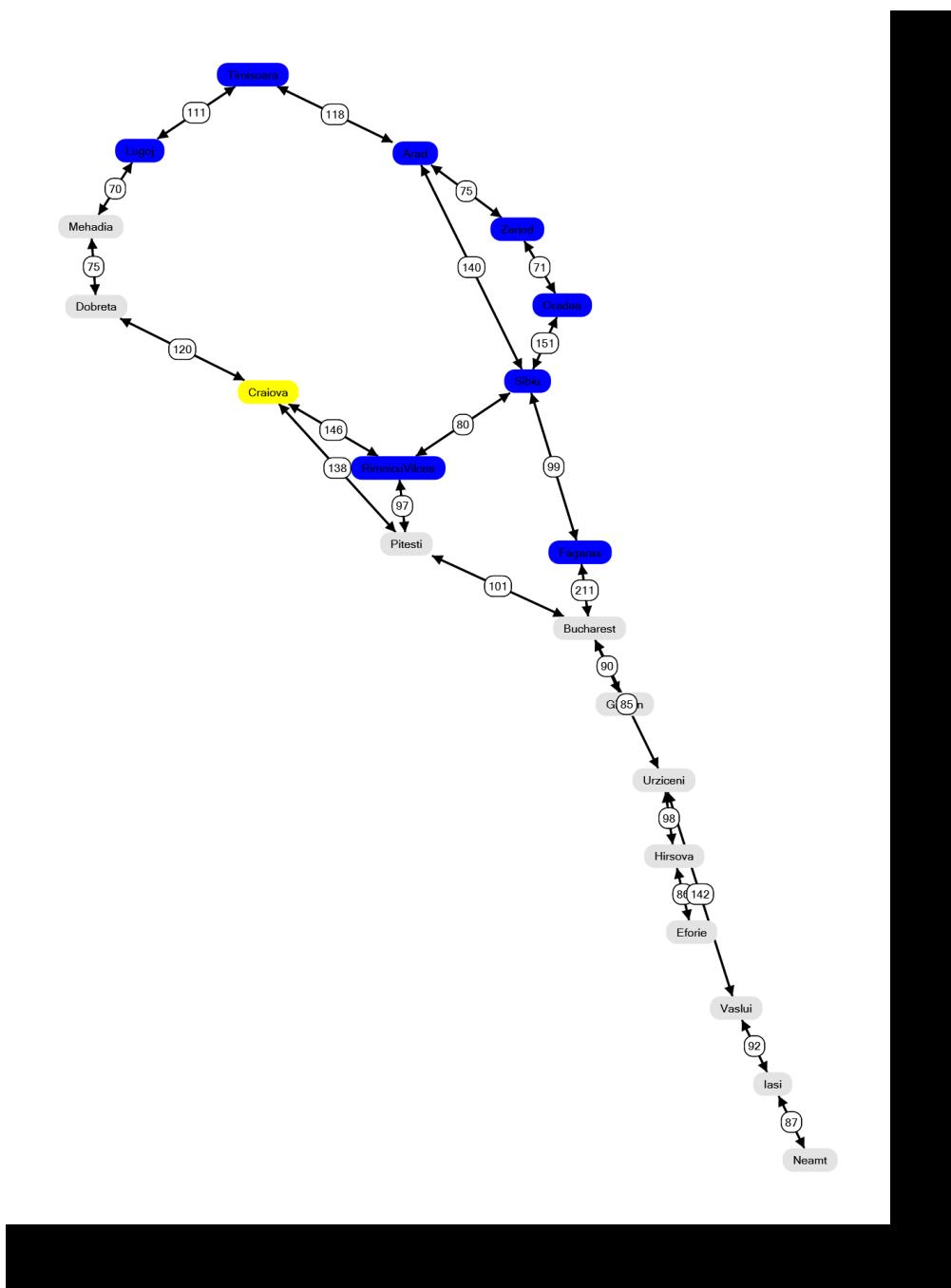
Post-Visit

The algorithm has closed the node Lugoj



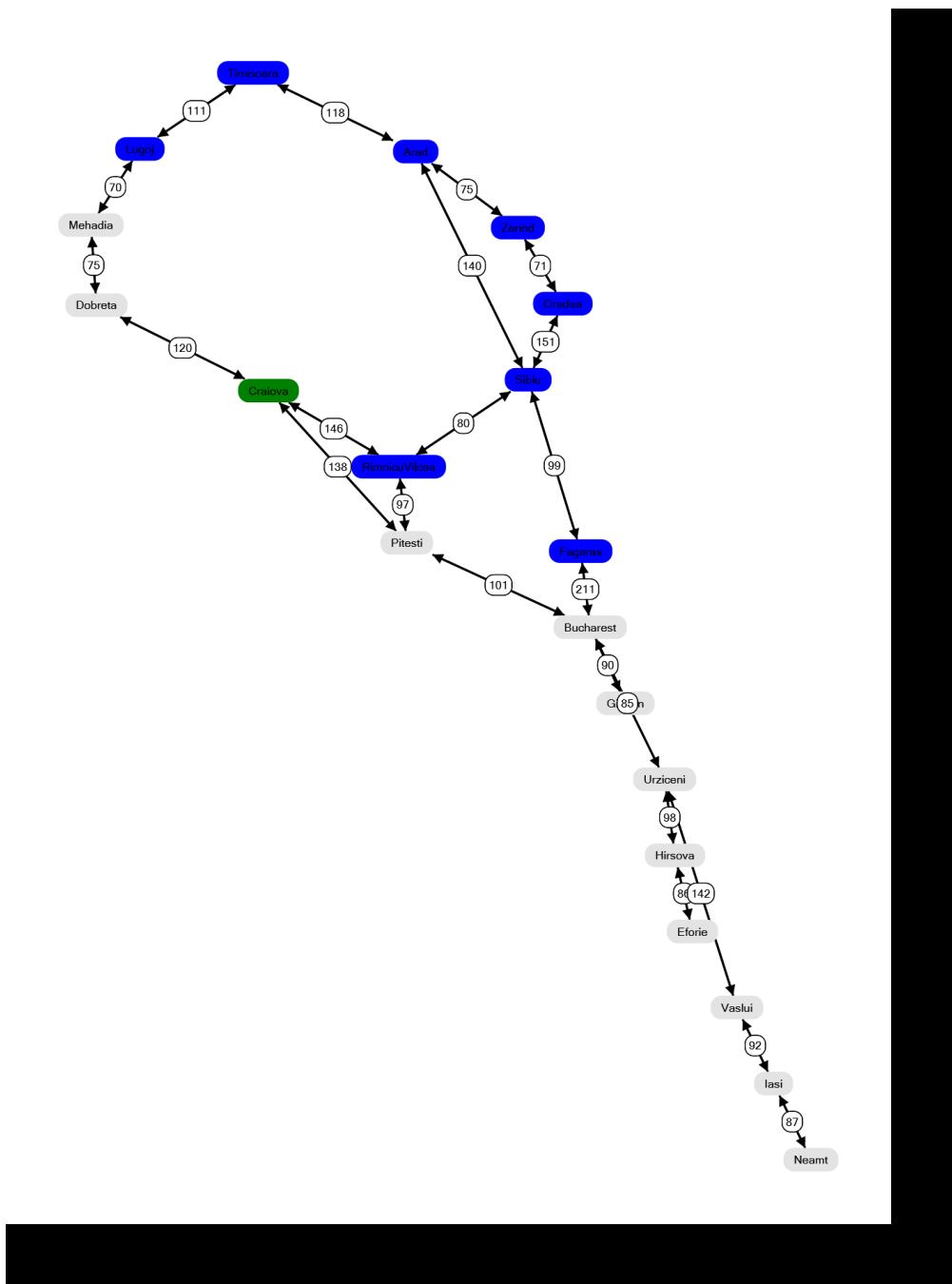
Pre-Visit

The algorithm has opened the node Craiova



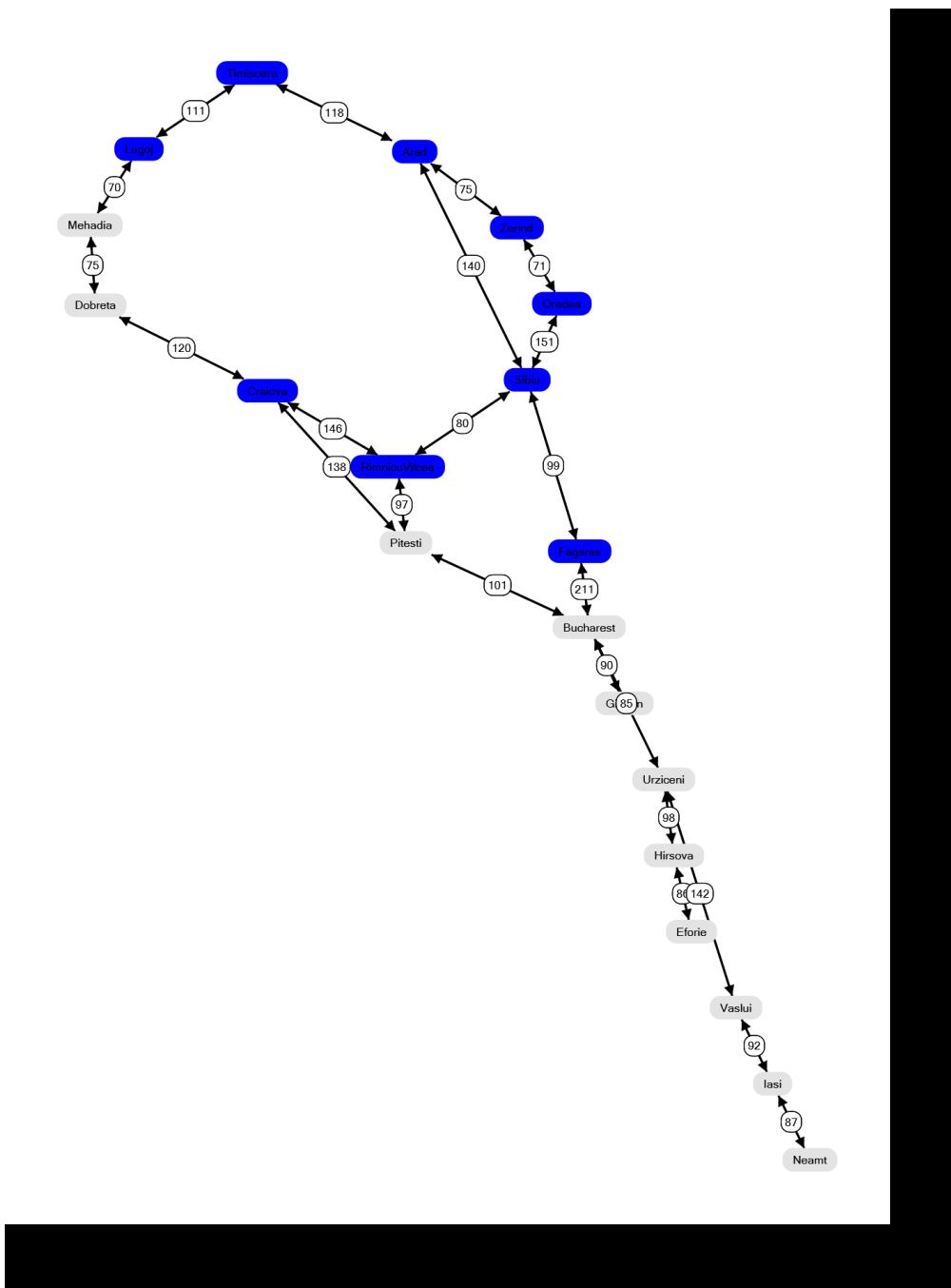
Visit

The algorithm has marked the node Craiova as visited



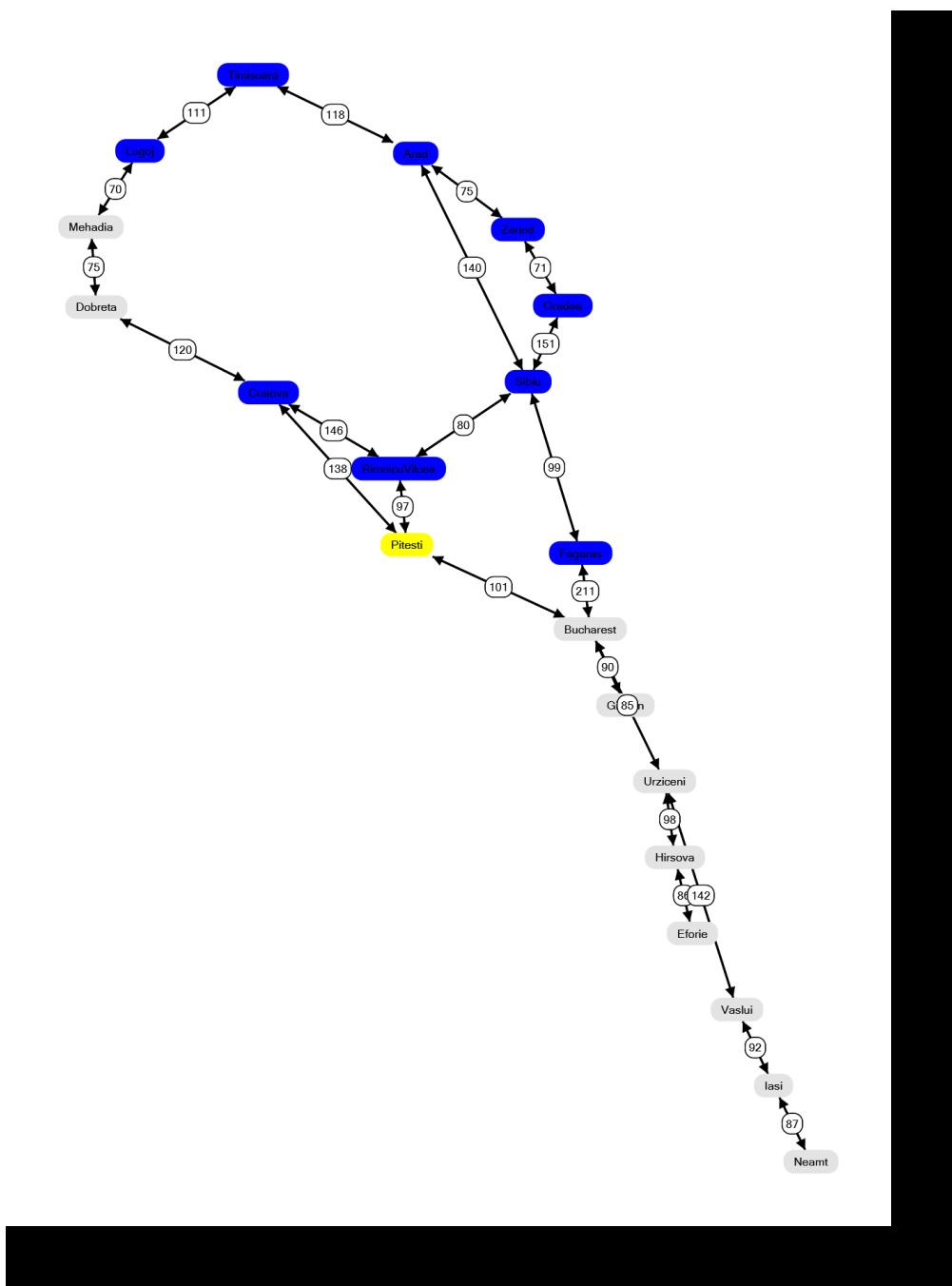
Post-Visit

The algorithm has closed the node Craiova



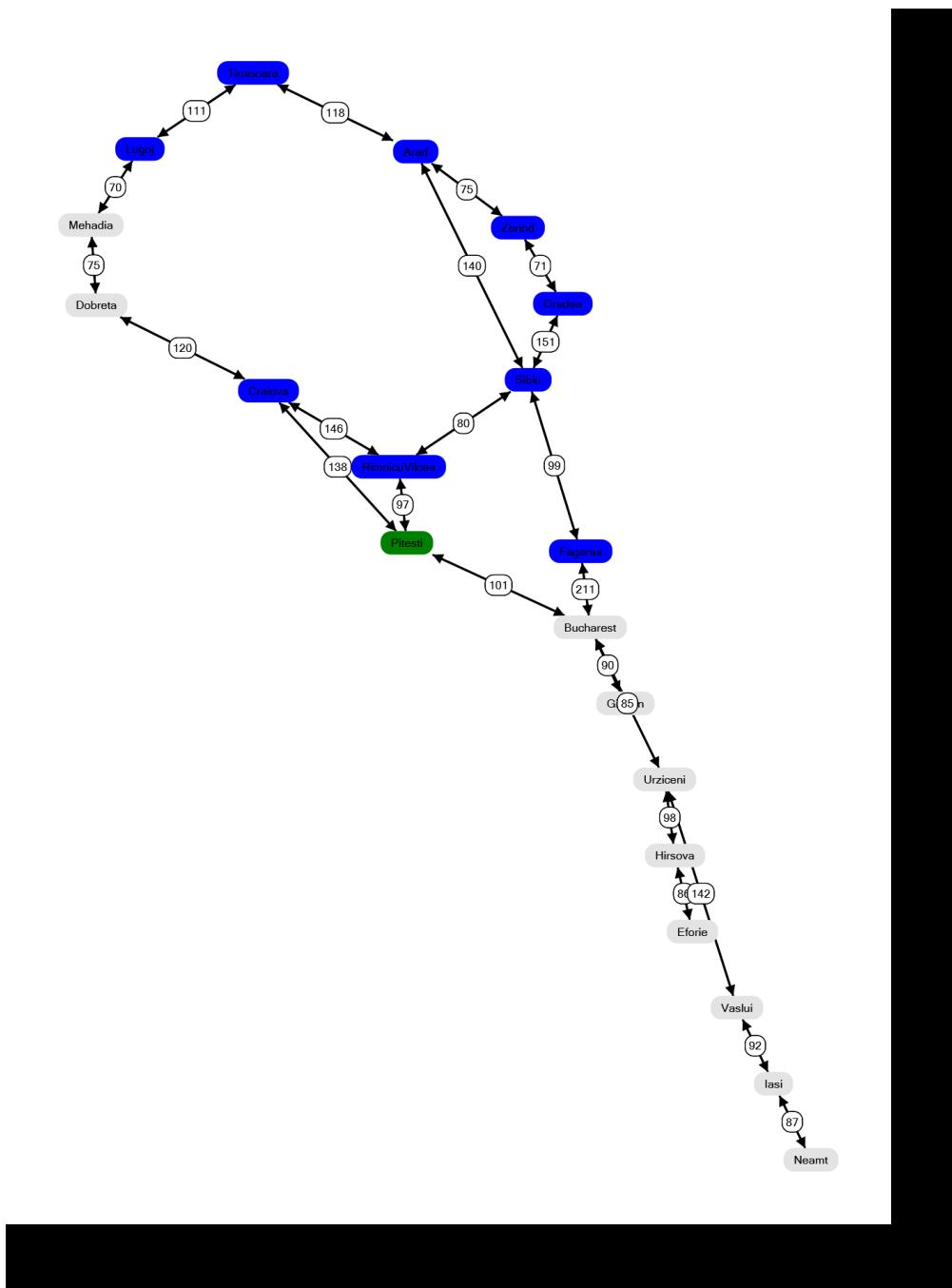
Pre-Visit

The algorithm has opened the node Pitesti



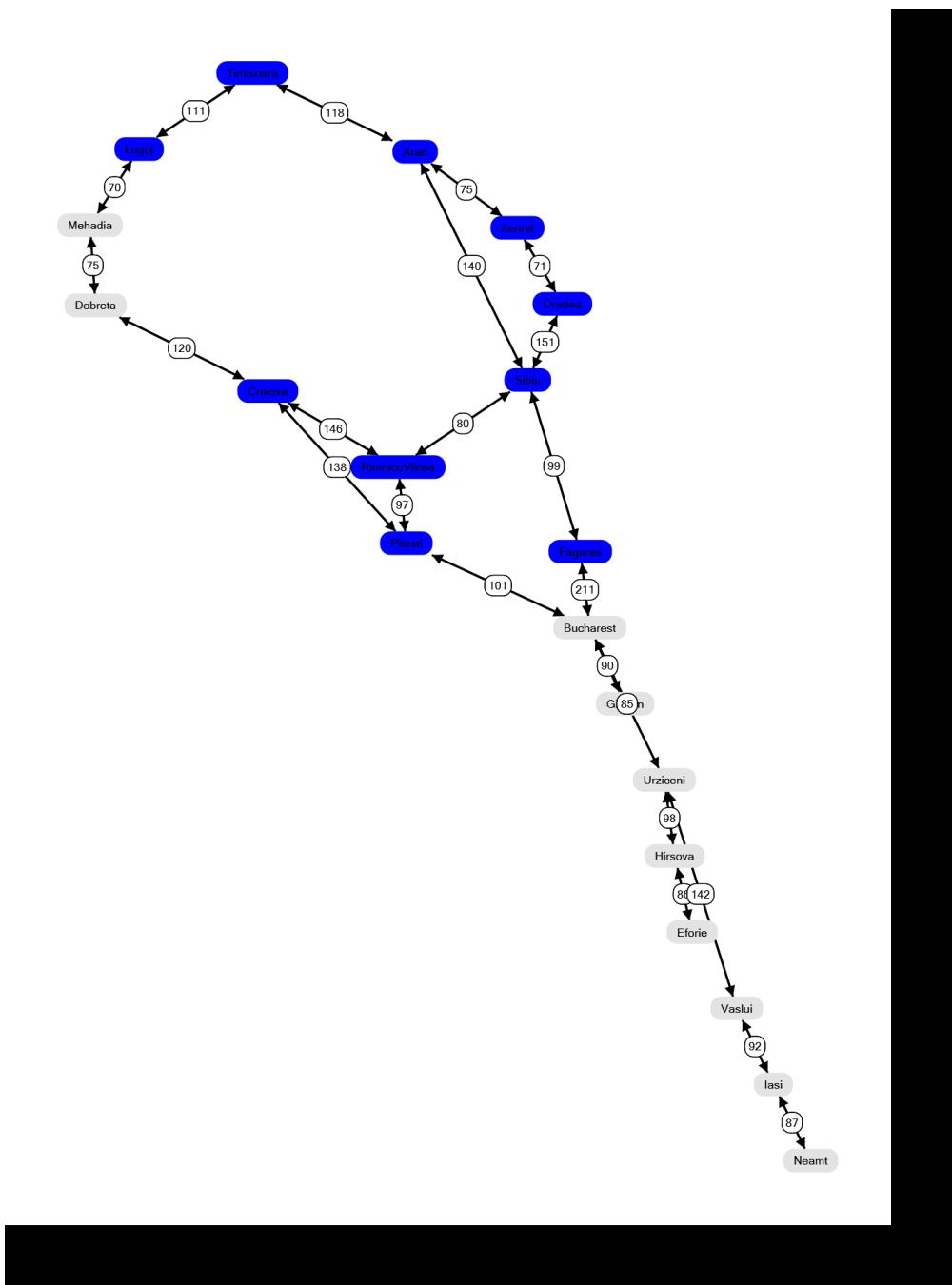
Visit

The algorithm has marked the node Pitesti as visited



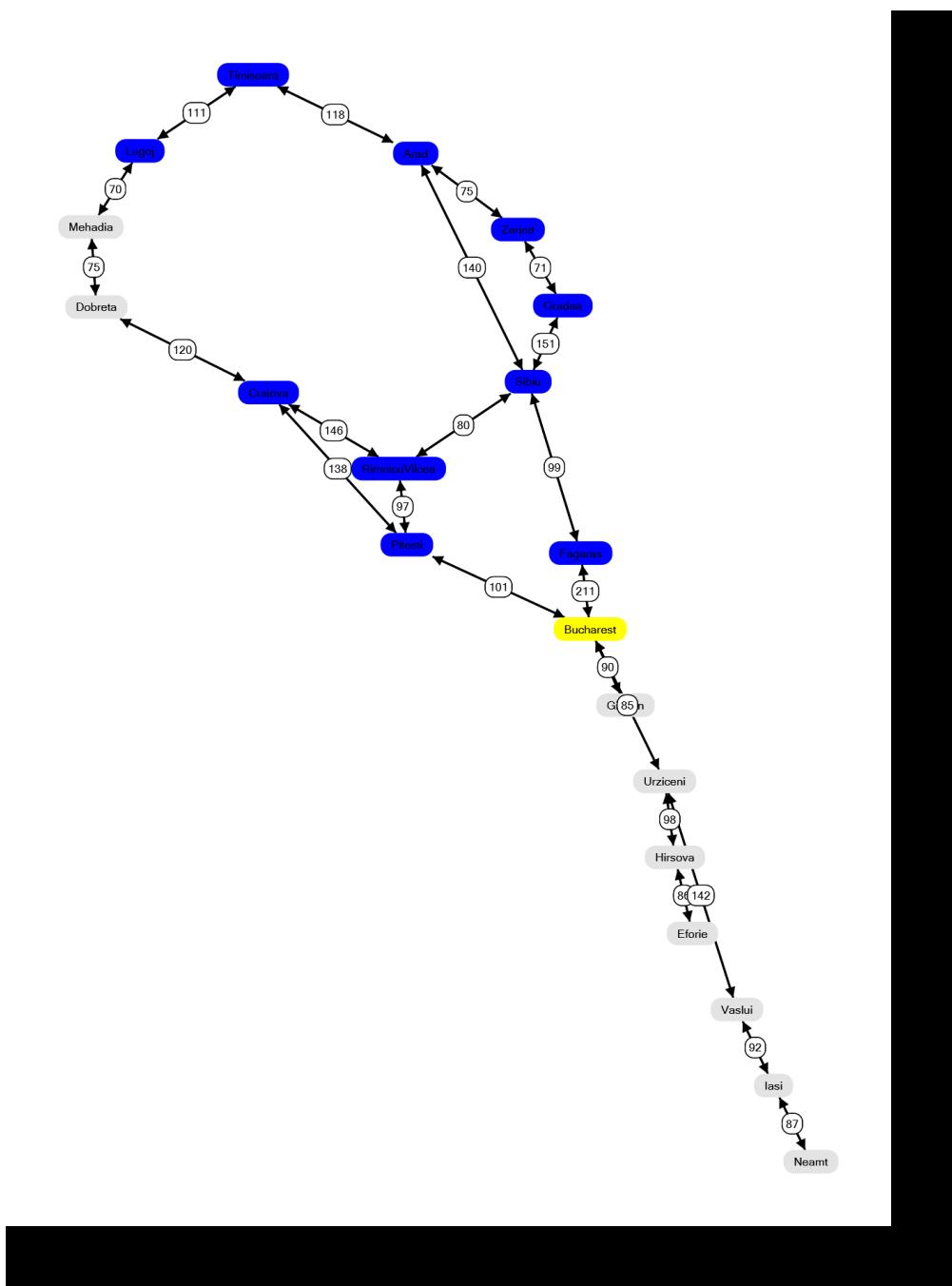
Post-Visit

The algorithm has closed the node Pitesti



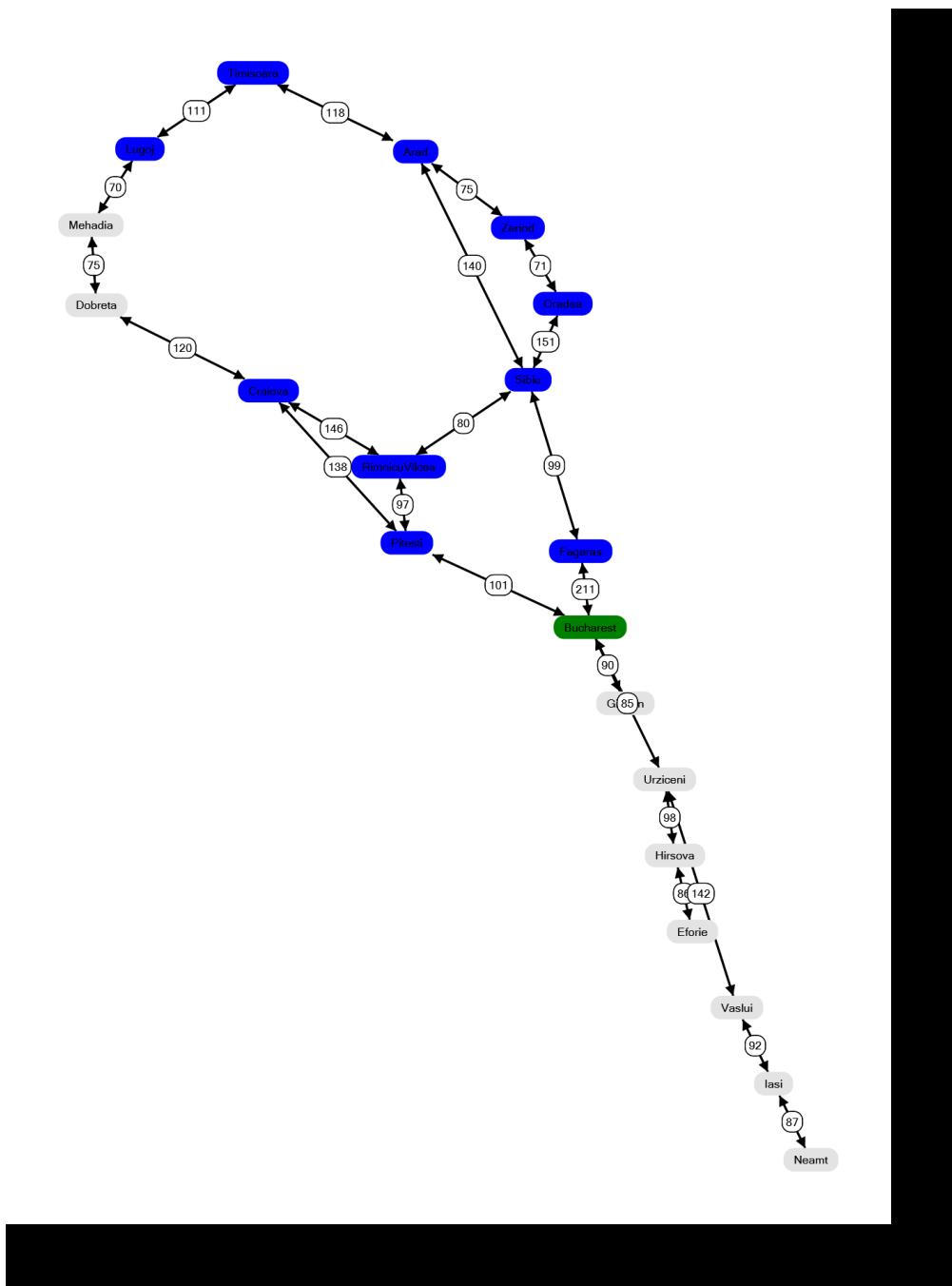
Pre-Visit

The algorithm has opened the node Bucharest



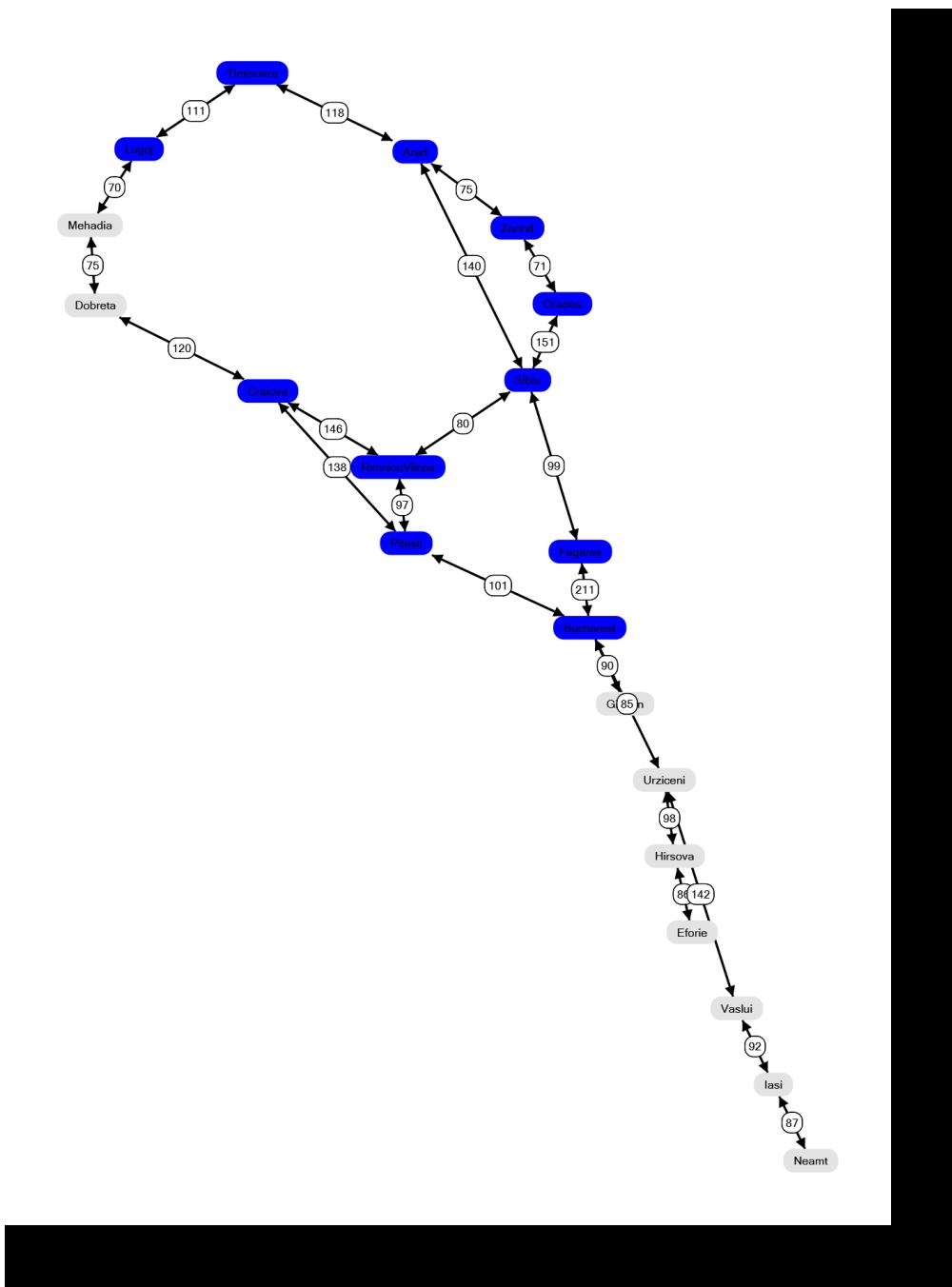
Visit

The algorithm has marked the node Bucharest as visited



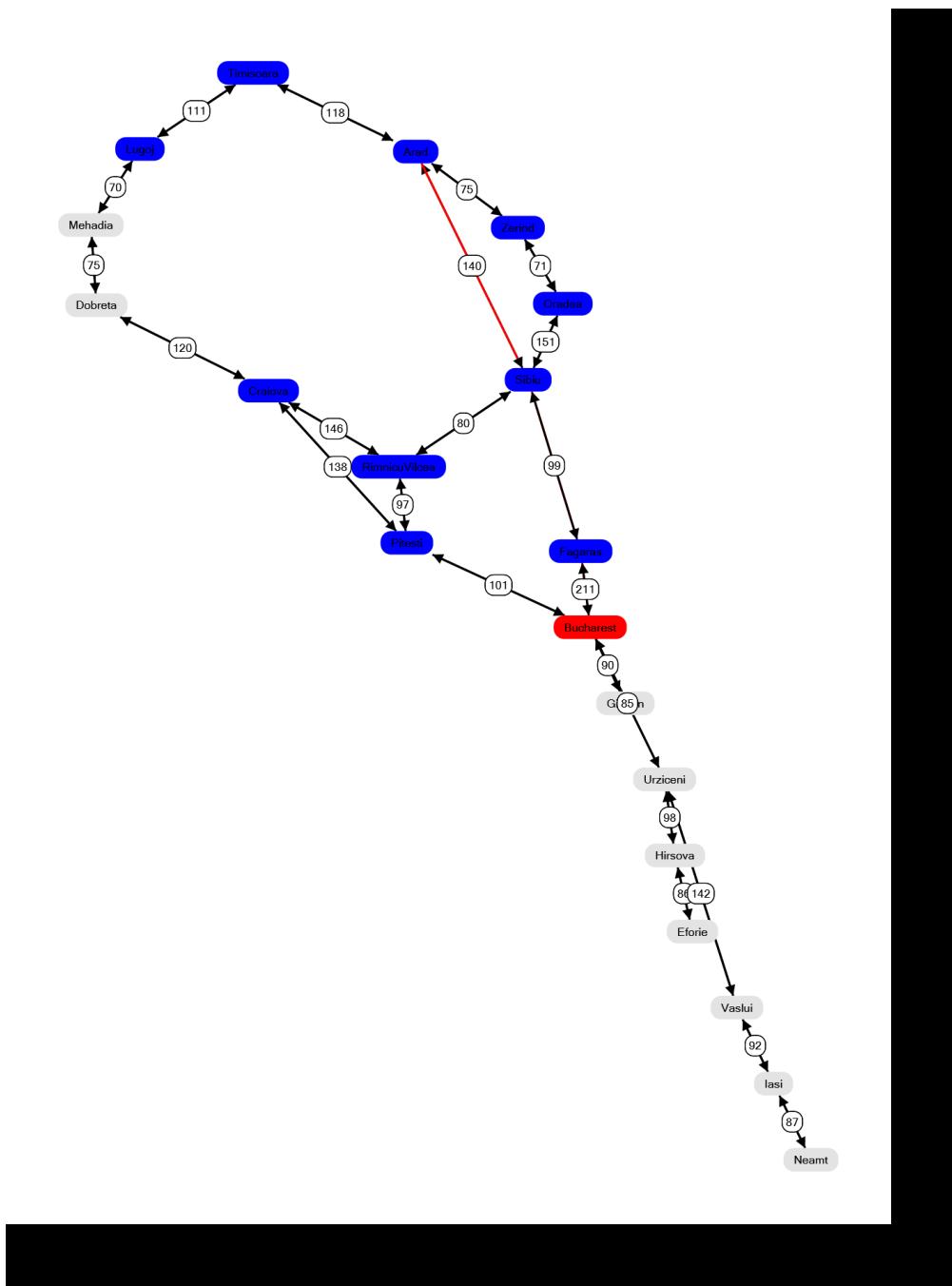
Post-Visit

The algorithm has closed the node Bucharest



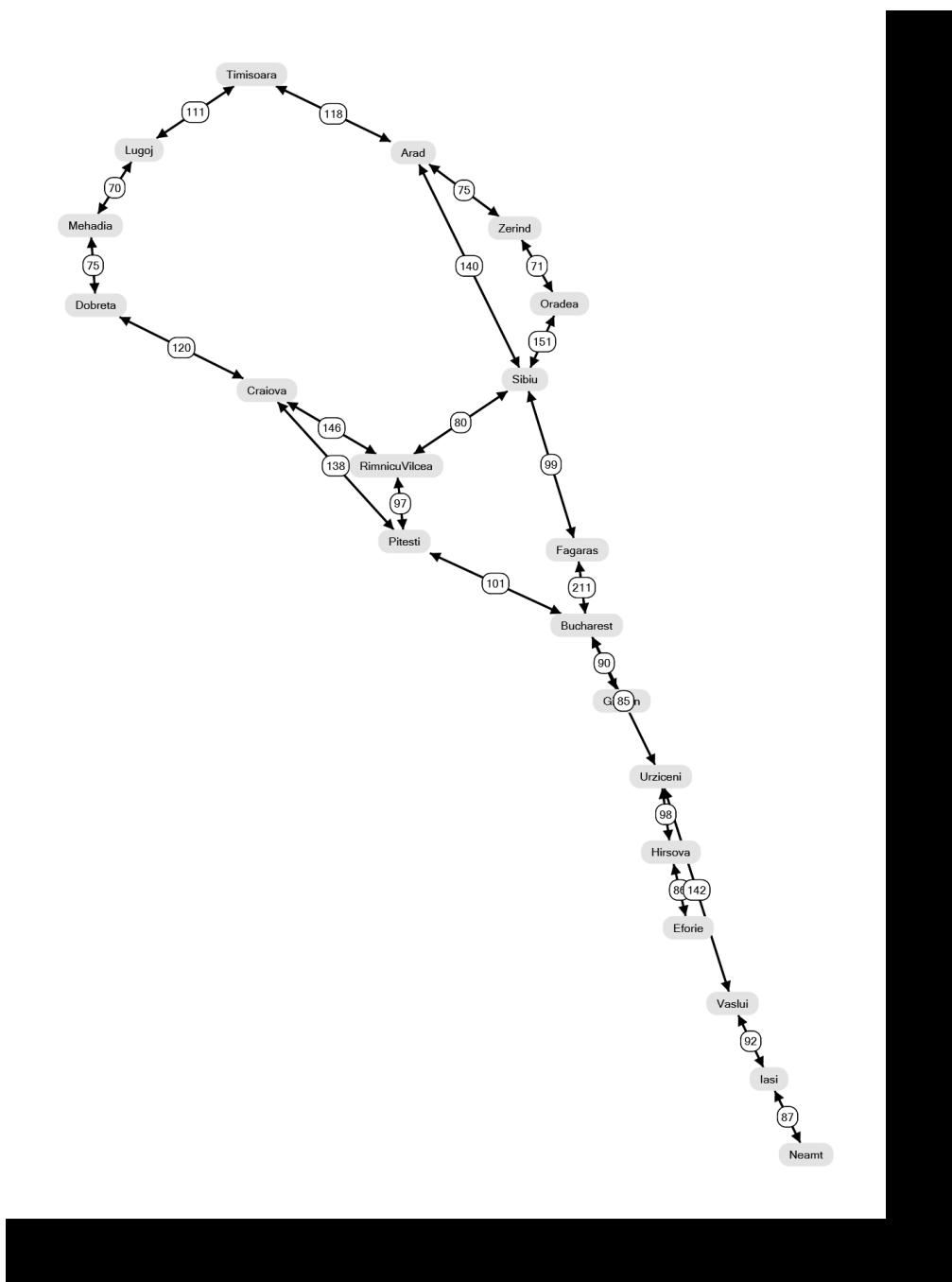
Result Found

The algorithm has found the target node Bucharest



Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



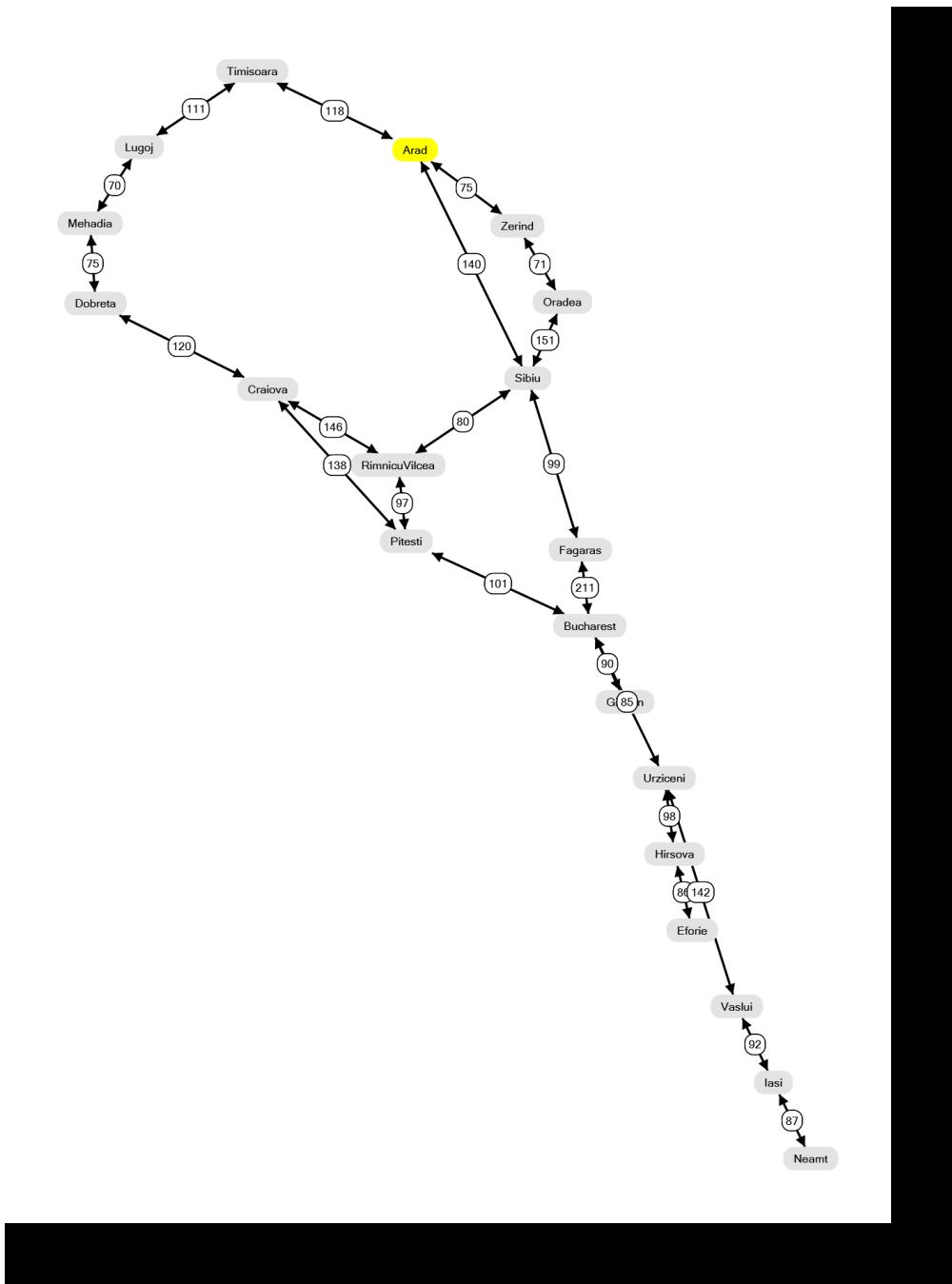
Depth-first search

Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. One starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking.

The algorithm took 00:00:00.0009414 to find the goal.

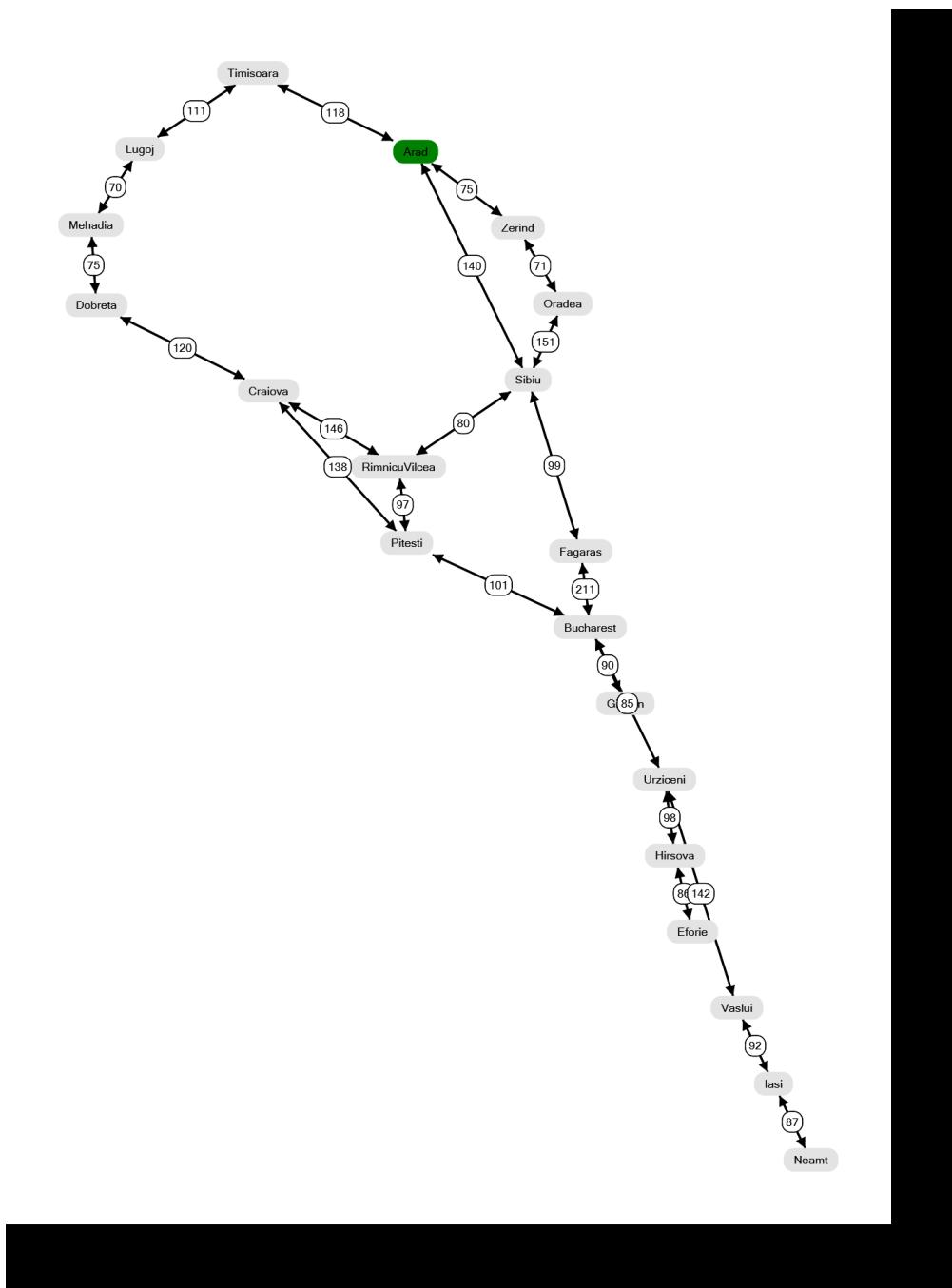
Pre-Visit

The algorithm has opened the node Arad



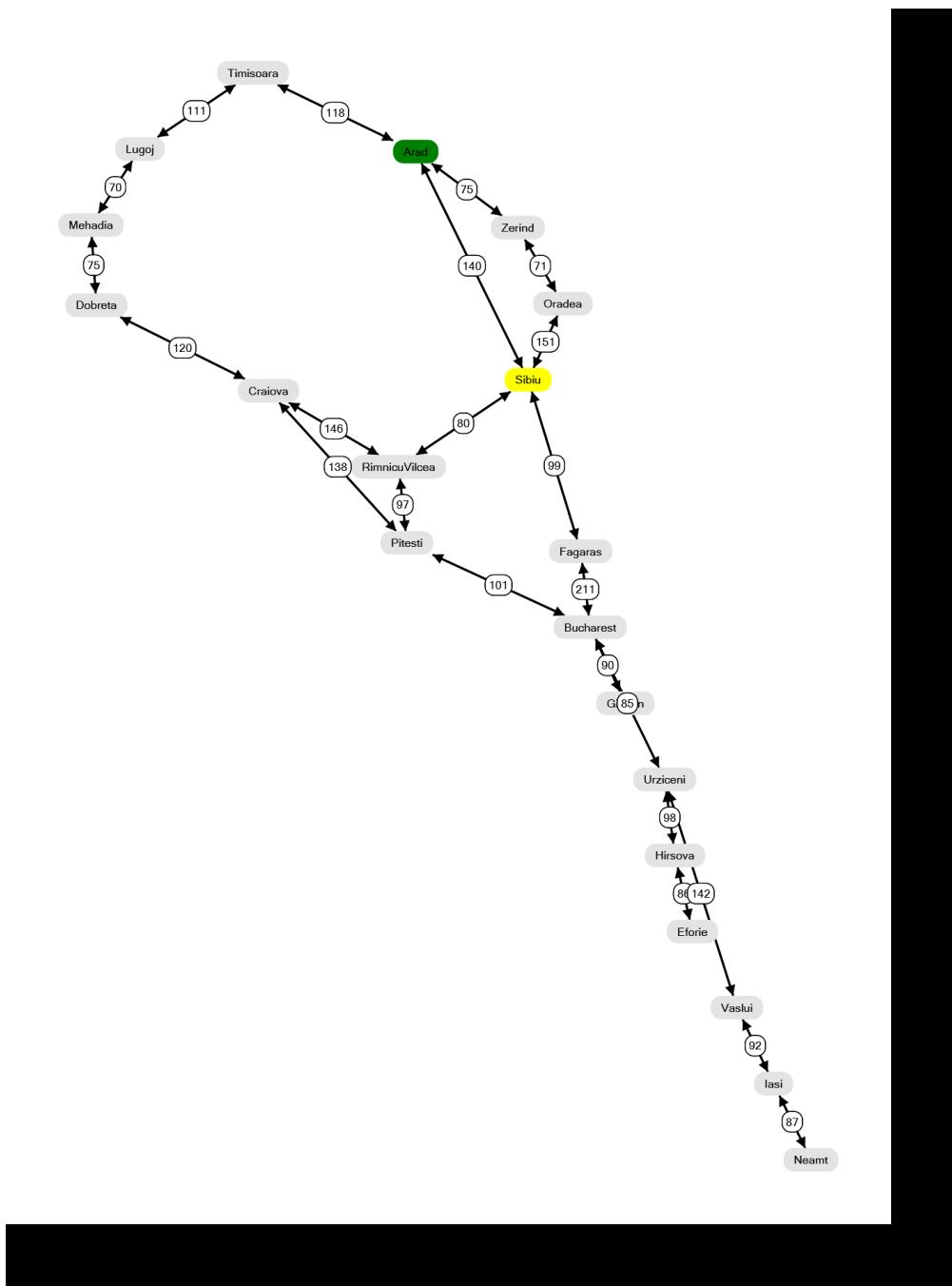
Visit

The algorithm has marked the node Arad as visited



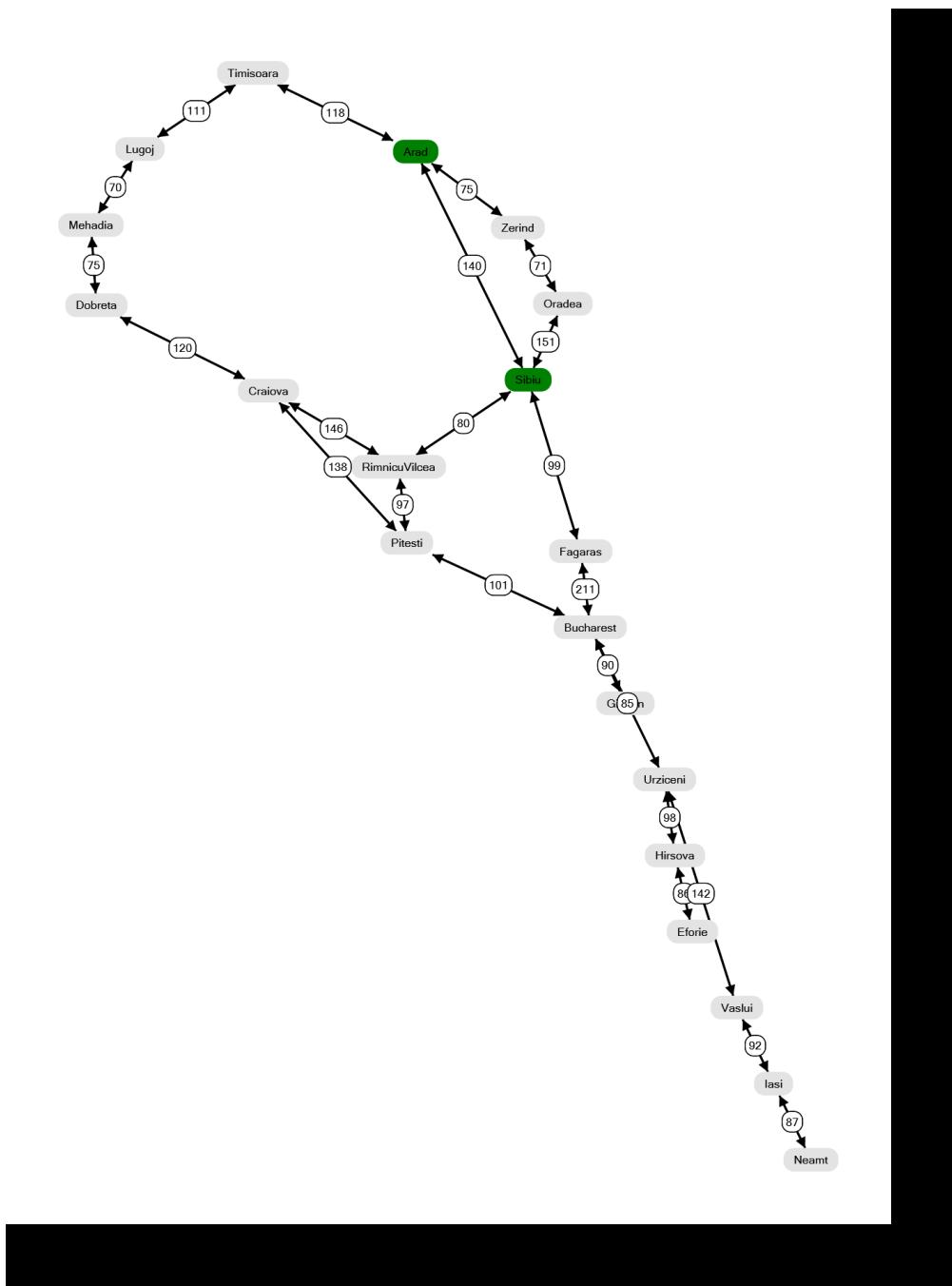
Pre-Visit

The algorithm has opened the node Sibiu



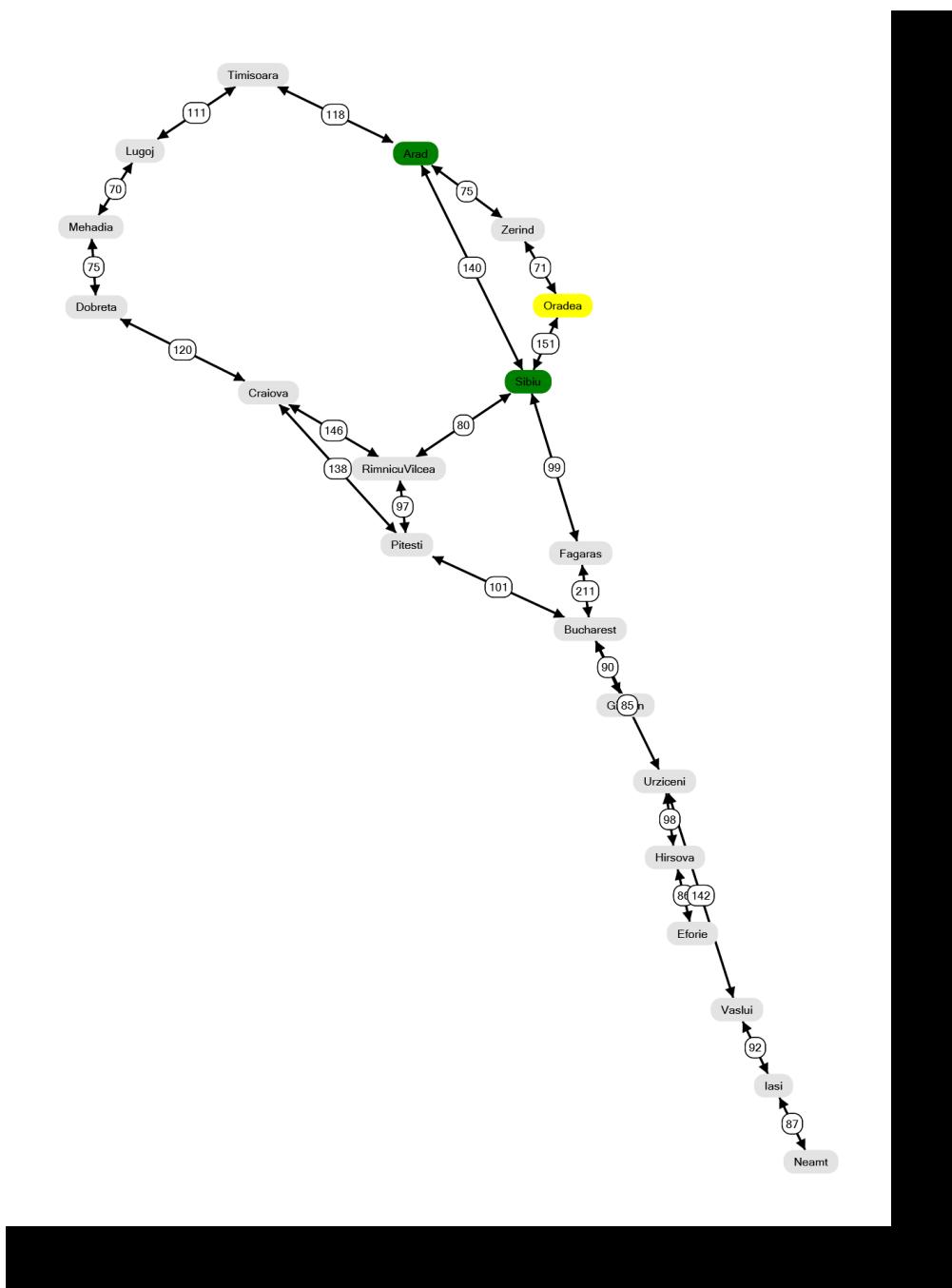
Visit

The algorithm has marked the node Sibiu as visited



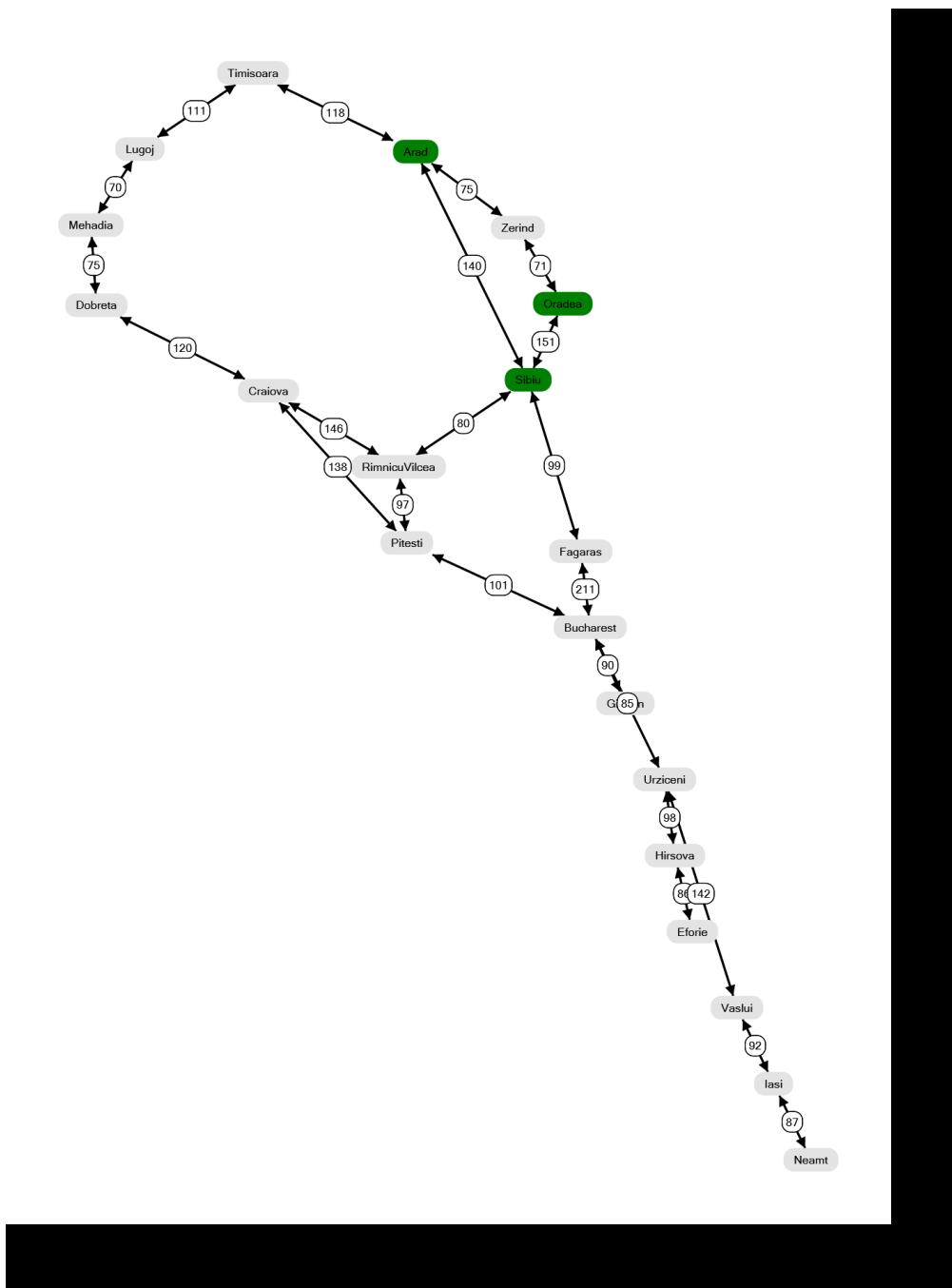
Pre-Visit

The algorithm has opened the node Oradea



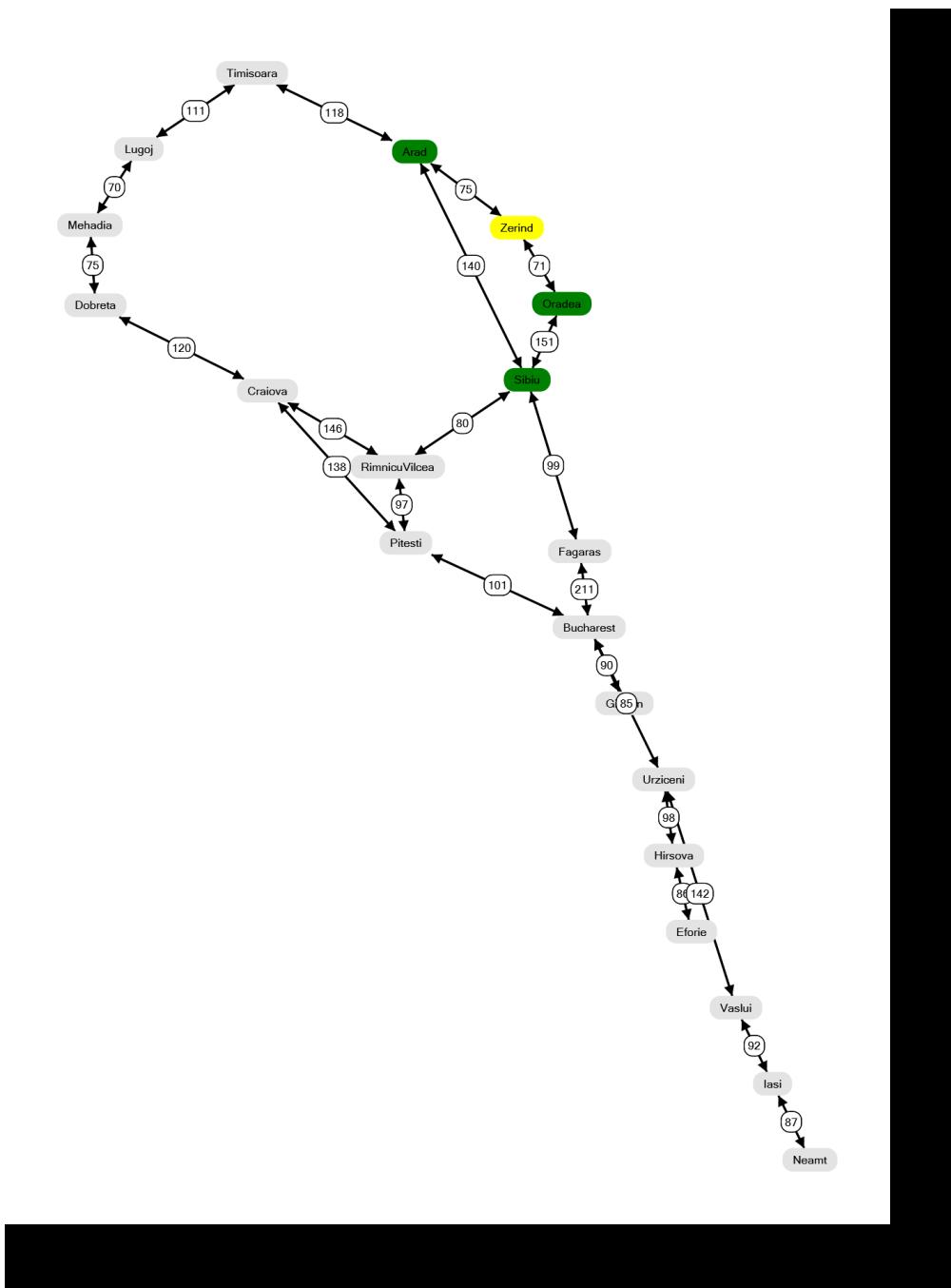
Visit

The algorithm has marked the node Oradea as visited



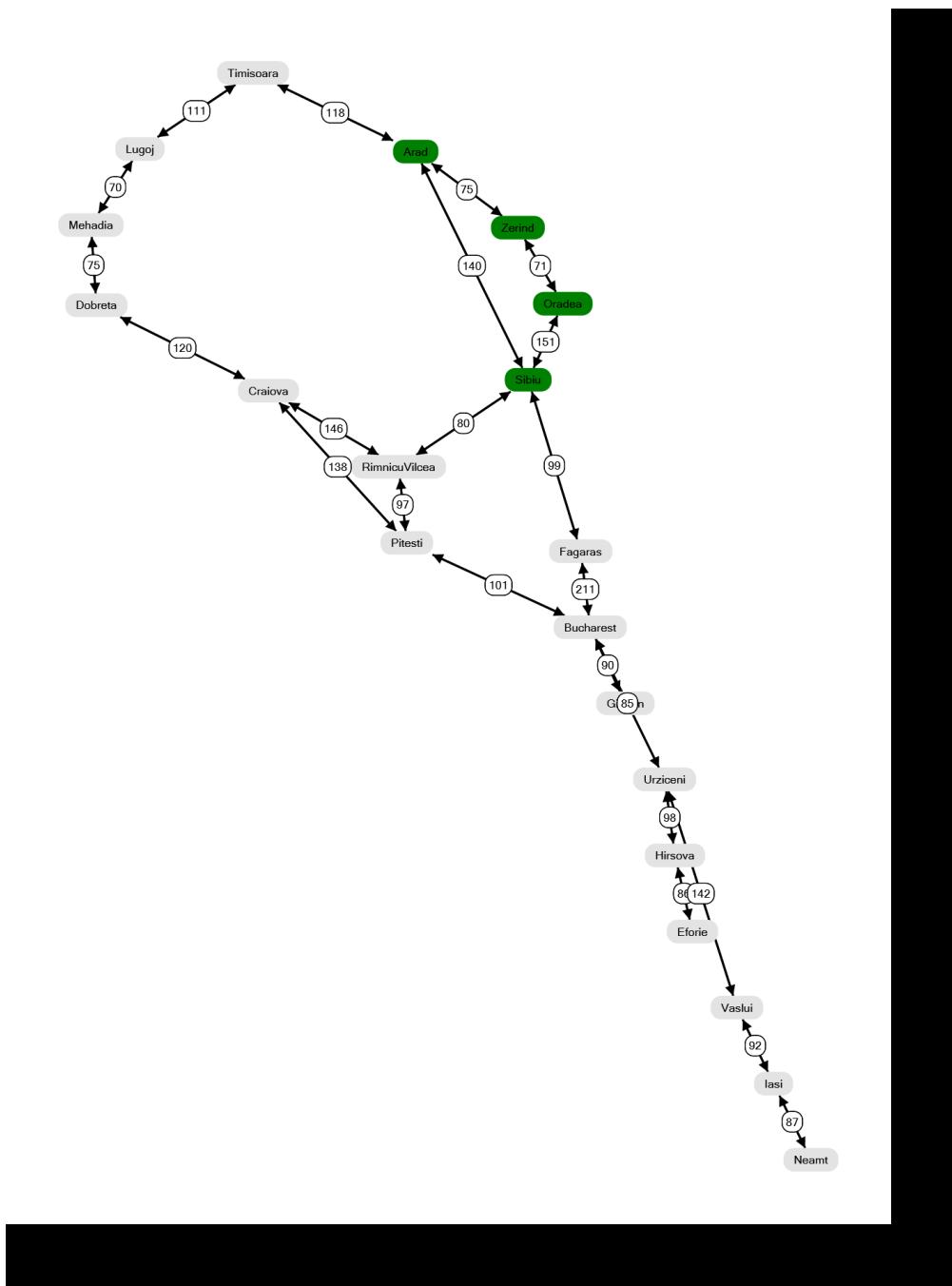
Pre-Visit

The algorithm has opened the node Zerind



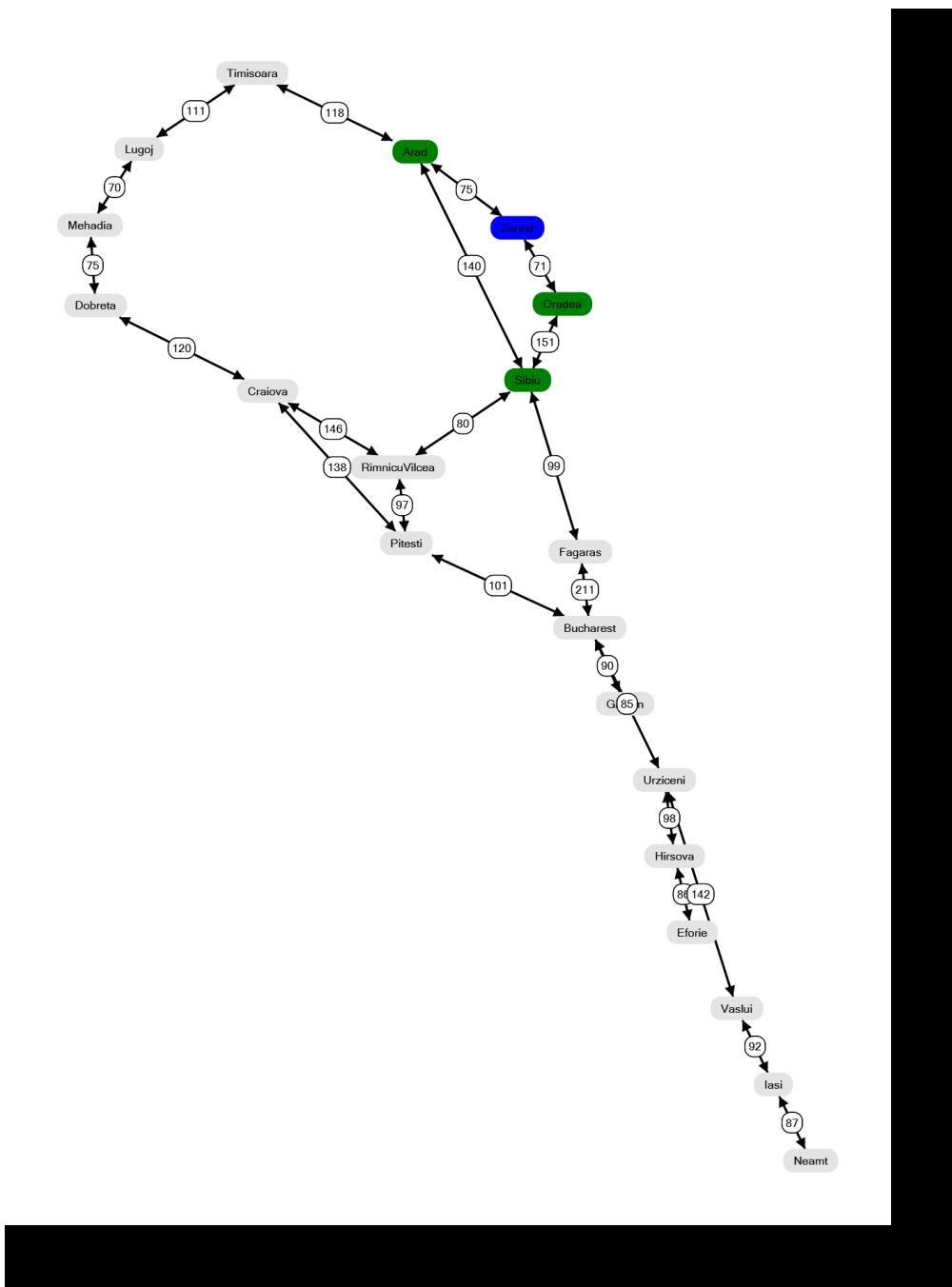
Visit

The algorithm has marked the node Zerind as visited



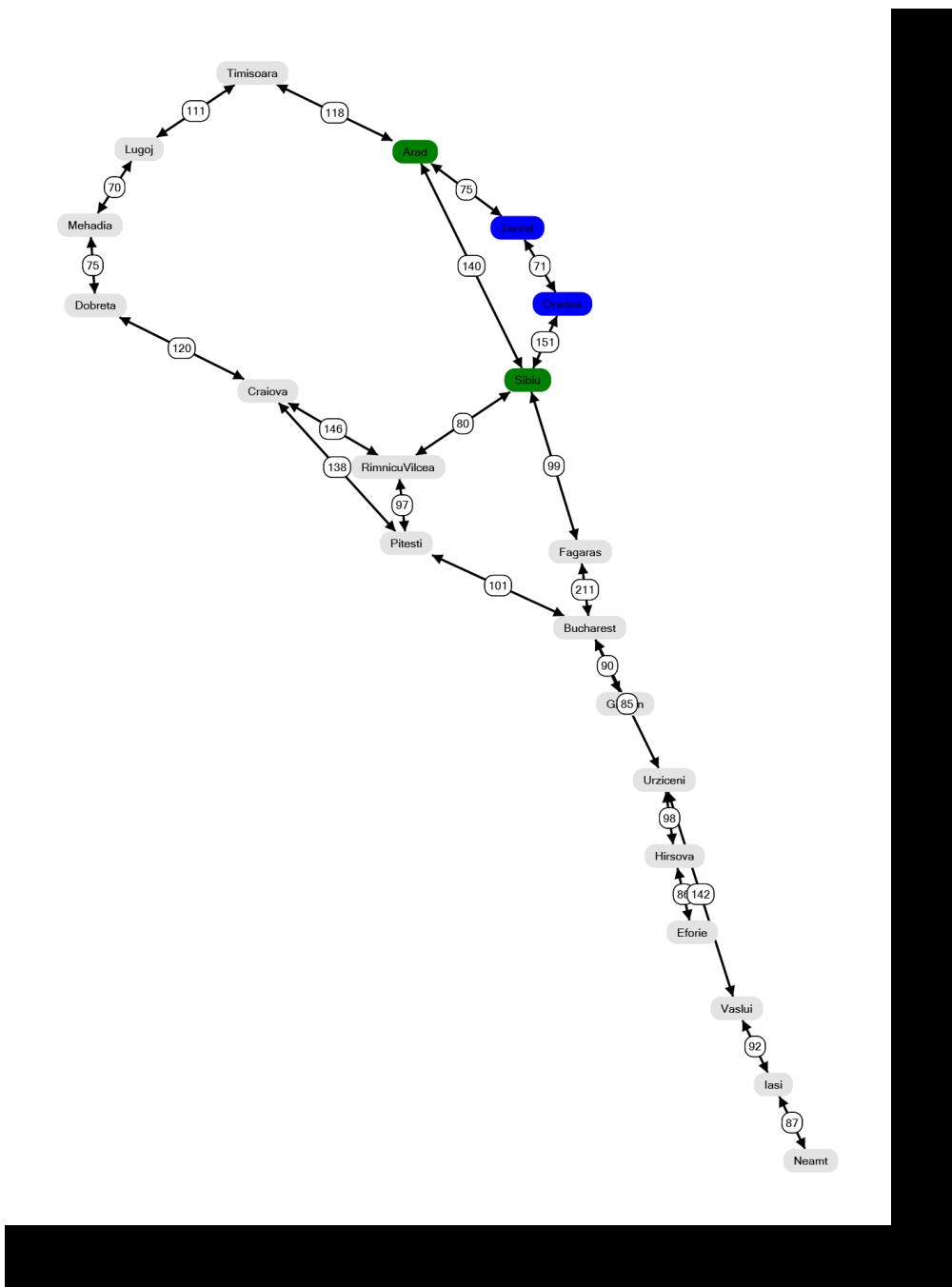
Post-Visit

The algorithm has closed the node Zerind



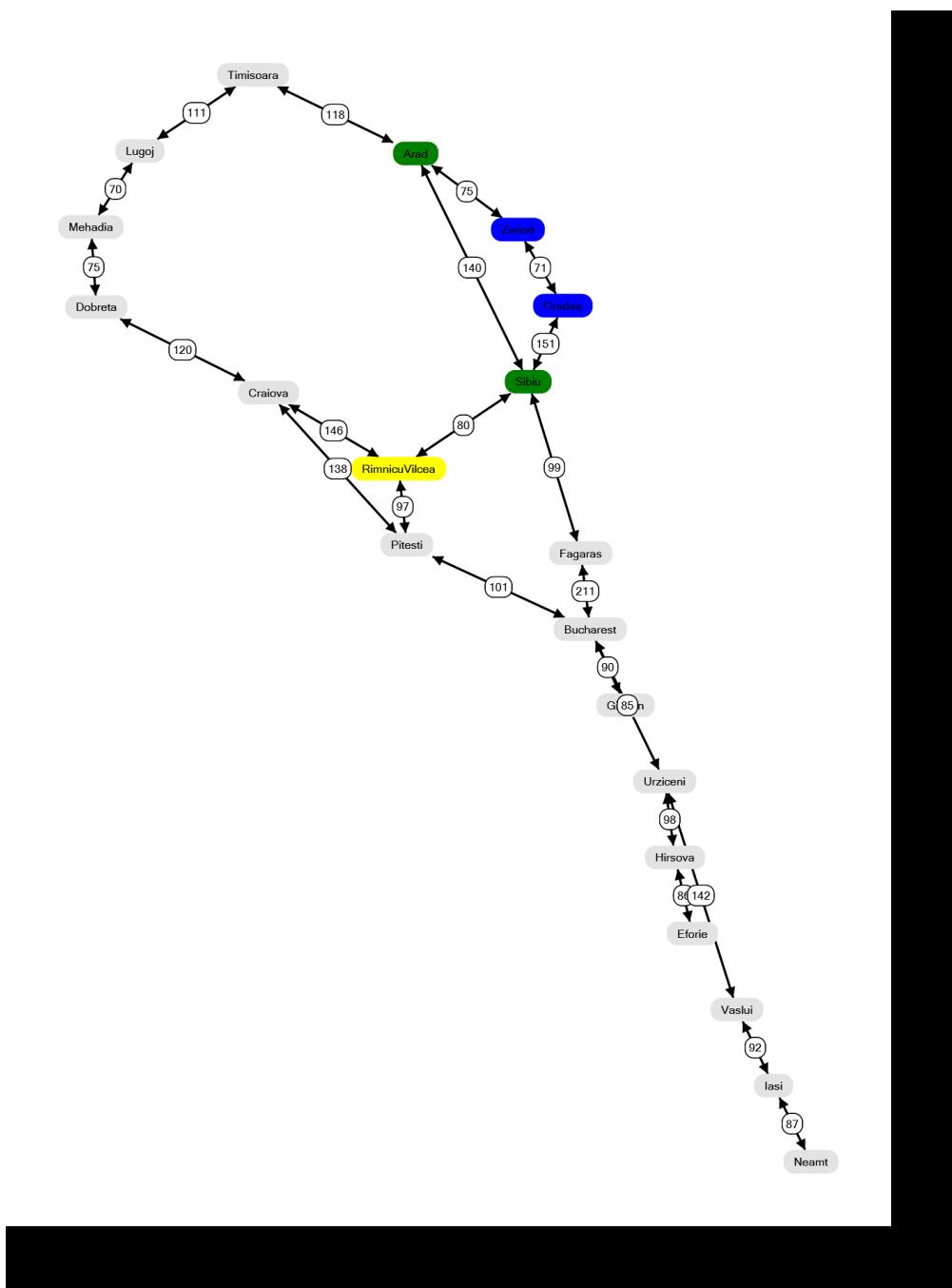
Post-Visit

The algorithm has closed the node Oradea



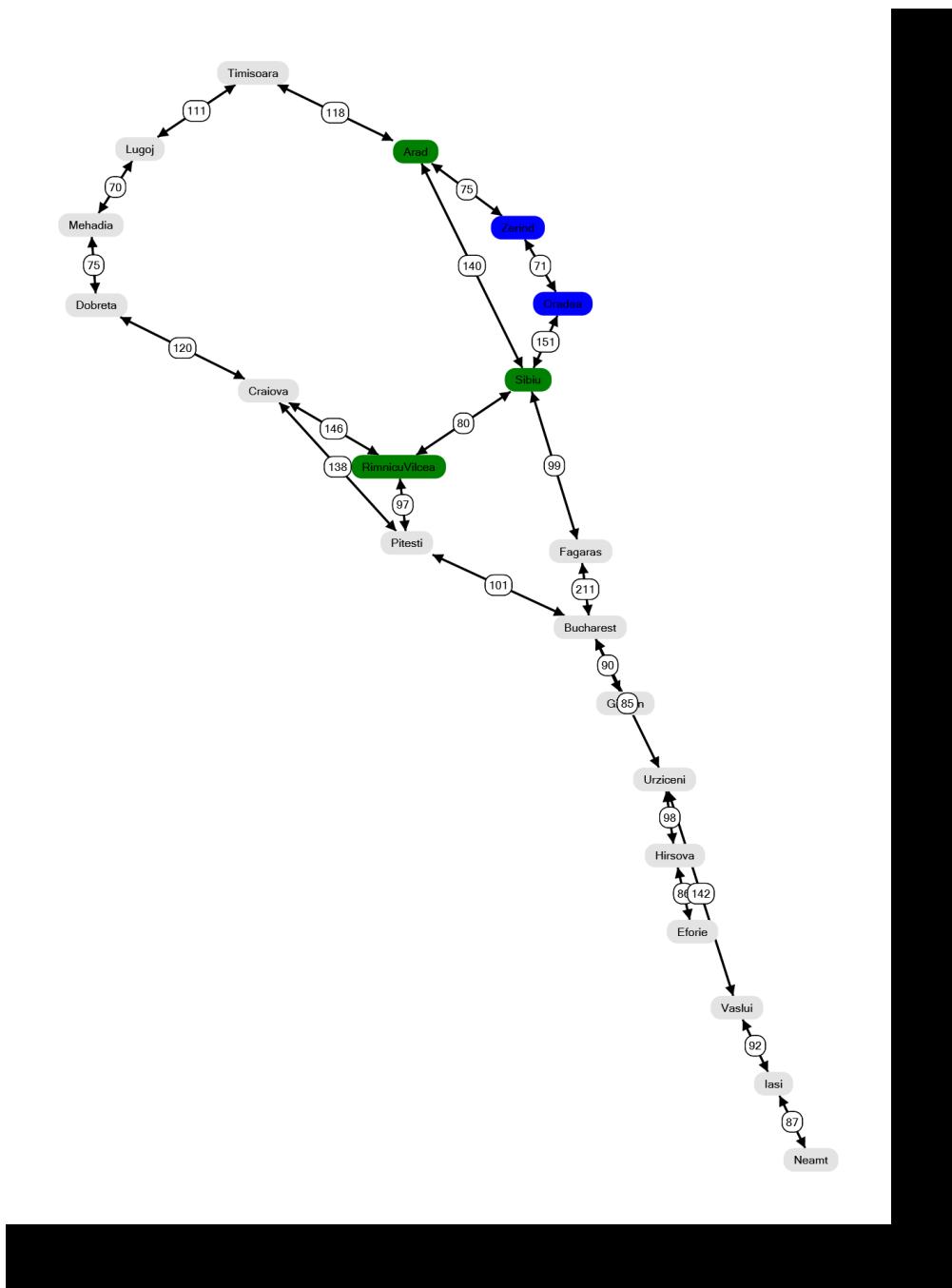
Pre-Visit

The algorithm has opened the node Rimnicu Vilcea



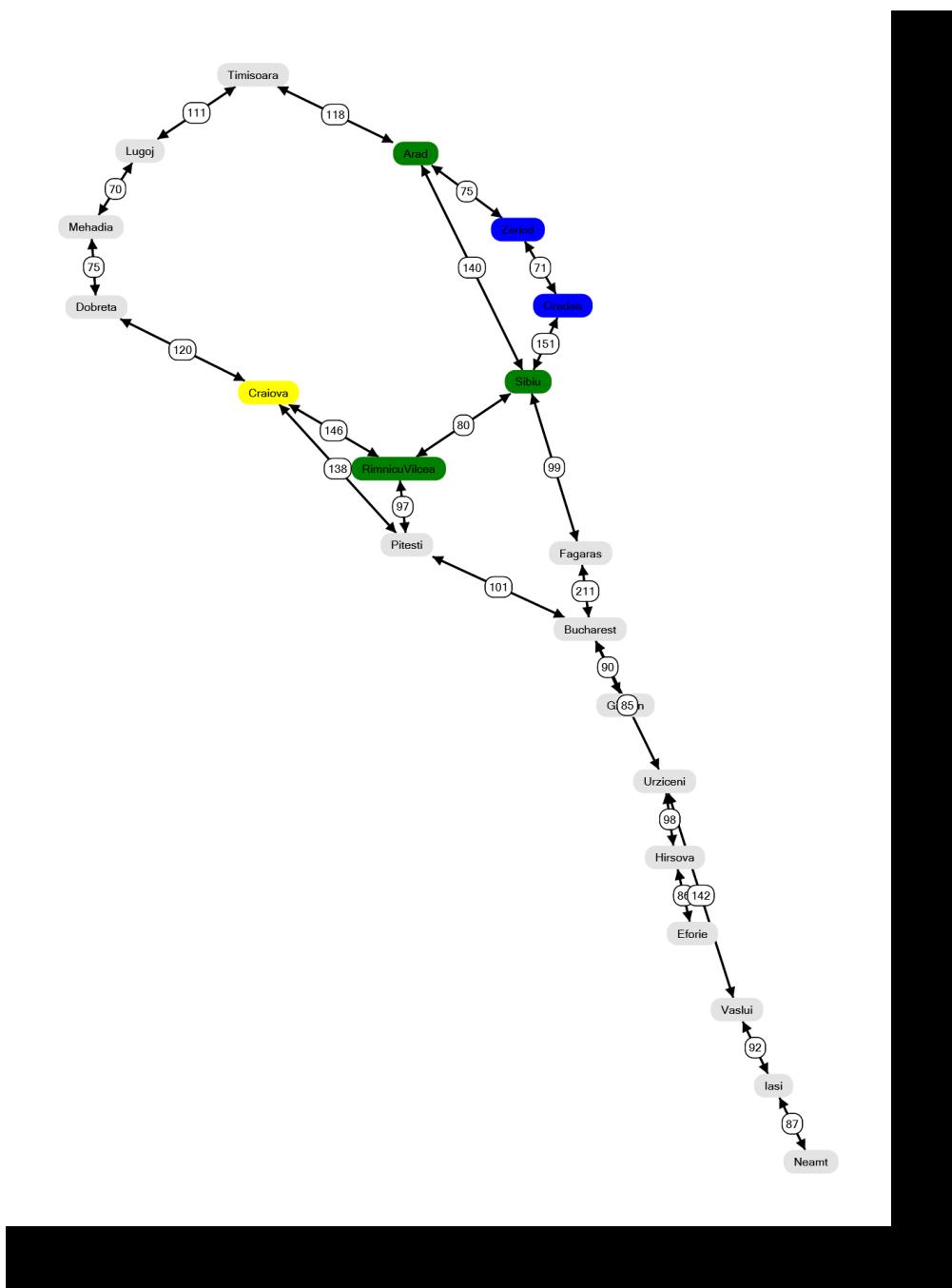
Visit

The algorithm has marked the node Rimnicu Vilcea as visited



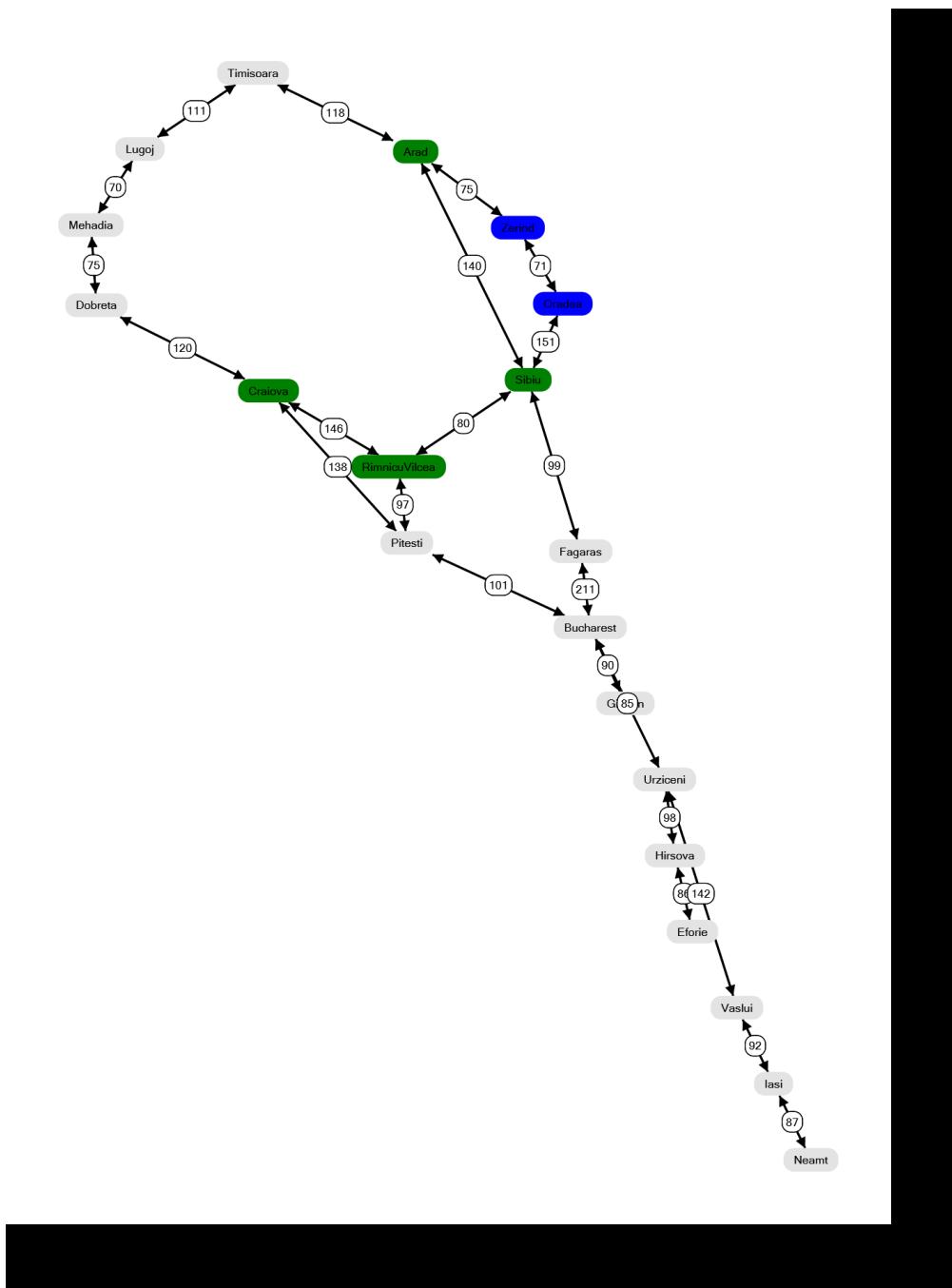
Pre-Visit

The algorithm has opened the node Craiova



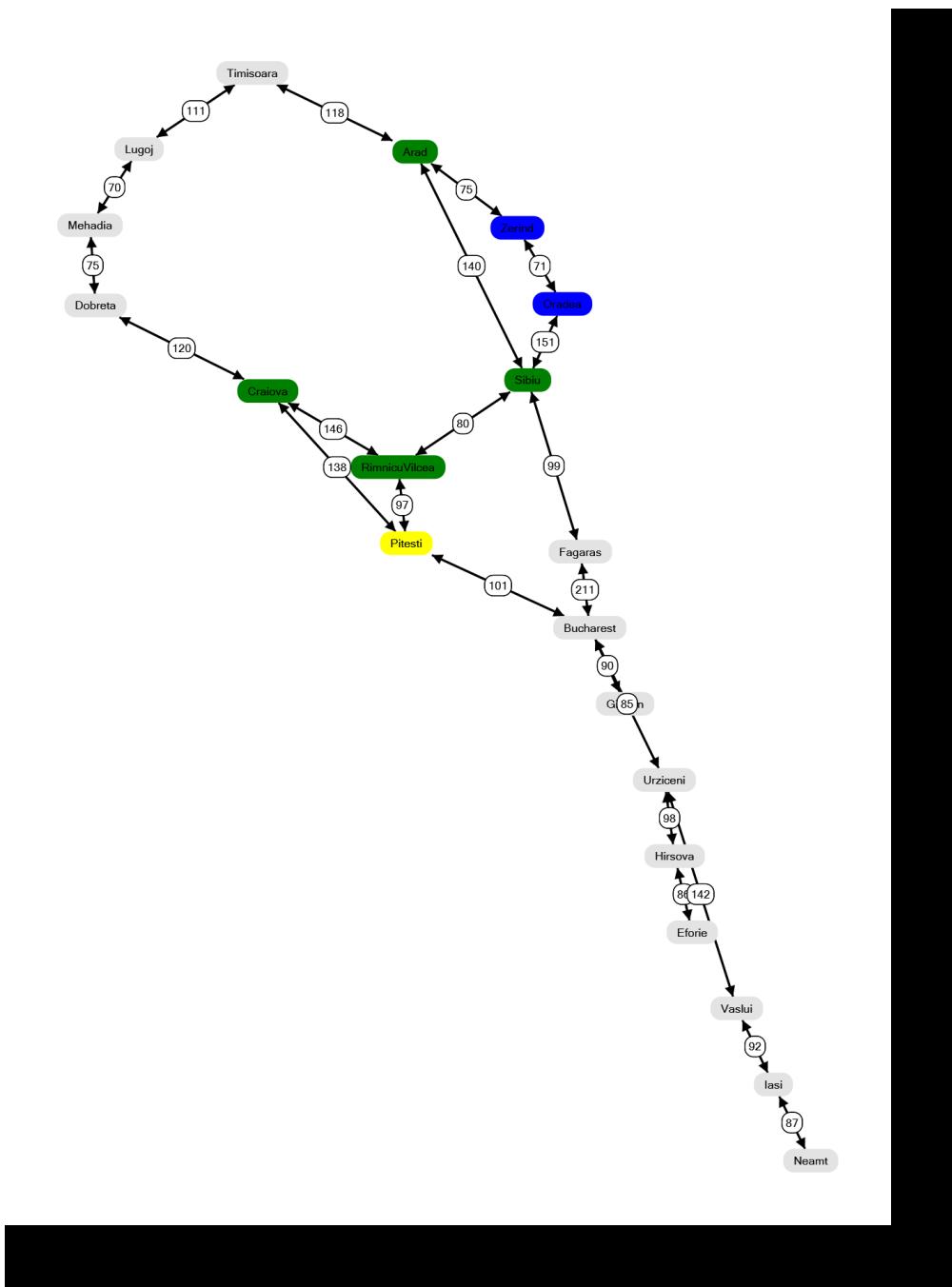
Visit

The algorithm has marked the node Craiova as visited



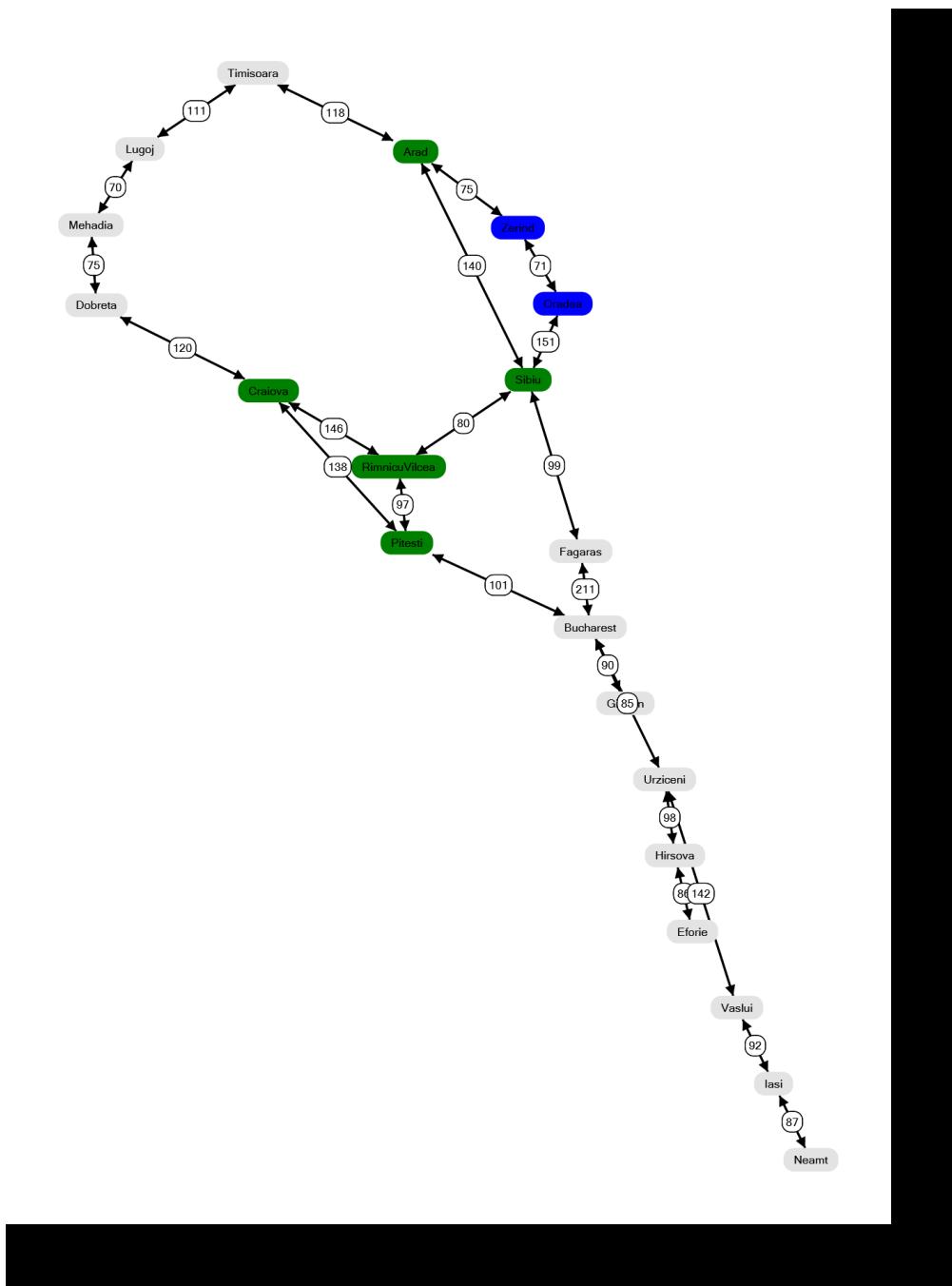
Pre-Visit

The algorithm has opened the node Pitesti



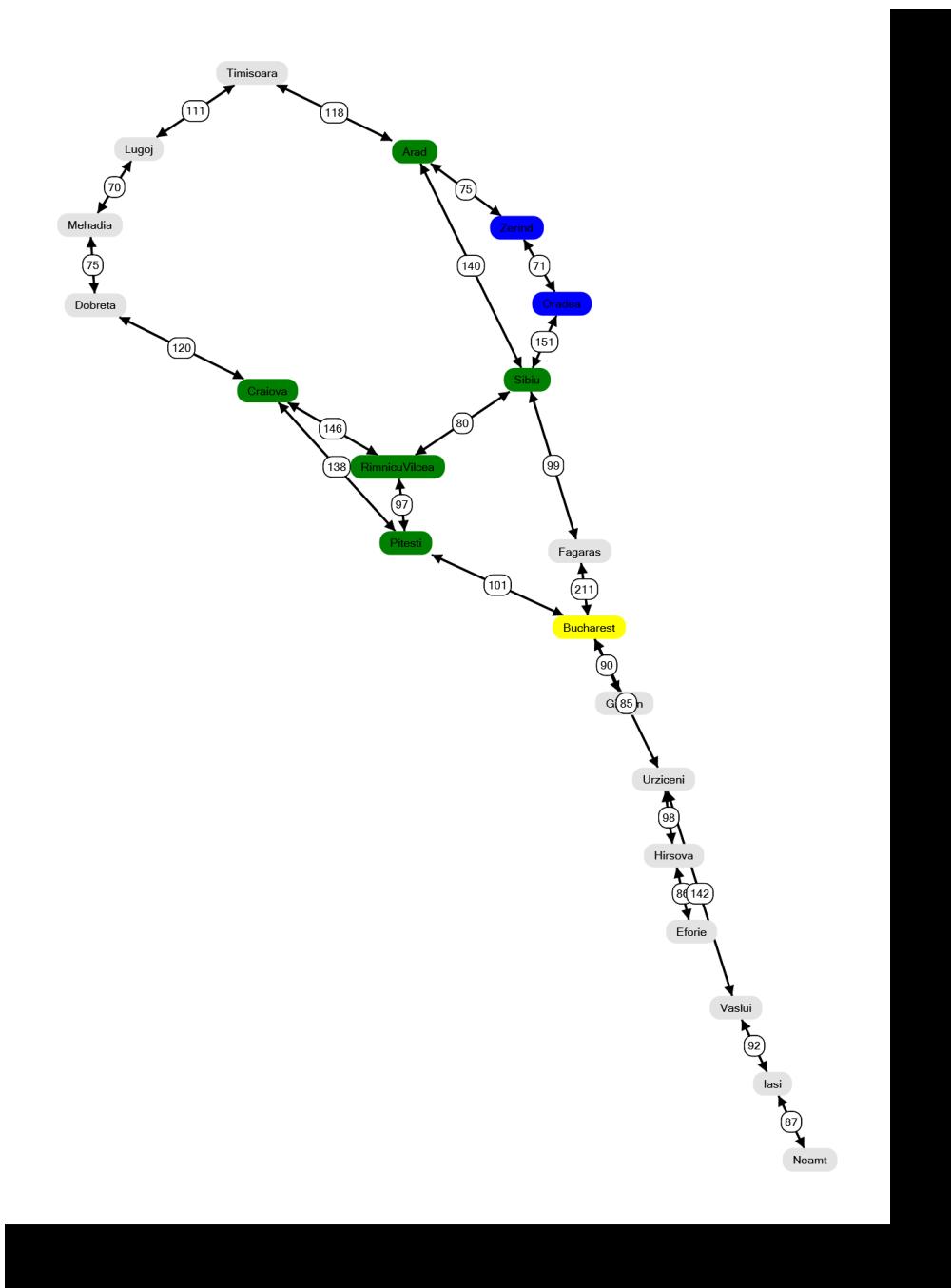
Visit

The algorithm has marked the node Pitesti as visited



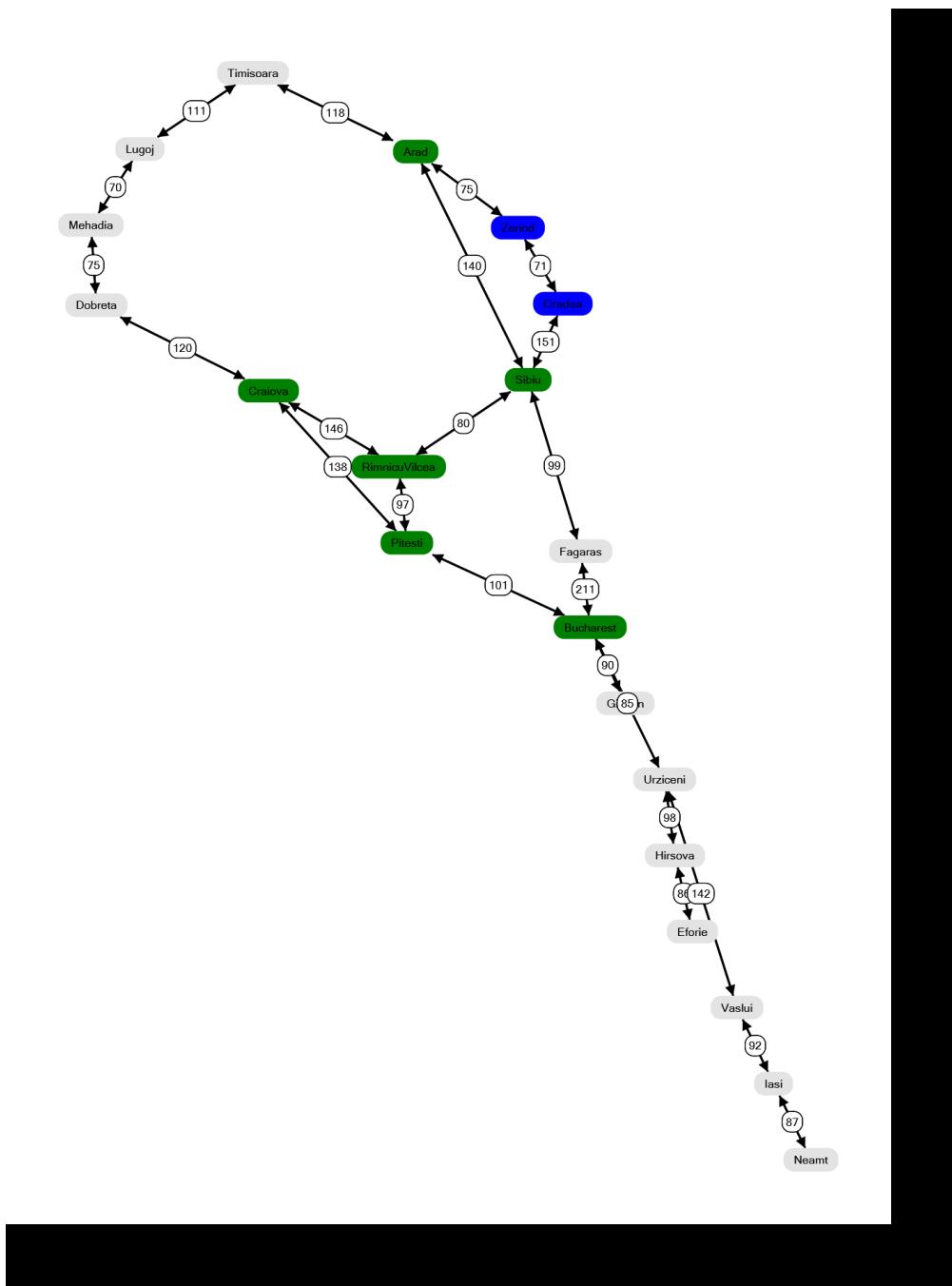
Pre-Visit

The algorithm has opened the node Bucharest



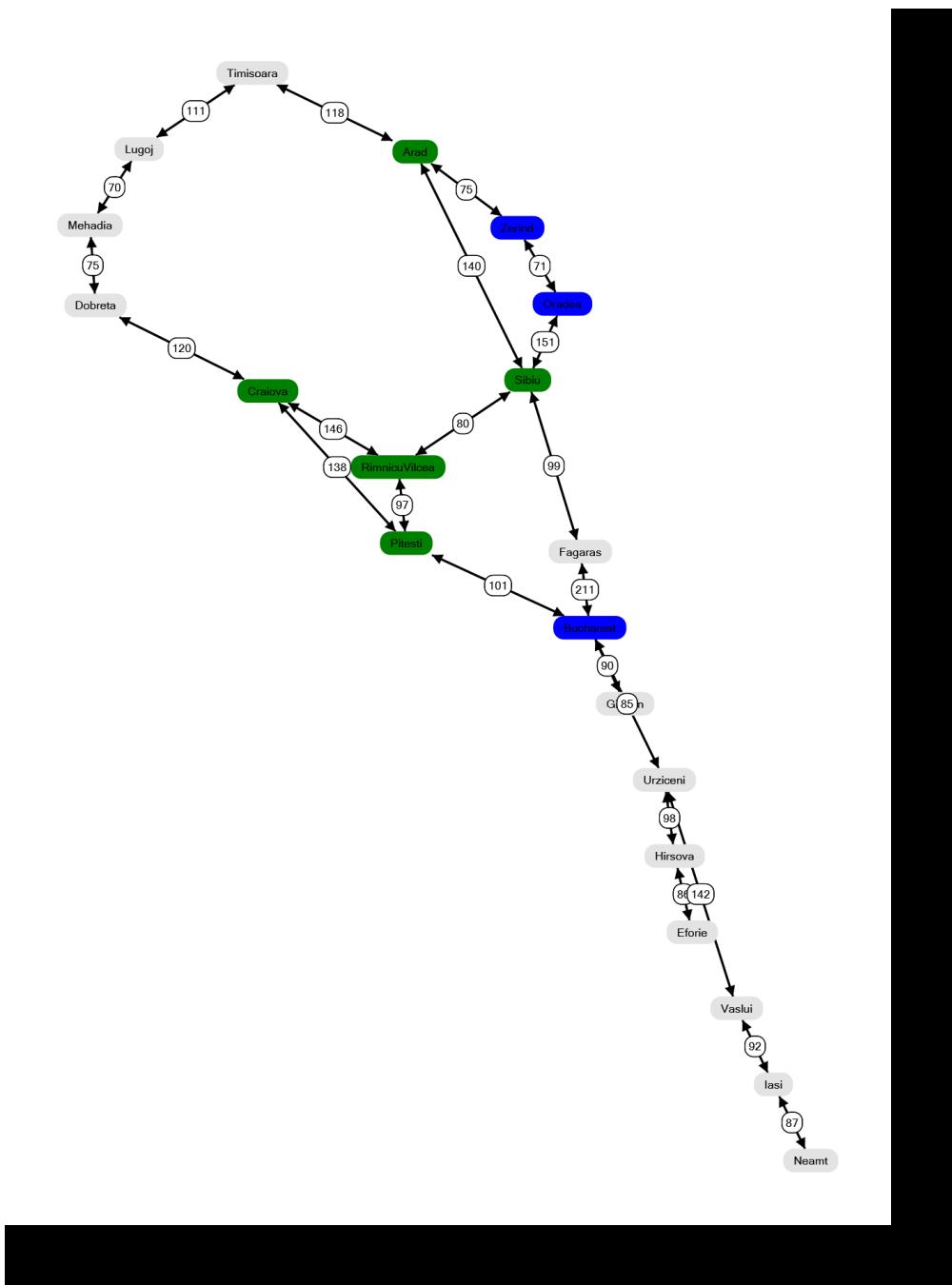
Visit

The algorithm has marked the node Bucharest as visited



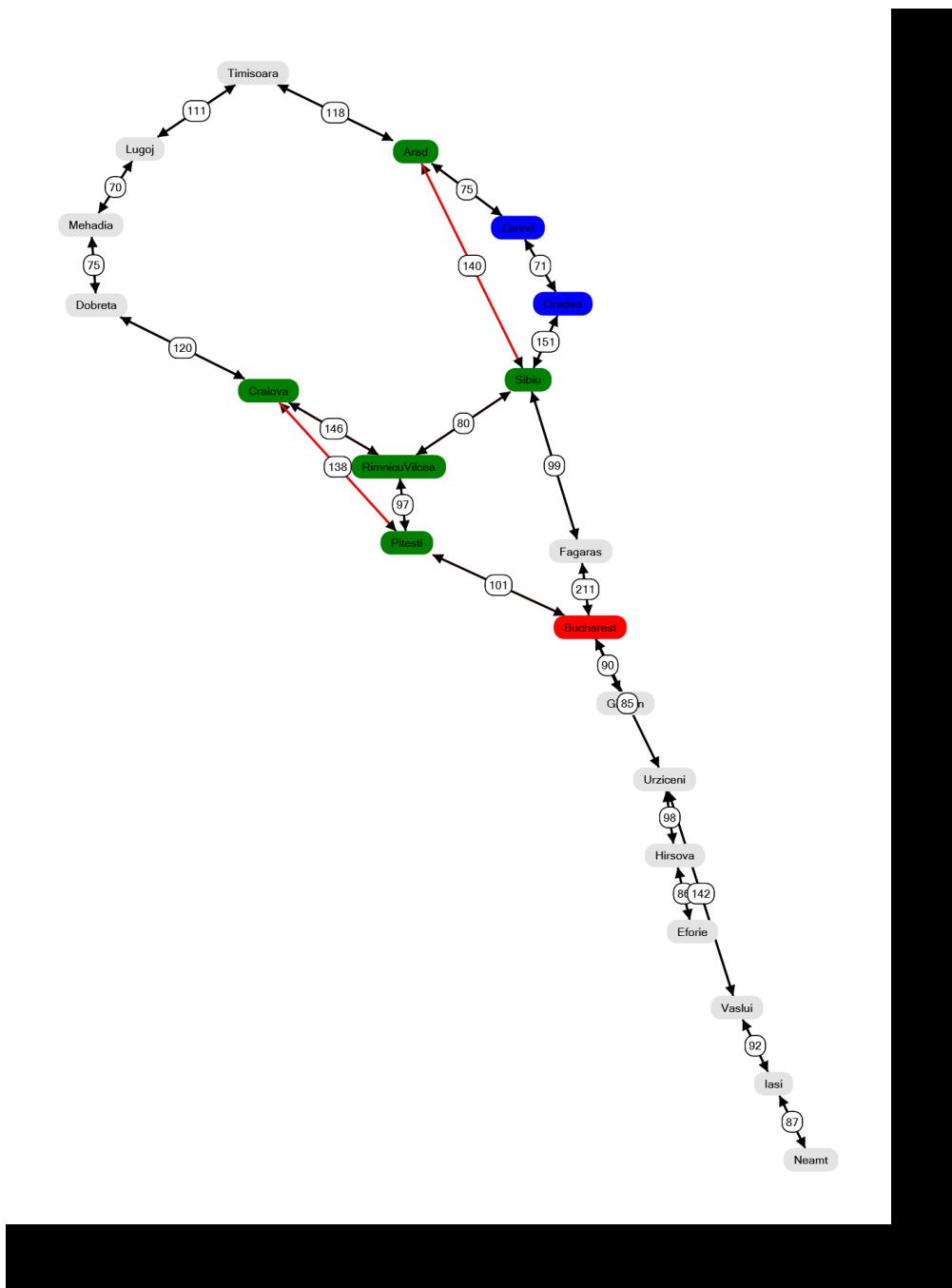
Post-Visit

The algorithm has closed the node Bucharest



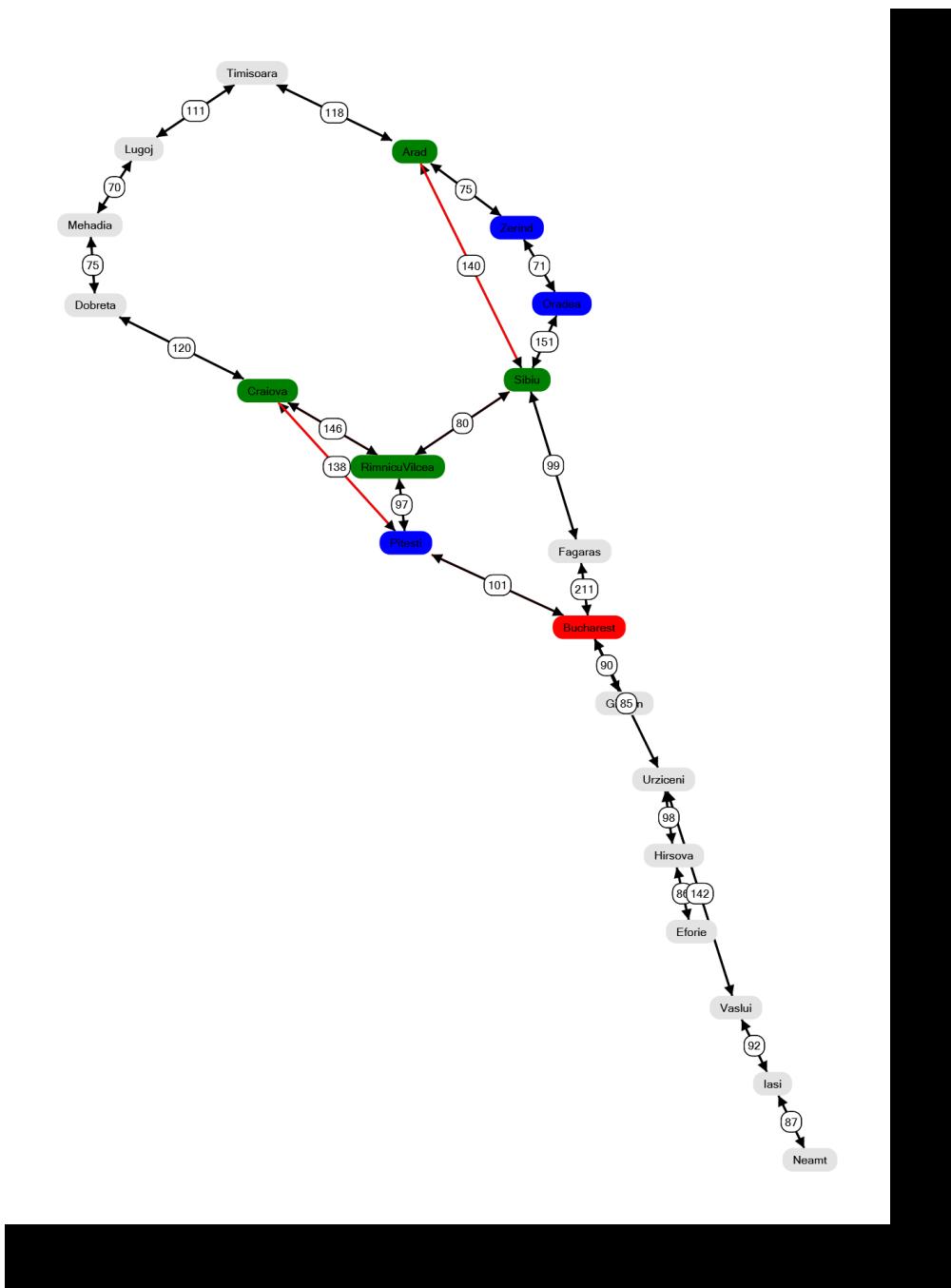
Result Found

The algorithm has found the target node Bucharest



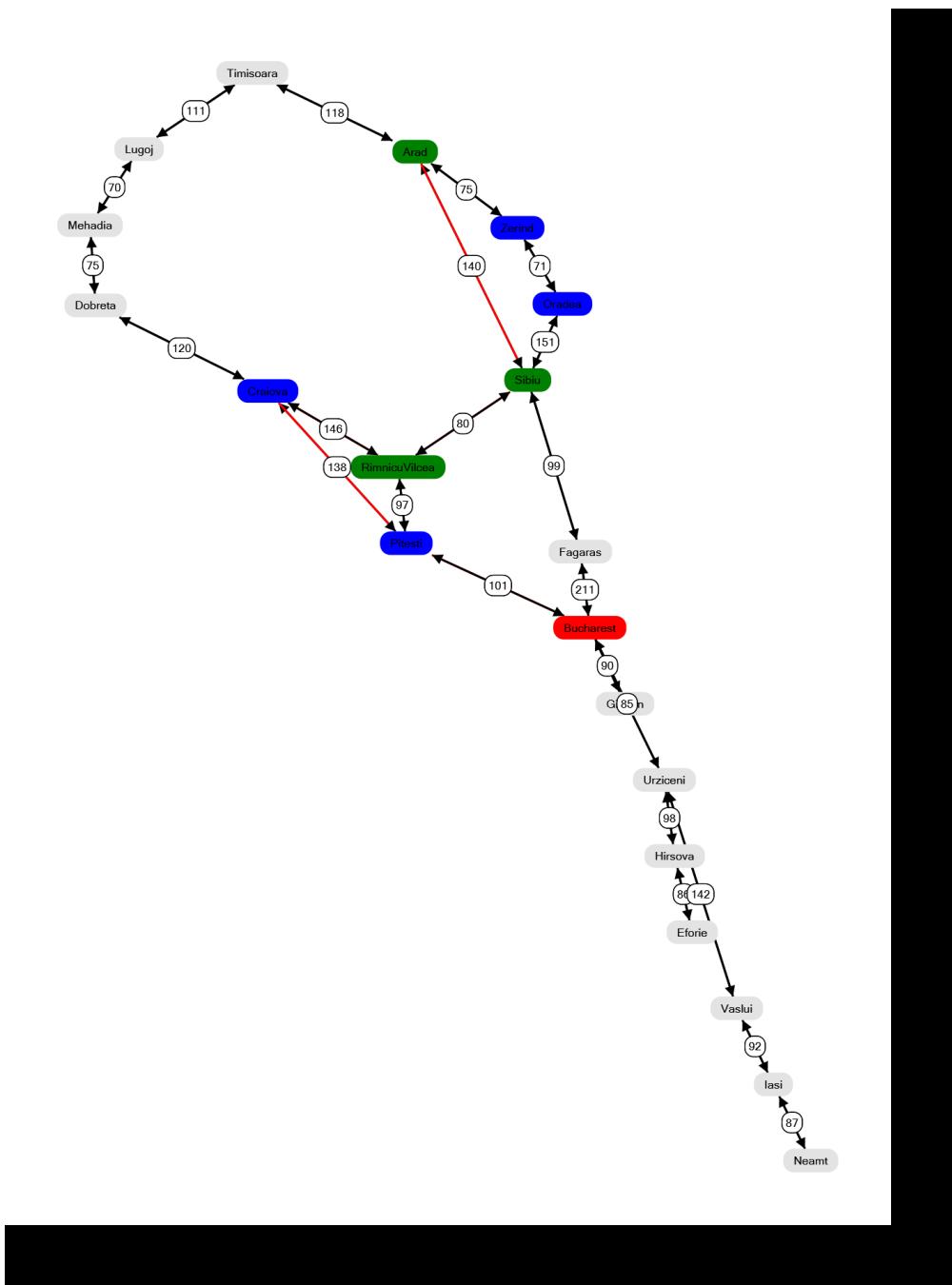
Post-Visit

The algorithm has closed the node Pitesti



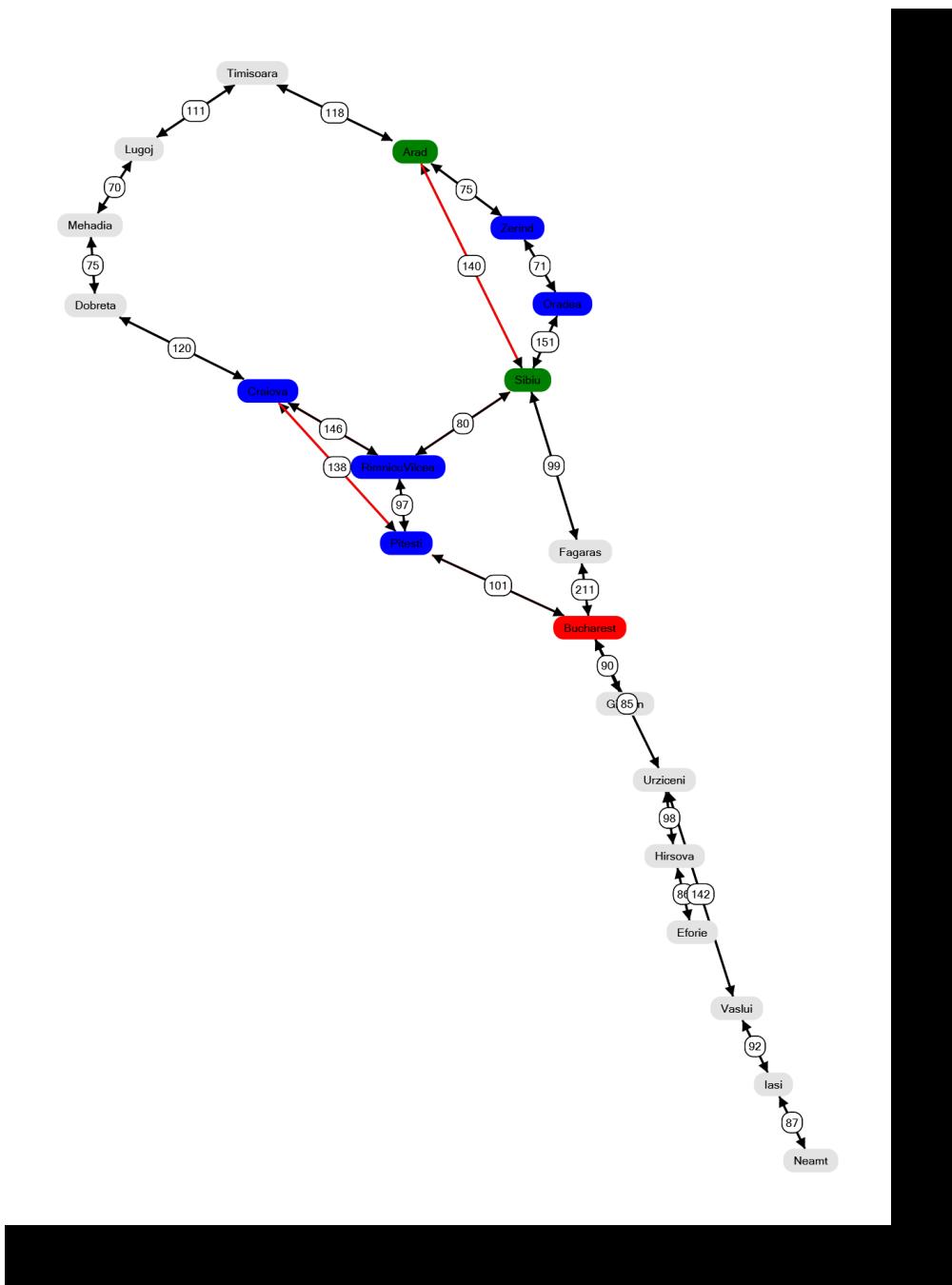
Post-Visit

The algorithm has closed the node Craiova



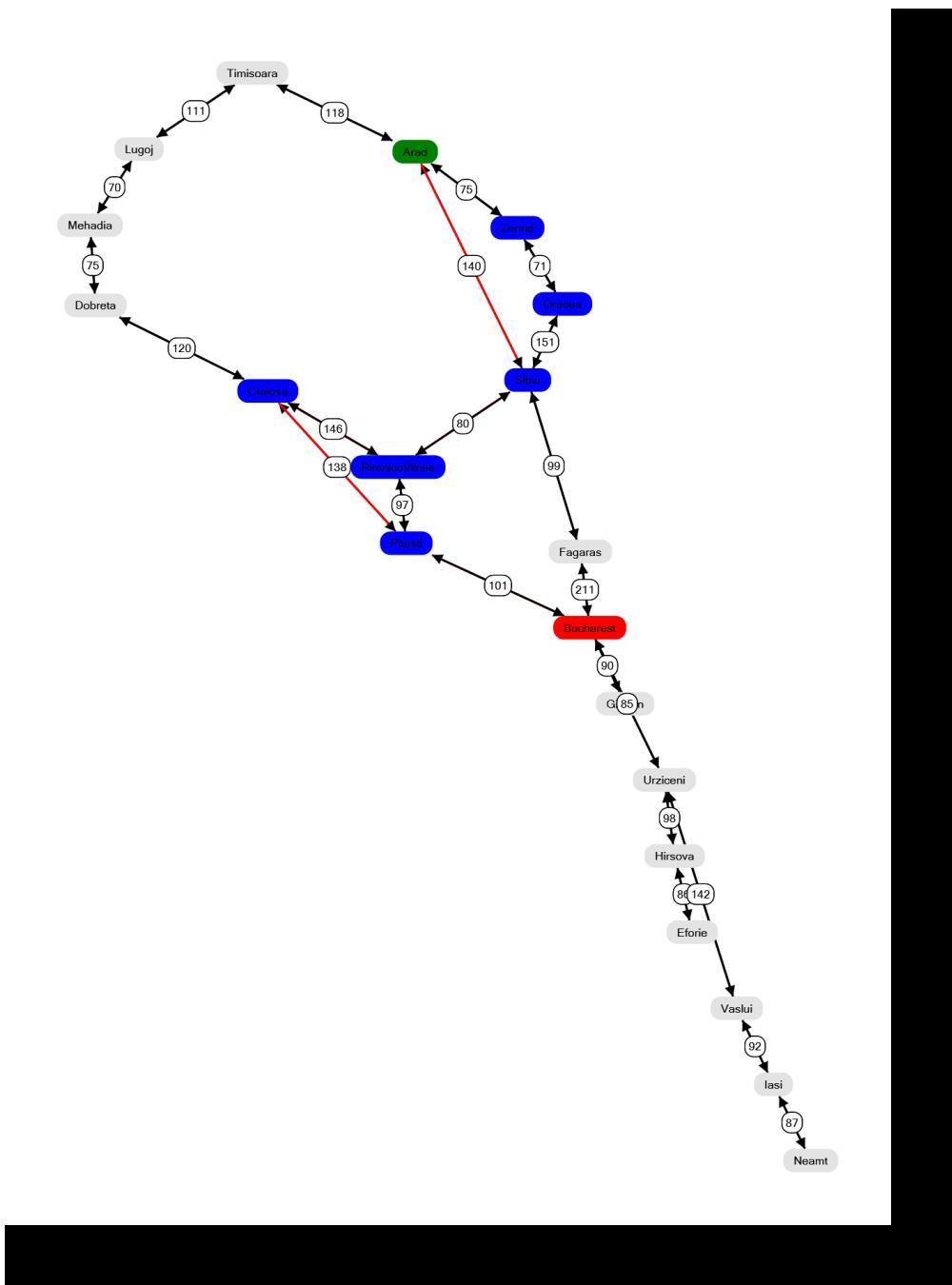
Post-Visit

The algorithm has closed the node Rimnicu Vilcea



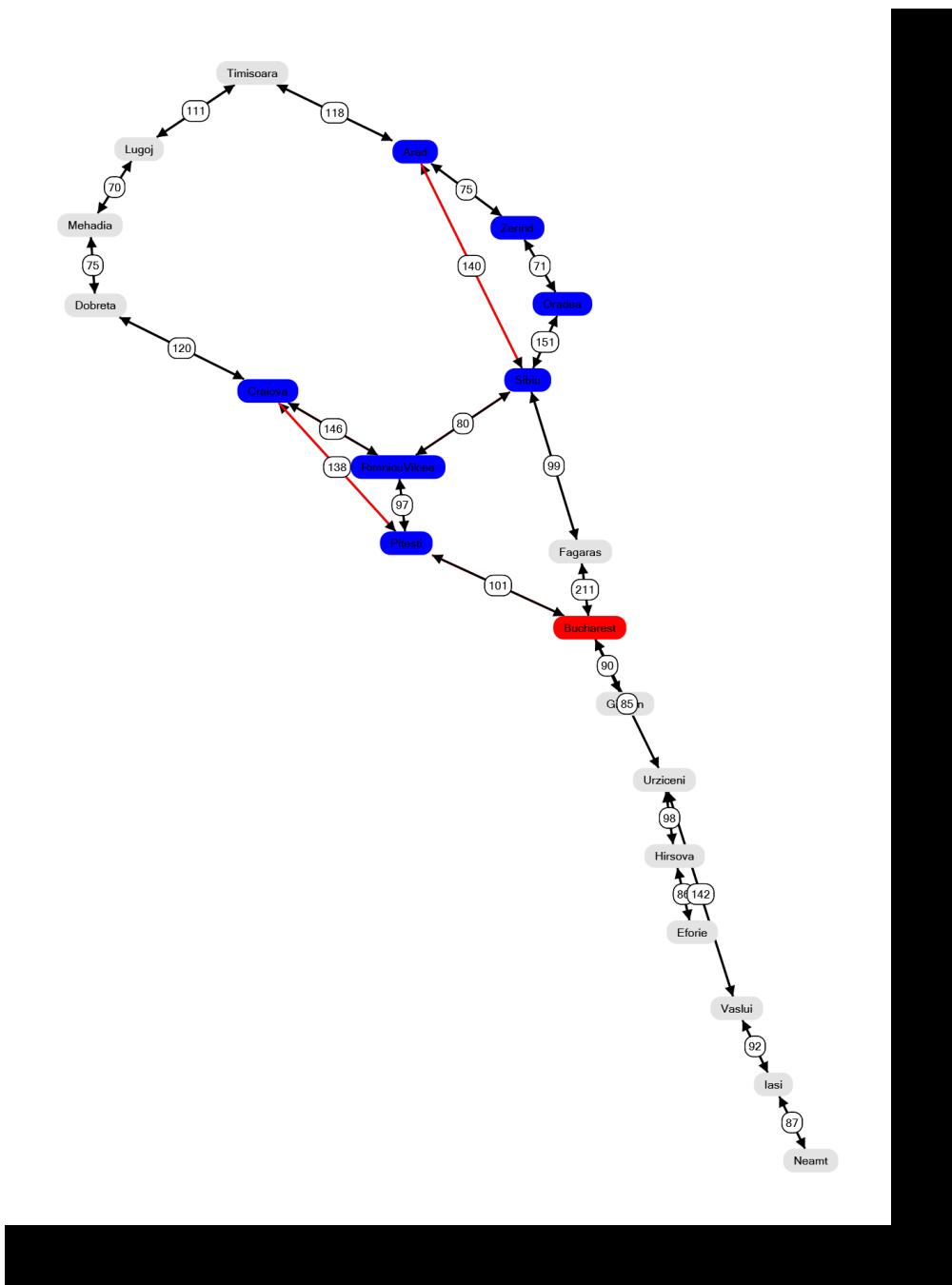
Post-Visit

The algorithm has closed the node Sibiu



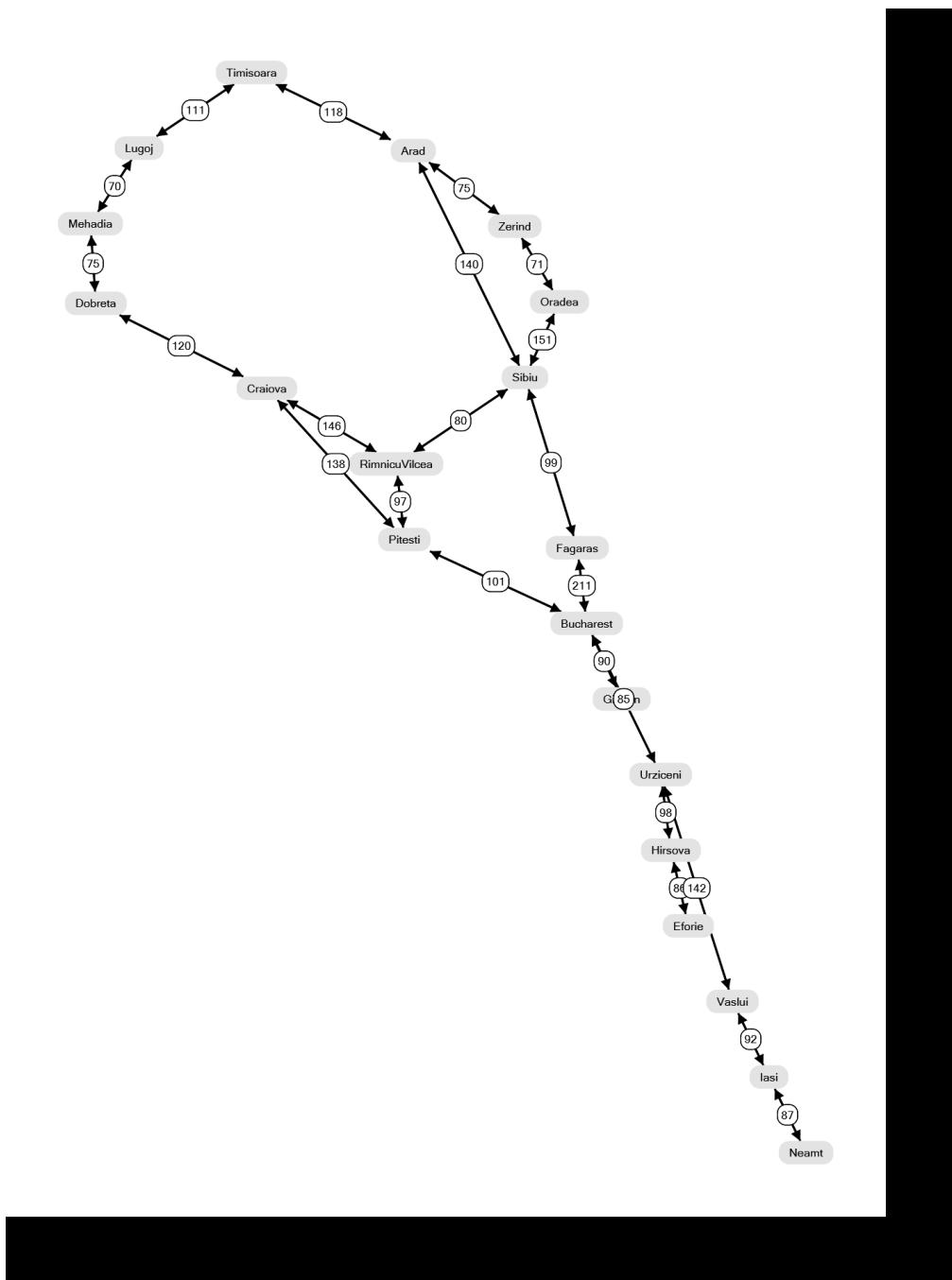
Post-Visit

The algorithm has closed the node Arad



Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



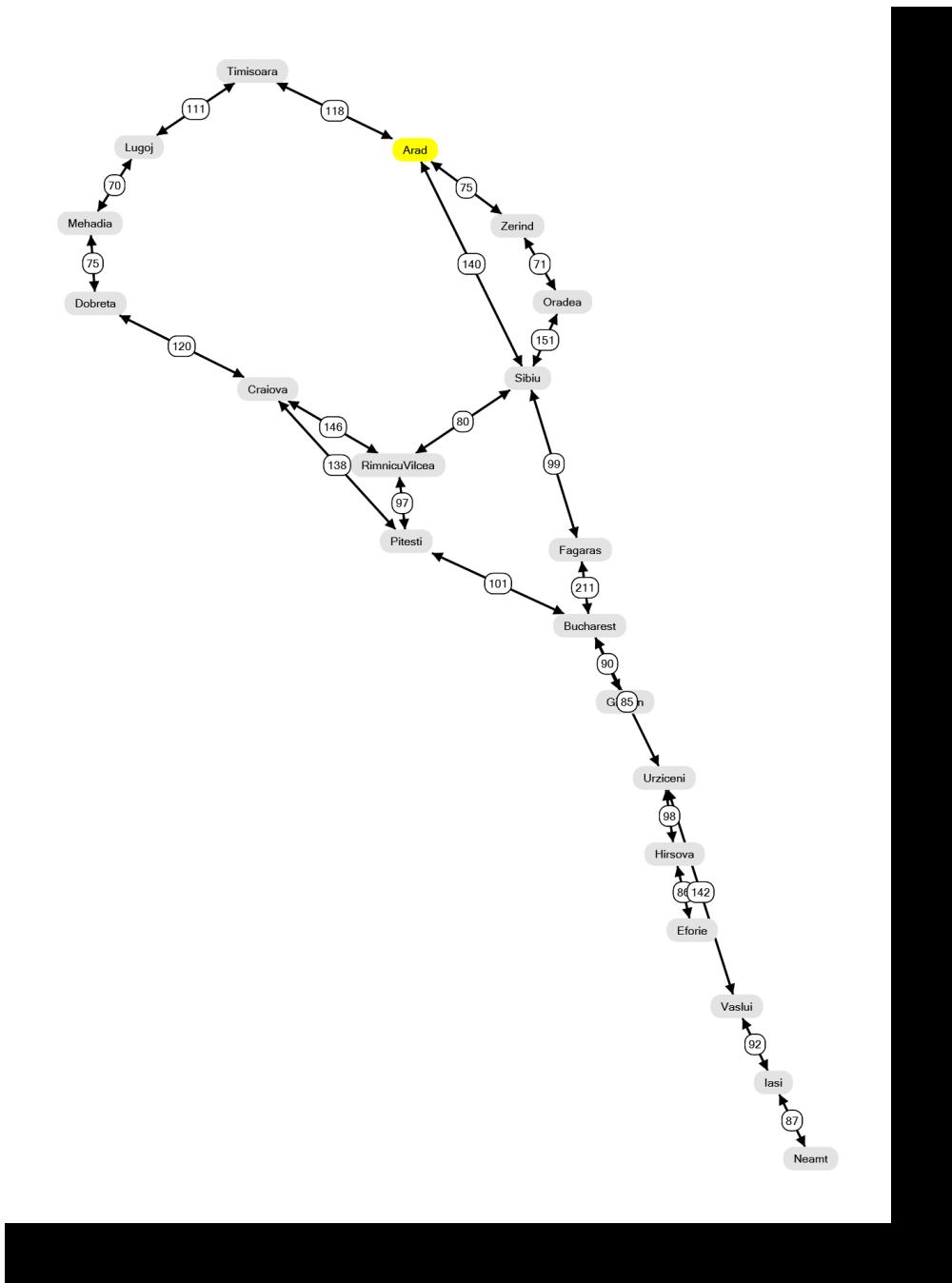
Uniform Cost Search

If all the edges in the search graph do not have the same cost then breadth-first search generalizes to uniform-cost search. Instead of expanding nodes in order of their depth from the root, uniform-cost search expands nodes in order of their cost from the root. At each step, the next step n to be expanded is one whose cost $g(n)$ is lowest where $g(n)$ is the sum of the edge costs from the root to node n . The nodes are stored in a priority queue. This algorithm is also known as Dijkstra's single-source shortest algorithm.

The algorithm took 00:00:00.0003160 to find the goal.

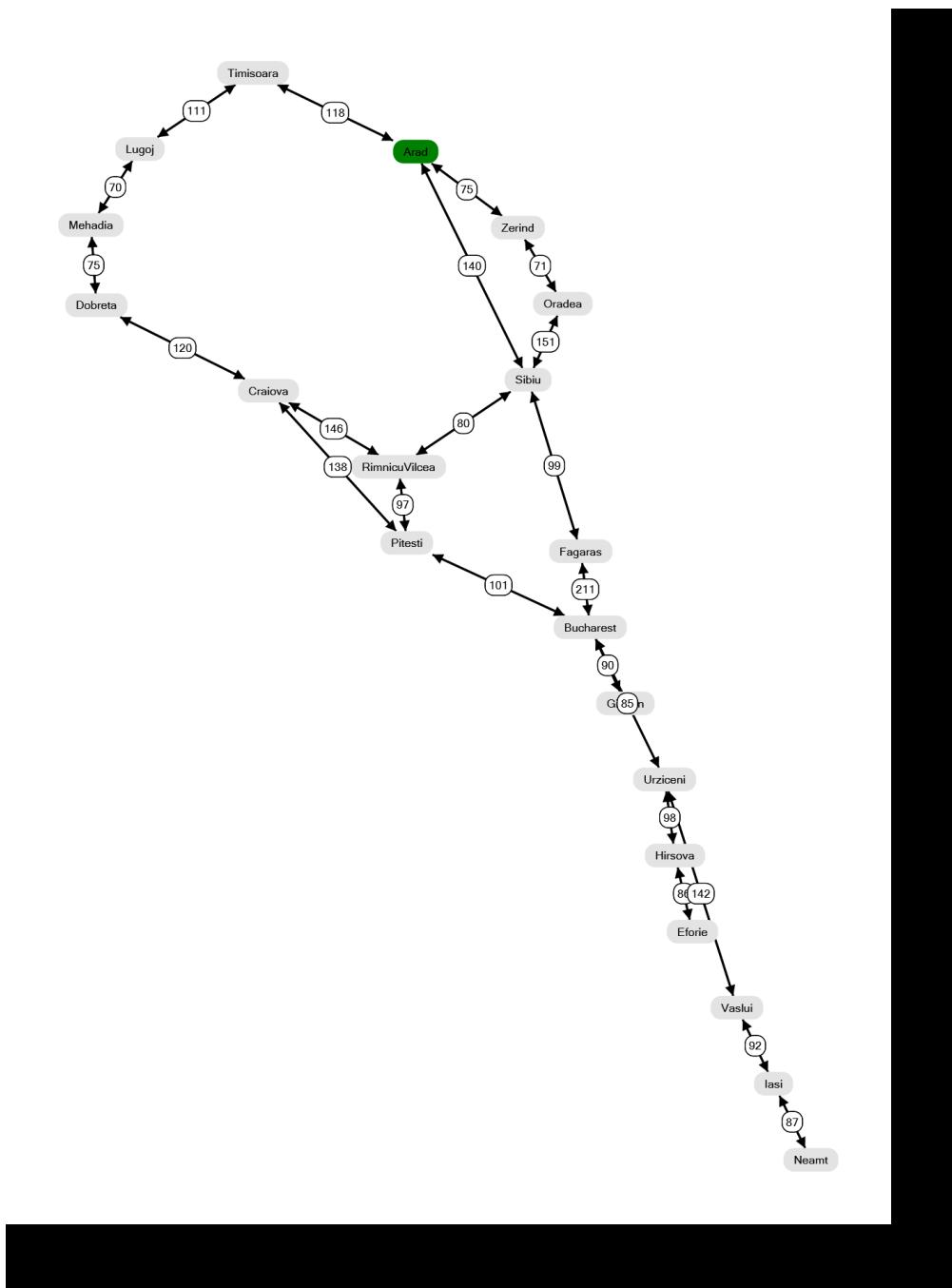
Pre-Visit

The algorithm has opened the node Arad



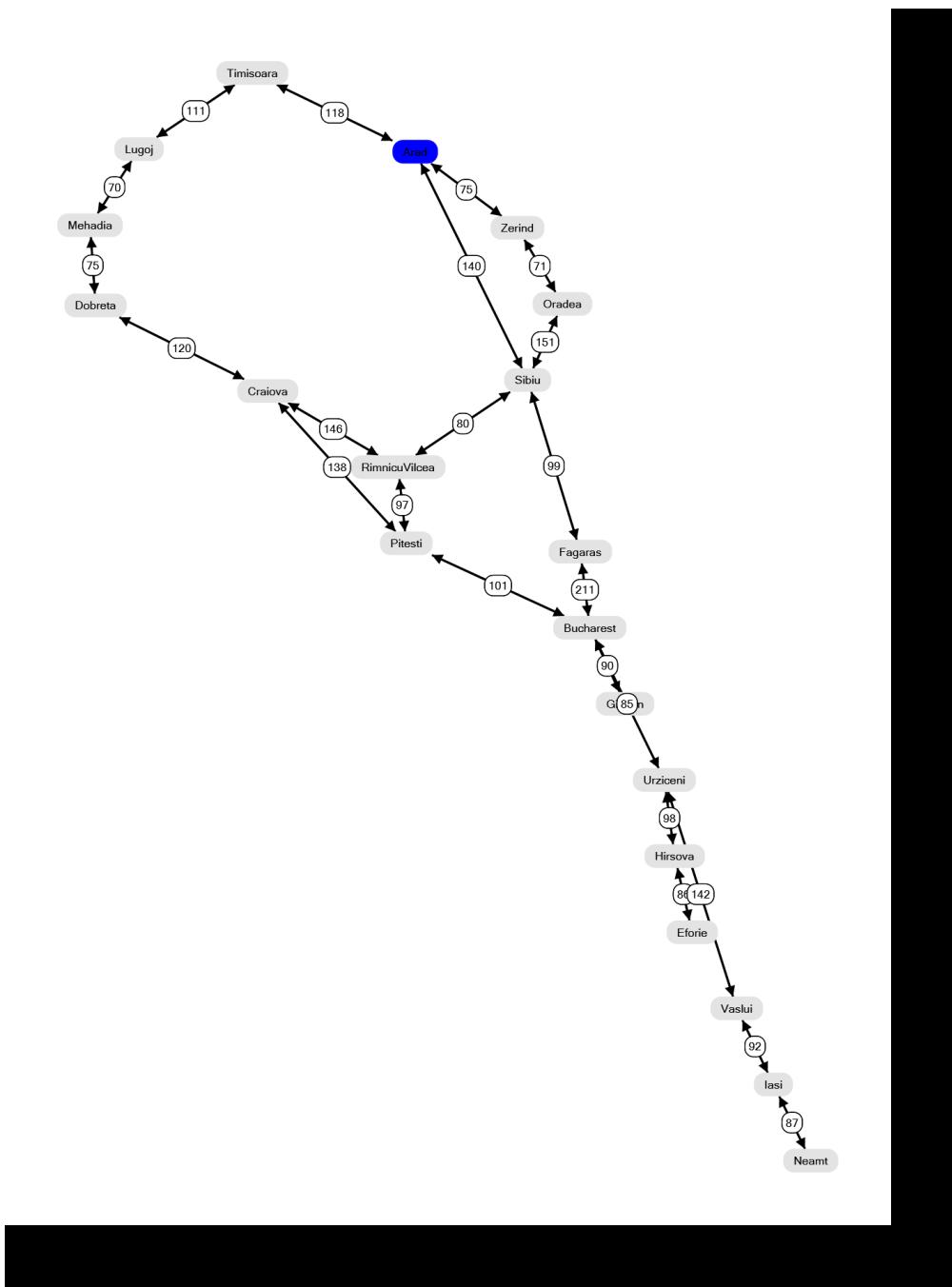
Visit

The algorithm has marked the node Arad as visited



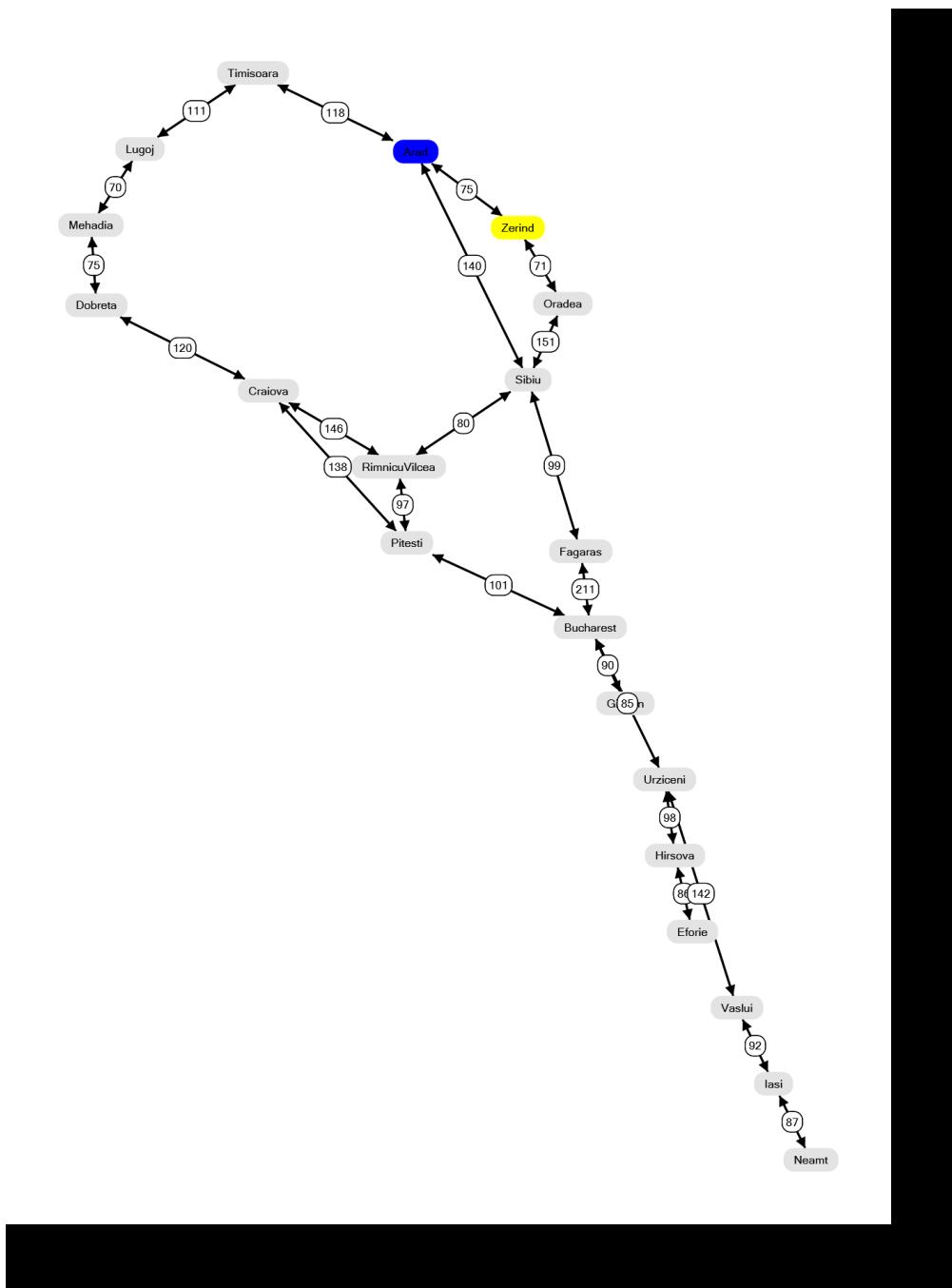
Post-Visit

The algorithm has closed the node Arad



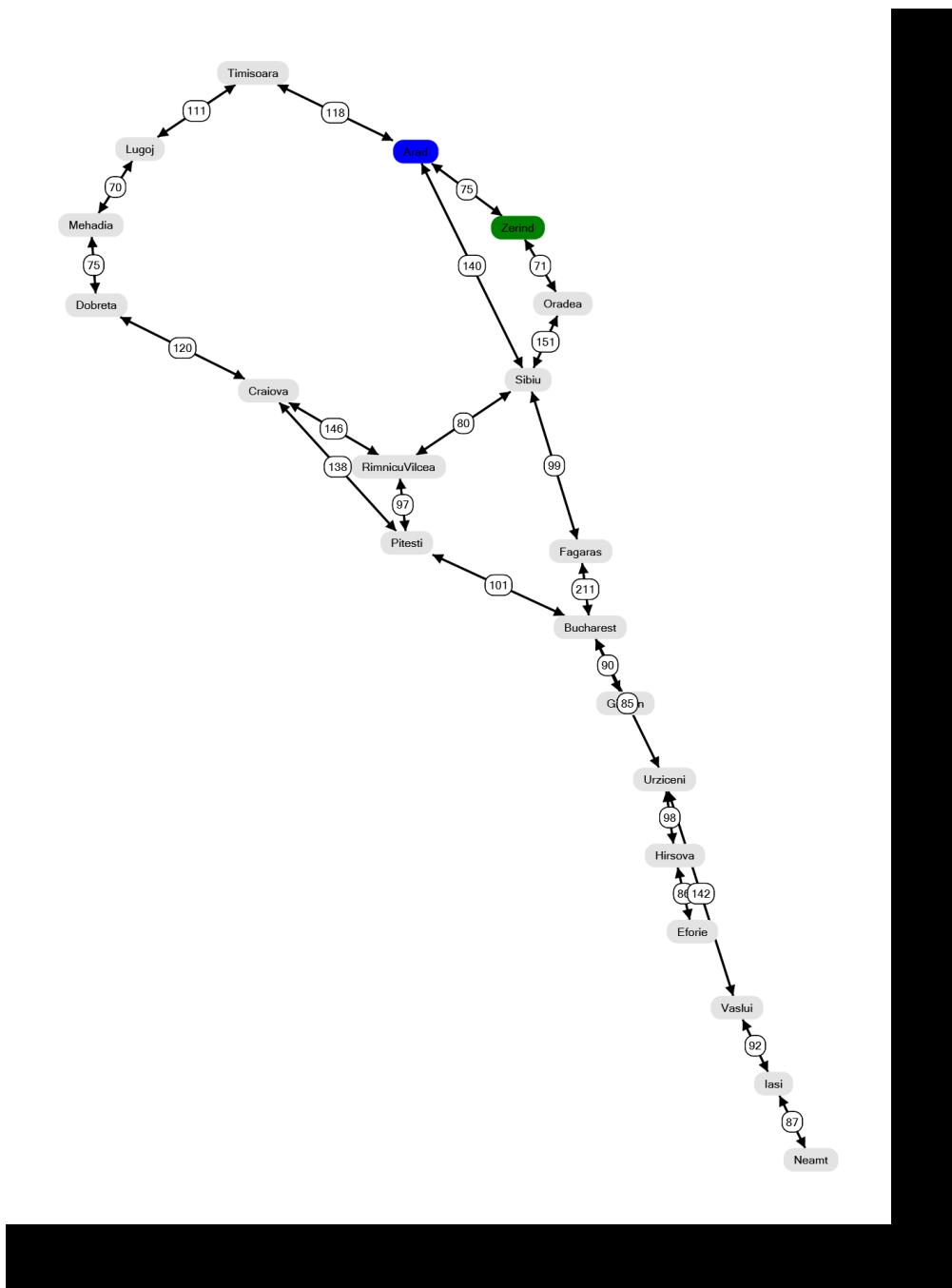
Pre-Visit

The algorithm has opened the node Zerind



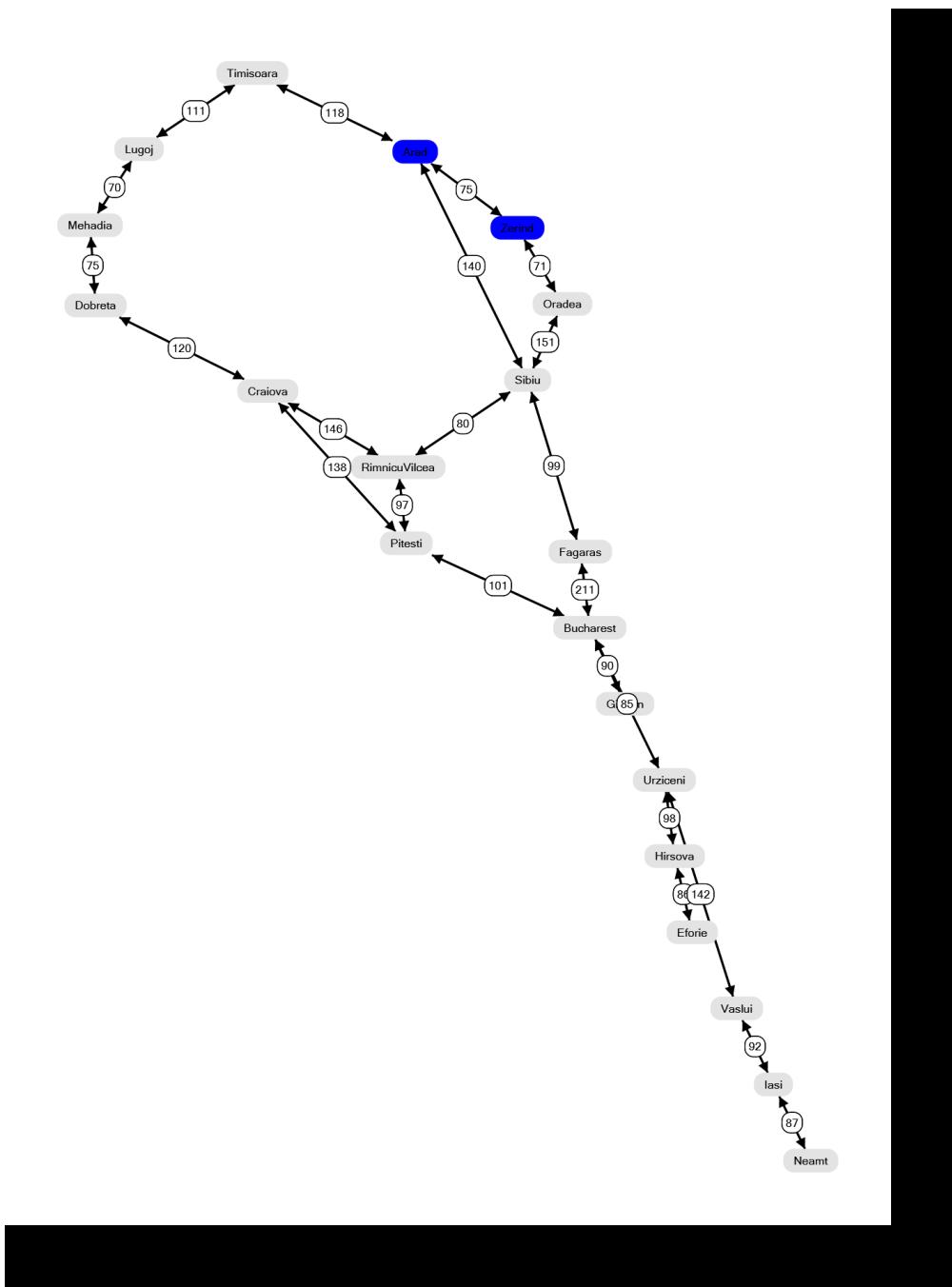
Visit

The algorithm has marked the node Zerind as visited



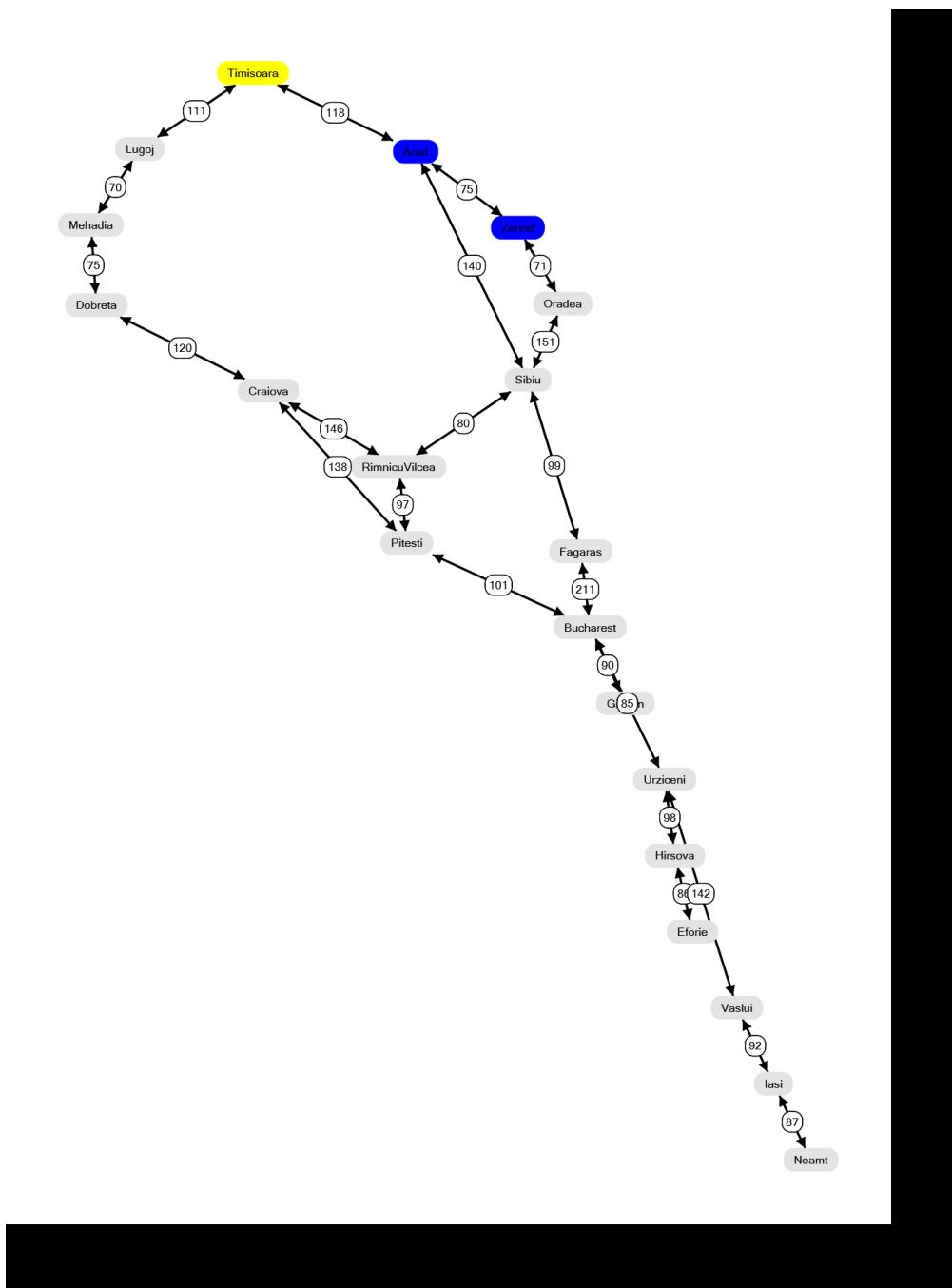
Post-Visit

The algorithm has closed the node Zerind



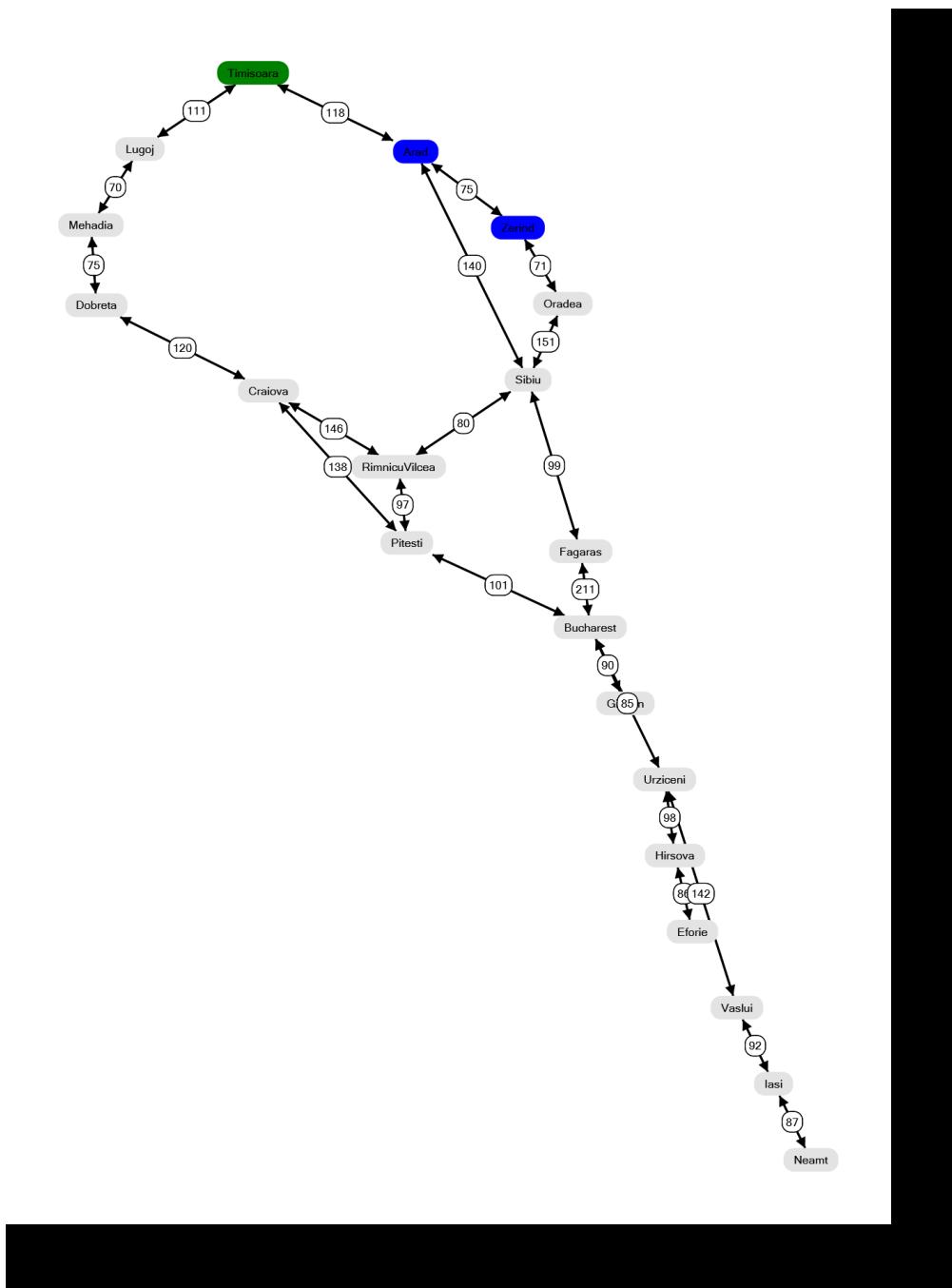
Pre-Visit

The algorithm has opened the node Timisoara



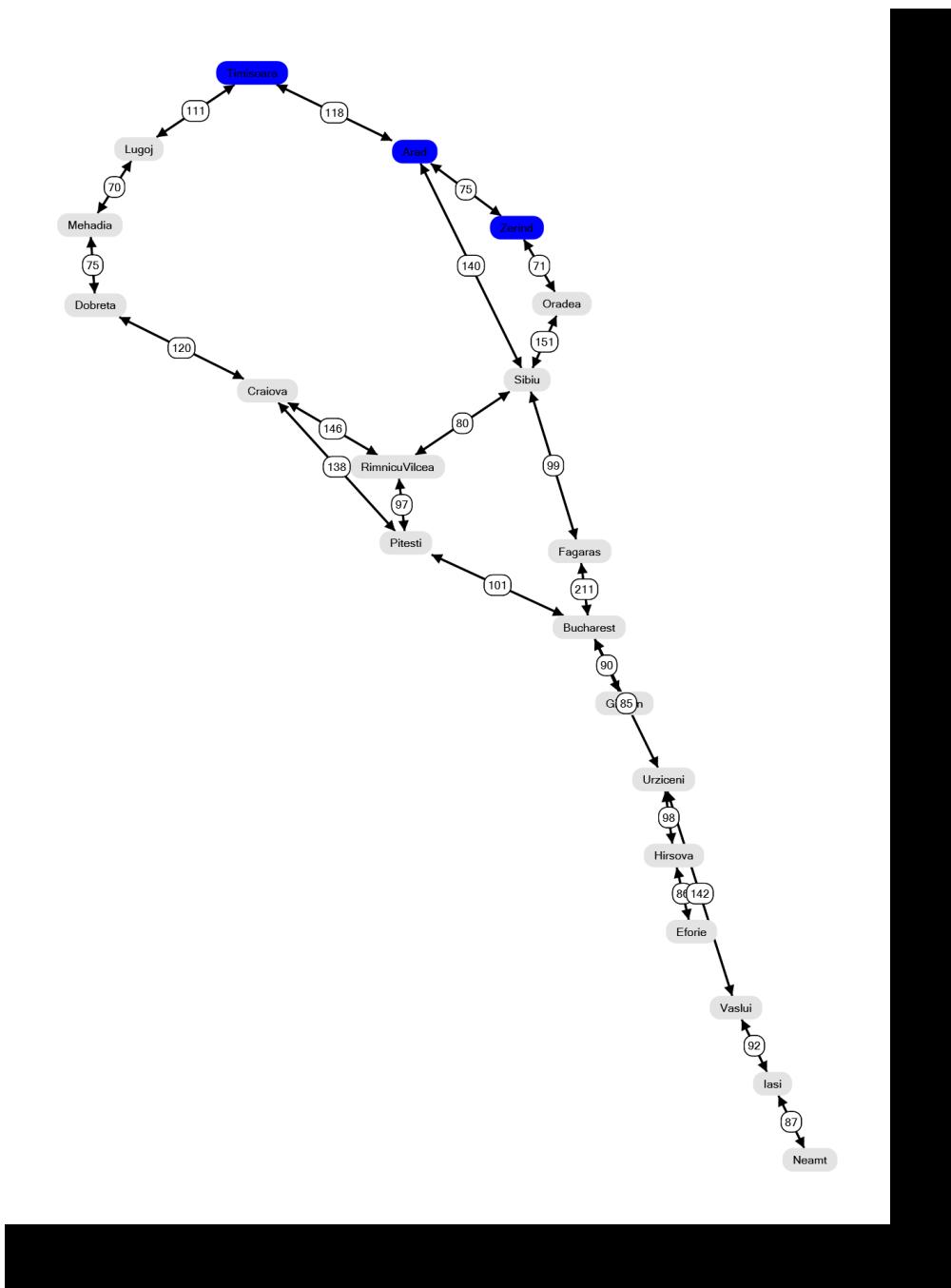
Visit

The algorithm has marked the node Timisoara as visited



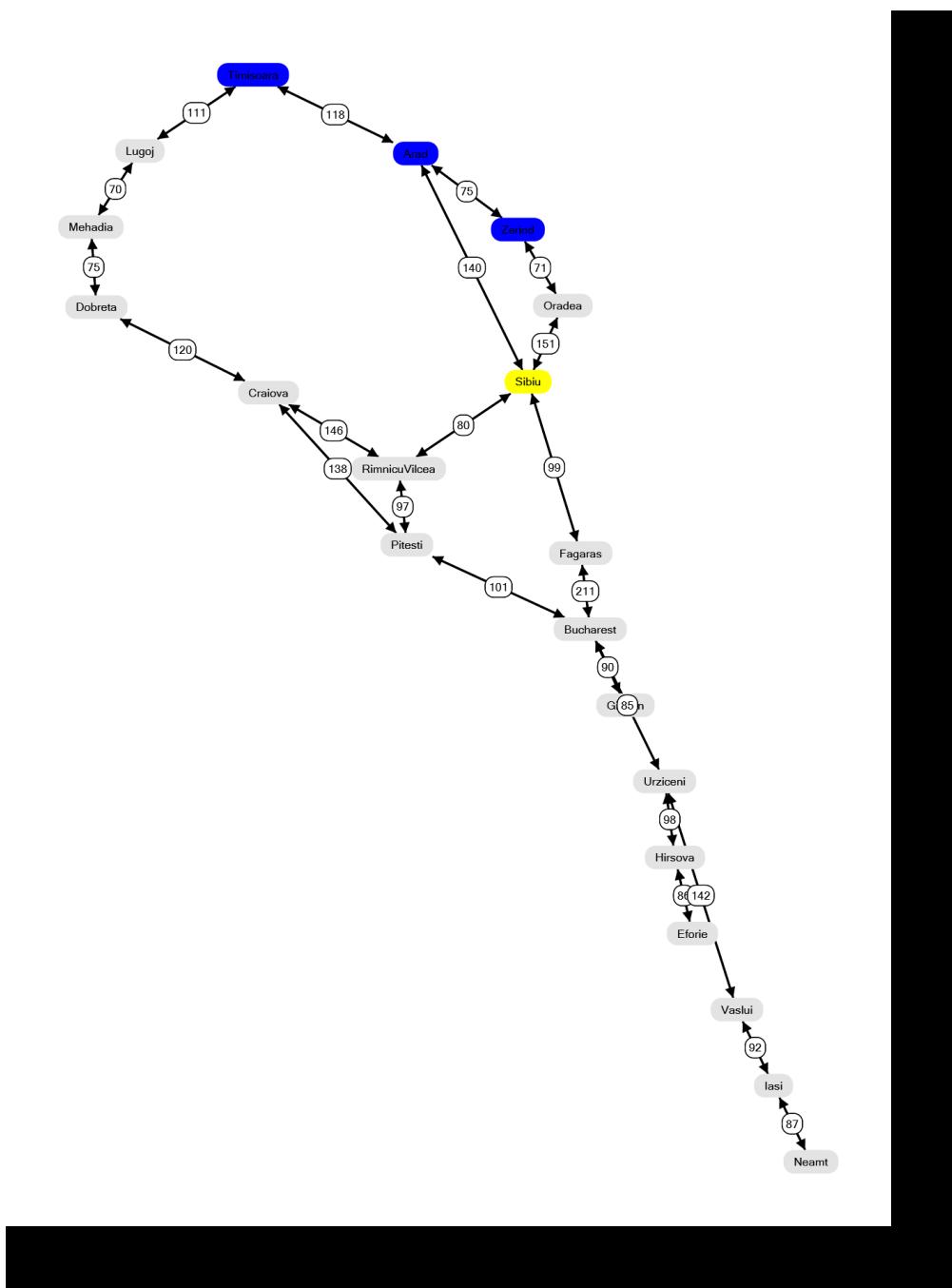
Post-Visit

The algorithm has closed the node Timisoara



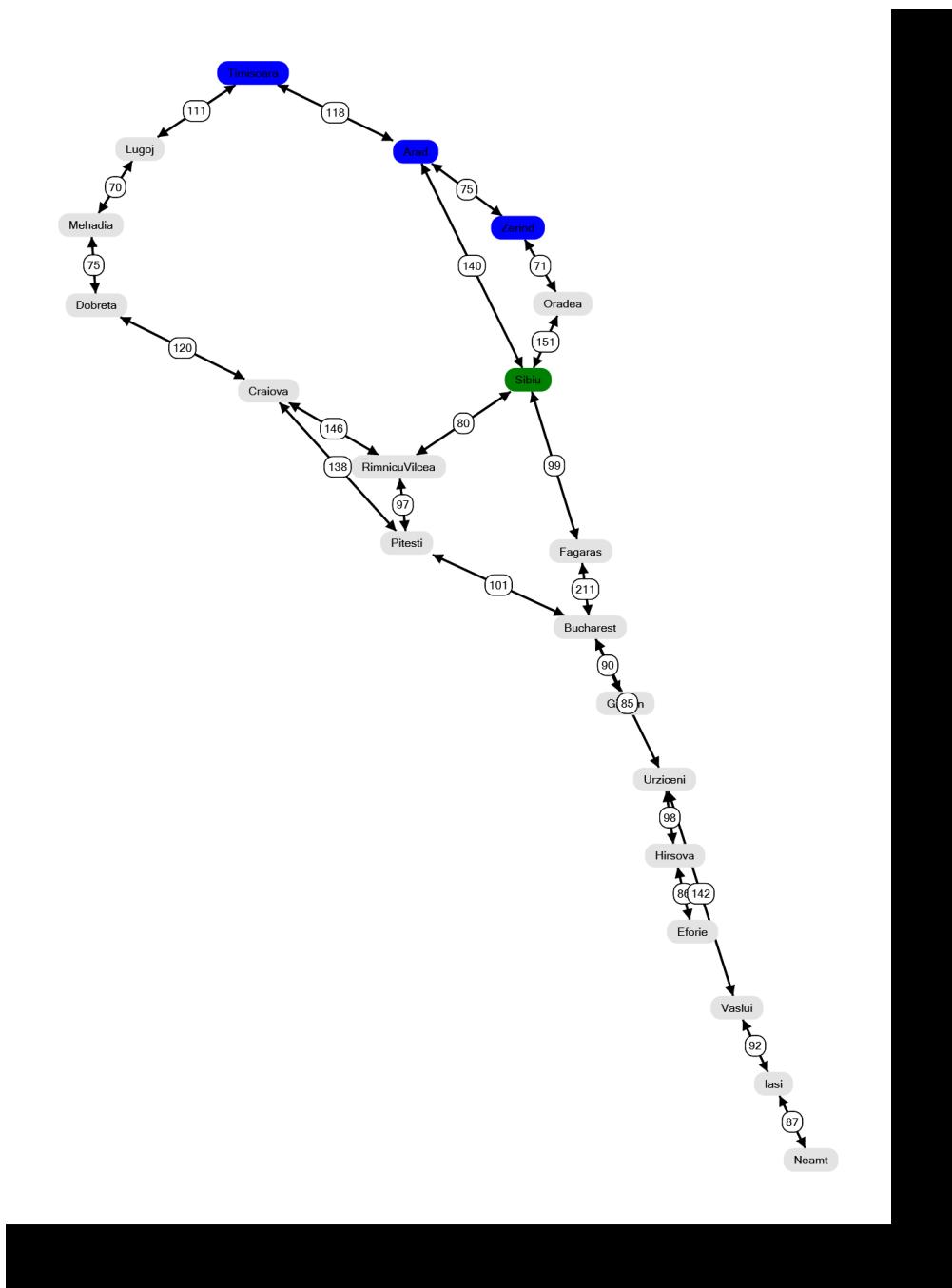
Pre-Visit

The algorithm has opened the node Sibiu



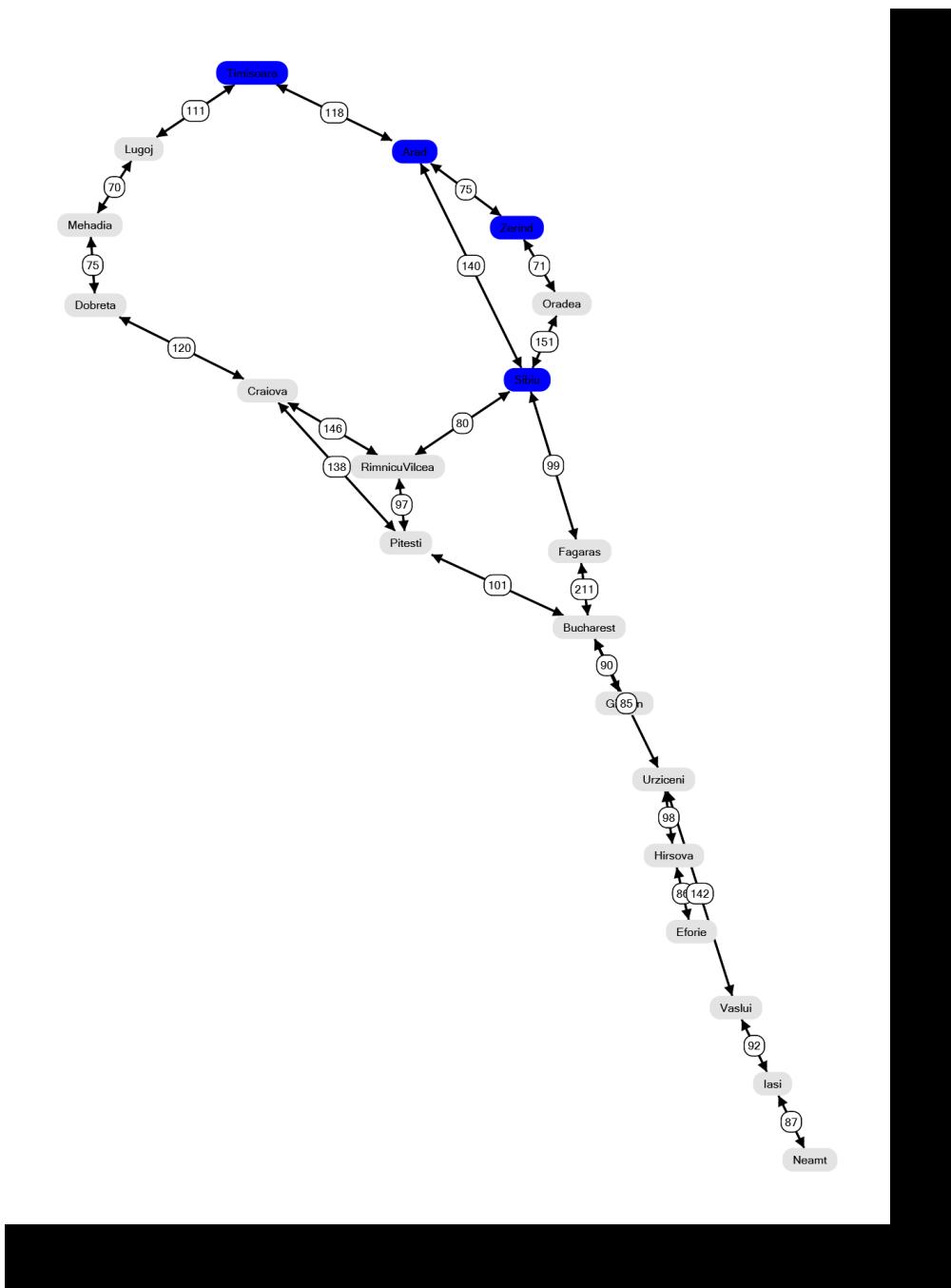
Visit

The algorithm has marked the node Sibiu as visited



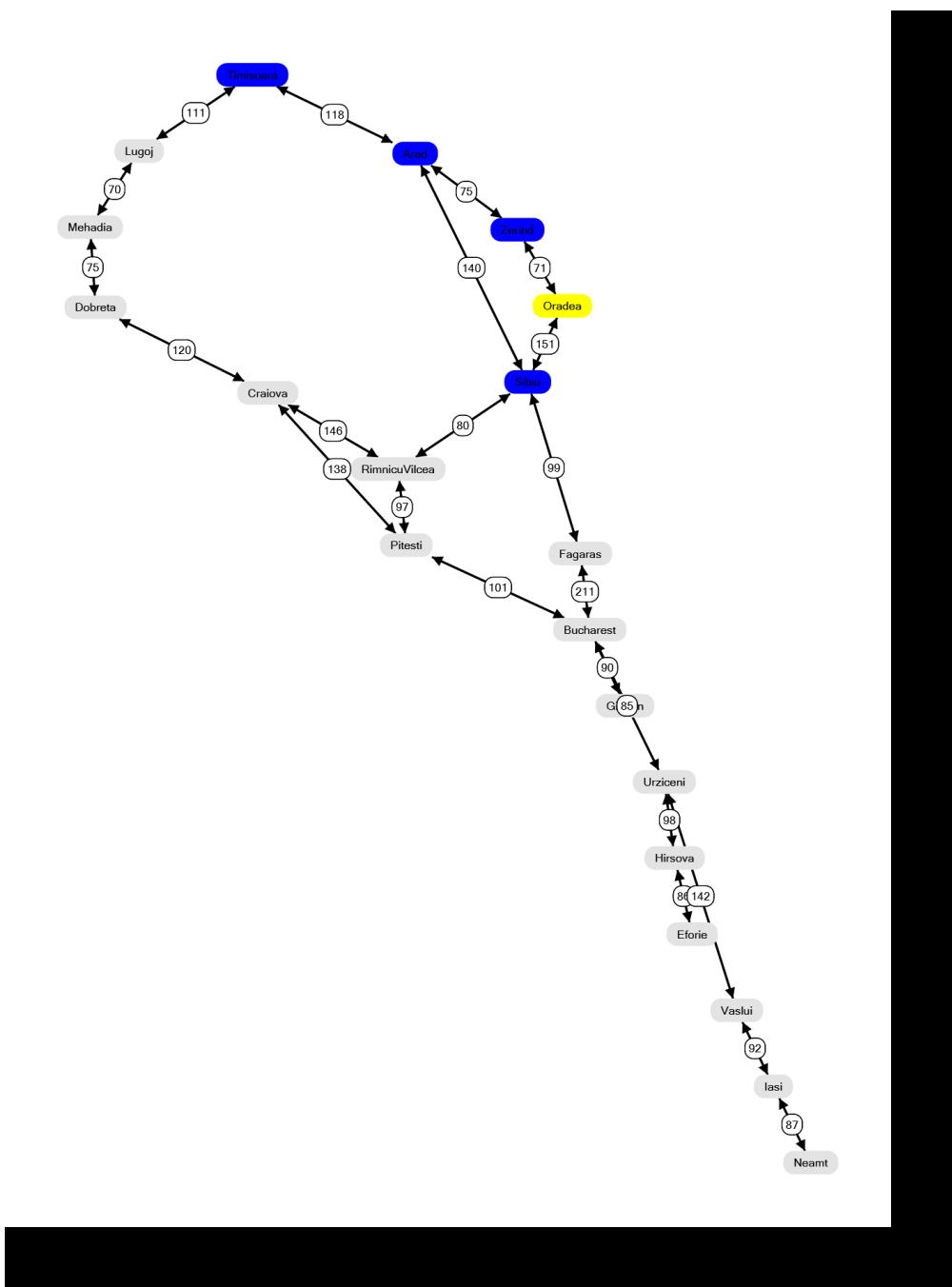
Post-Visit

The algorithm has closed the node Sibiu



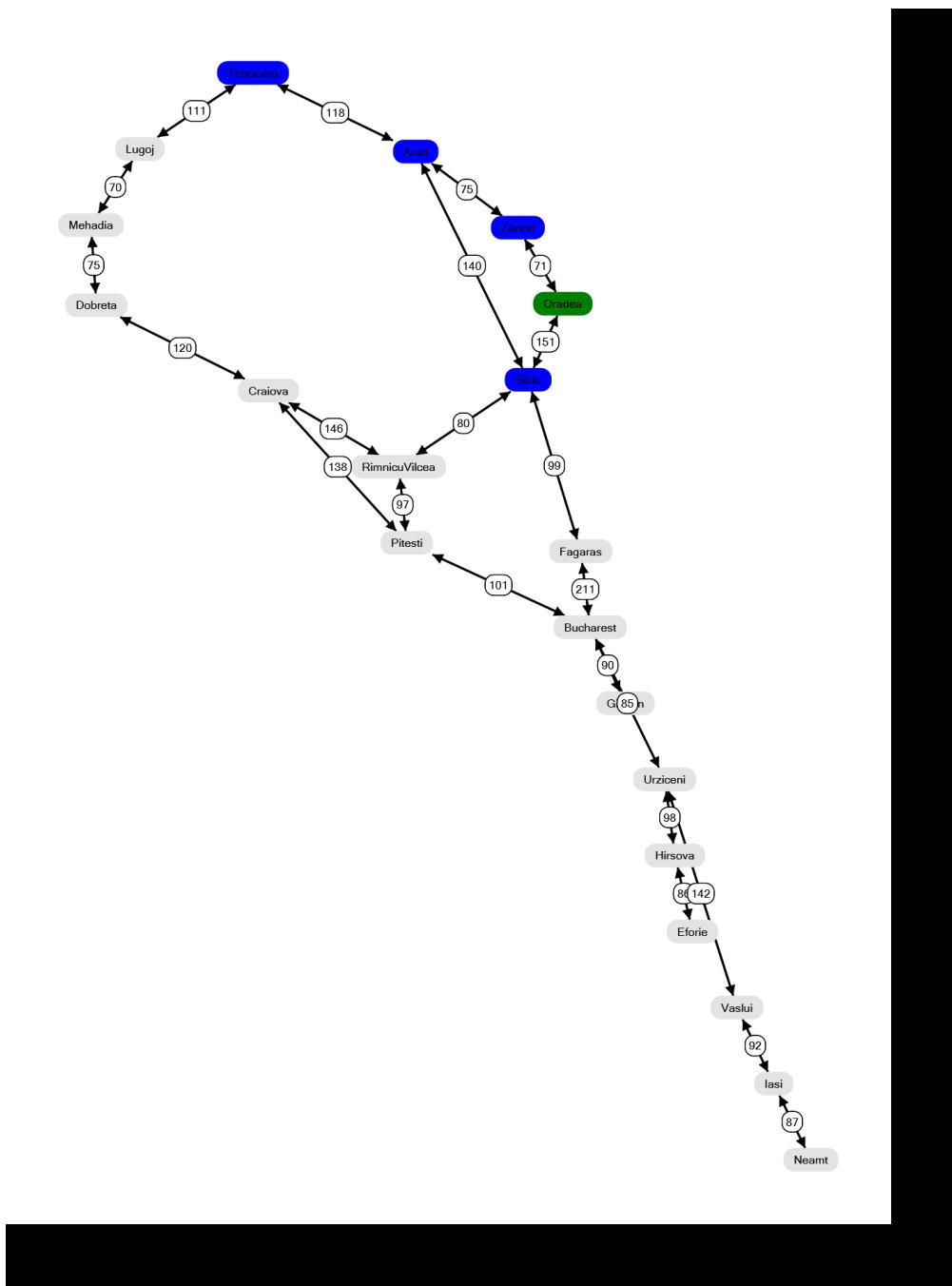
Pre-Visit

The algorithm has opened the node Oradea



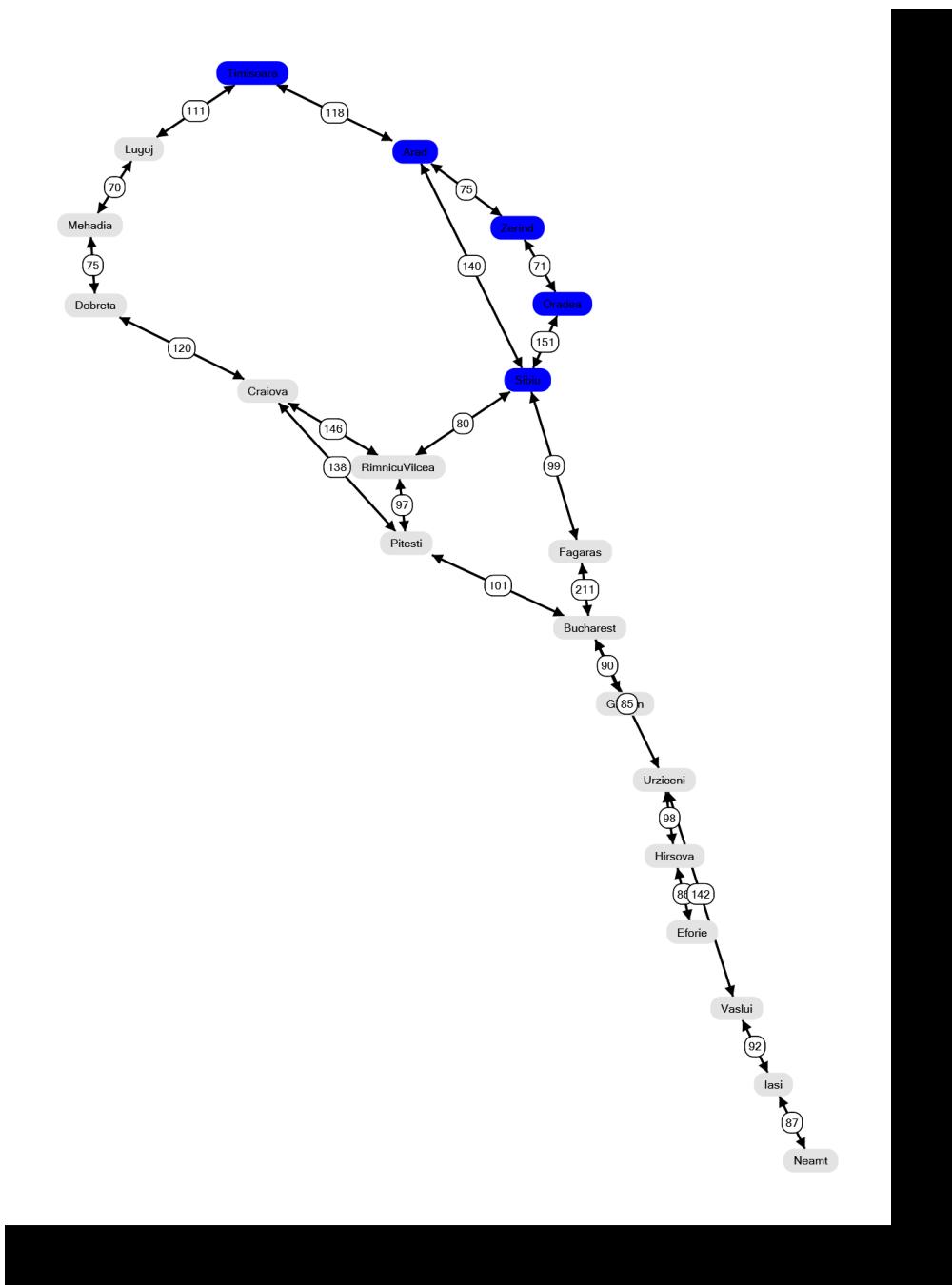
Visit

The algorithm has marked the node Oradea as visited



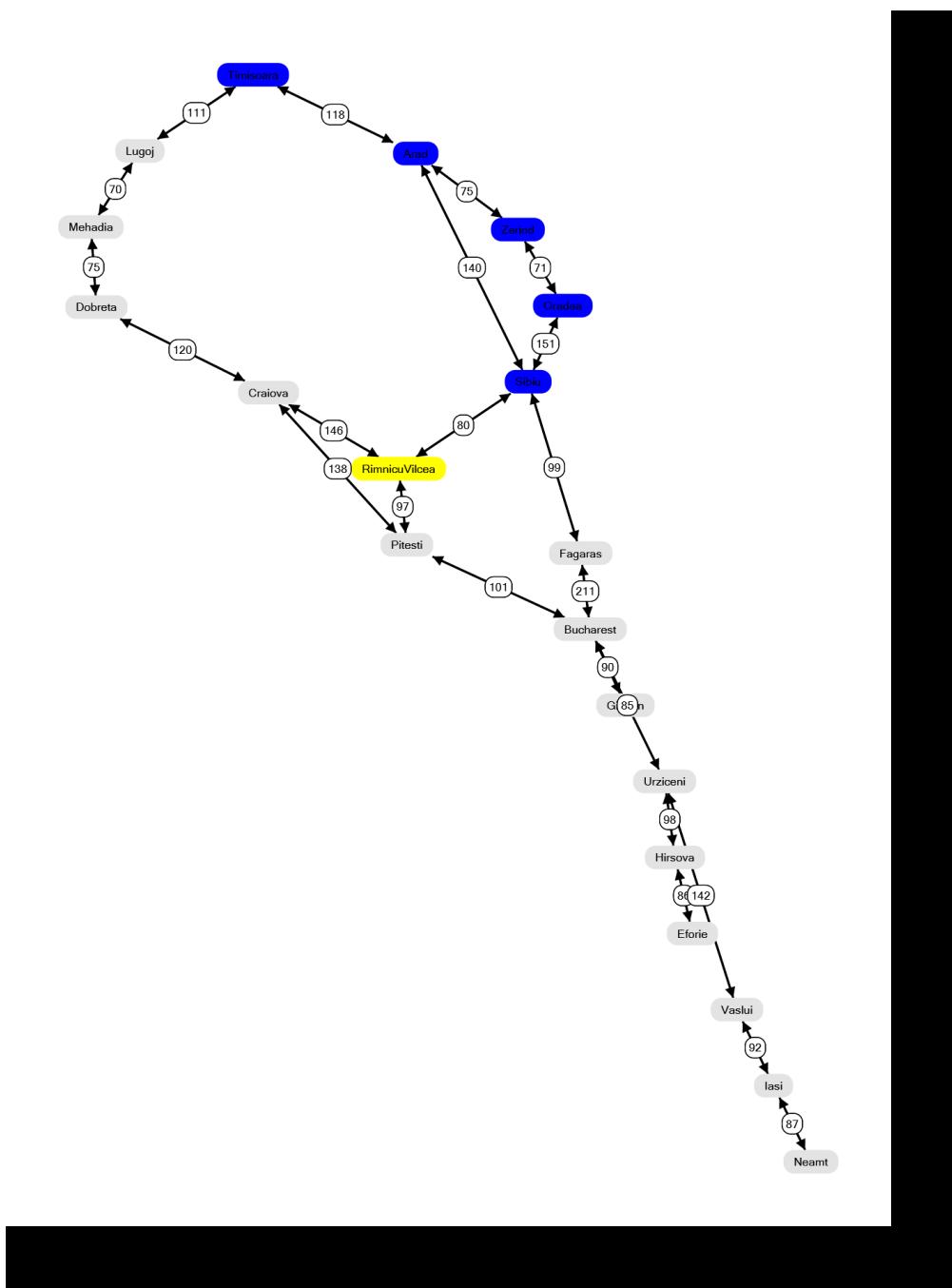
Post-Visit

The algorithm has closed the node Oradea



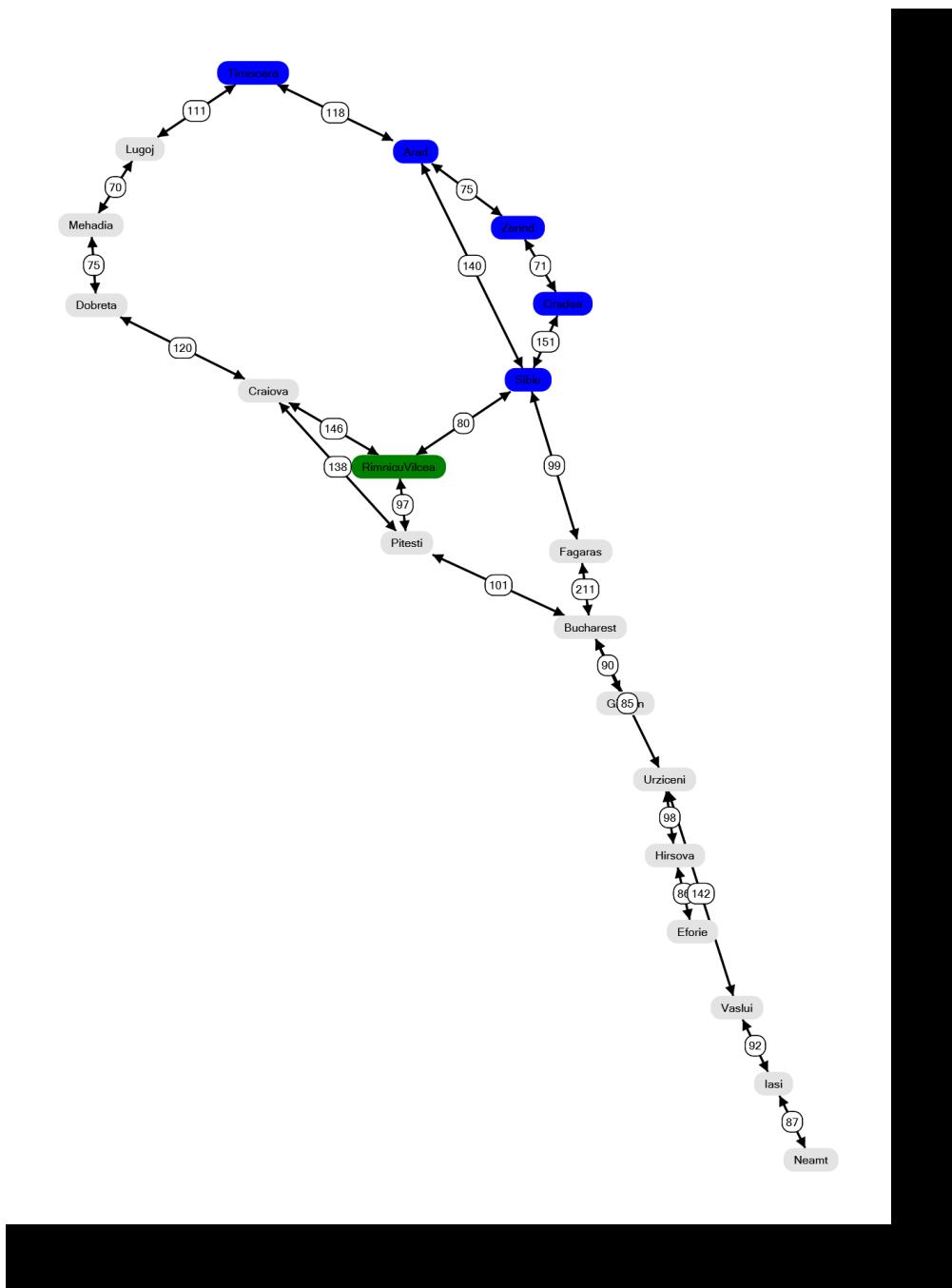
Pre-Visit

The algorithm has opened the node Rimnicu Vilcea



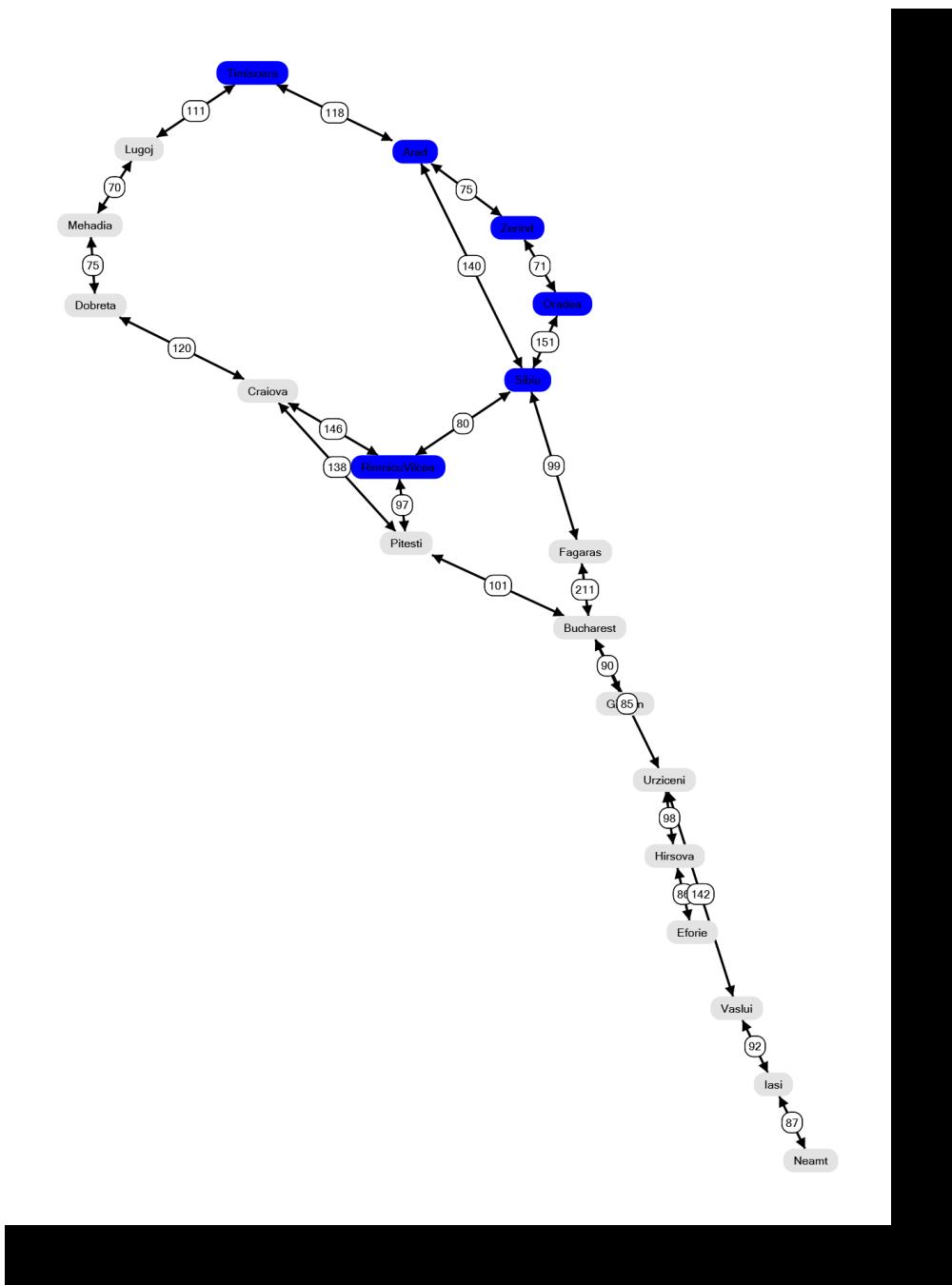
Visit

The algorithm has marked the node Rimnicu Vilcea as visited



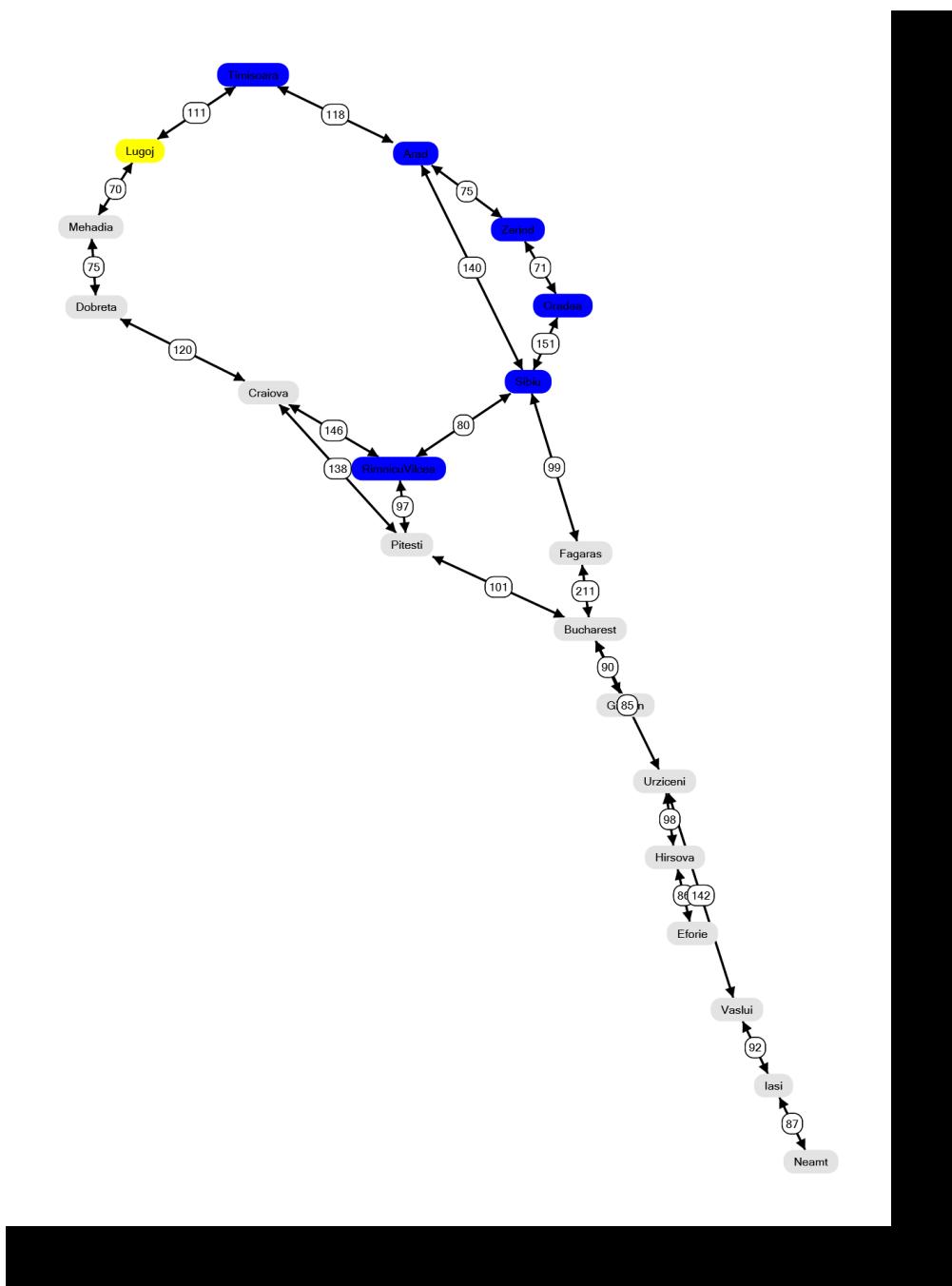
Post-Visit

The algorithm has closed the node Rimnicu Vilcea



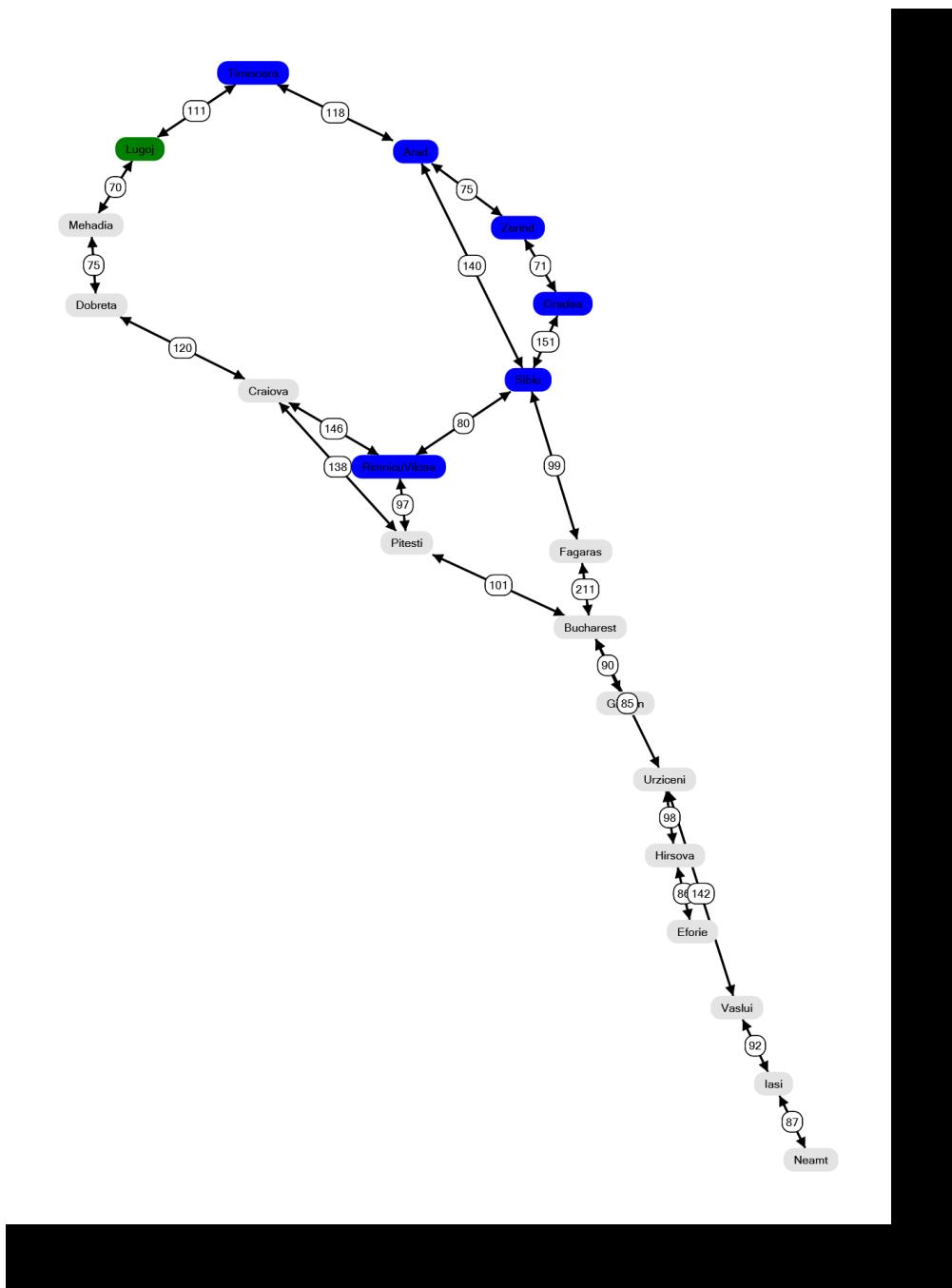
Pre-Visit

The algorithm has opened the node Lugoj



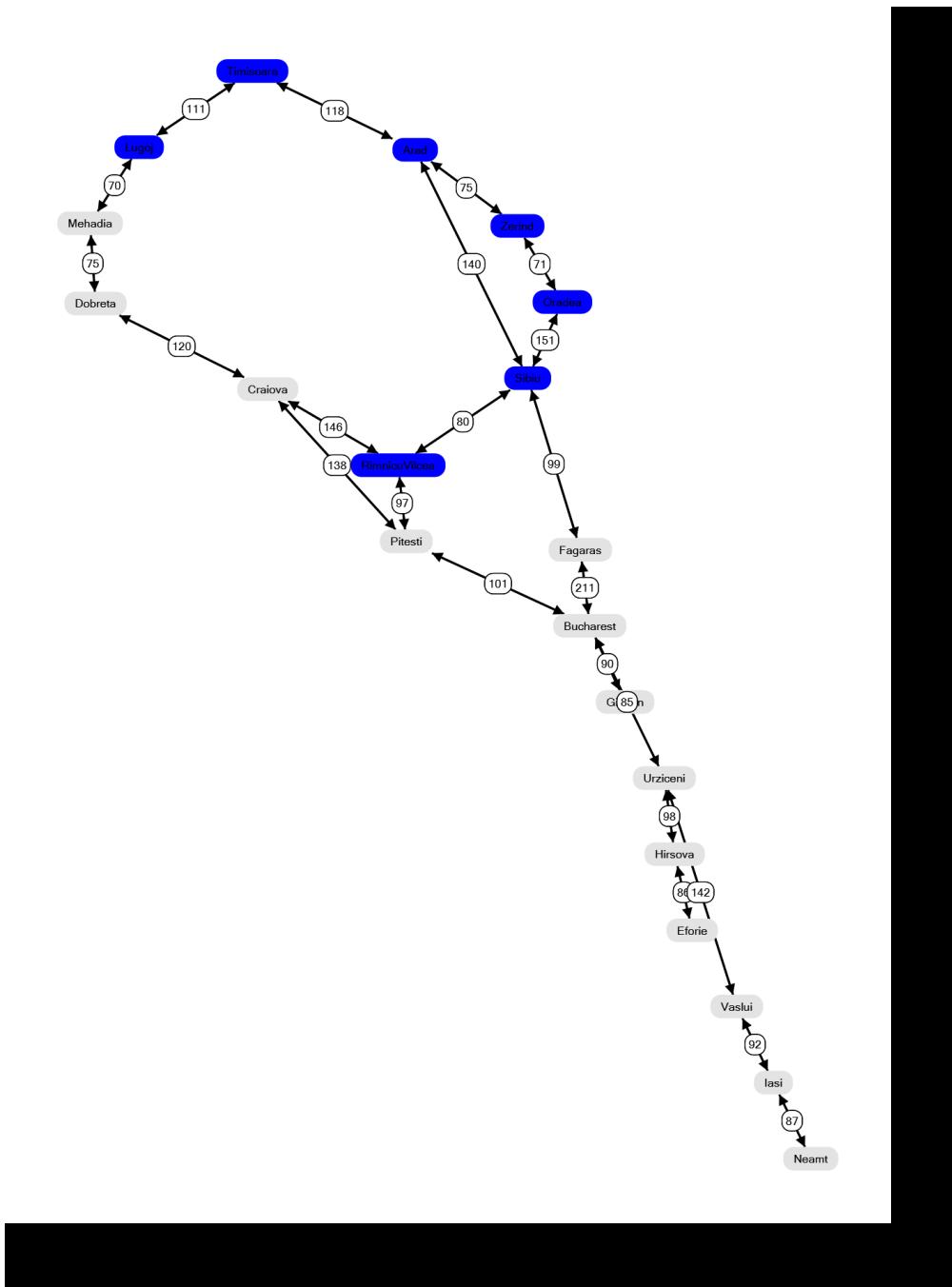
Visit

The algorithm has marked the node Lugoj as visited



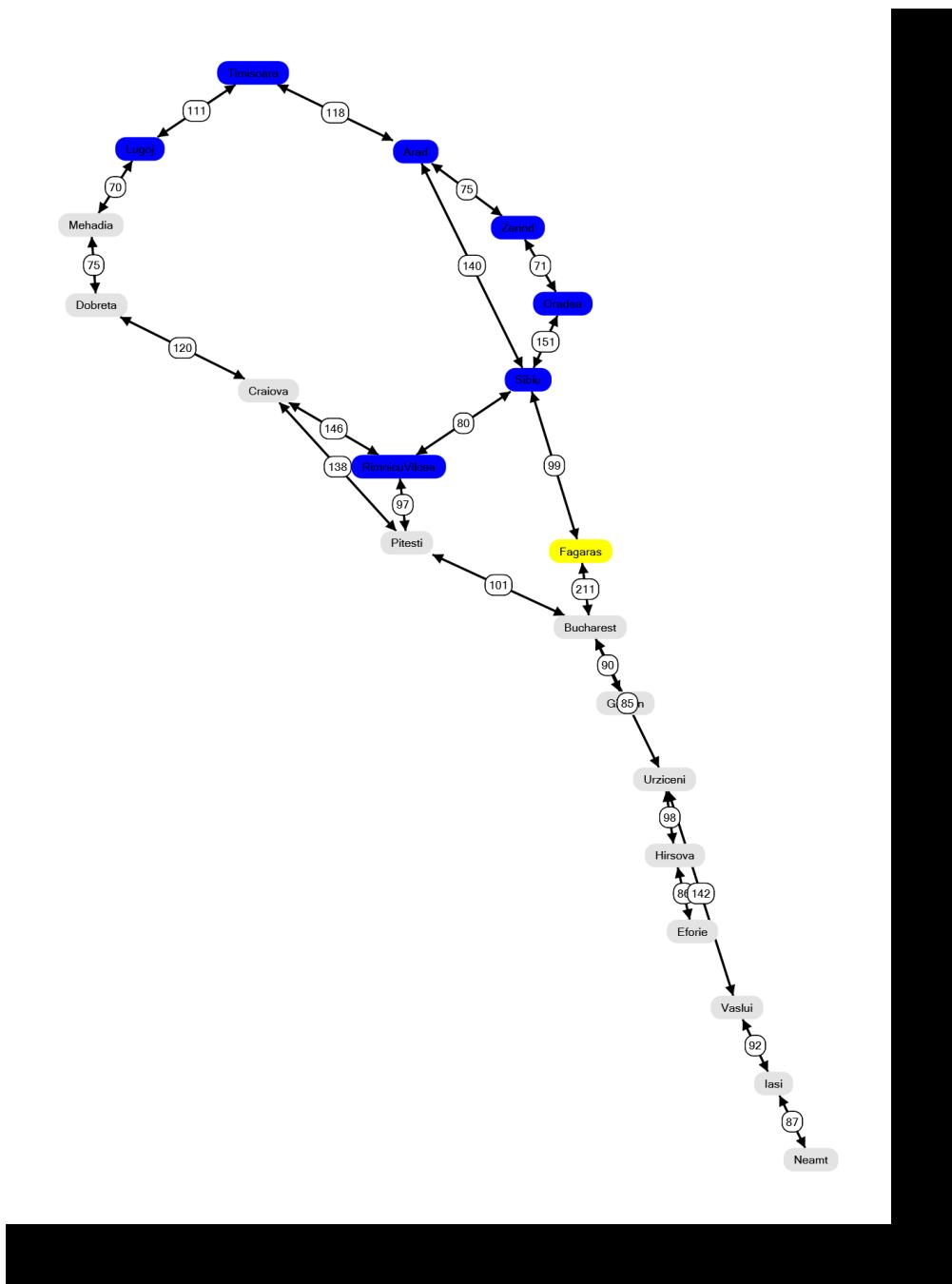
Post-Visit

The algorithm has closed the node Lugoj



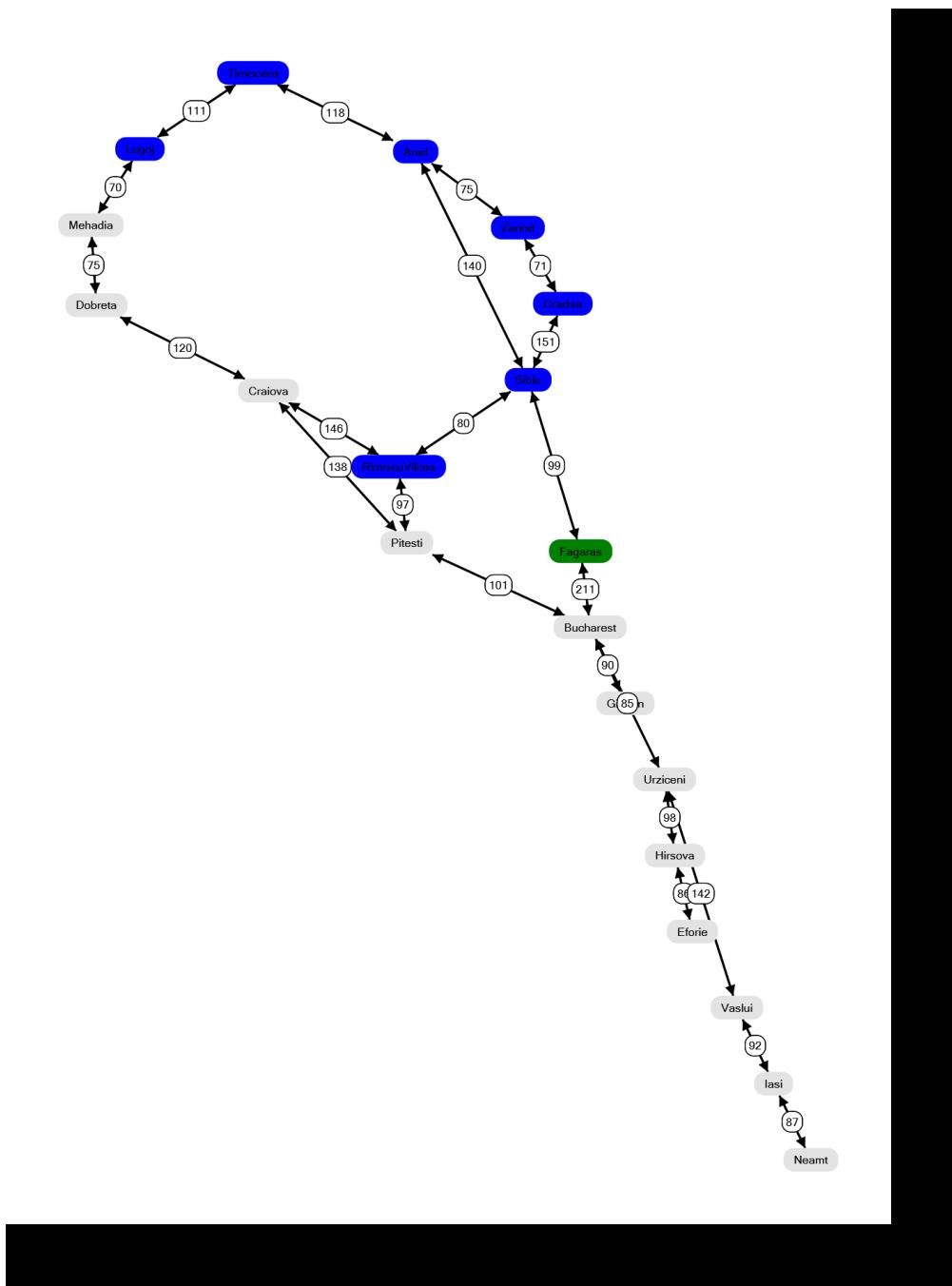
Pre-Visit

The algorithm has opened the node Fagaras



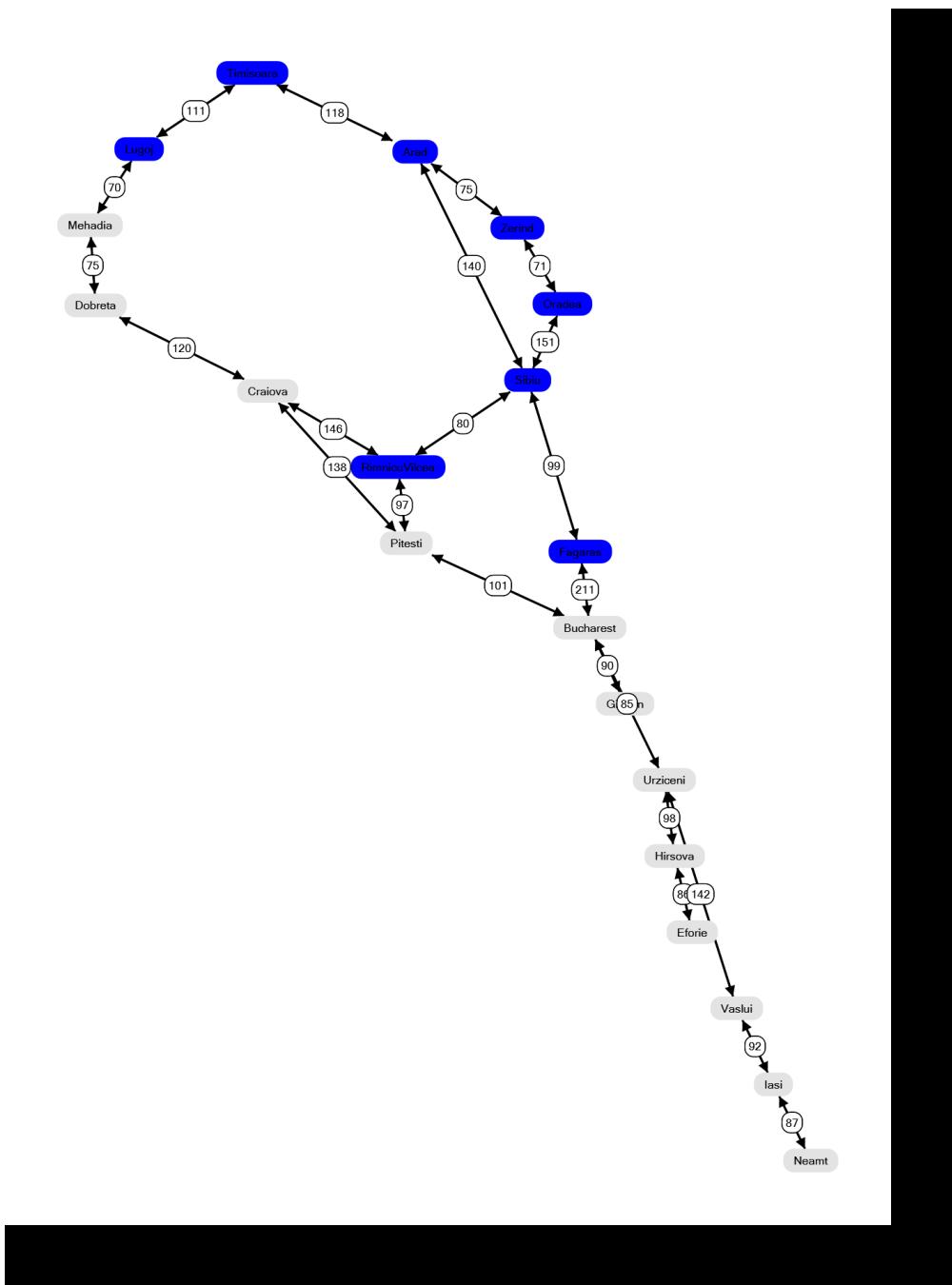
Visit

The algorithm has marked the node Fagaras as visited



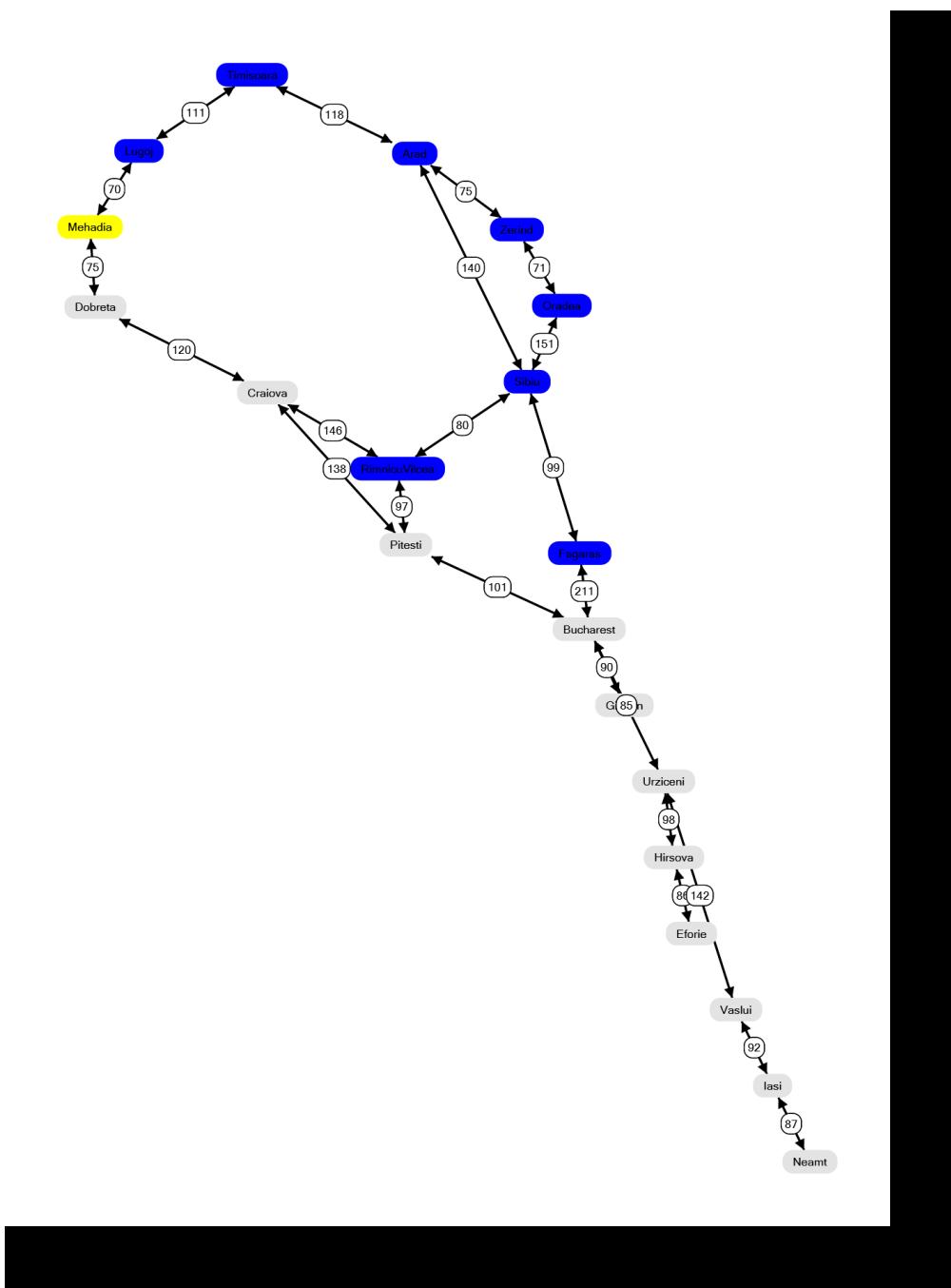
Post-Visit

The algorithm has closed the node Fagaras



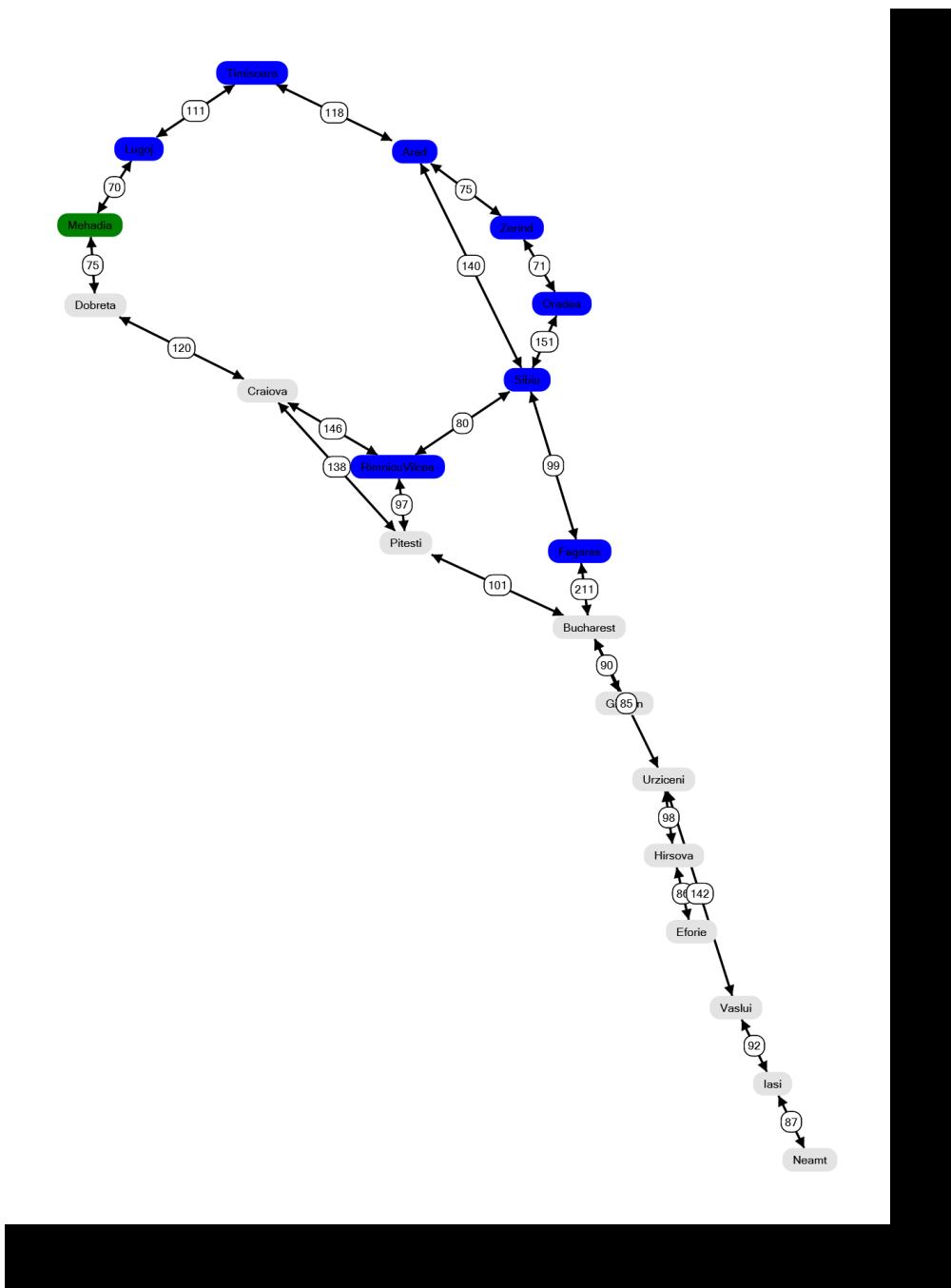
Pre-Visit

The algorithm has opened the node Mehadia



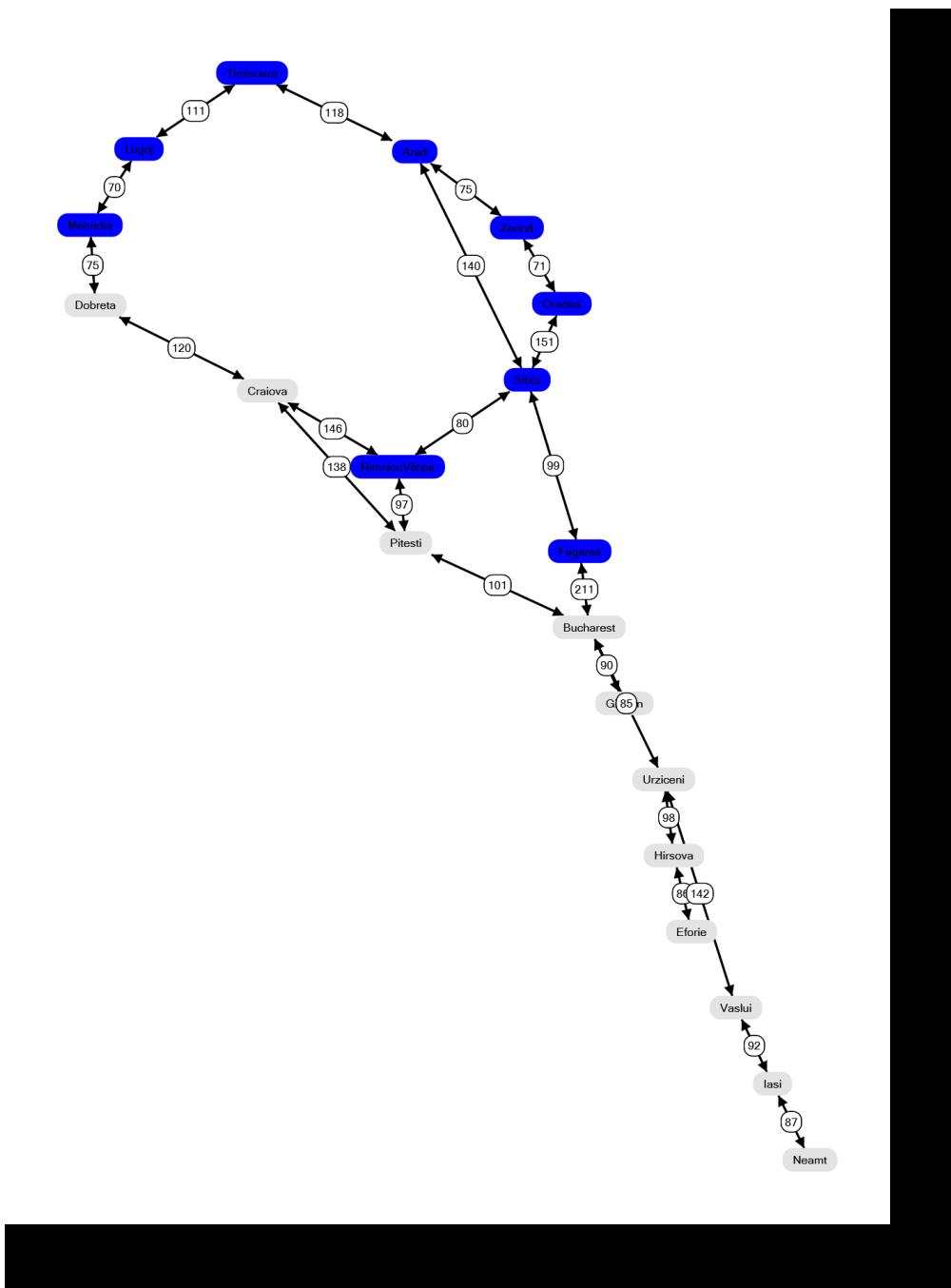
Visit

The algorithm has marked the node Mehadia as visited



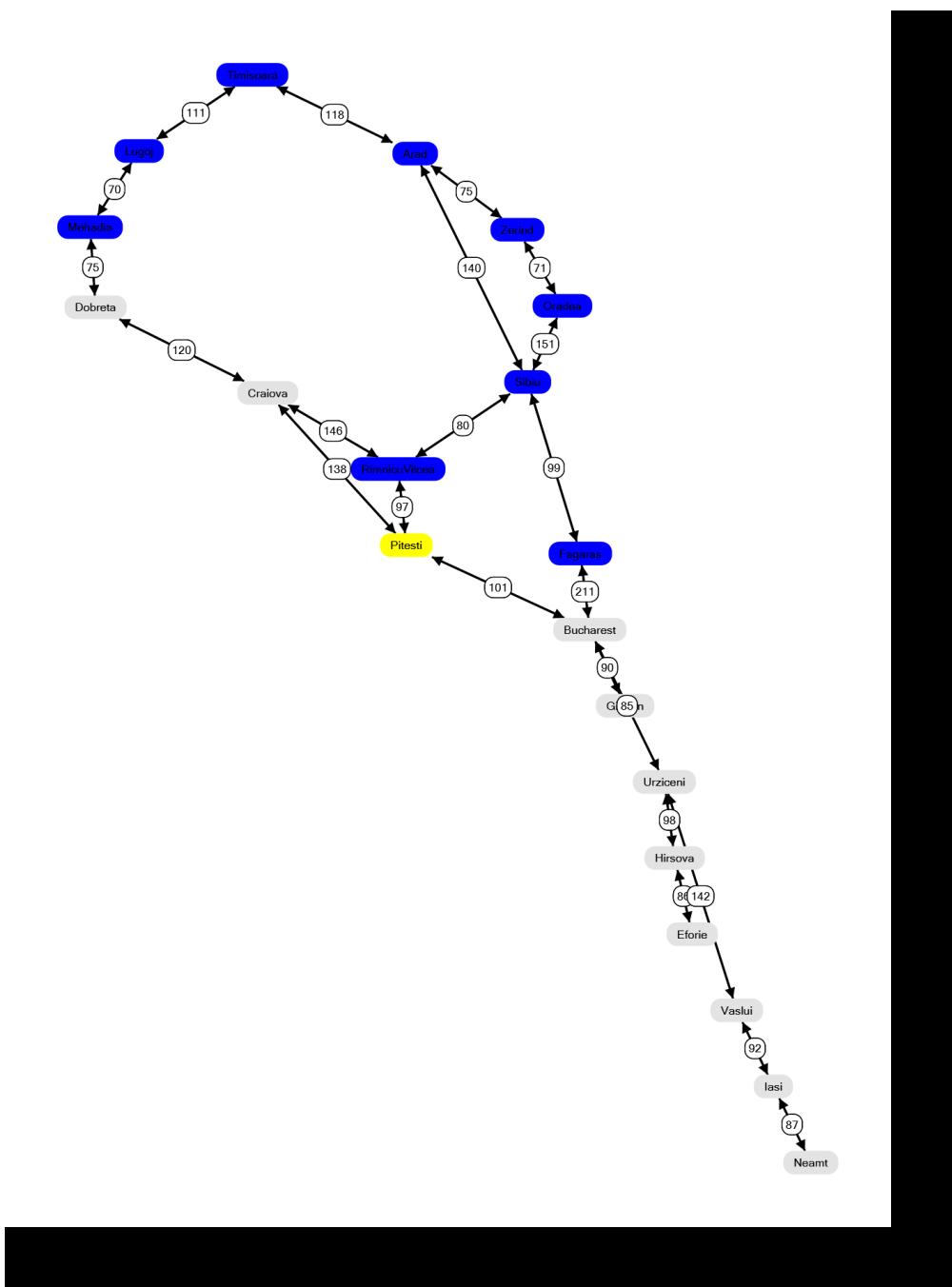
Post-Visit

The algorithm has closed the node Mehadia



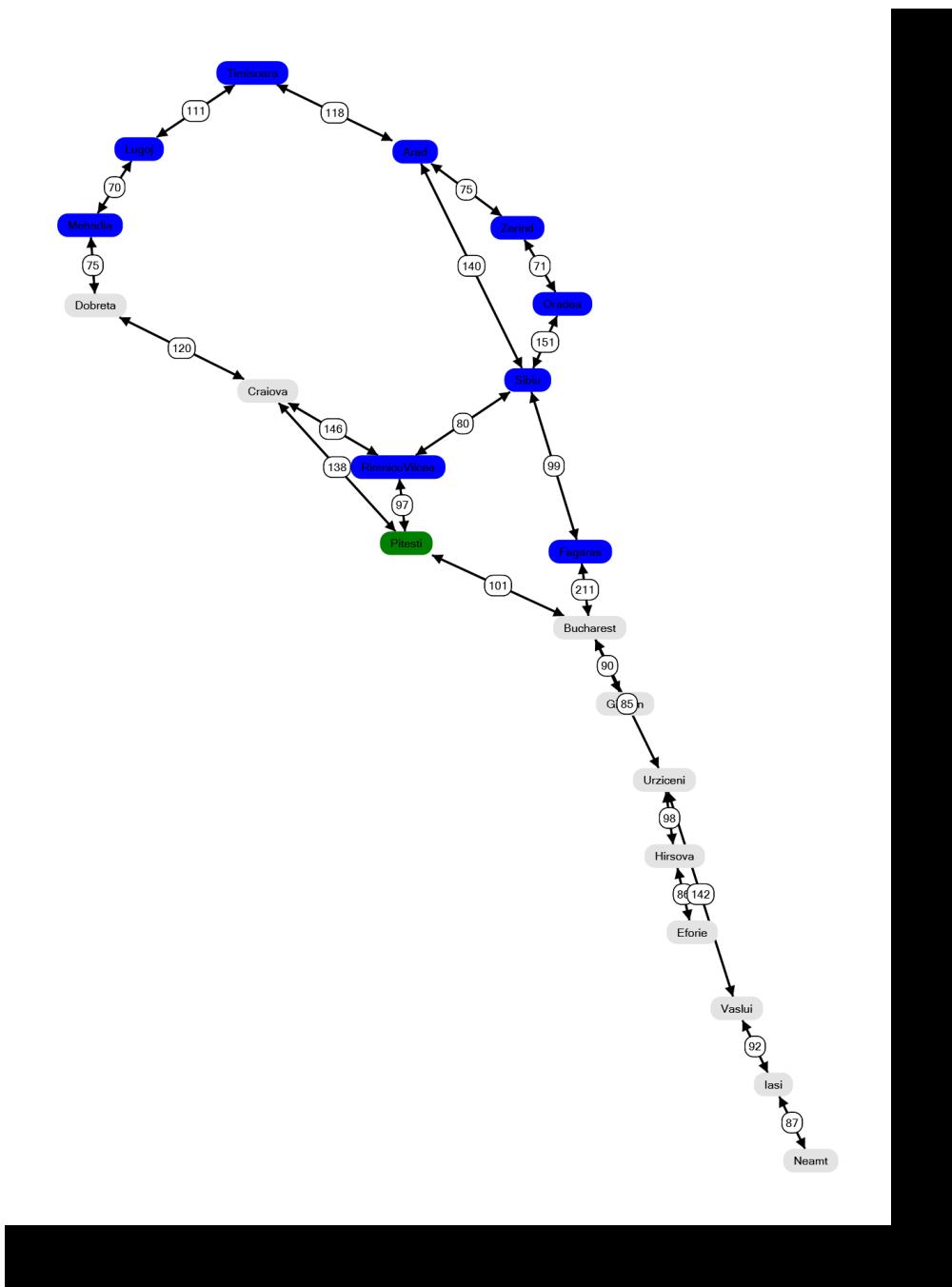
Pre-Visit

The algorithm has opened the node Pitesti



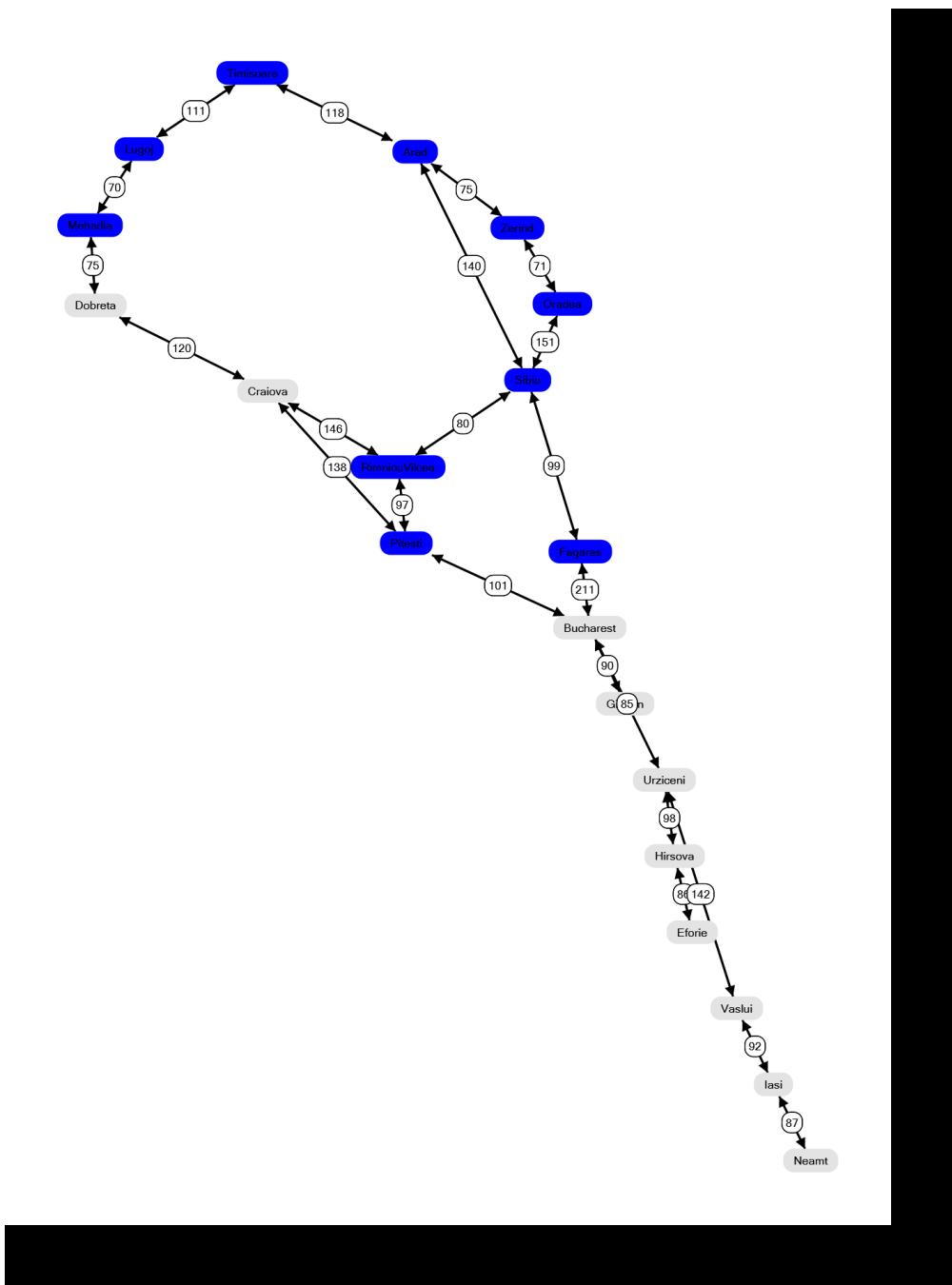
Visit

The algorithm has marked the node Pitesti as visited



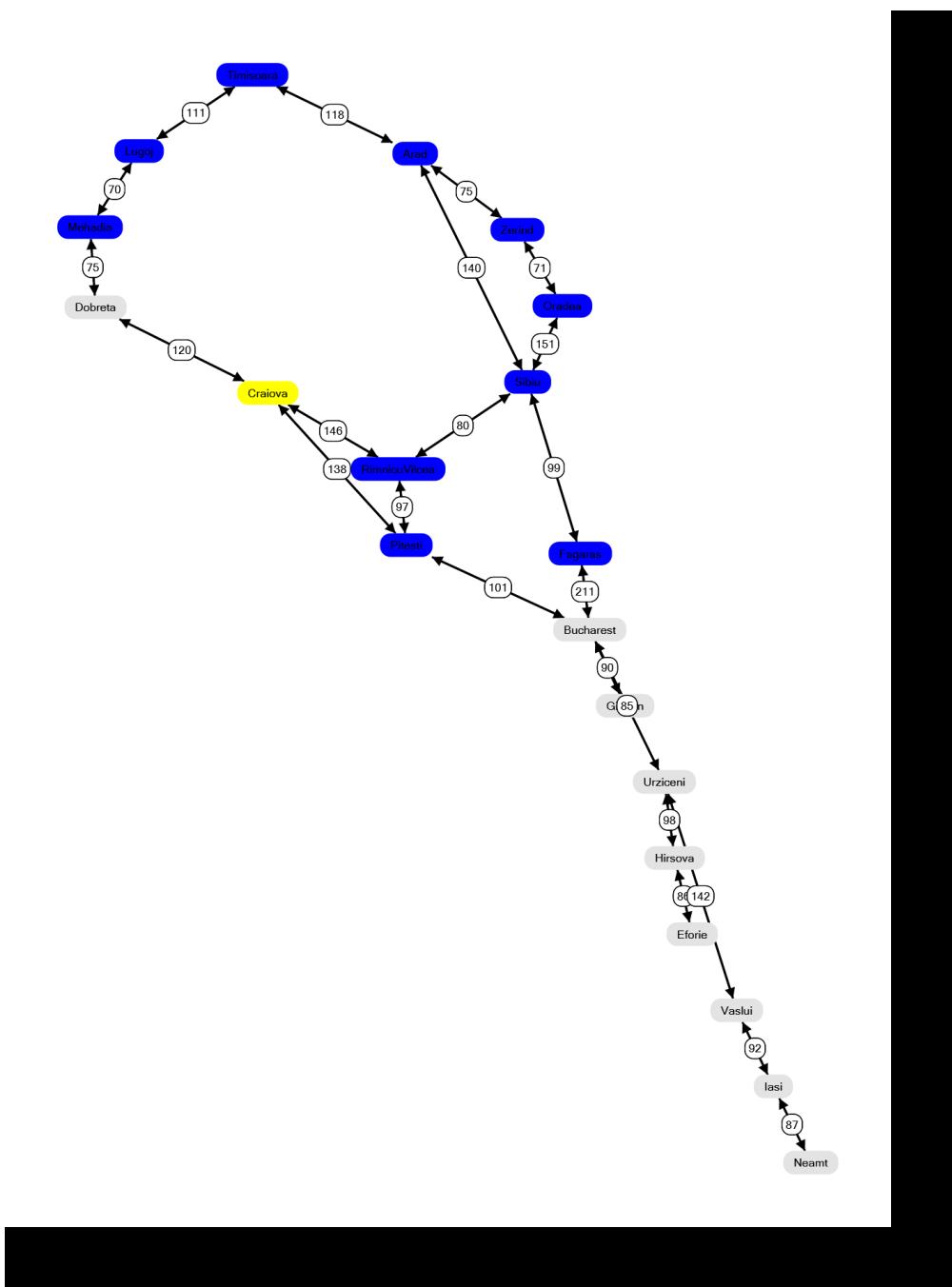
Post-Visit

The algorithm has closed the node Pitesti



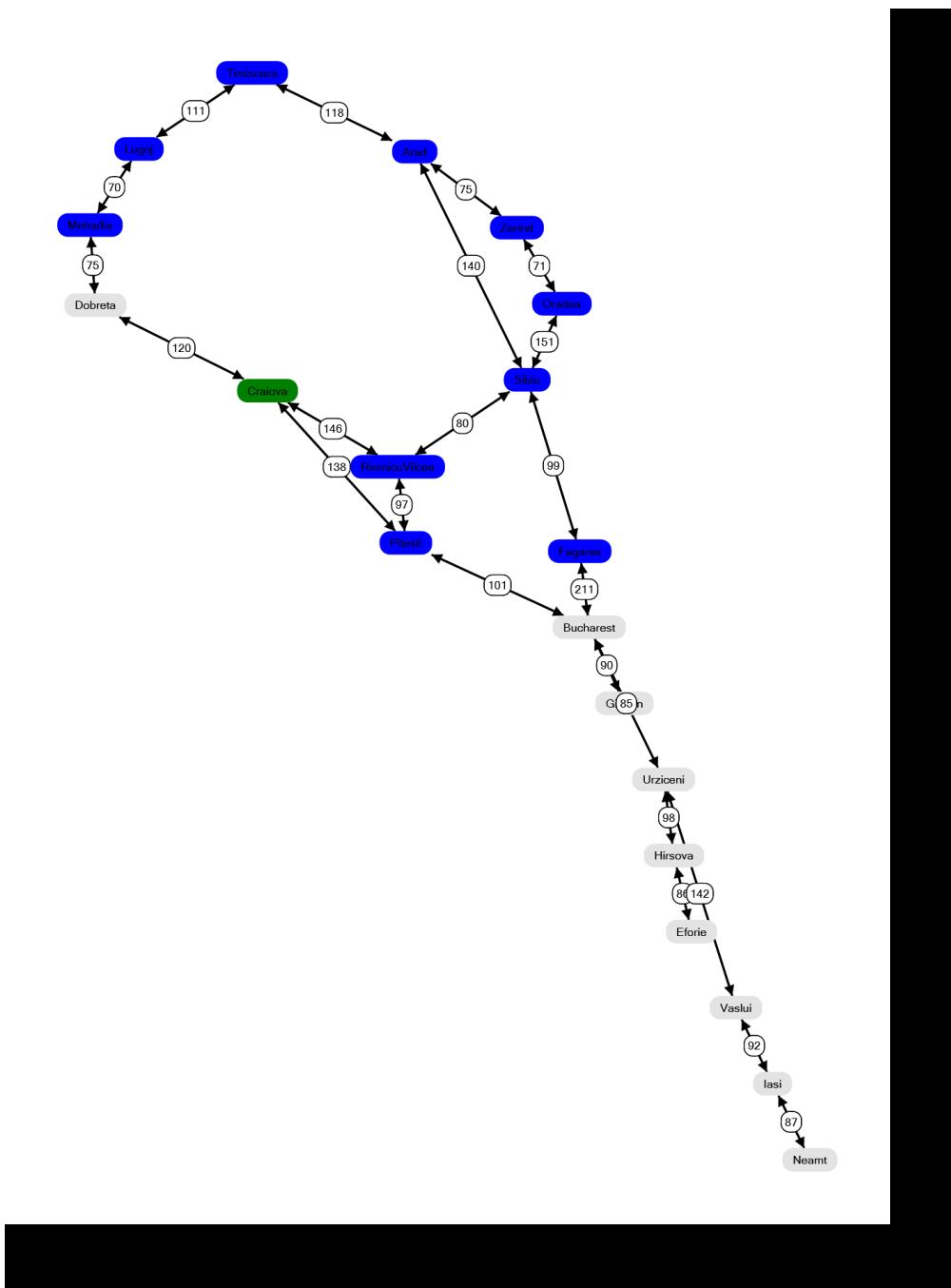
Pre-Visit

The algorithm has opened the node Craiova



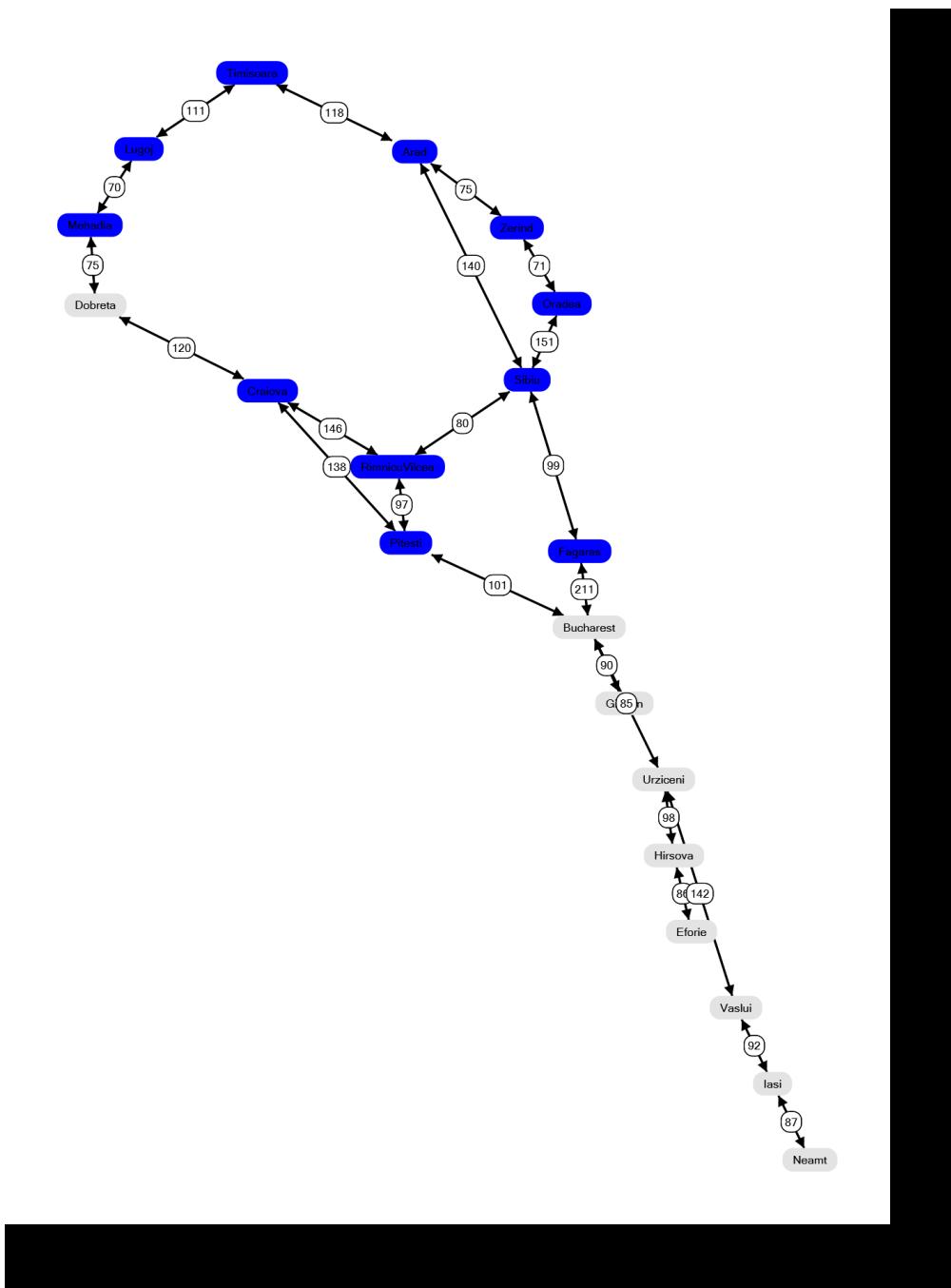
Visit

The algorithm has marked the node Craiova as visited



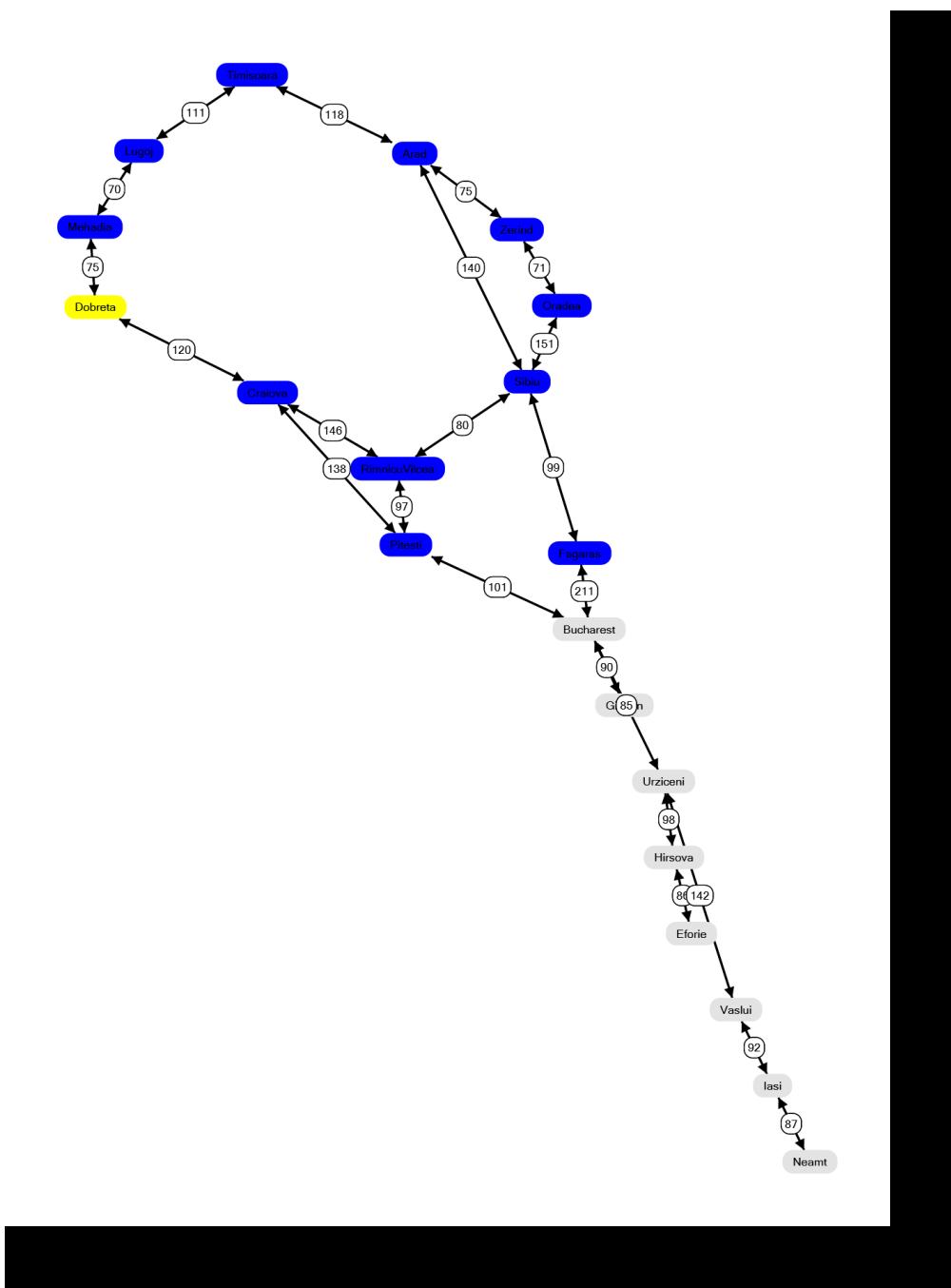
Post-Visit

The algorithm has closed the node Craiova



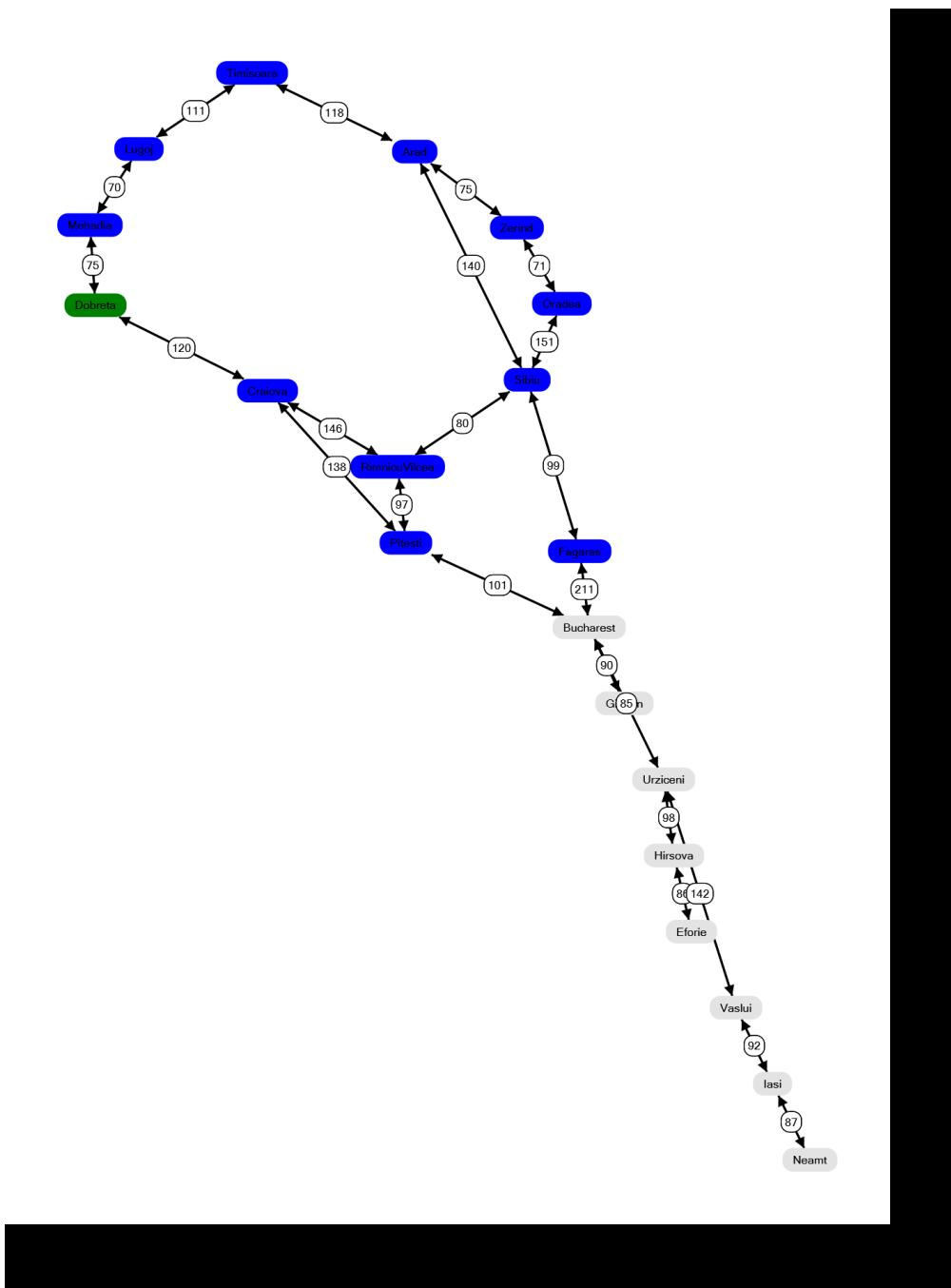
Pre-Visit

The algorithm has opened the node Dobreta



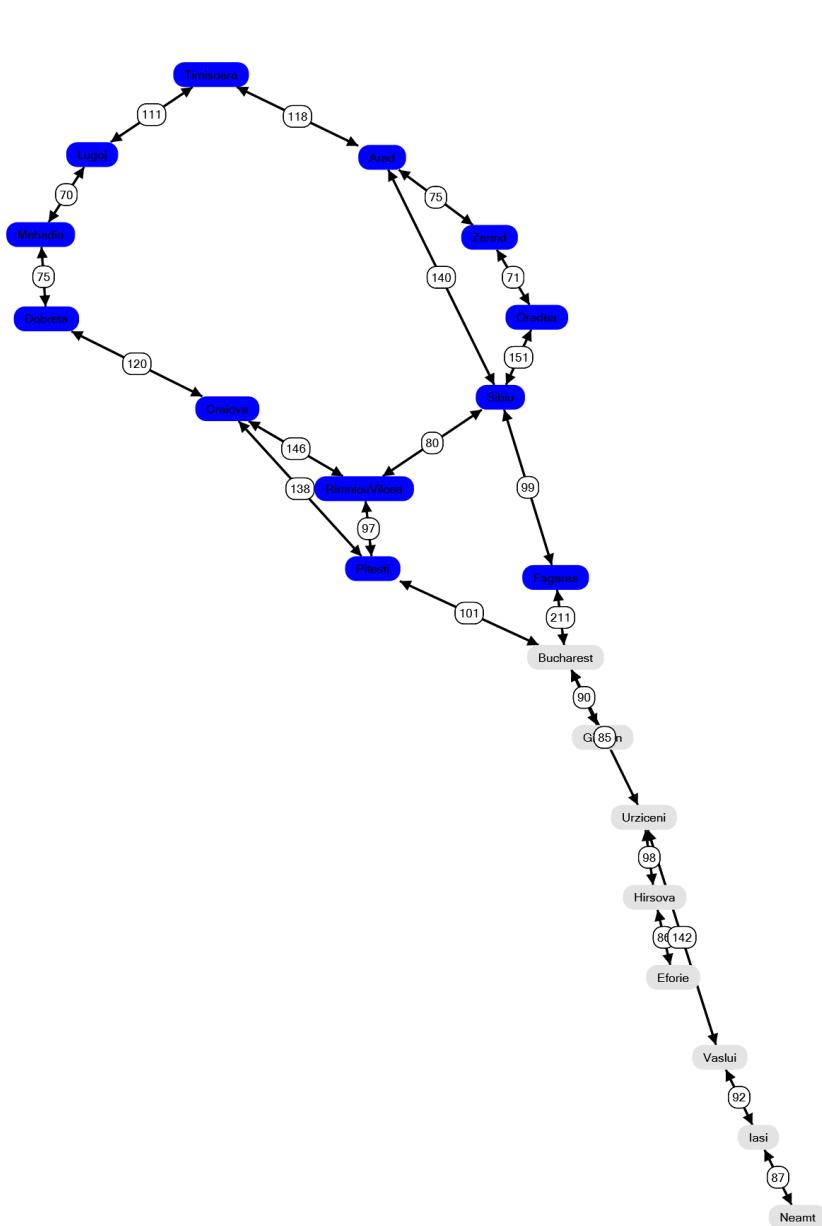
Visit

The algorithm has marked the node Dobreta as visited



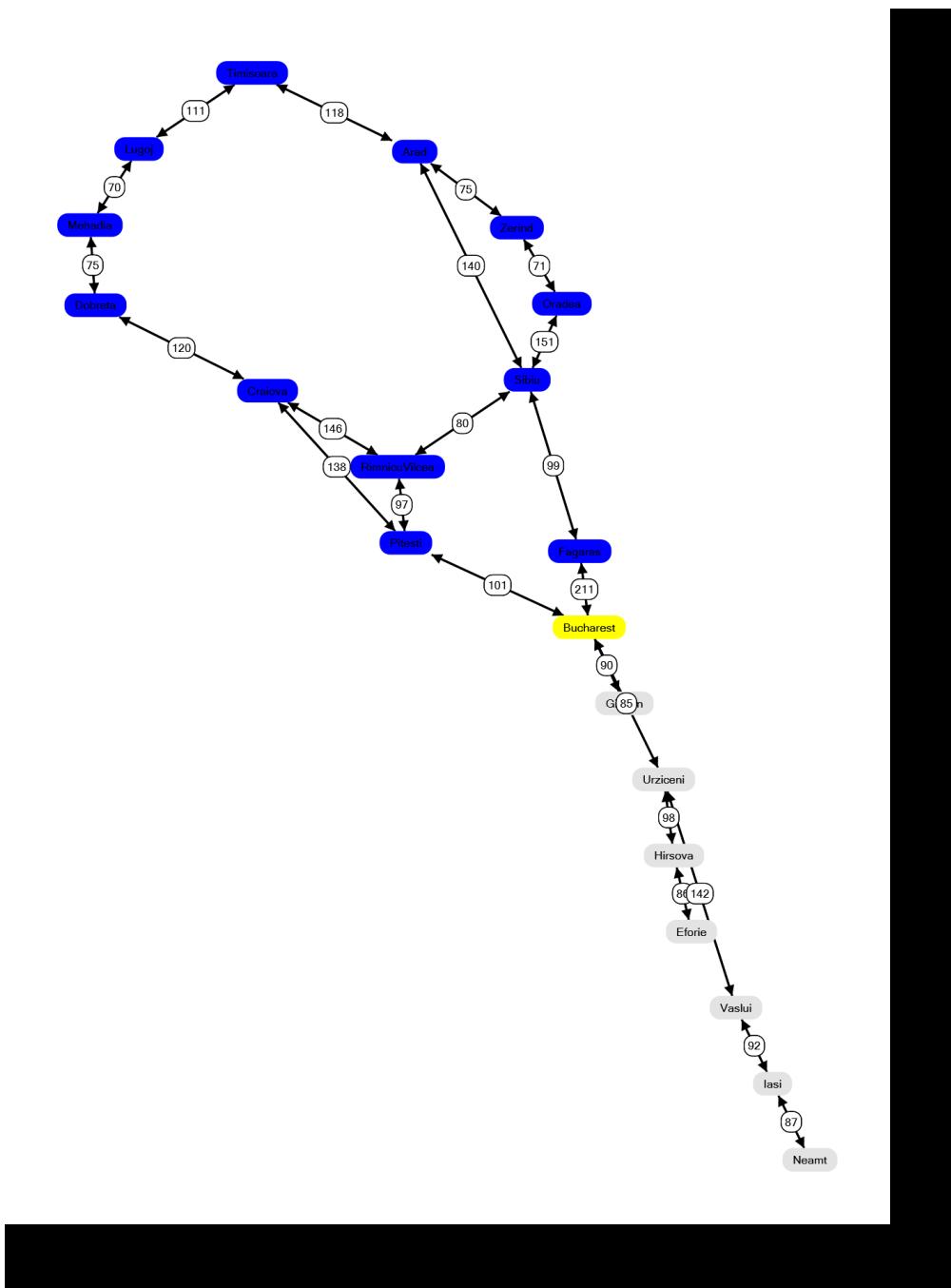
Post-Visit

The algorithm has closed the node Dobreta



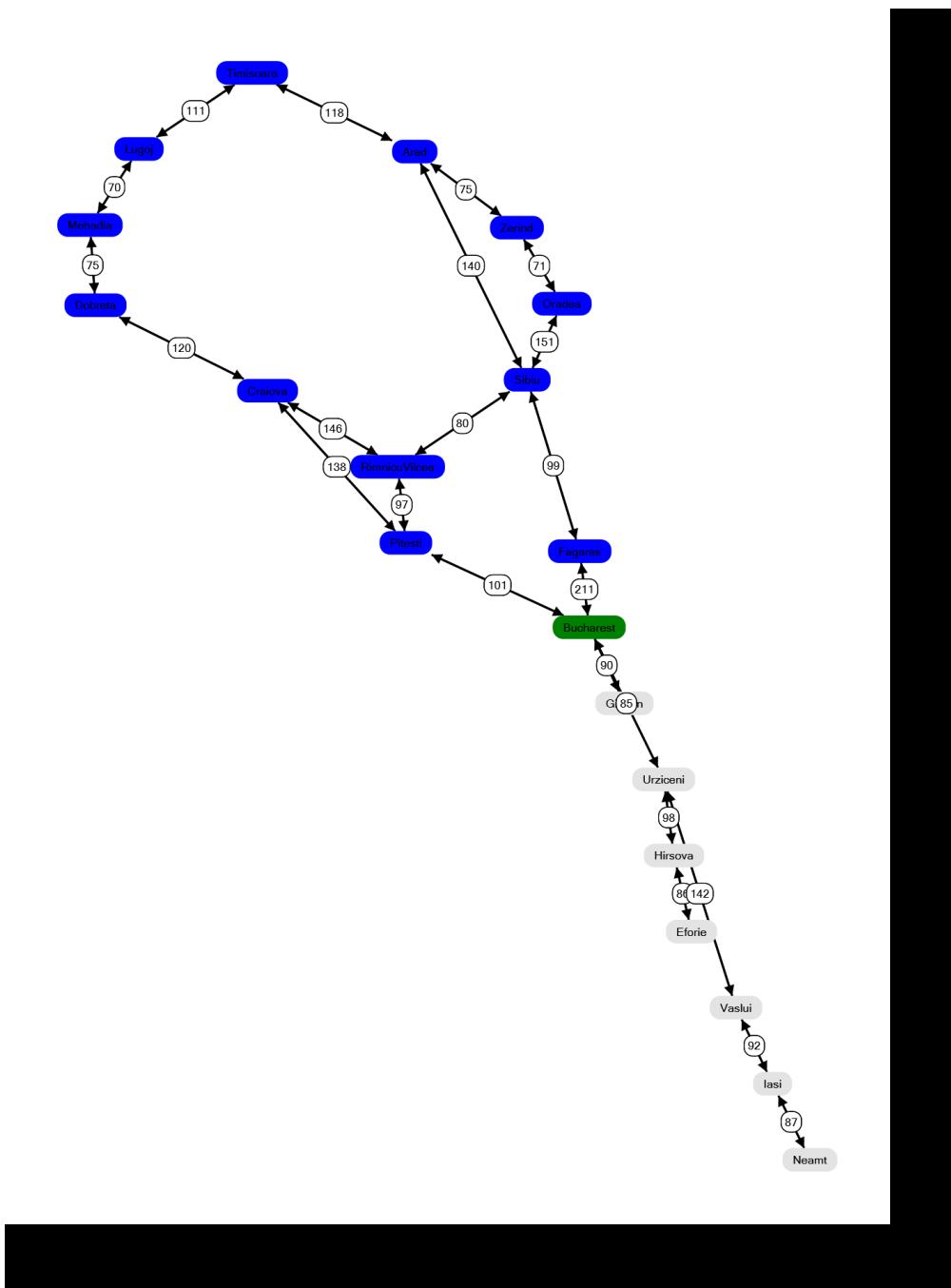
Pre-Visit

The algorithm has opened the node Bucharest



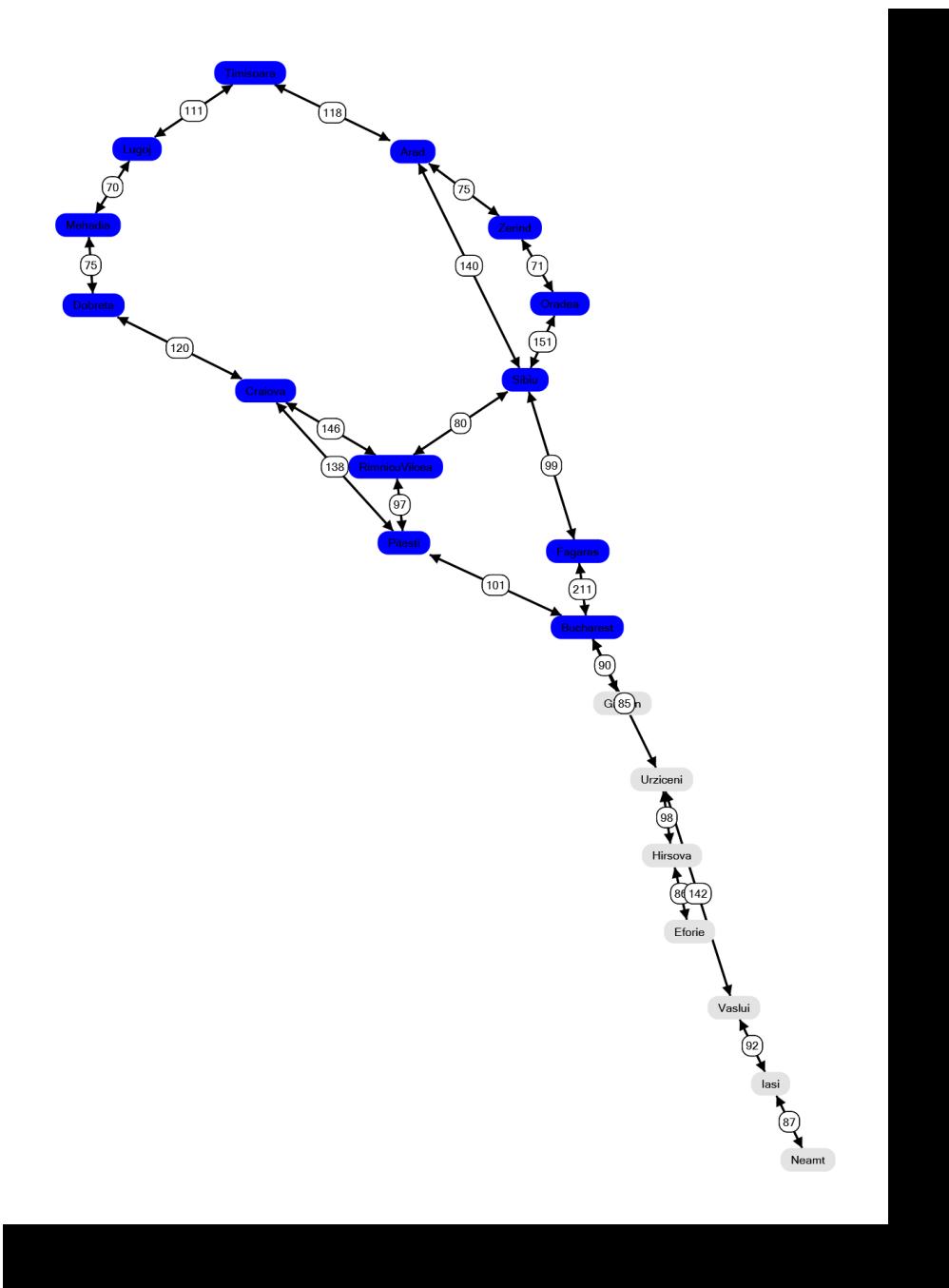
Visit

The algorithm has marked the node Bucharest as visited



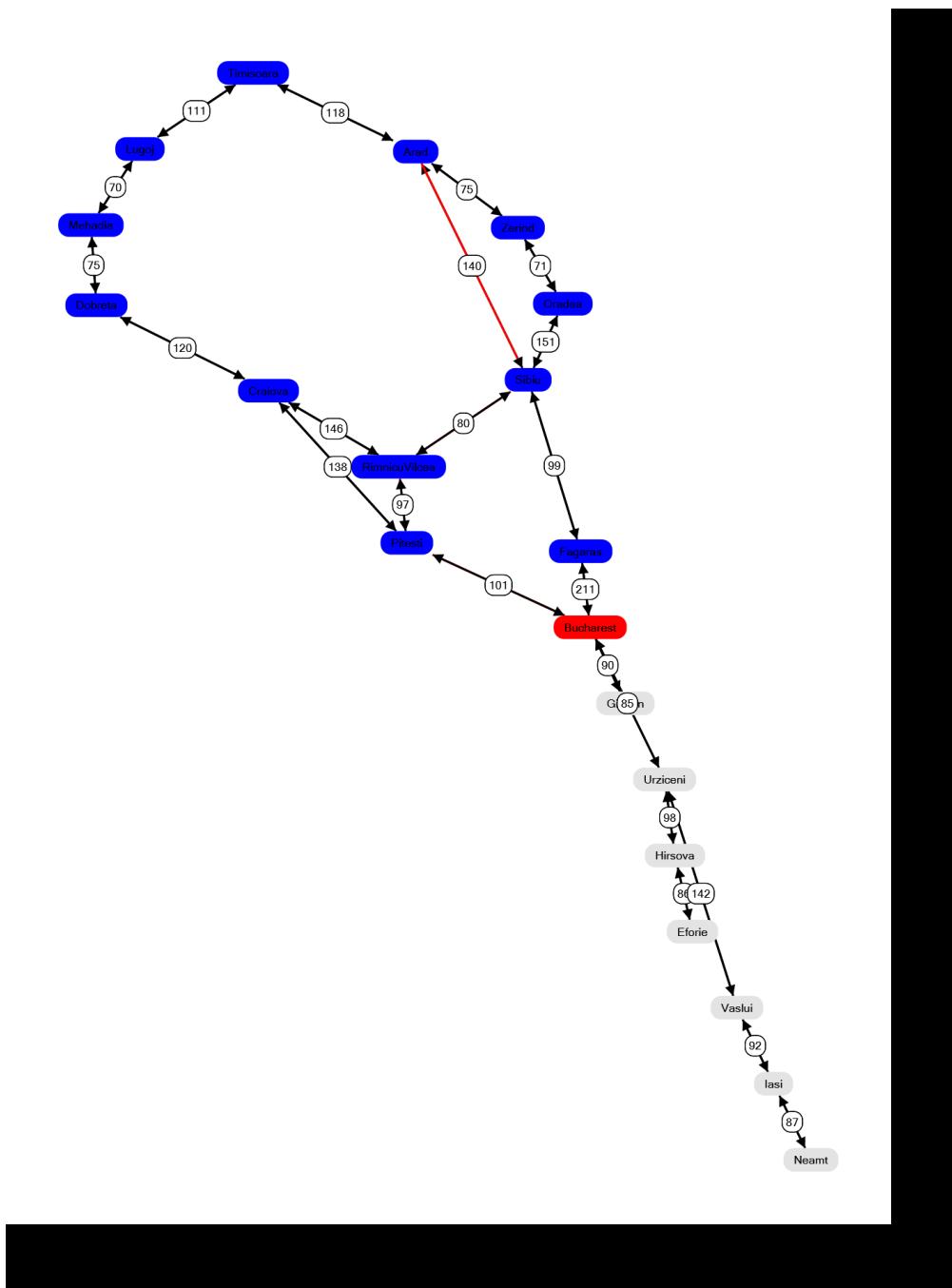
Post-Visit

The algorithm has closed the node Bucharest



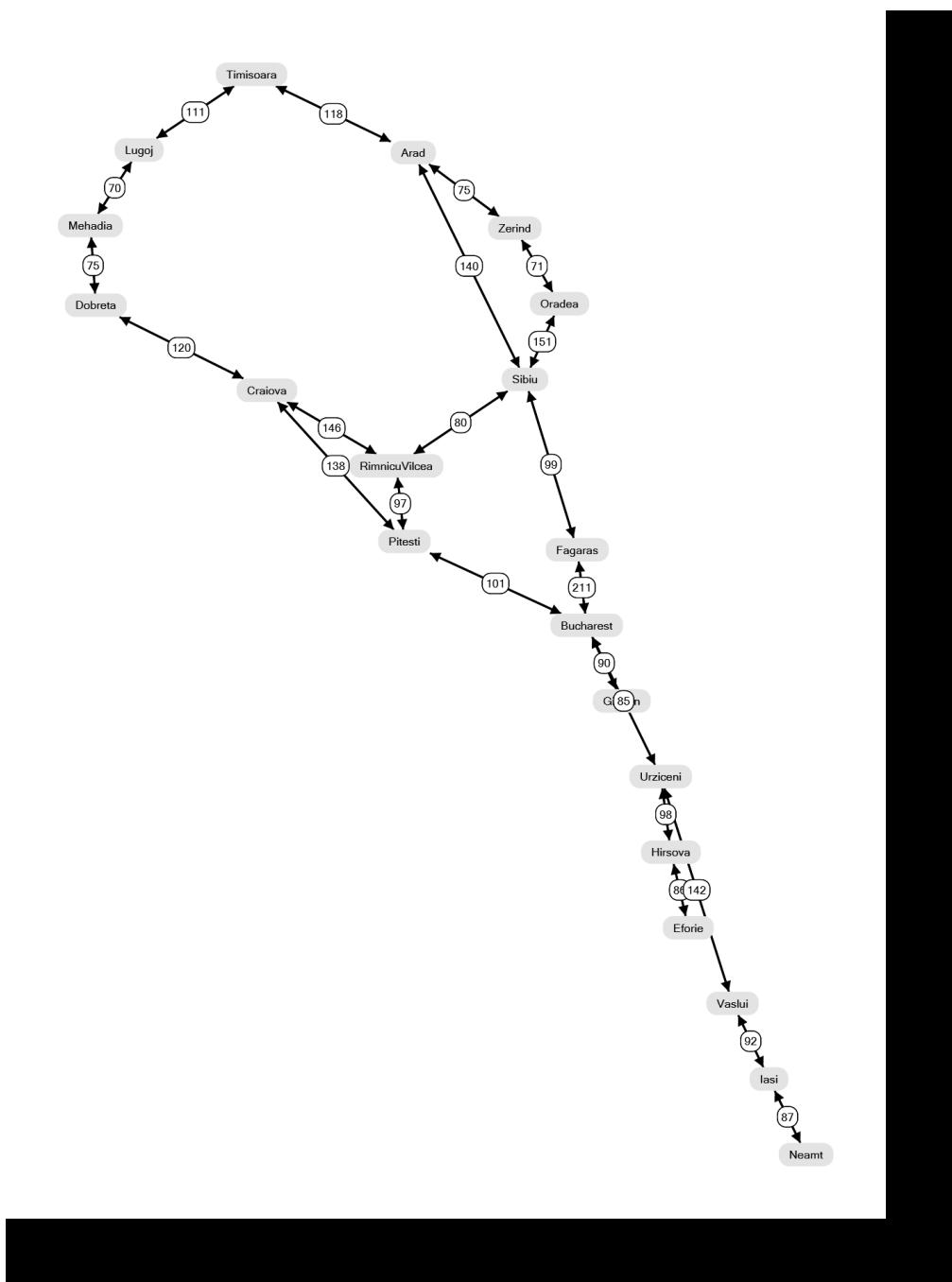
Result Found

The algorithm has found the target node Bucharest



Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))

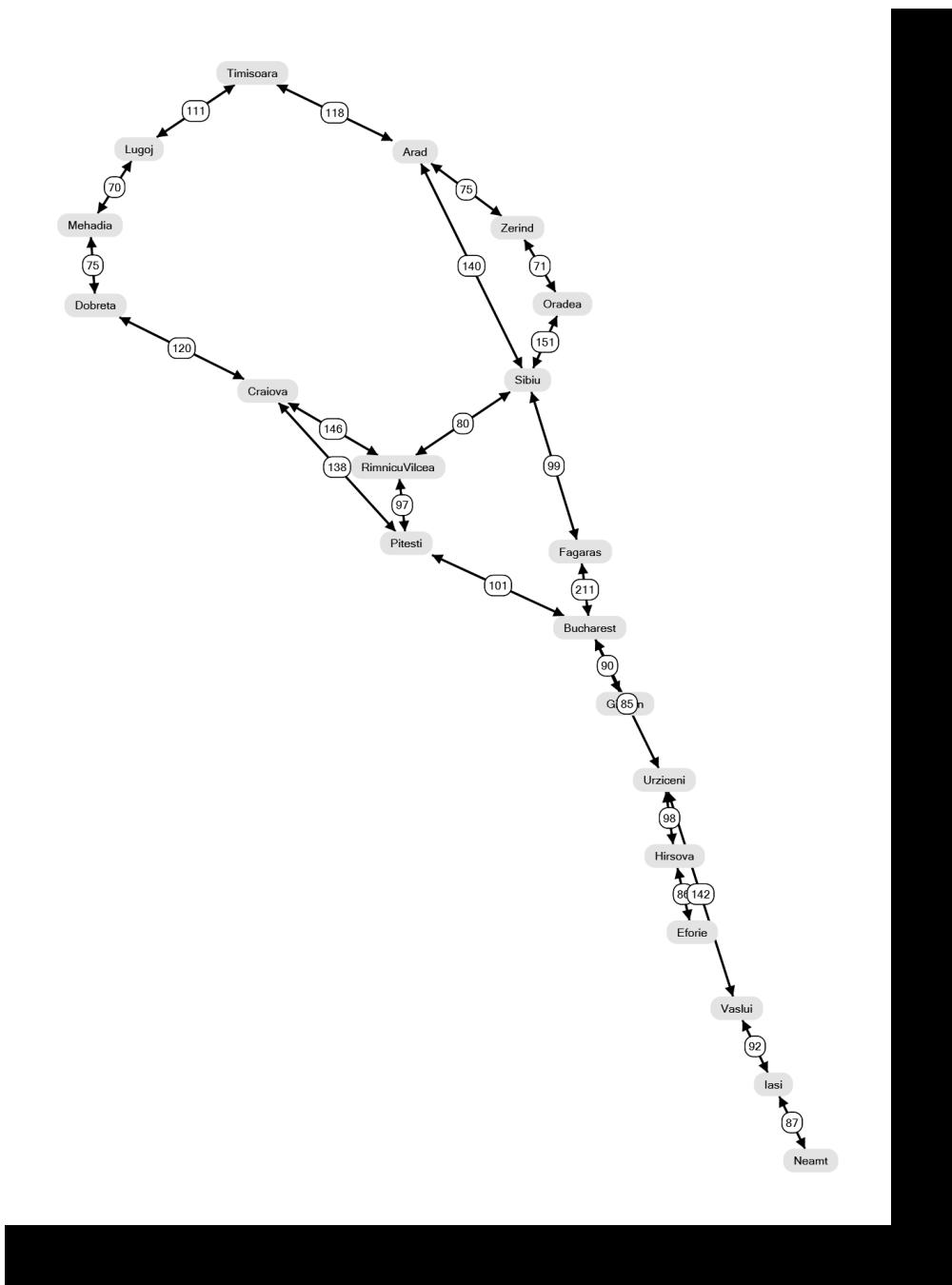


Iterative Depth limited search

In computer science, iterative deepening search or more specifically iterative deepening depth-first search (IDS or IDDFS) is a state space/graph search strategy in which a depth-limited version of depth-first search is run repeatedly with increasing depth limits until the goal is found. IDDFS is equivalent to breadth-first search, but uses much less memory; on each iteration, it visits the nodes in the search tree in the same order as depth-first search, but the cumulative order in which nodes are first visited is effectively breadth-first. The algorithm took 00:00:00.0012527 to find the goal.

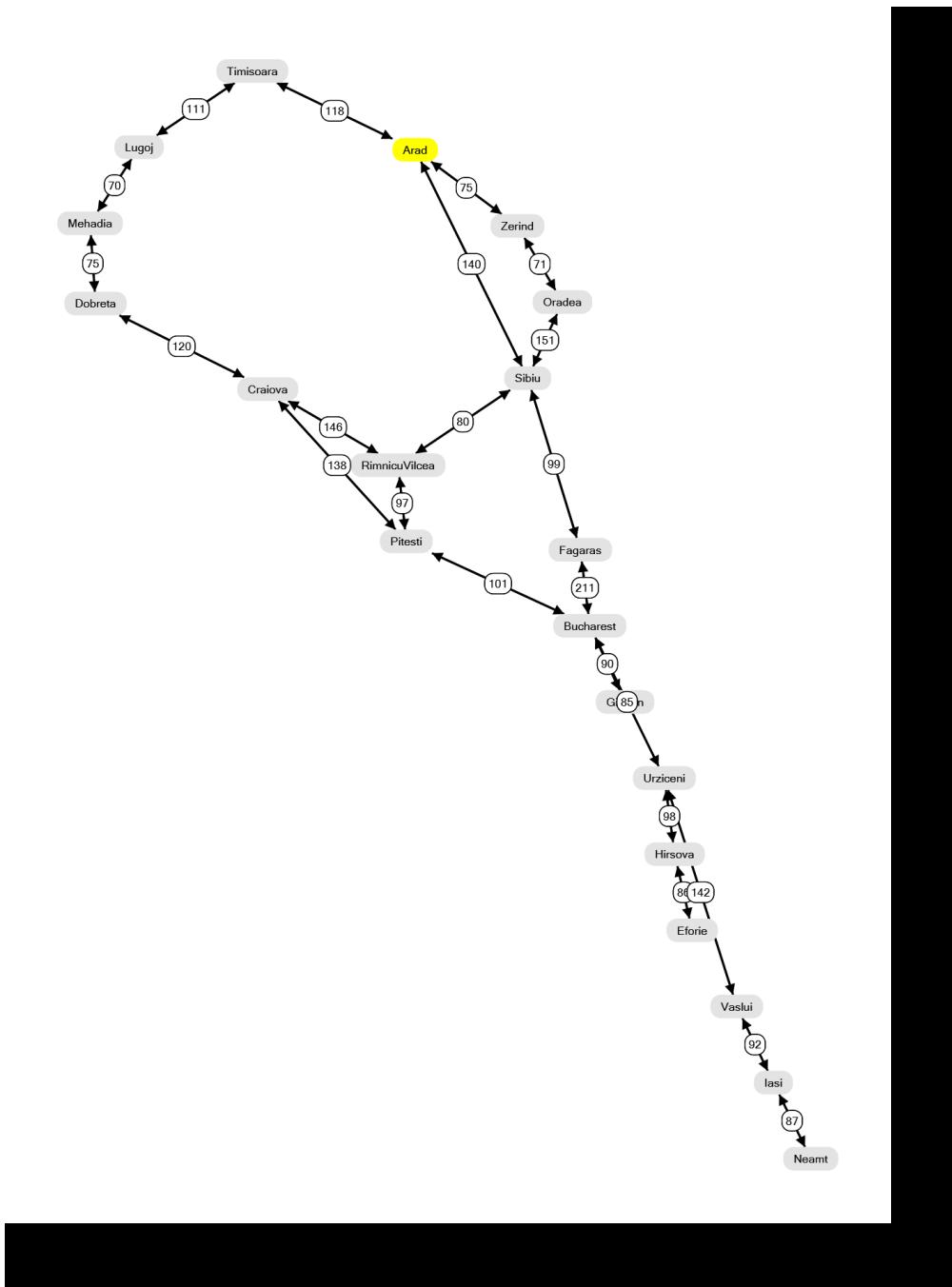
Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



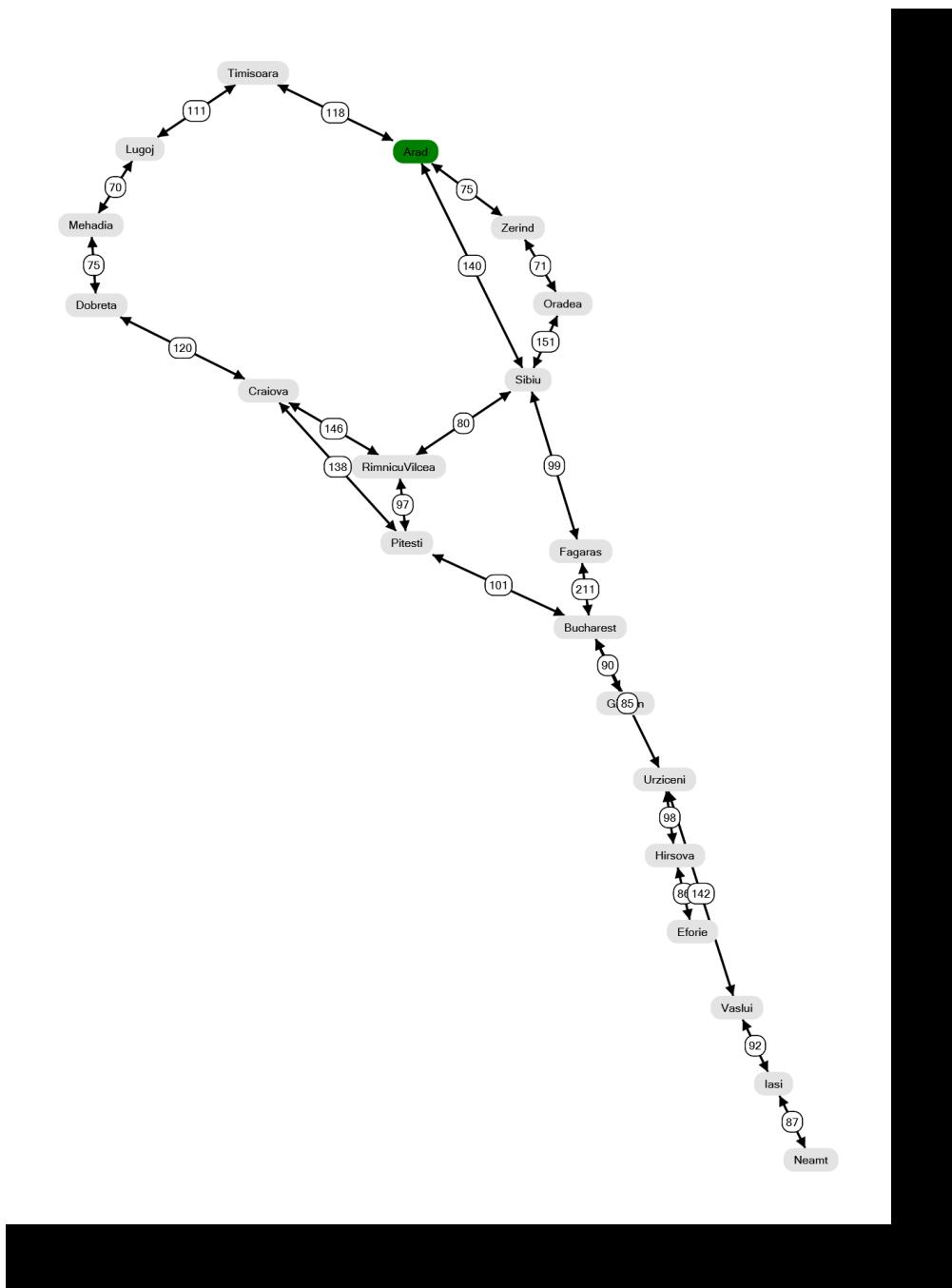
Pre-Visit

The algorithm has opened the node Arad



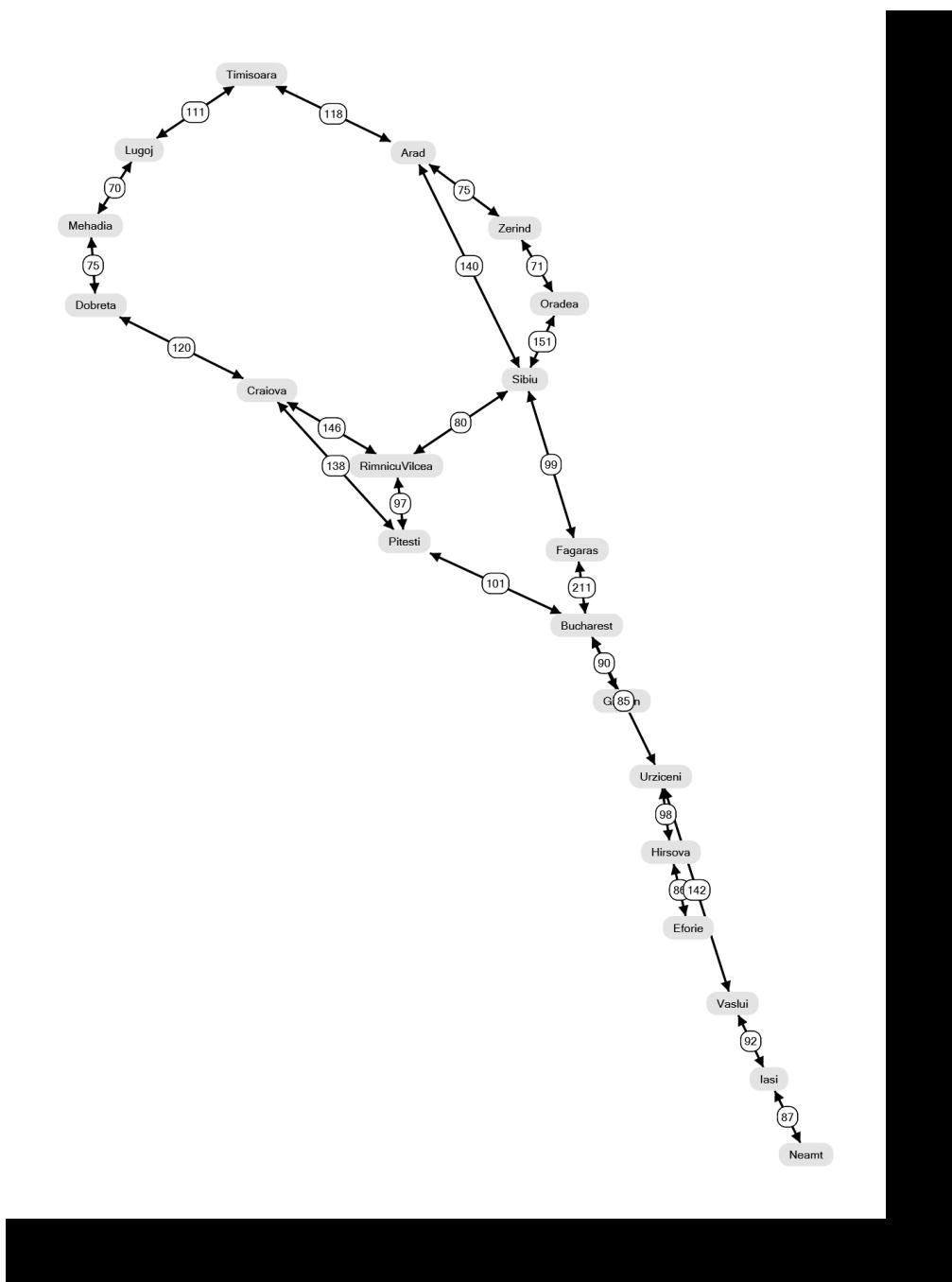
Visit

The algorithm has marked the node Arad as visited



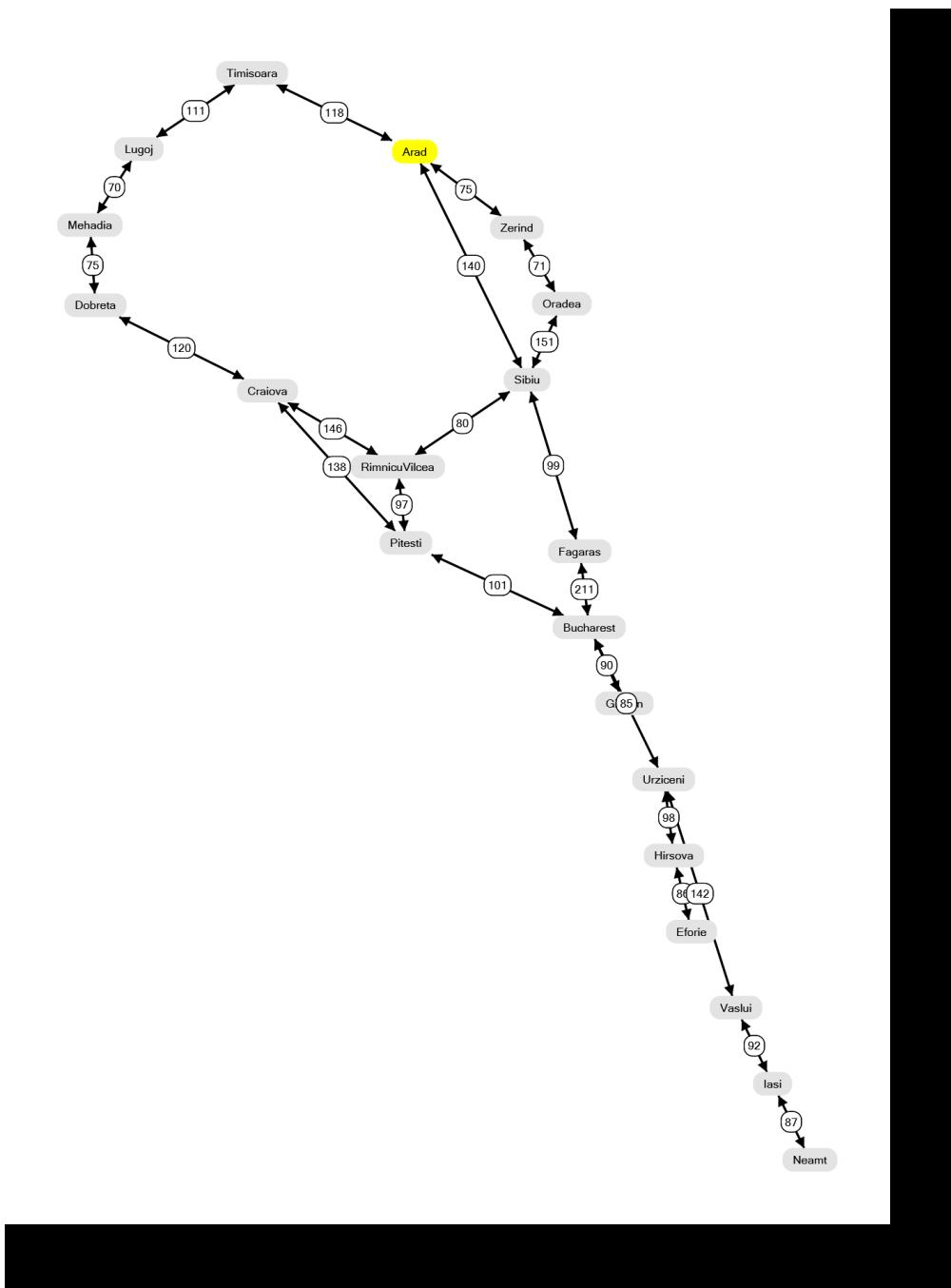
Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



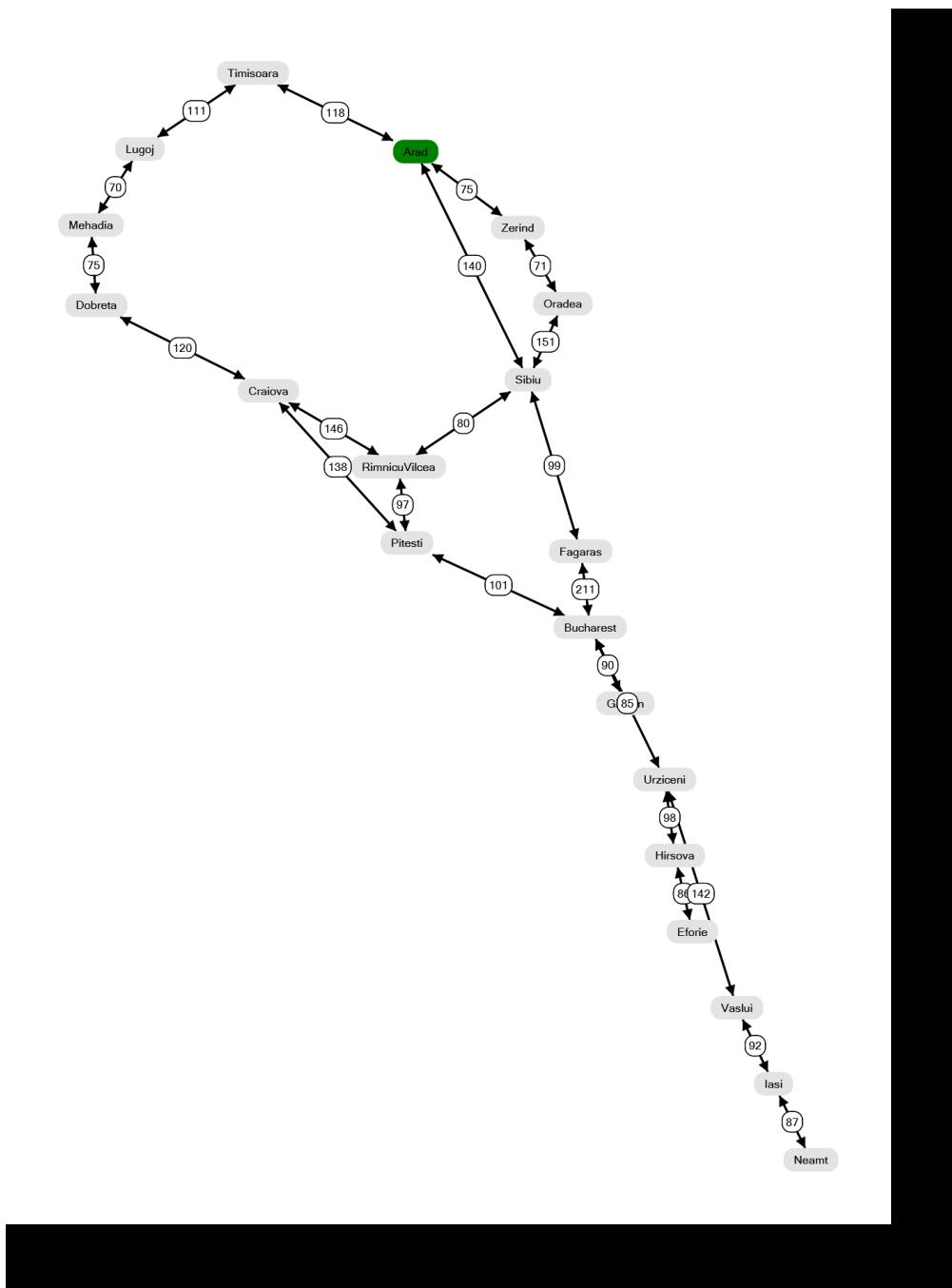
Pre-Visit

The algorithm has opened the node Arad



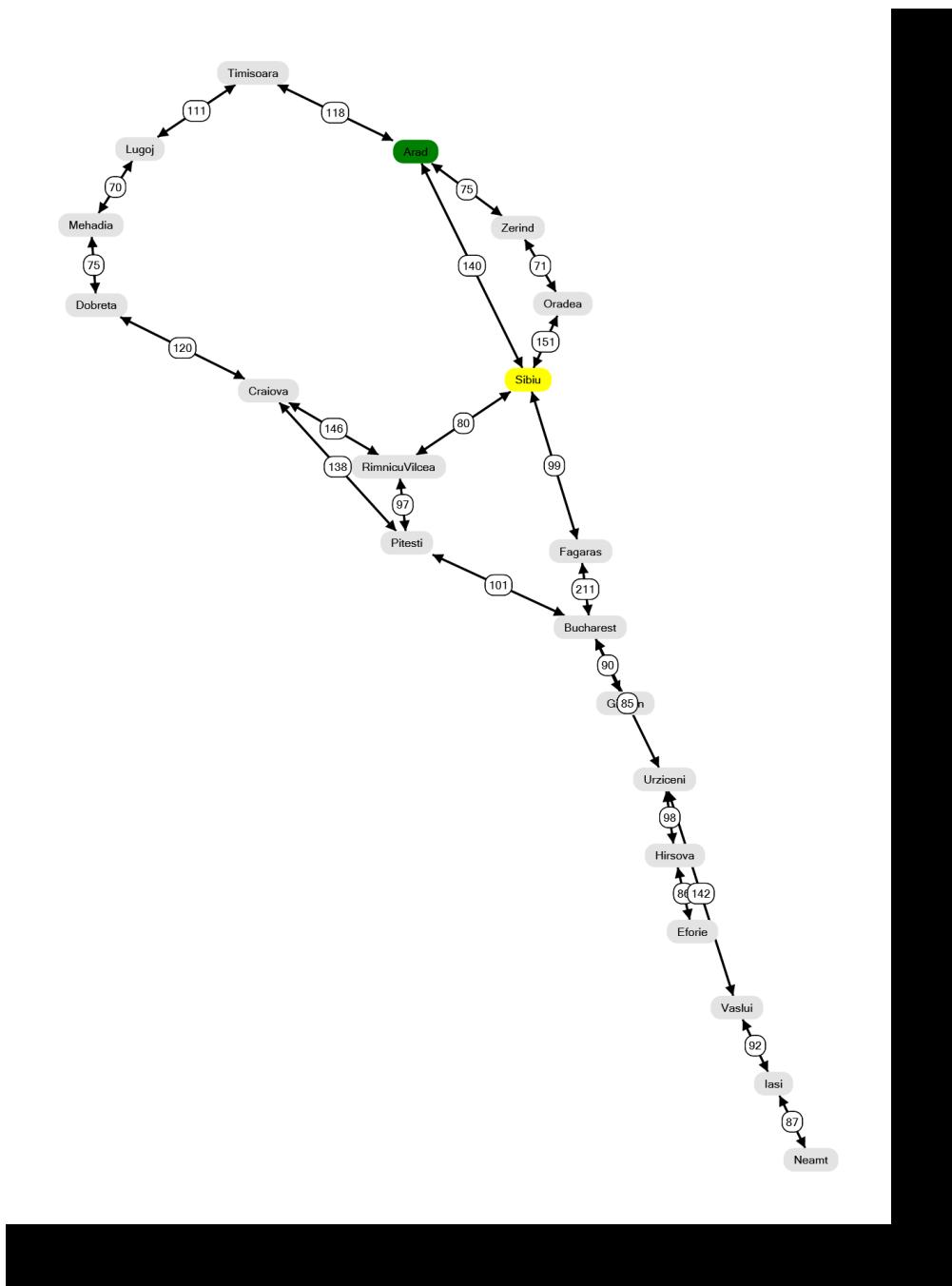
Visit

The algorithm has marked the node Arad as visited



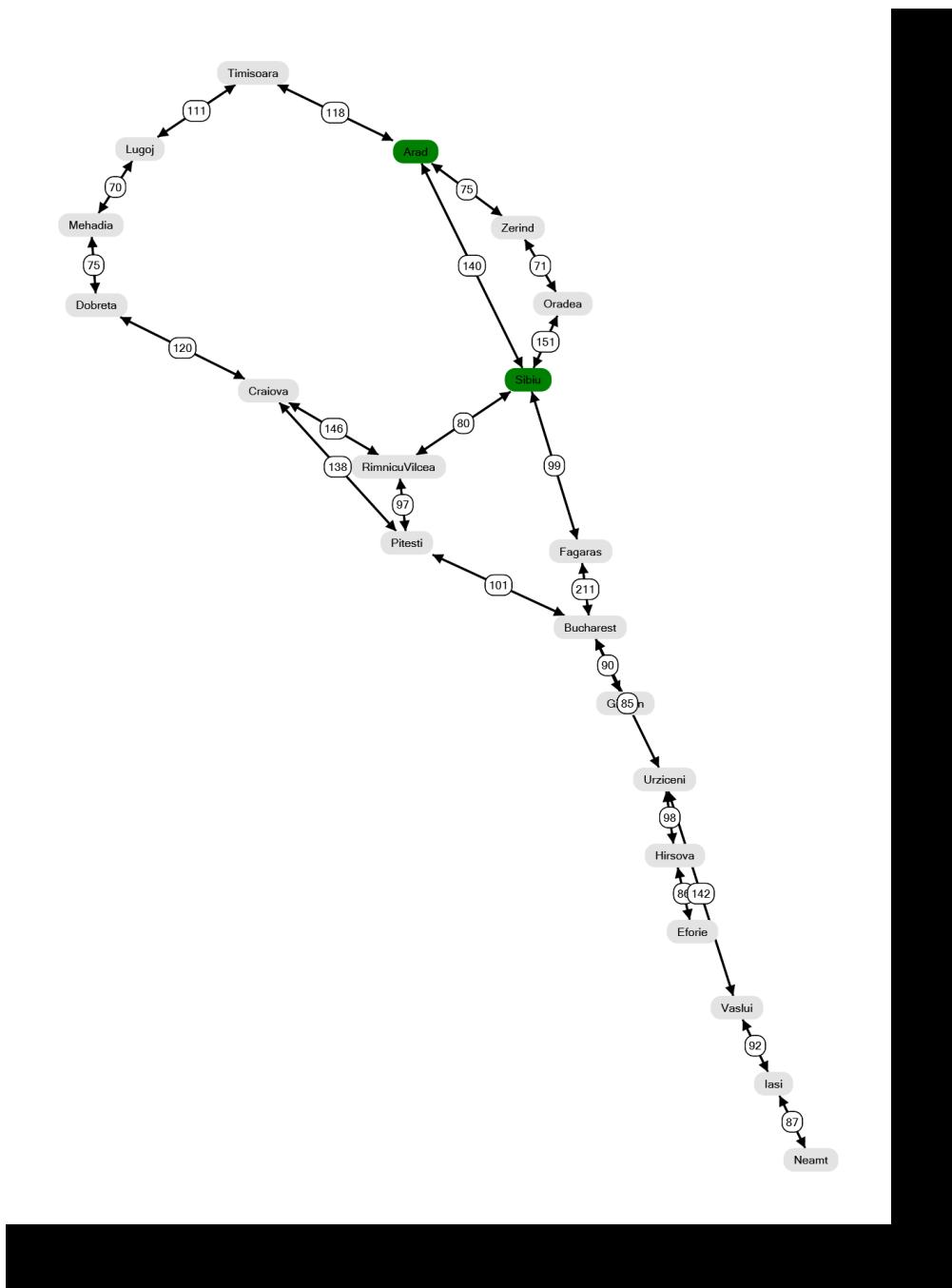
Pre-Visit

The algorithm has opened the node Sibiu



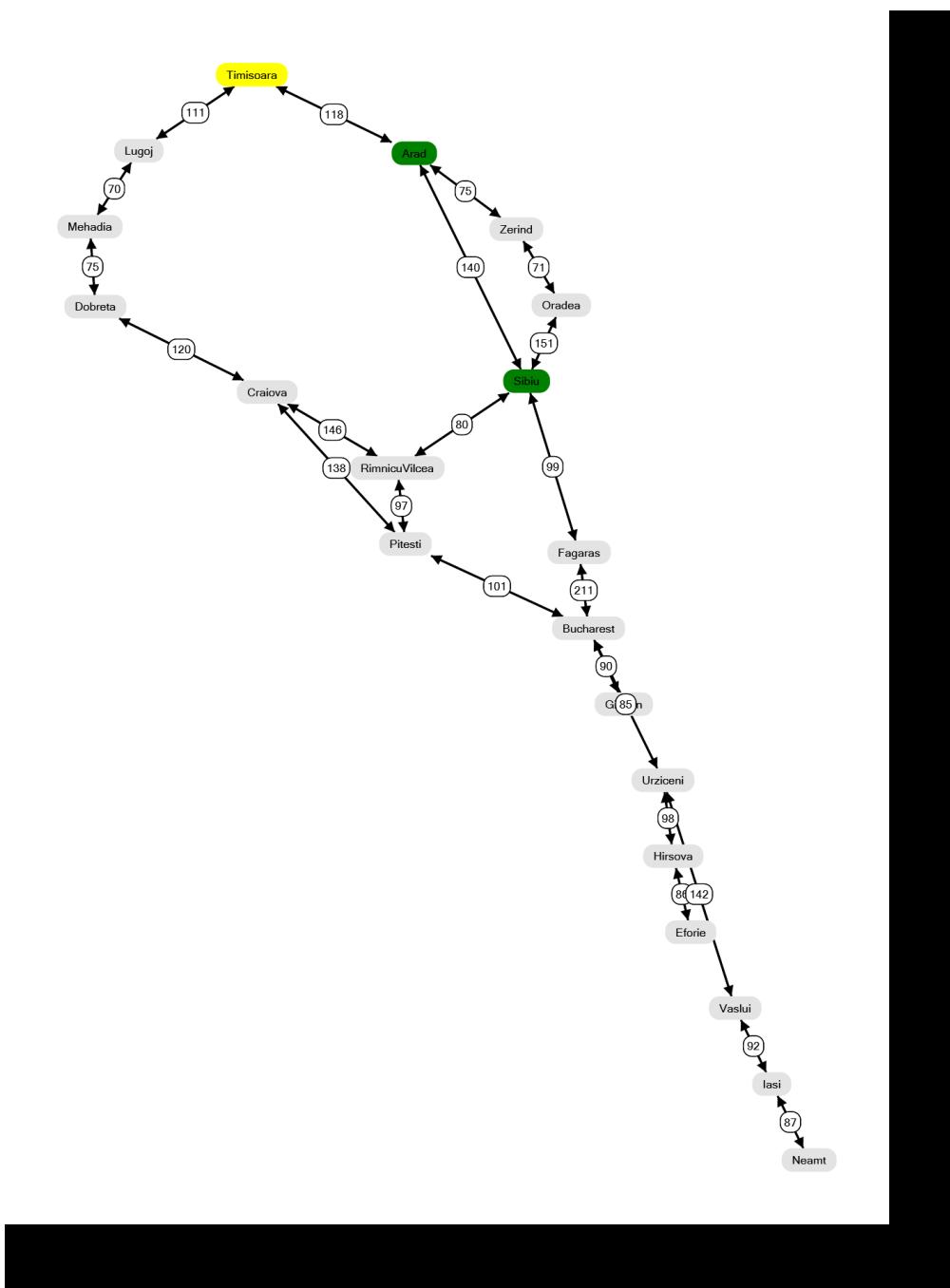
Visit

The algorithm has marked the node Sibiu as visited



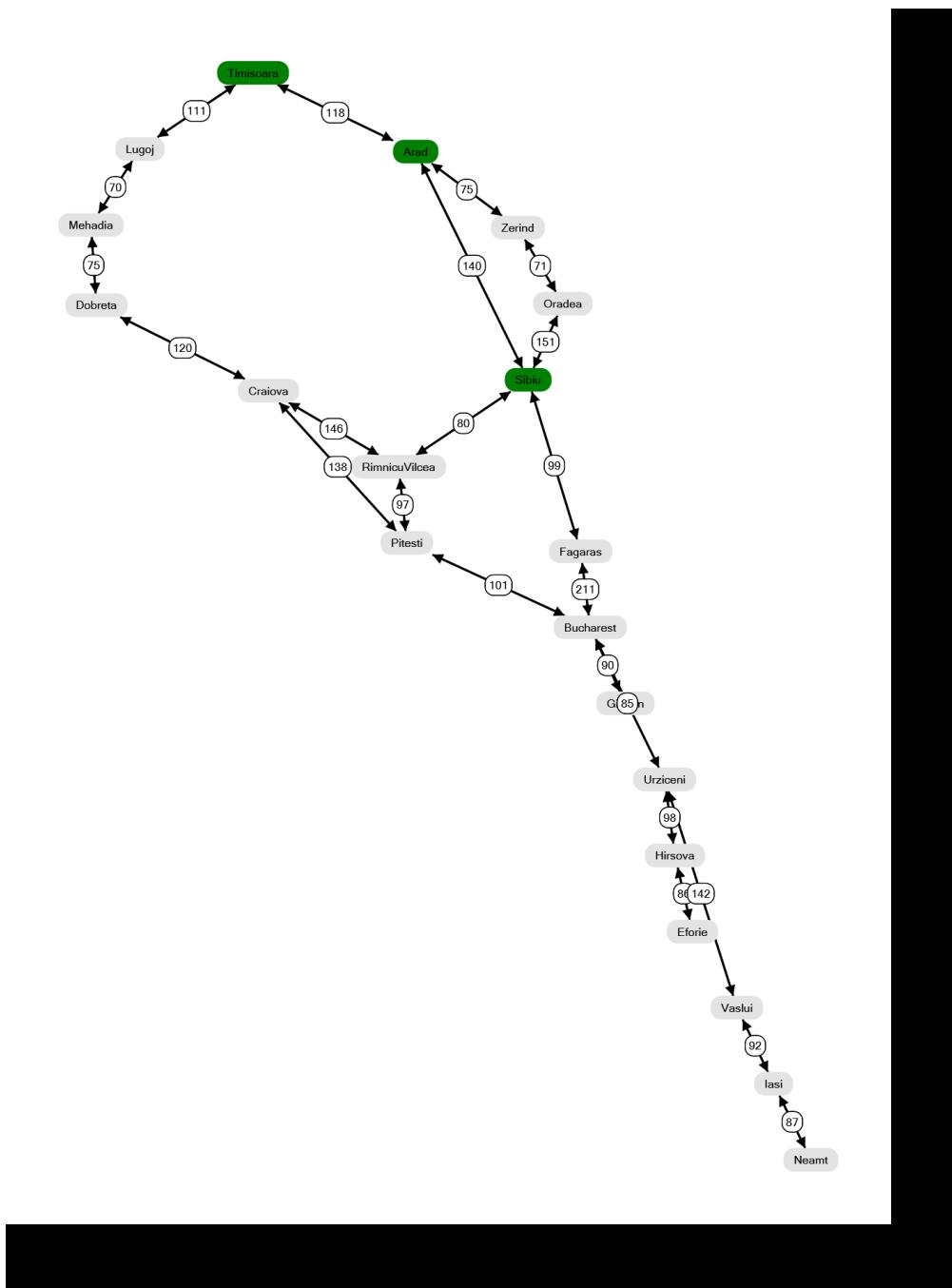
Pre-Visit

The algorithm has opened the node Timisoara



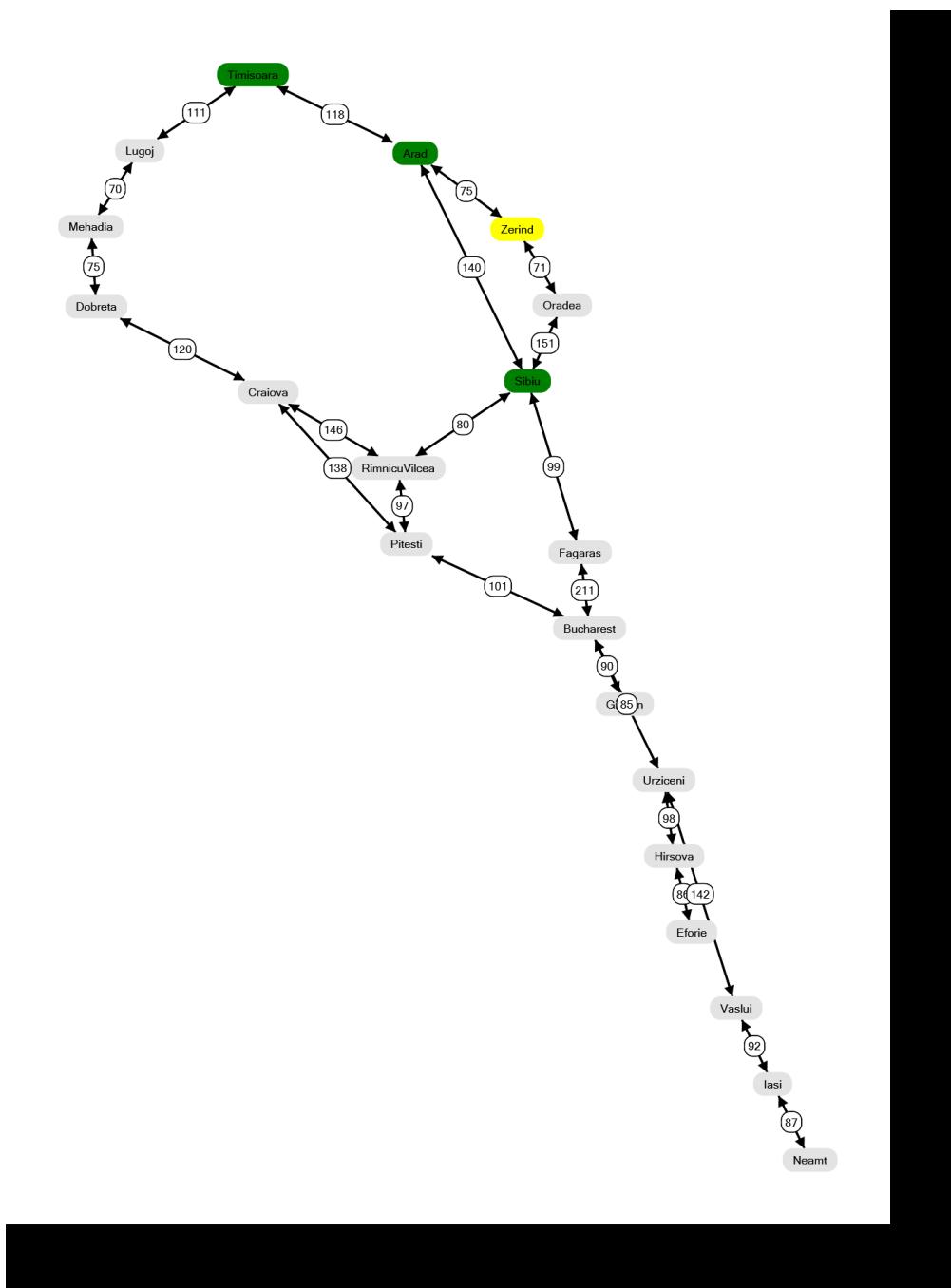
Visit

The algorithm has marked the node Timisoara as visited



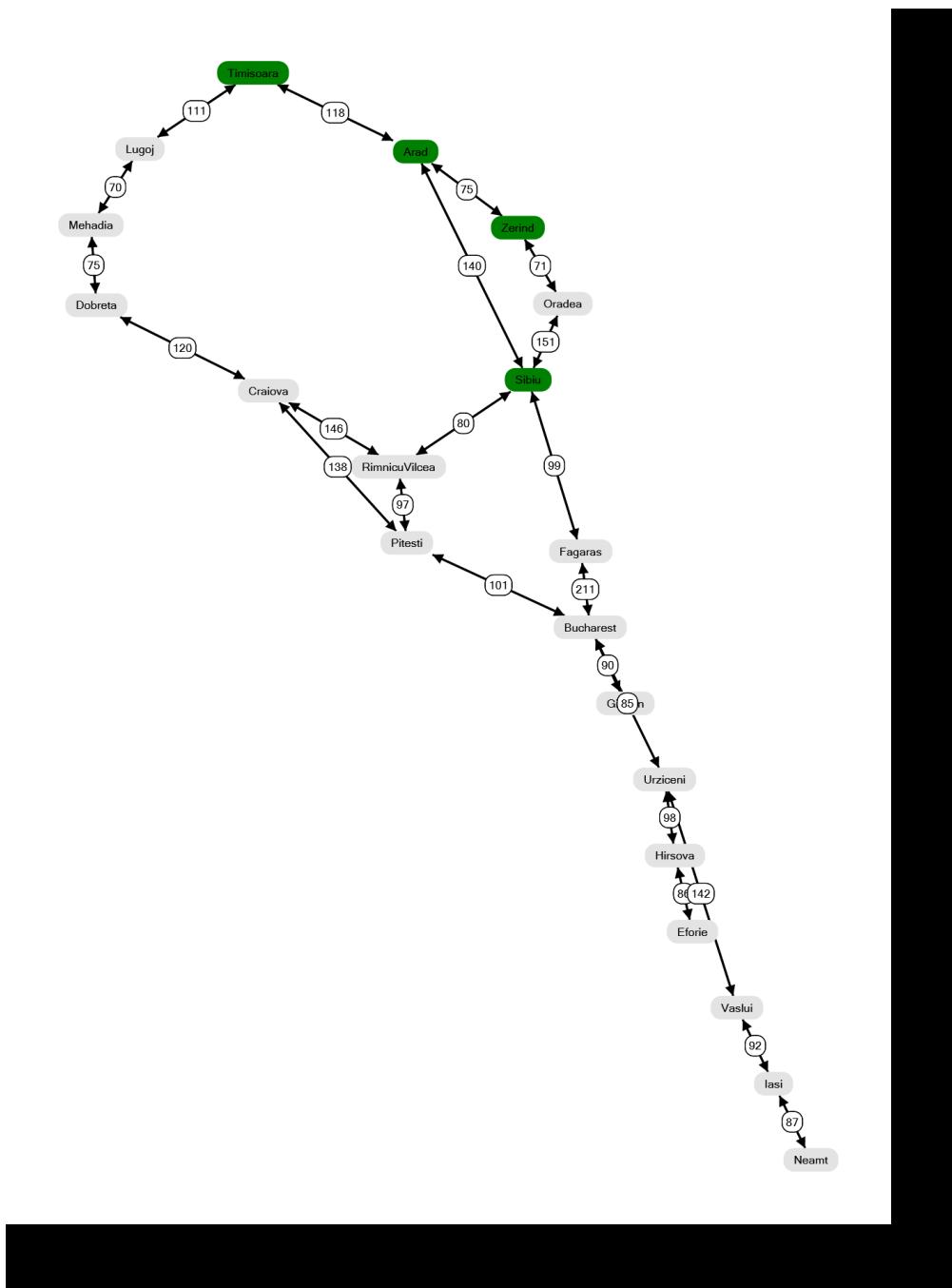
Pre-Visit

The algorithm has opened the node Zerind



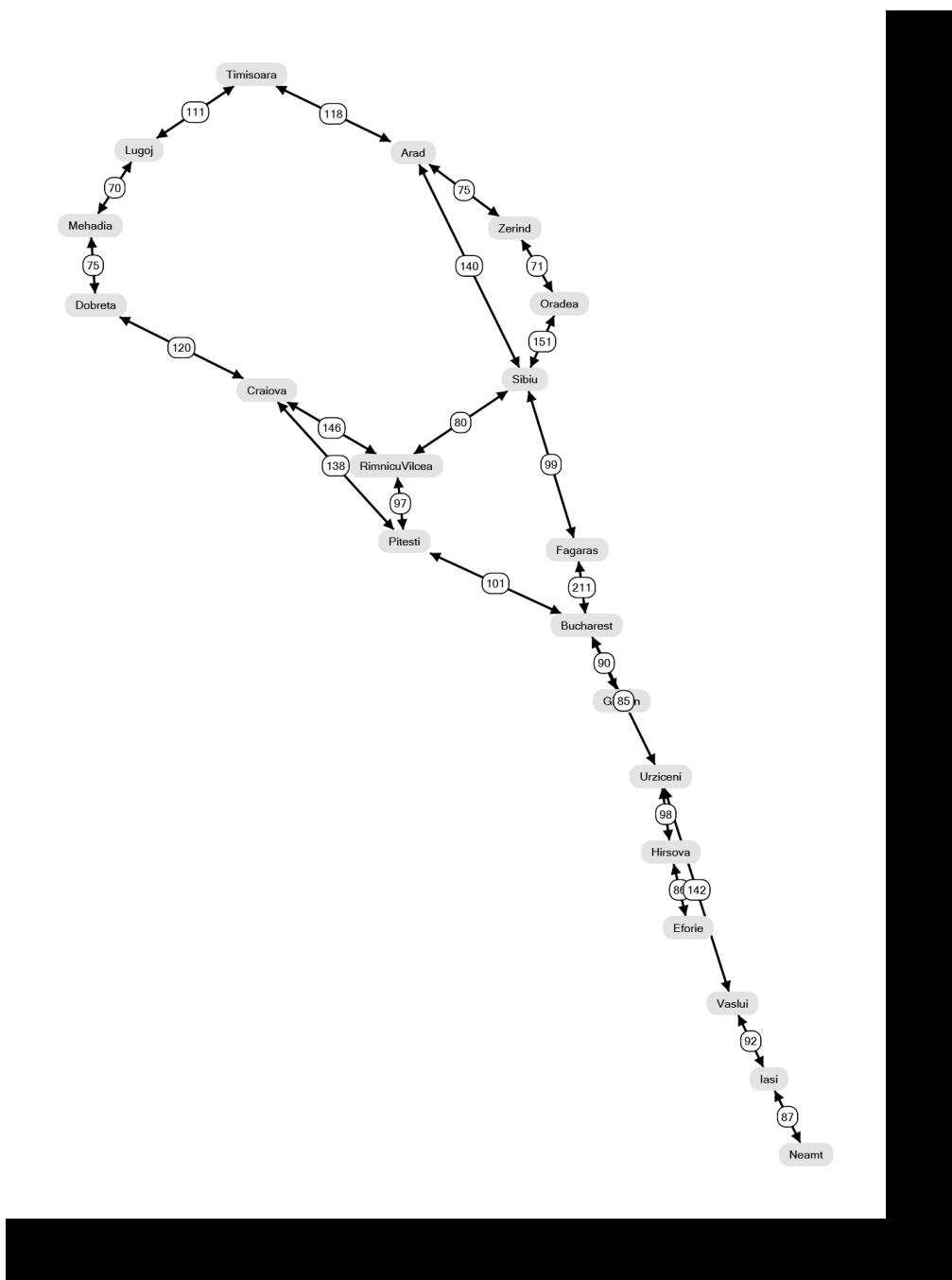
Visit

The algorithm has marked the node Zerind as visited



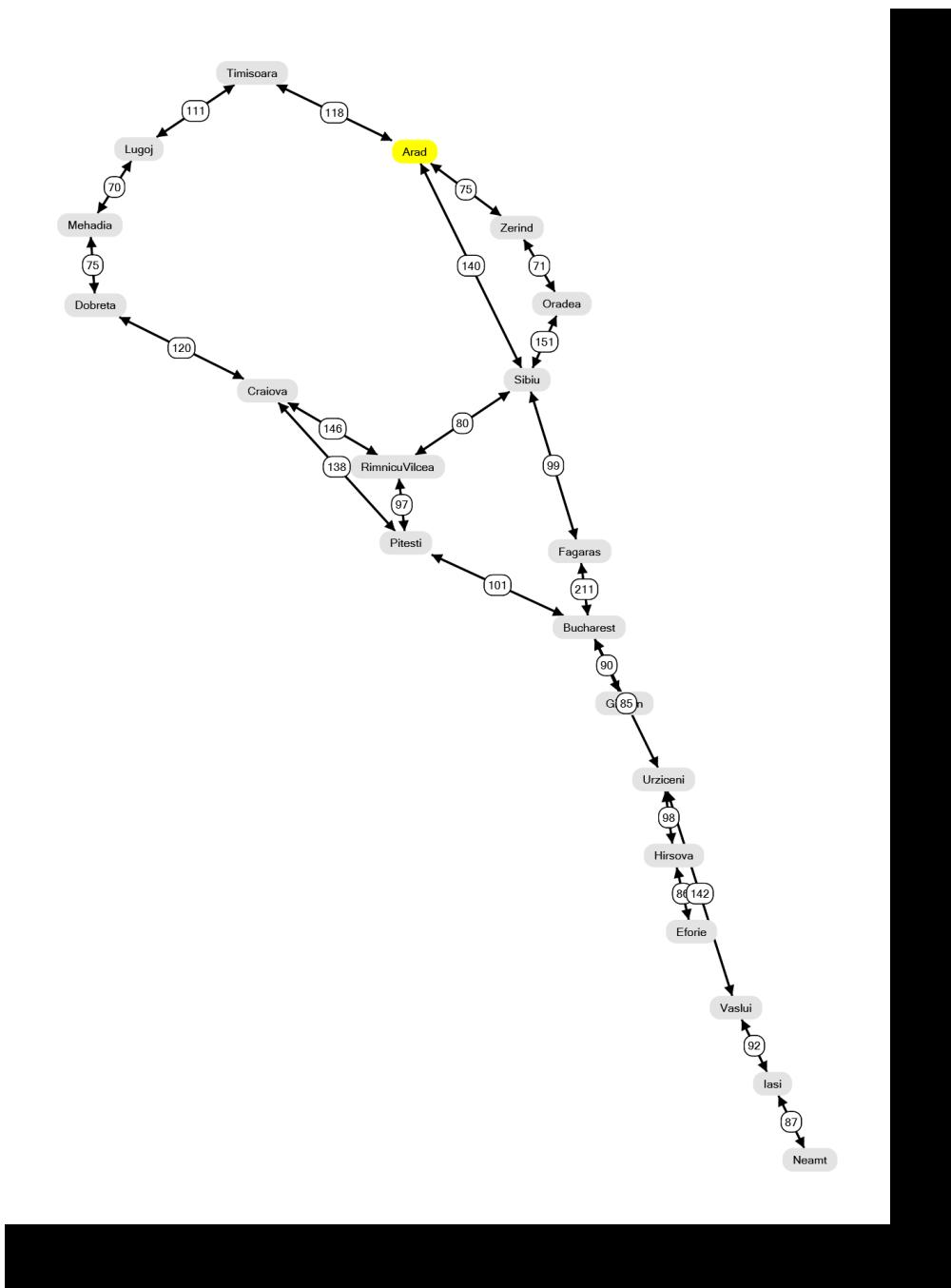
Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



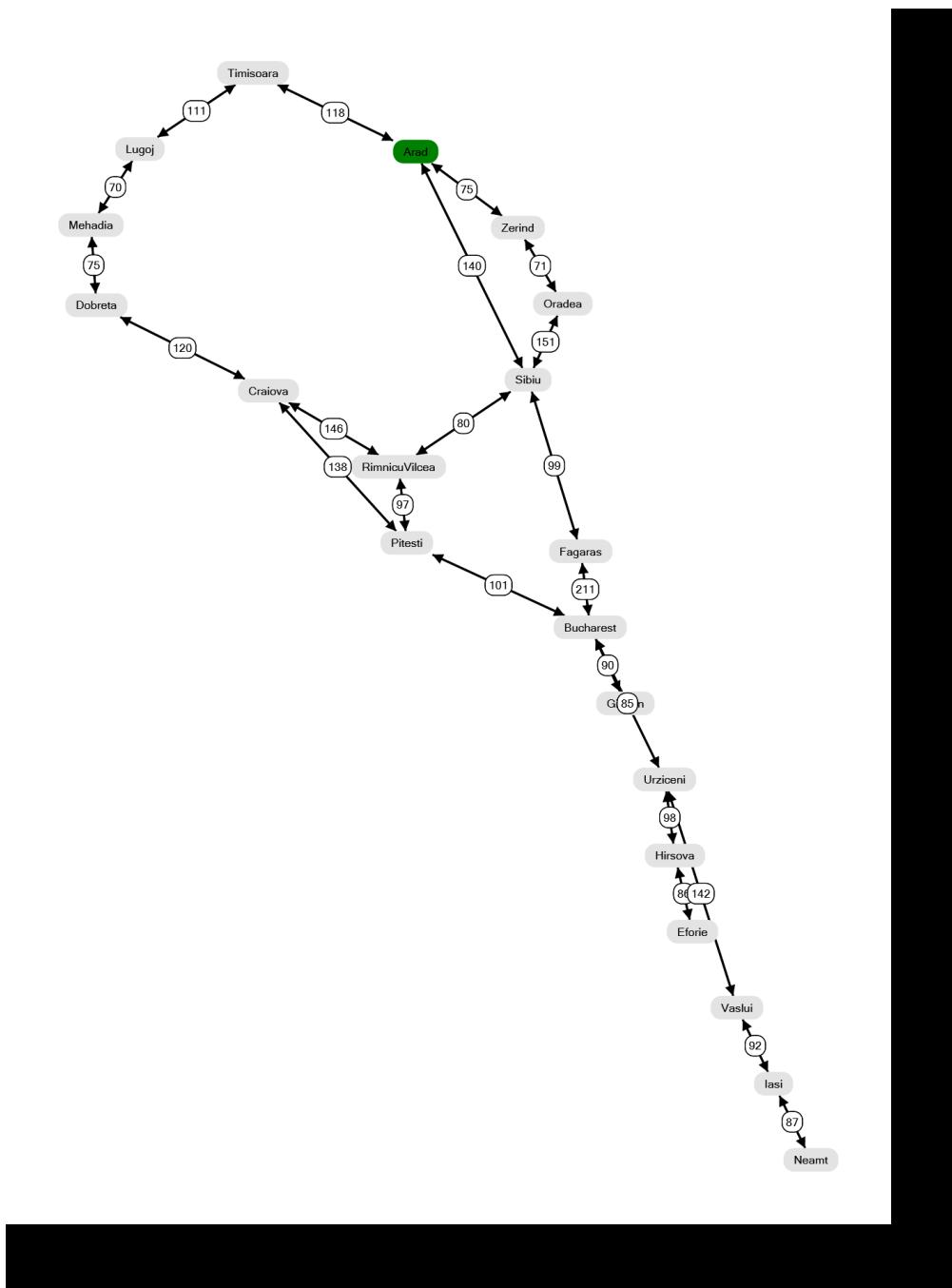
Pre-Visit

The algorithm has opened the node Arad



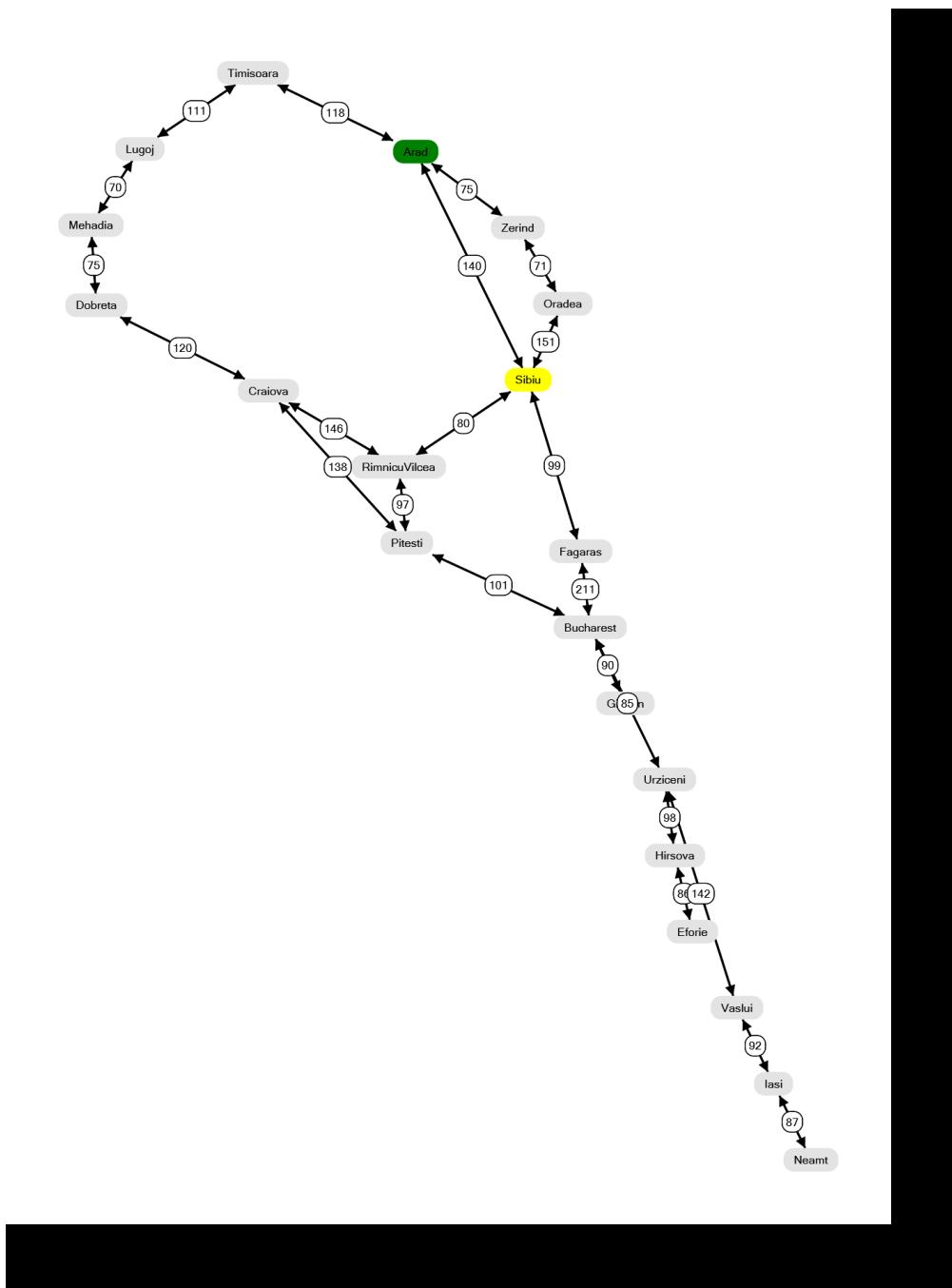
Visit

The algorithm has marked the node Arad as visited



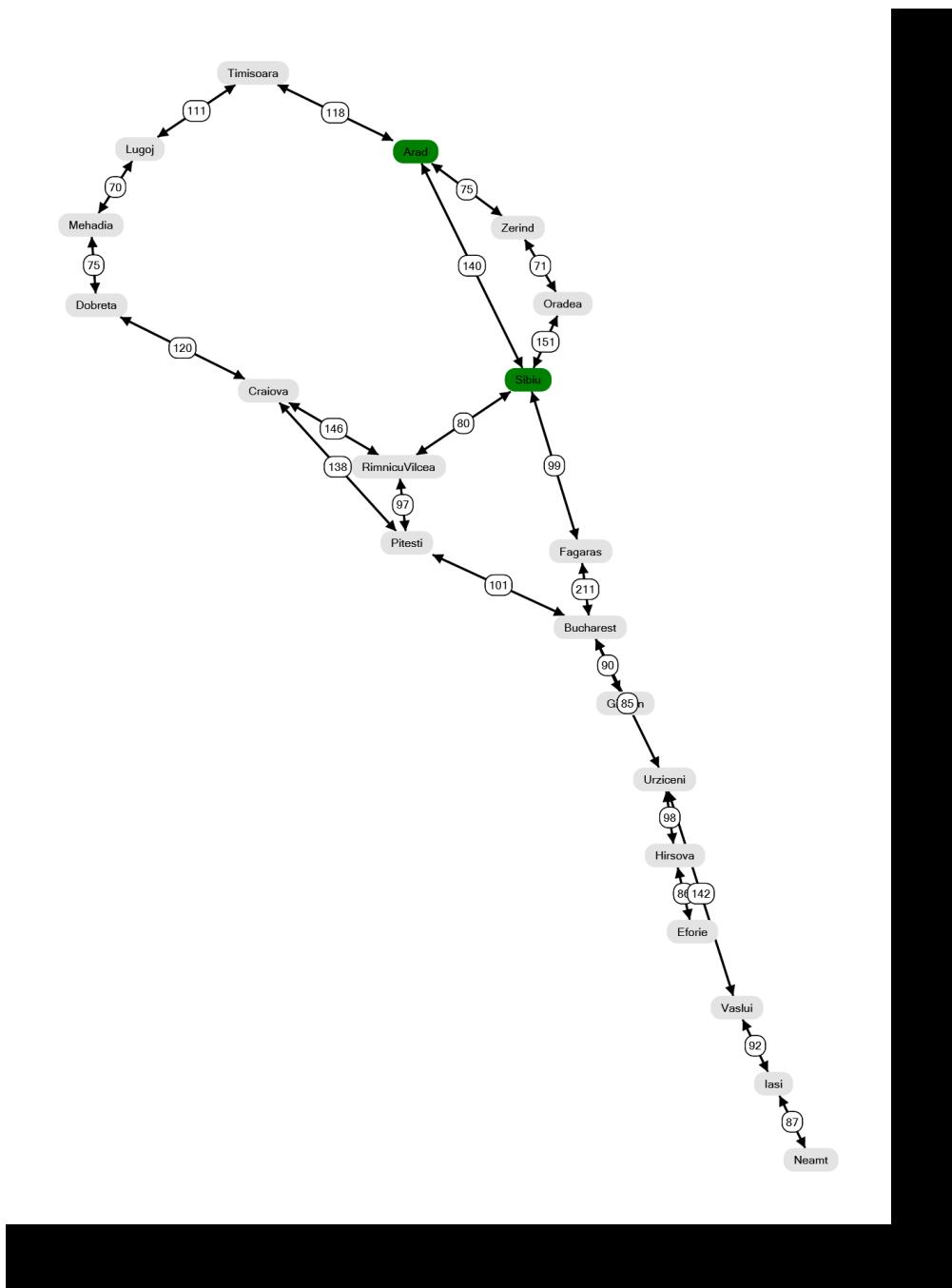
Pre-Visit

The algorithm has opened the node Sibiu



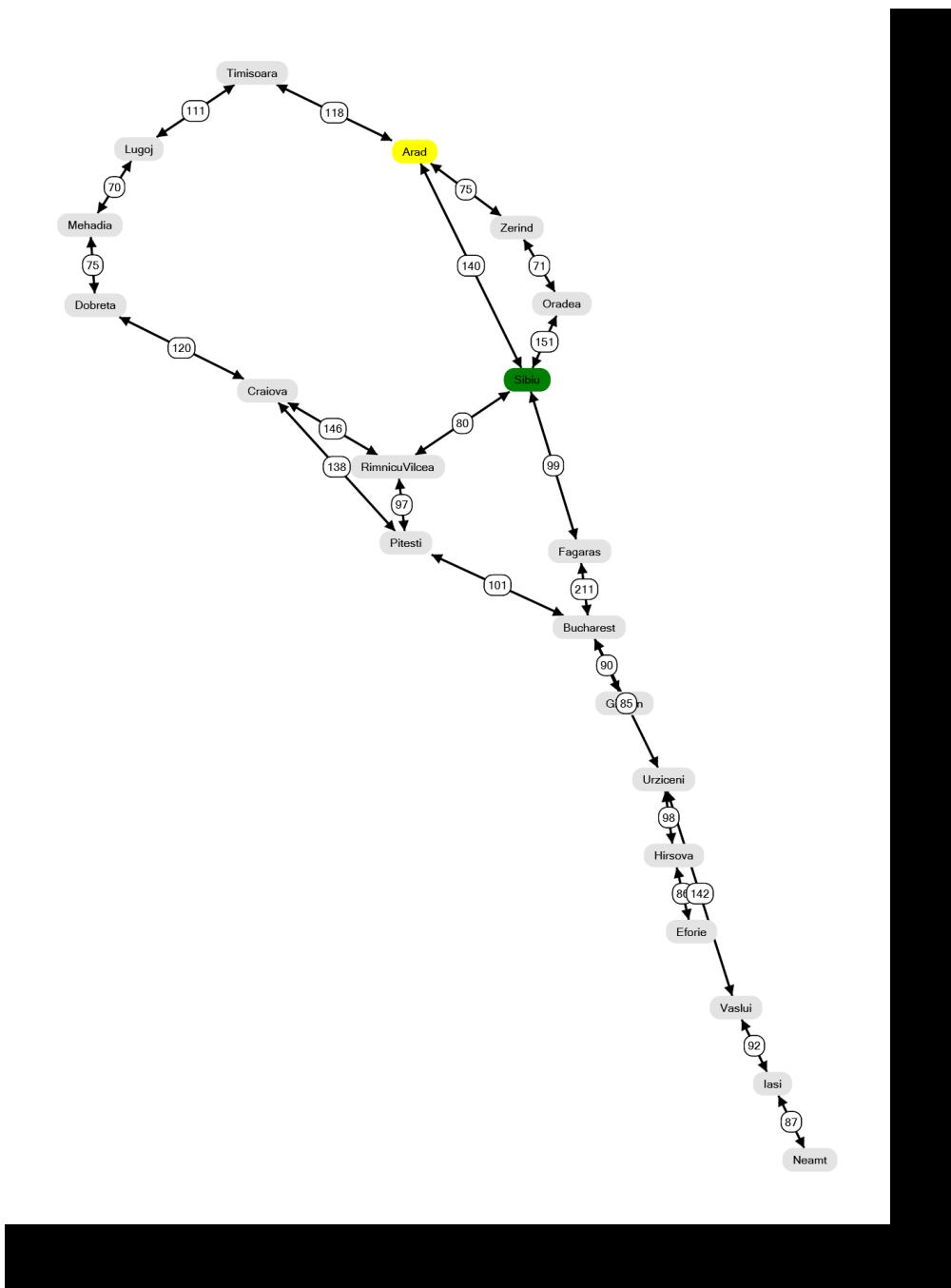
Visit

The algorithm has marked the node Sibiu as visited



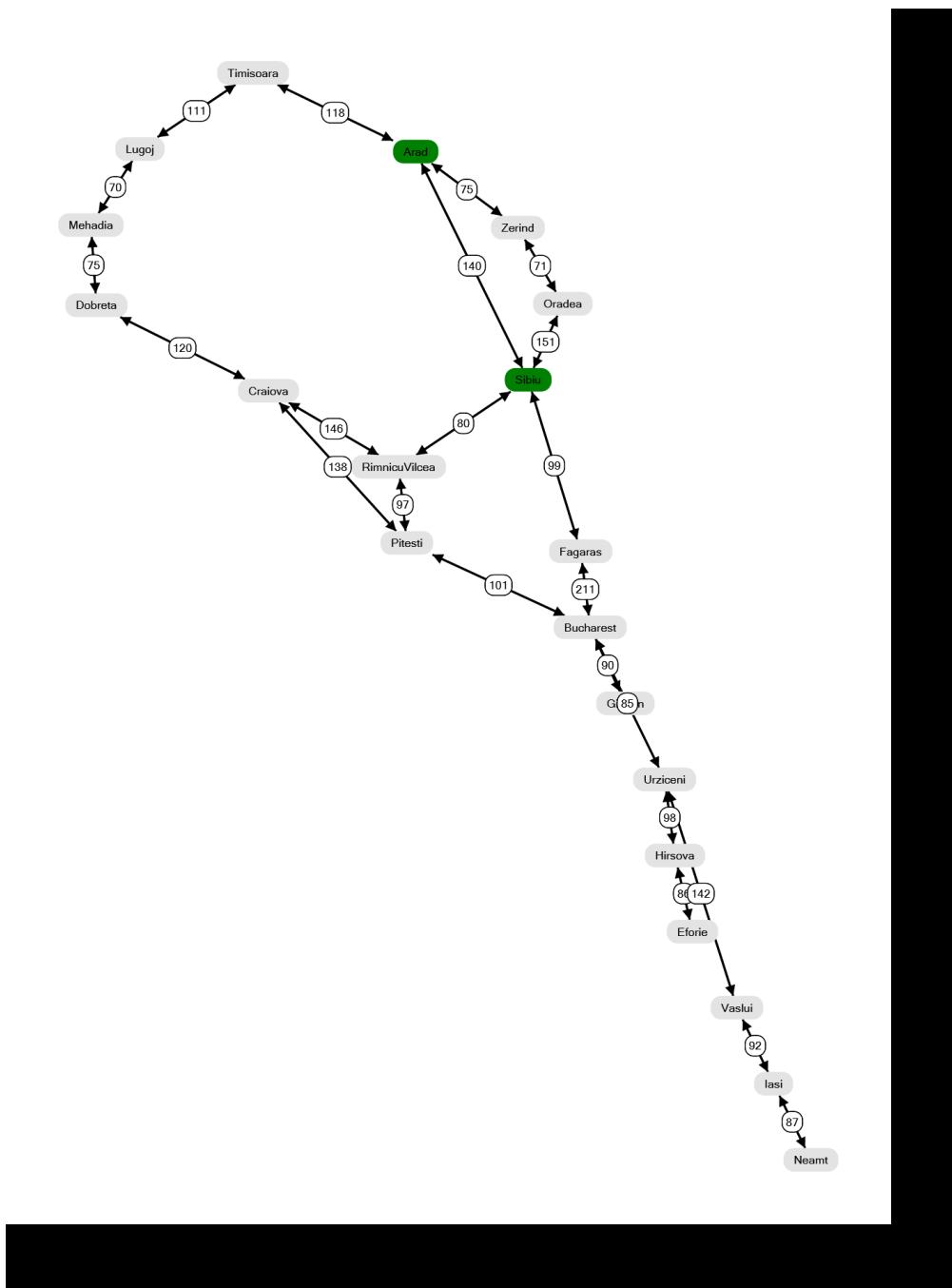
Pre-Visit

The algorithm has opened the node Arad



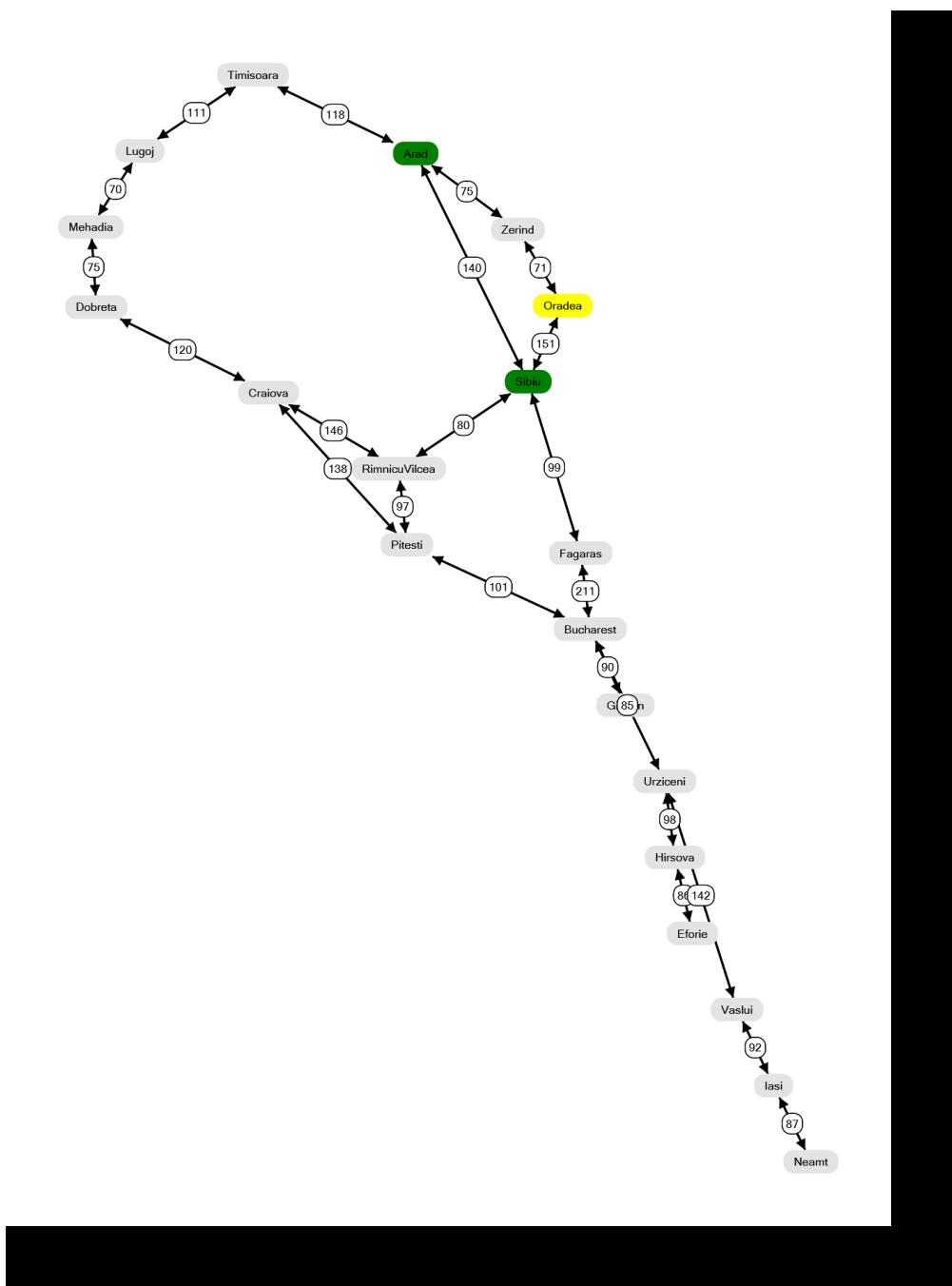
Visit

The algorithm has marked the node Arad as visited



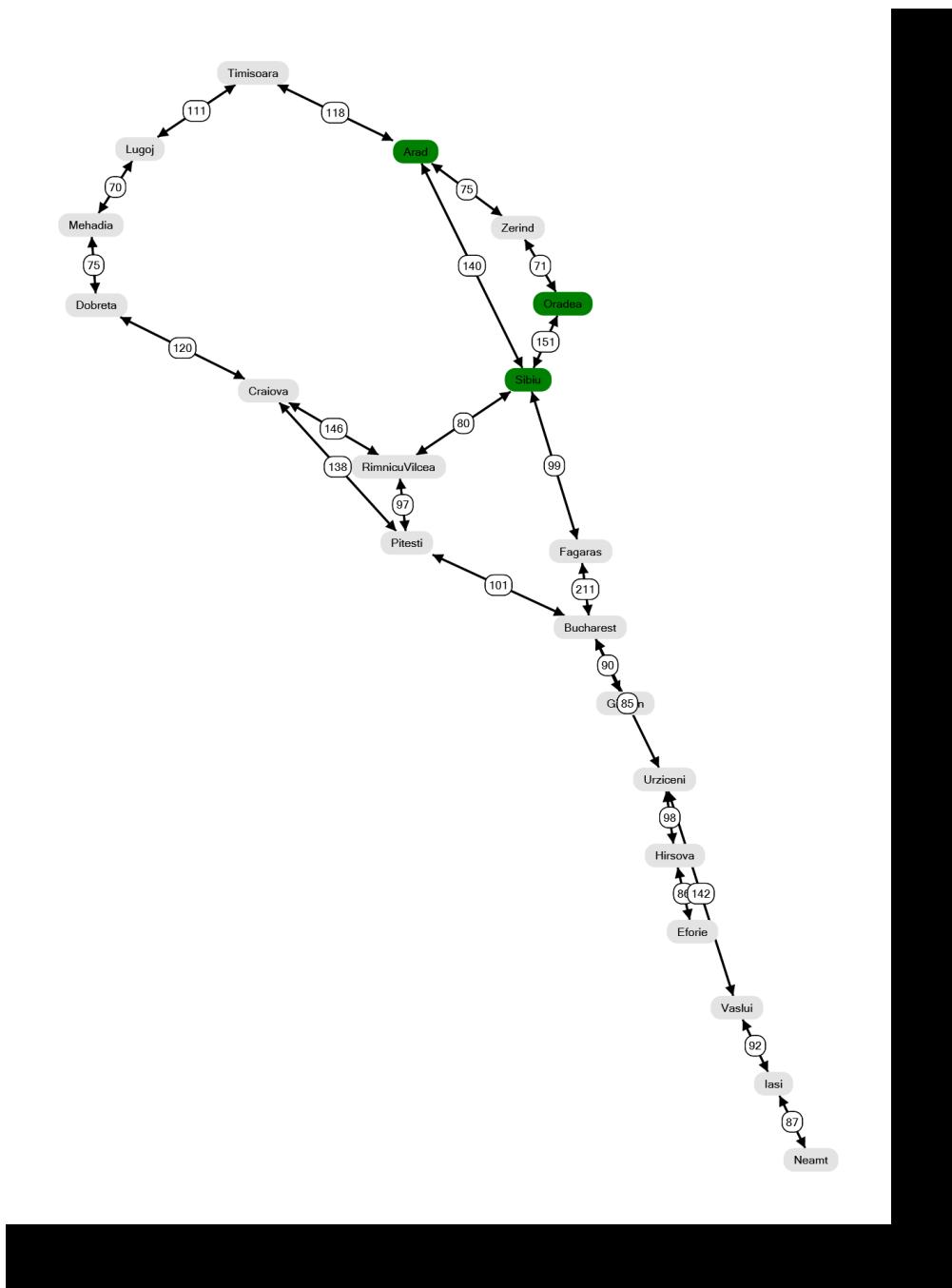
Pre-Visit

The algorithm has opened the node Oradea



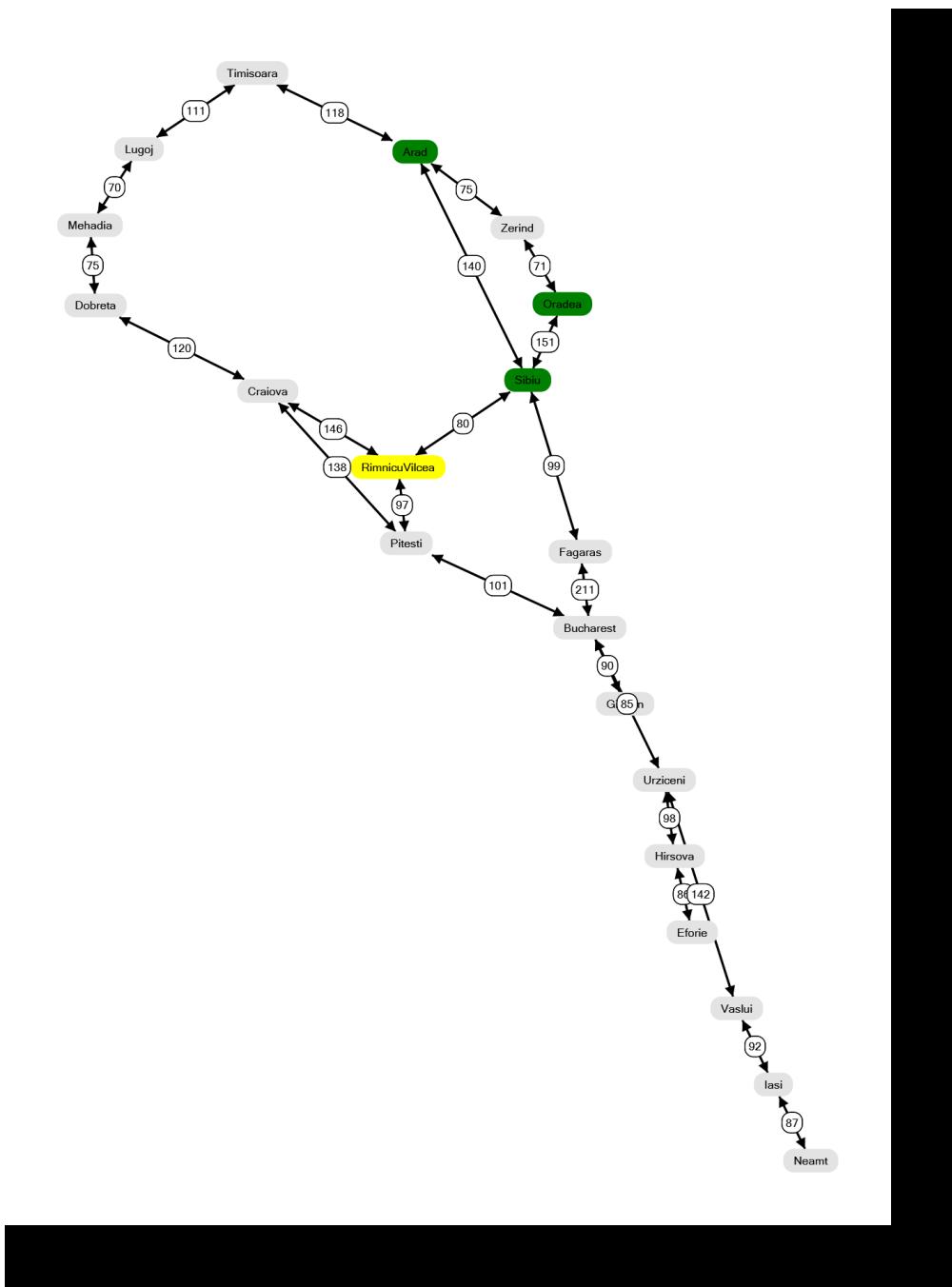
Visit

The algorithm has marked the node Oradea as visited



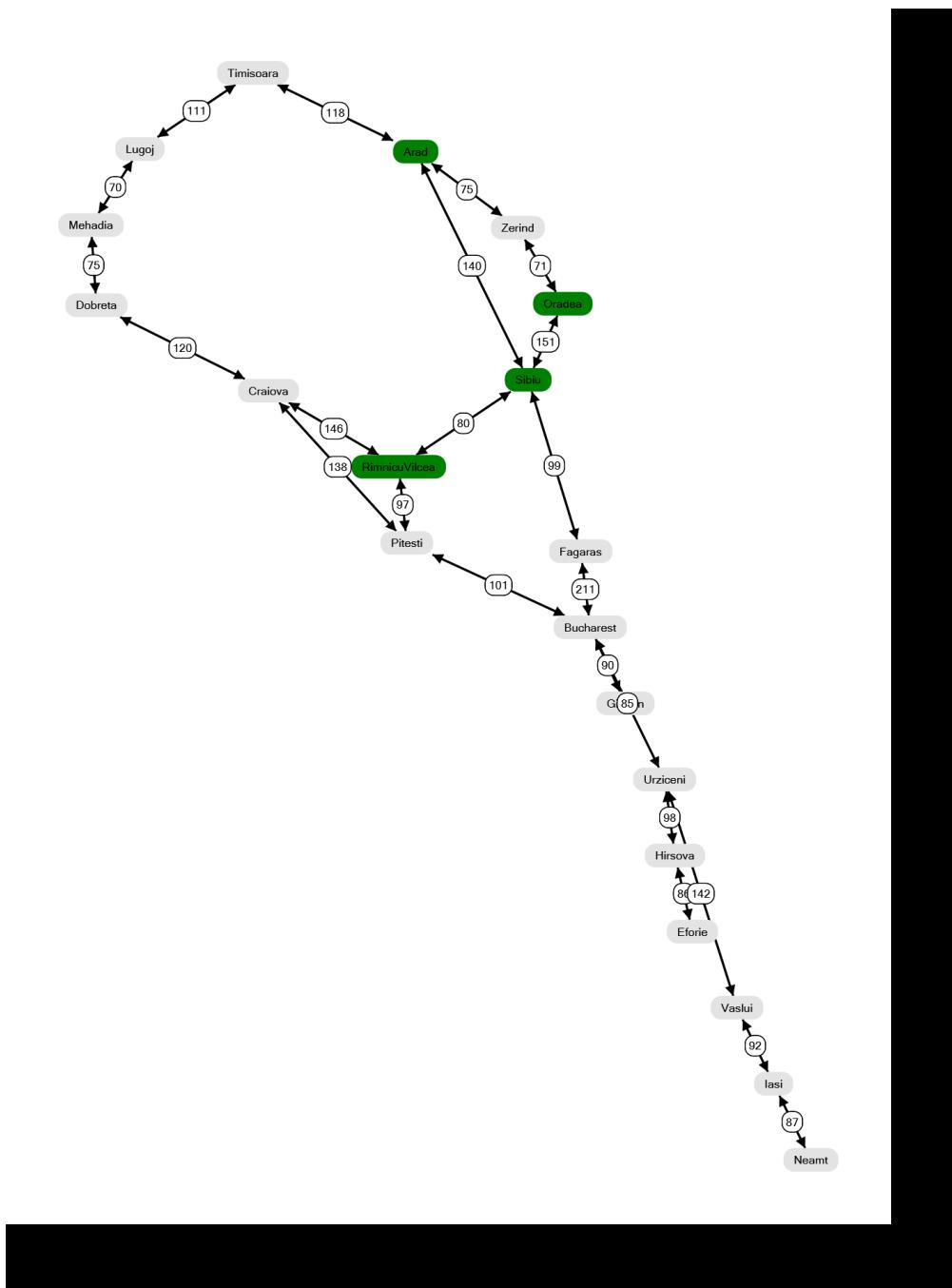
Pre-Visit

The algorithm has opened the node Rimnicu Vilcea



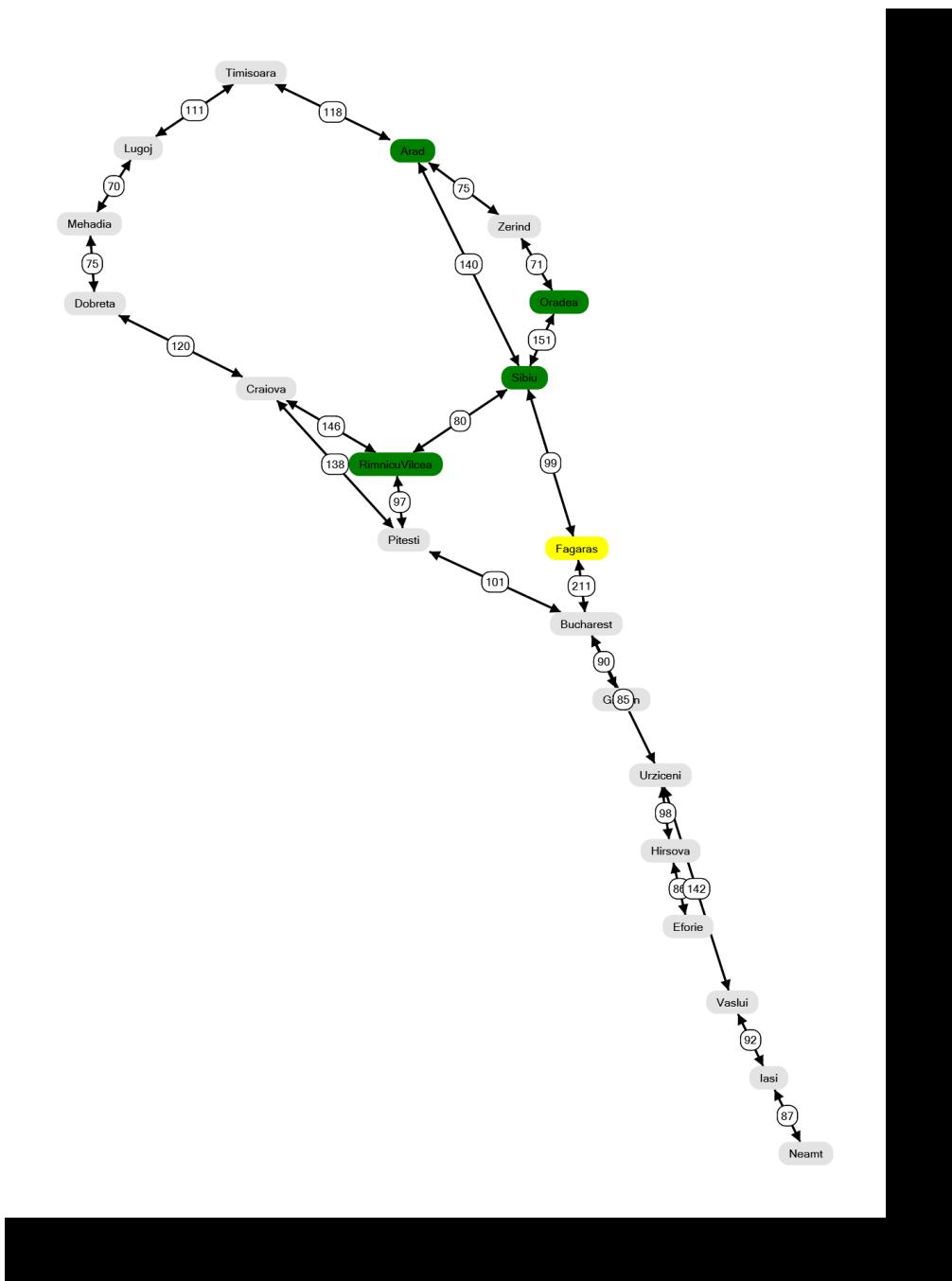
Visit

The algorithm has marked the node Rimnicu Vilcea as visited



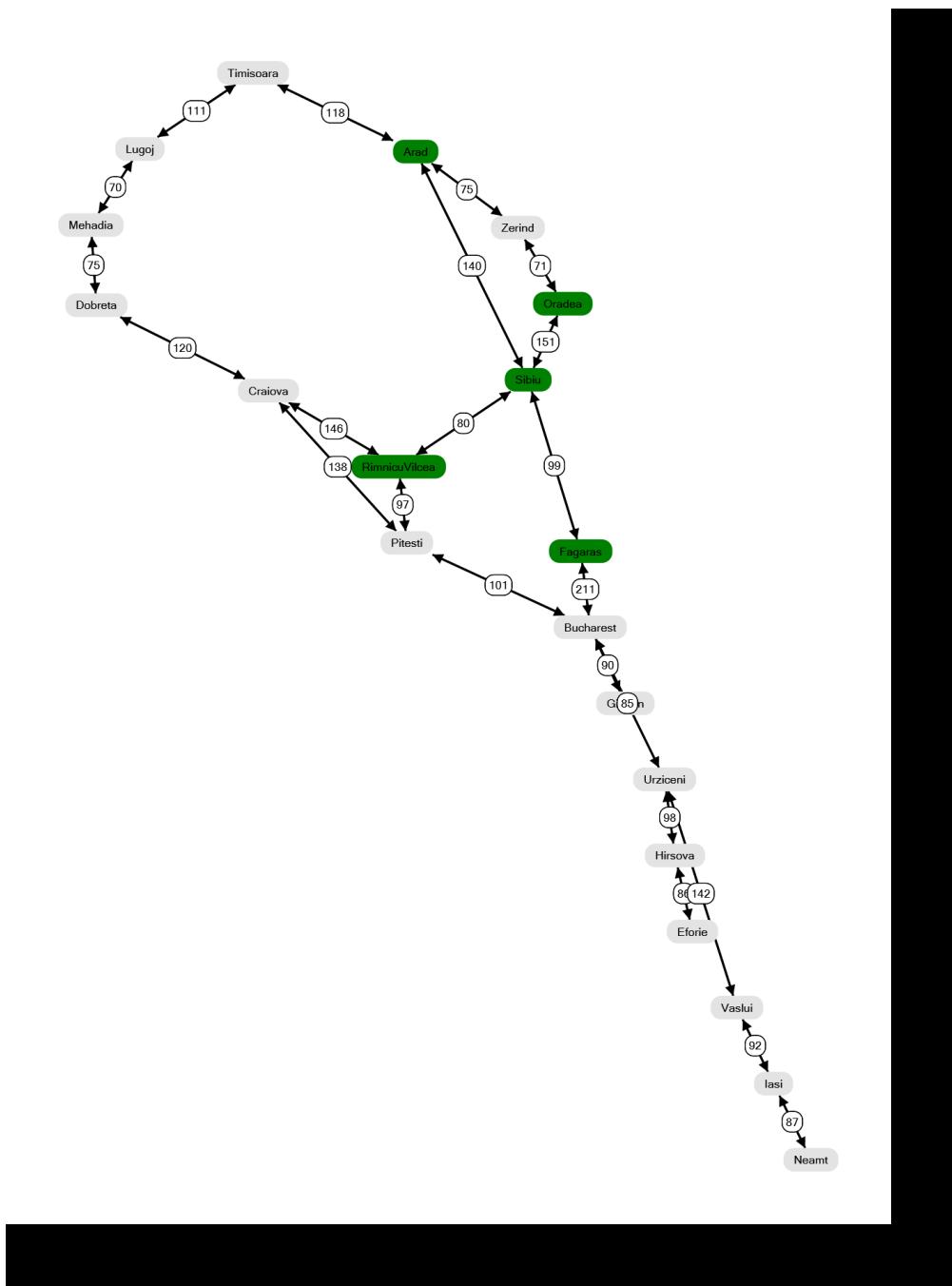
Pre-Visit

The algorithm has opened the node Fagaras



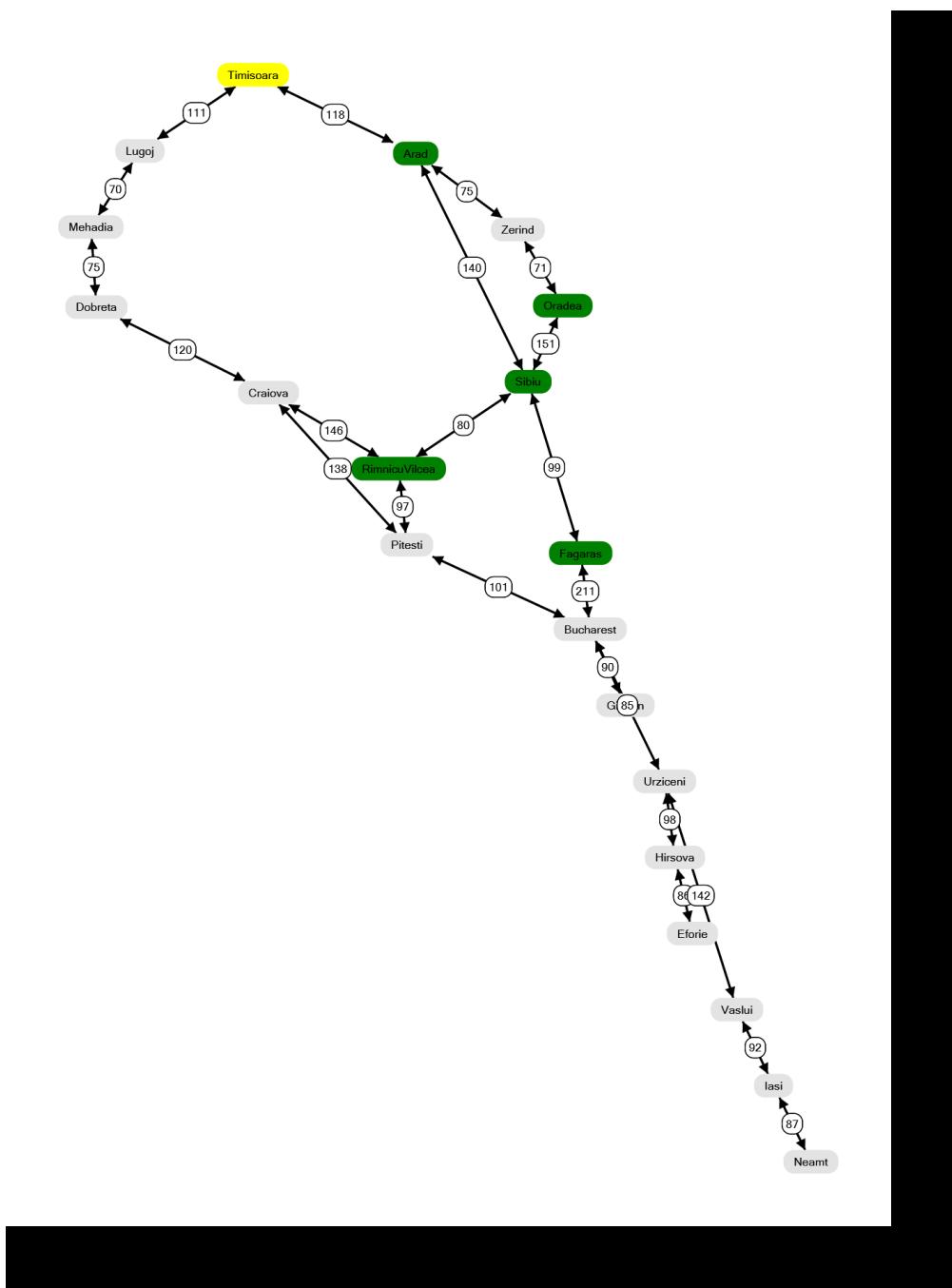
Visit

The algorithm has marked the node Fagaras as visited



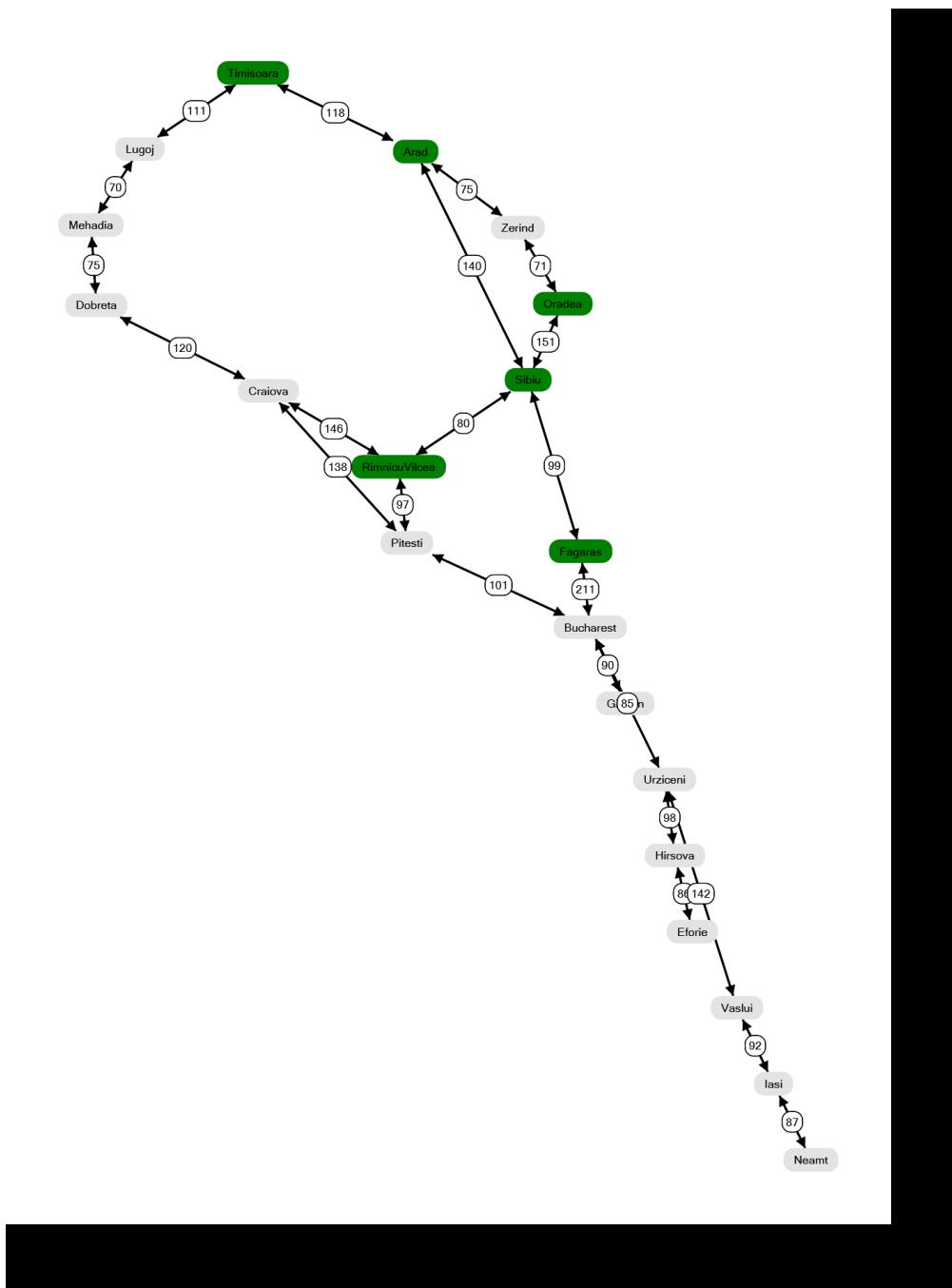
Pre-Visit

The algorithm has opened the node Timisoara



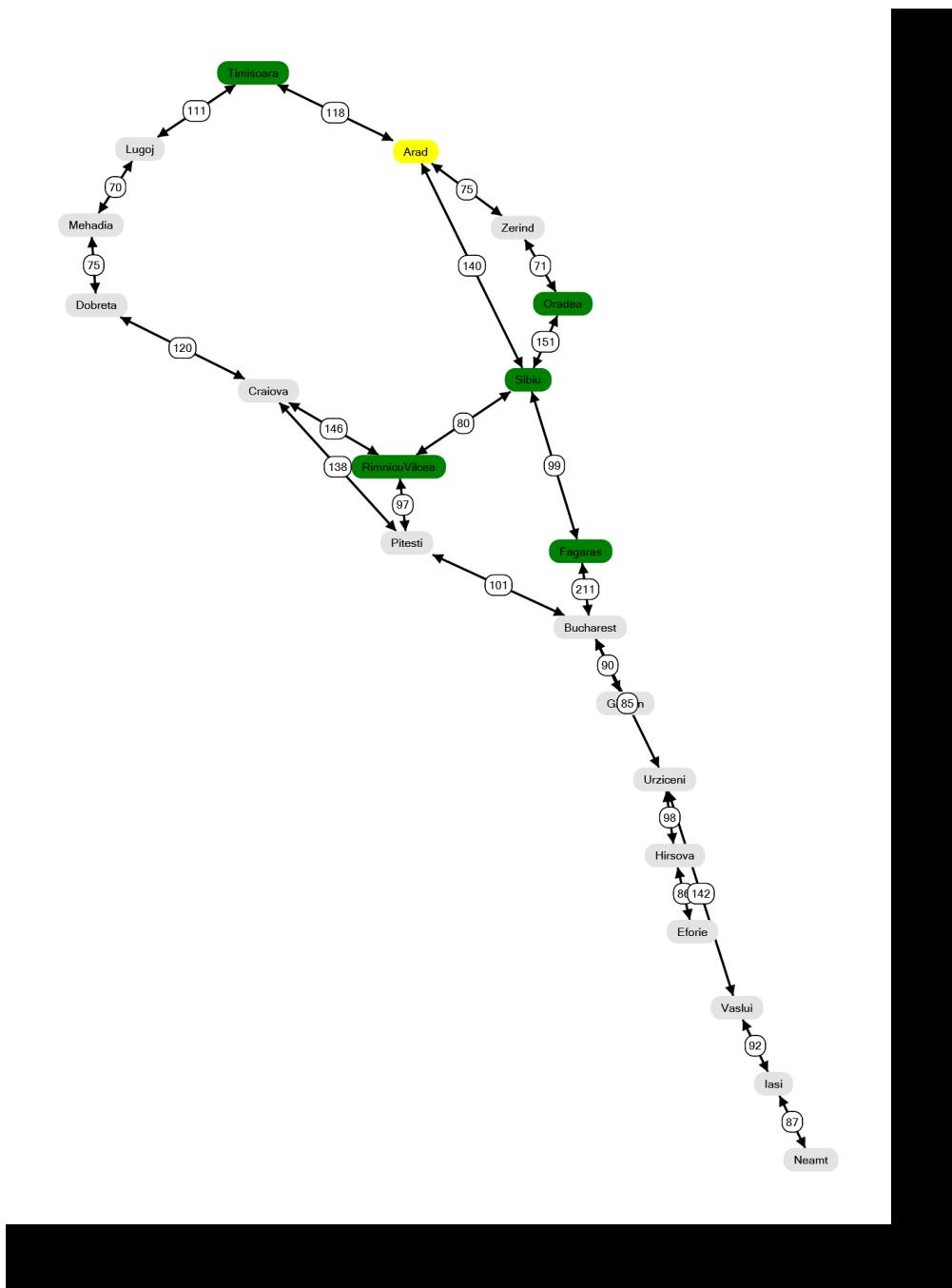
Visit

The algorithm has marked the node Timisoara as visited



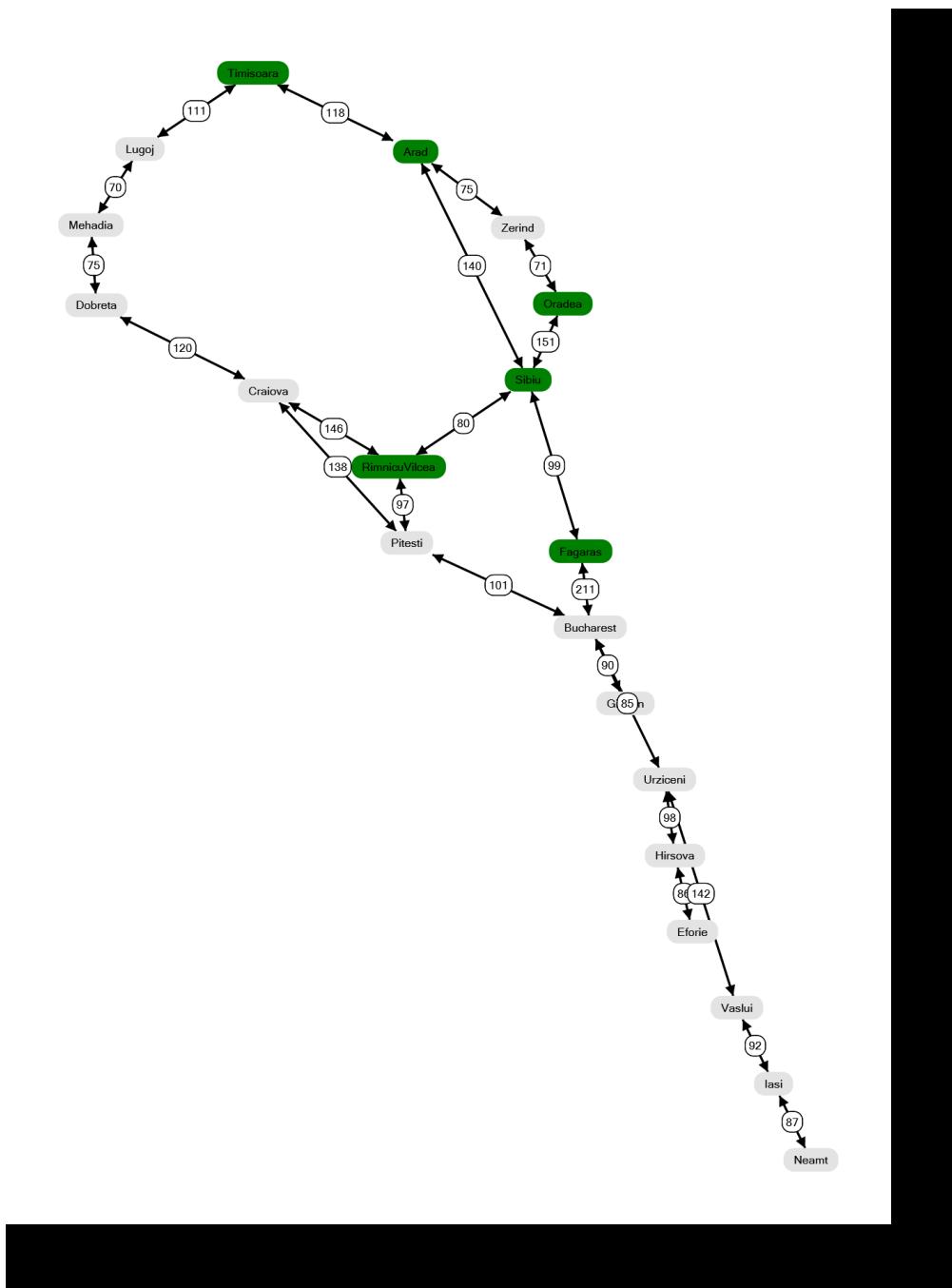
Pre-Visit

The algorithm has opened the node Arad



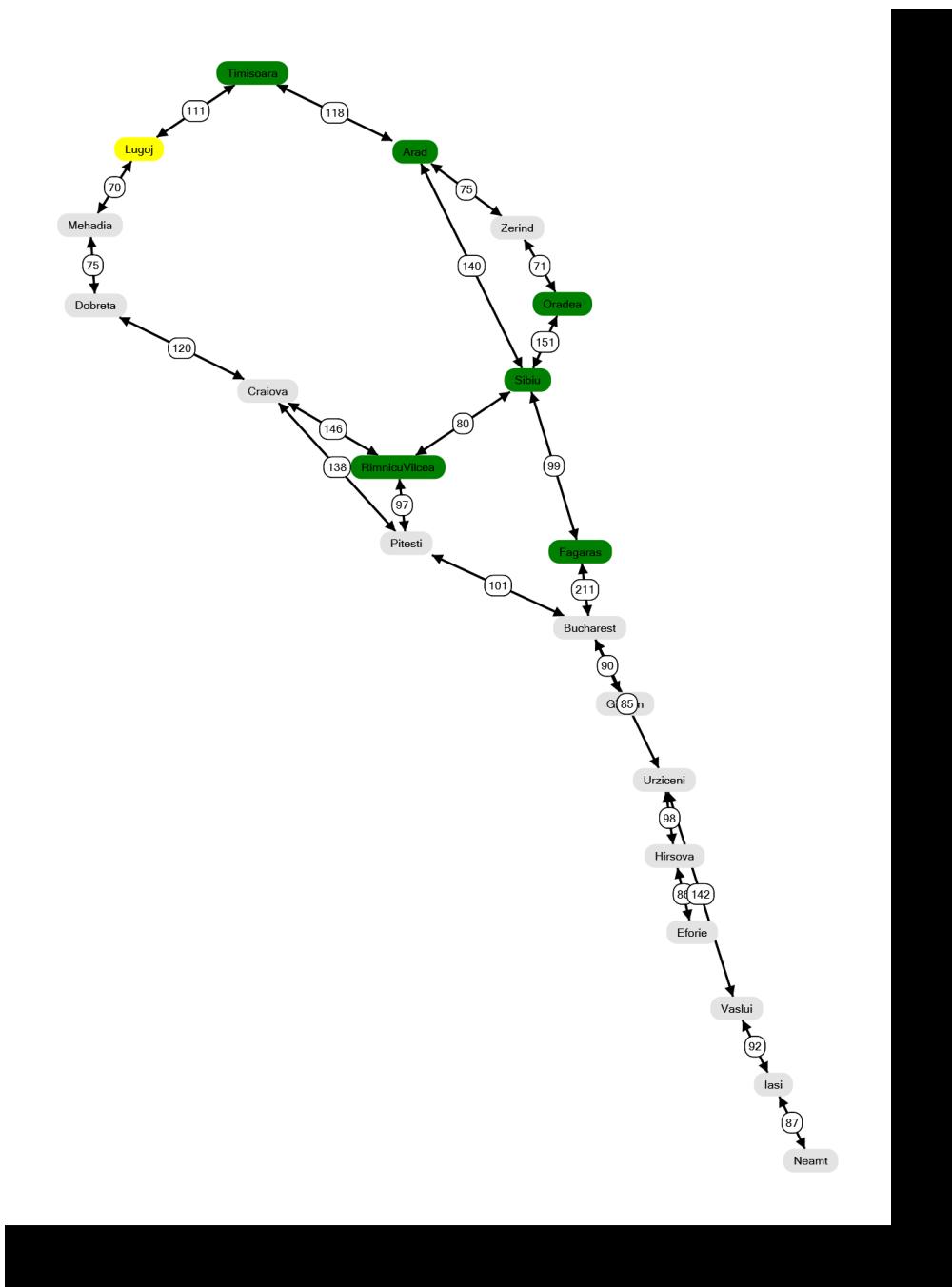
Visit

The algorithm has marked the node Arad as visited



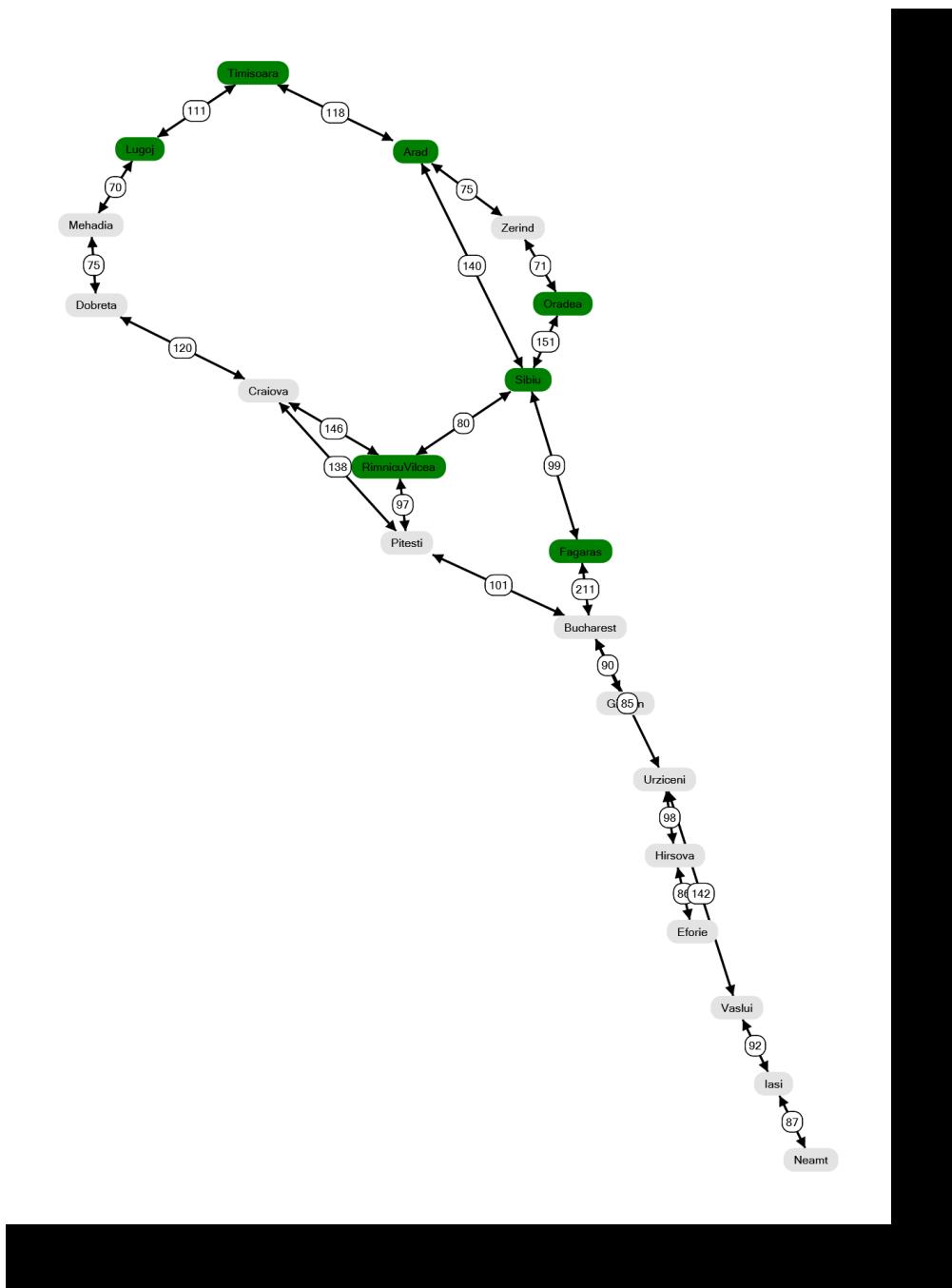
Pre-Visit

The algorithm has opened the node Lugoj



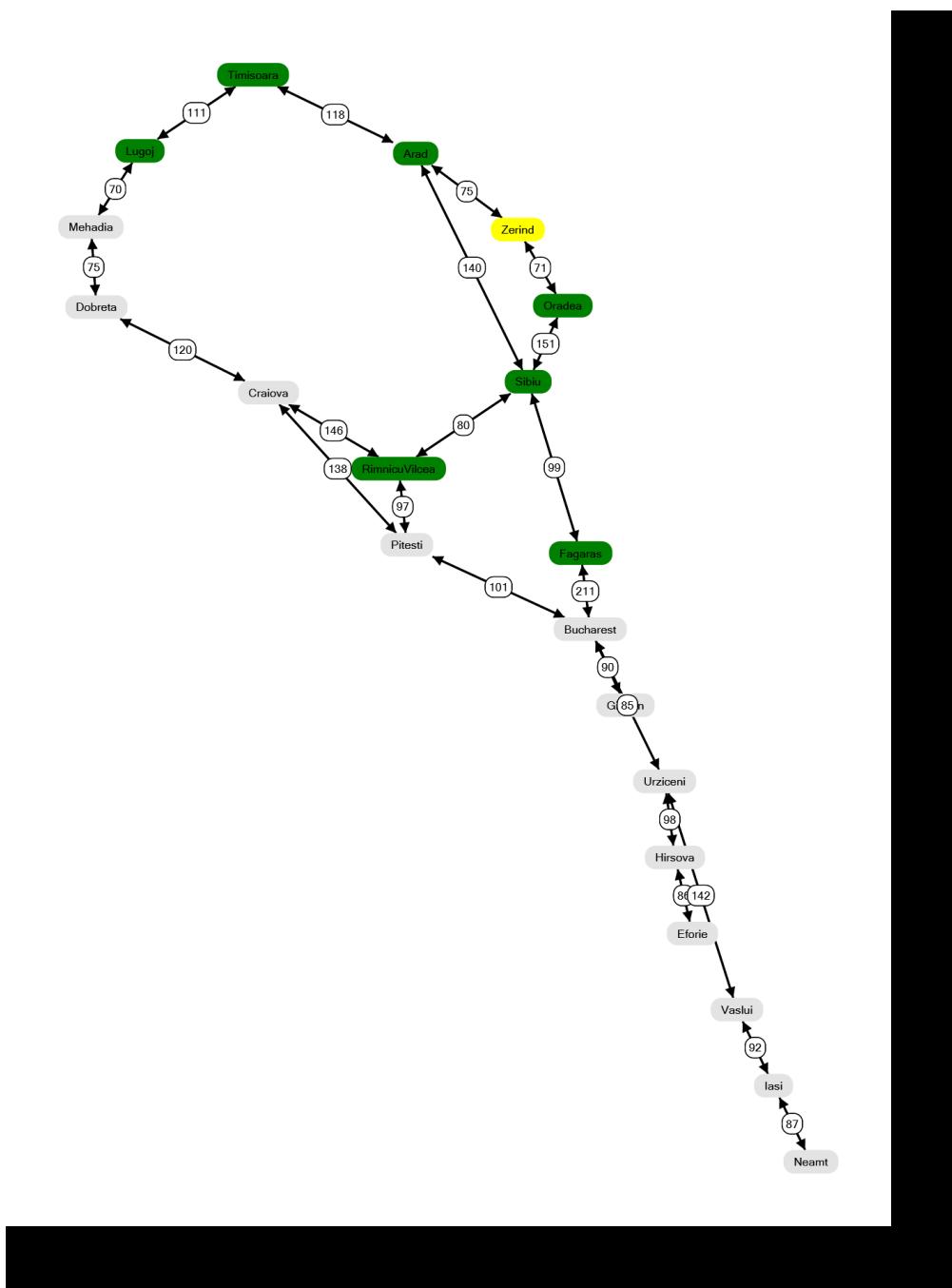
Visit

The algorithm has marked the node Lugoj as visited



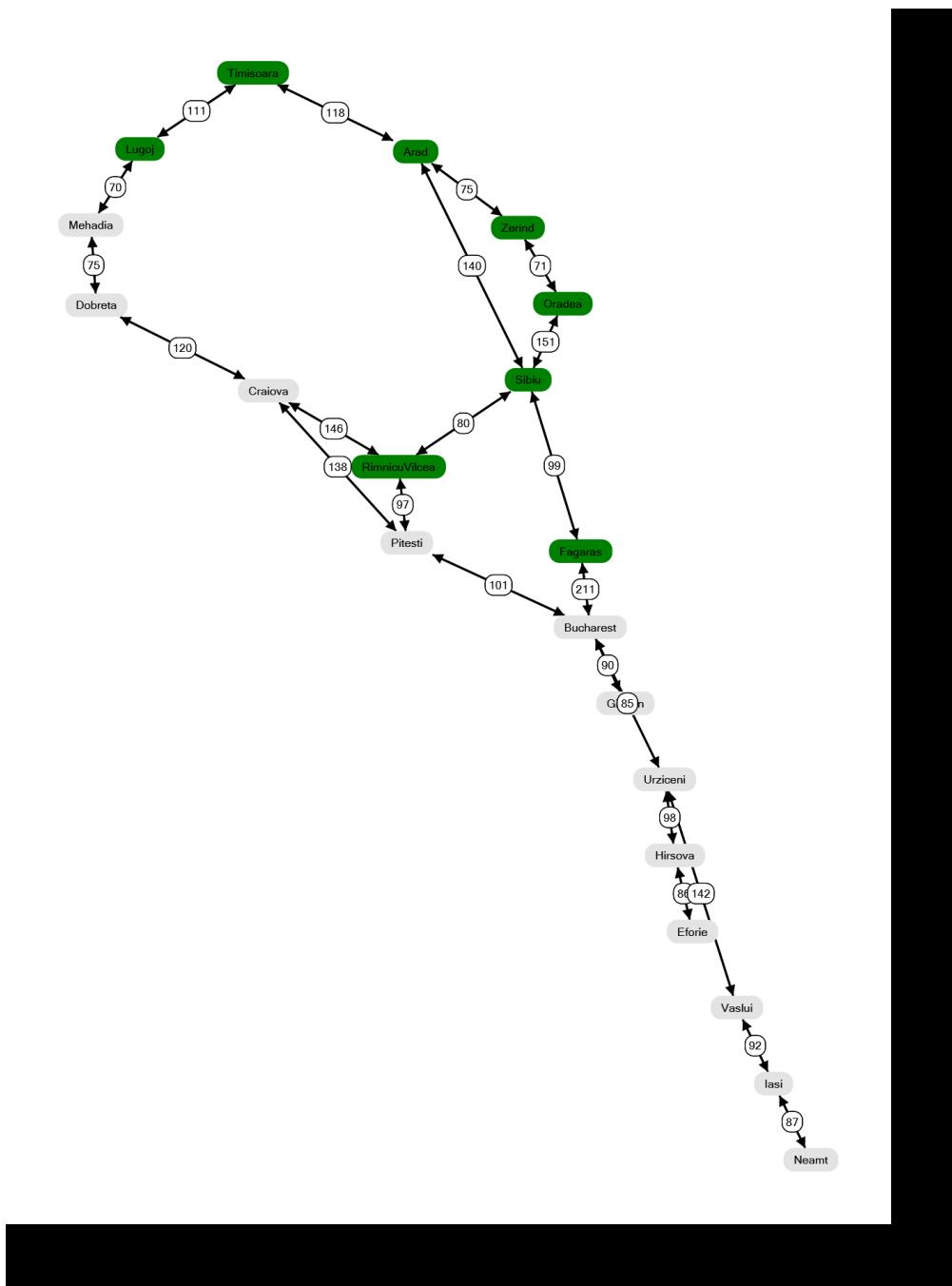
Pre-Visit

The algorithm has opened the node Zerind



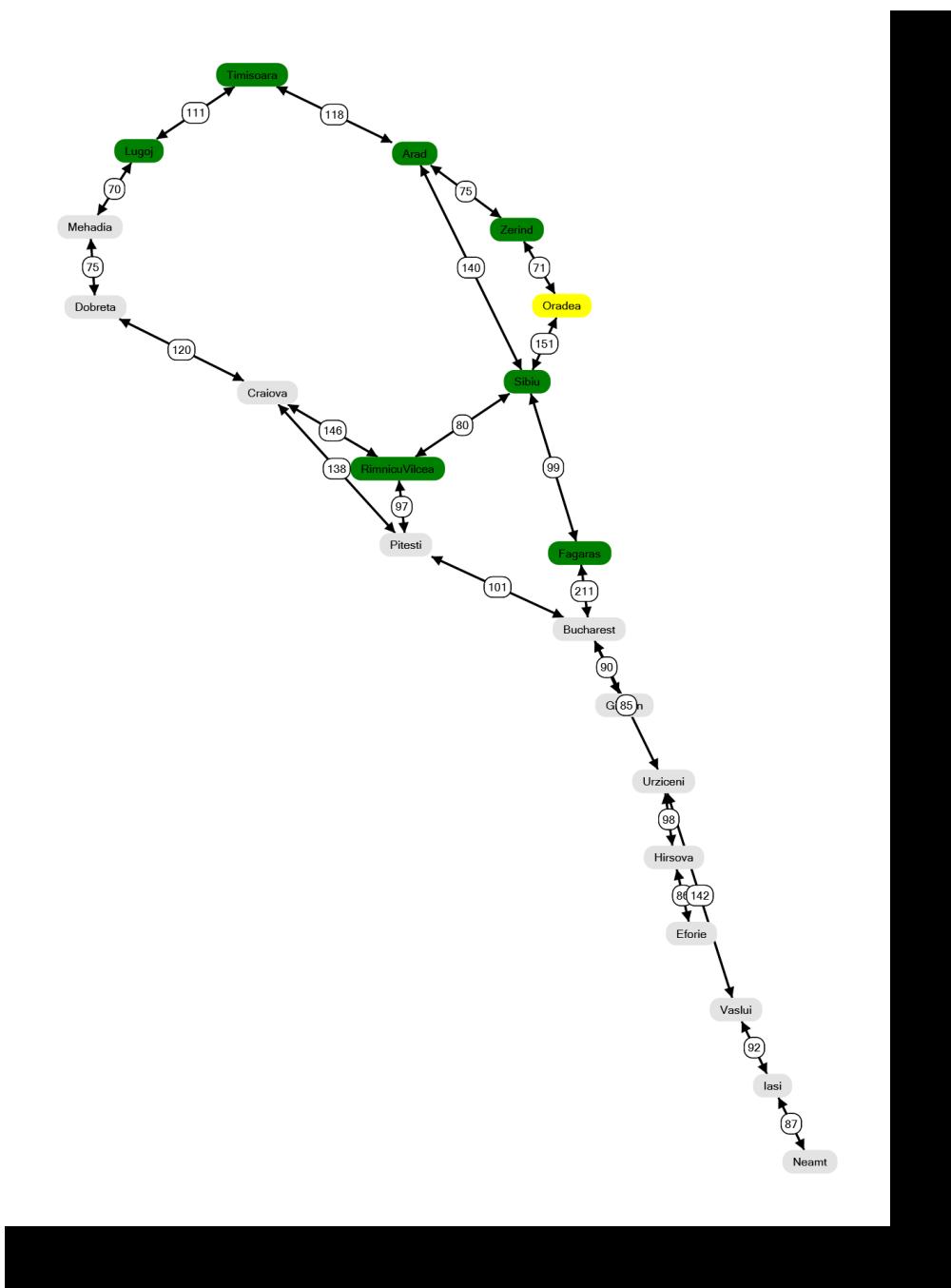
Visit

The algorithm has marked the node Zerind as visited



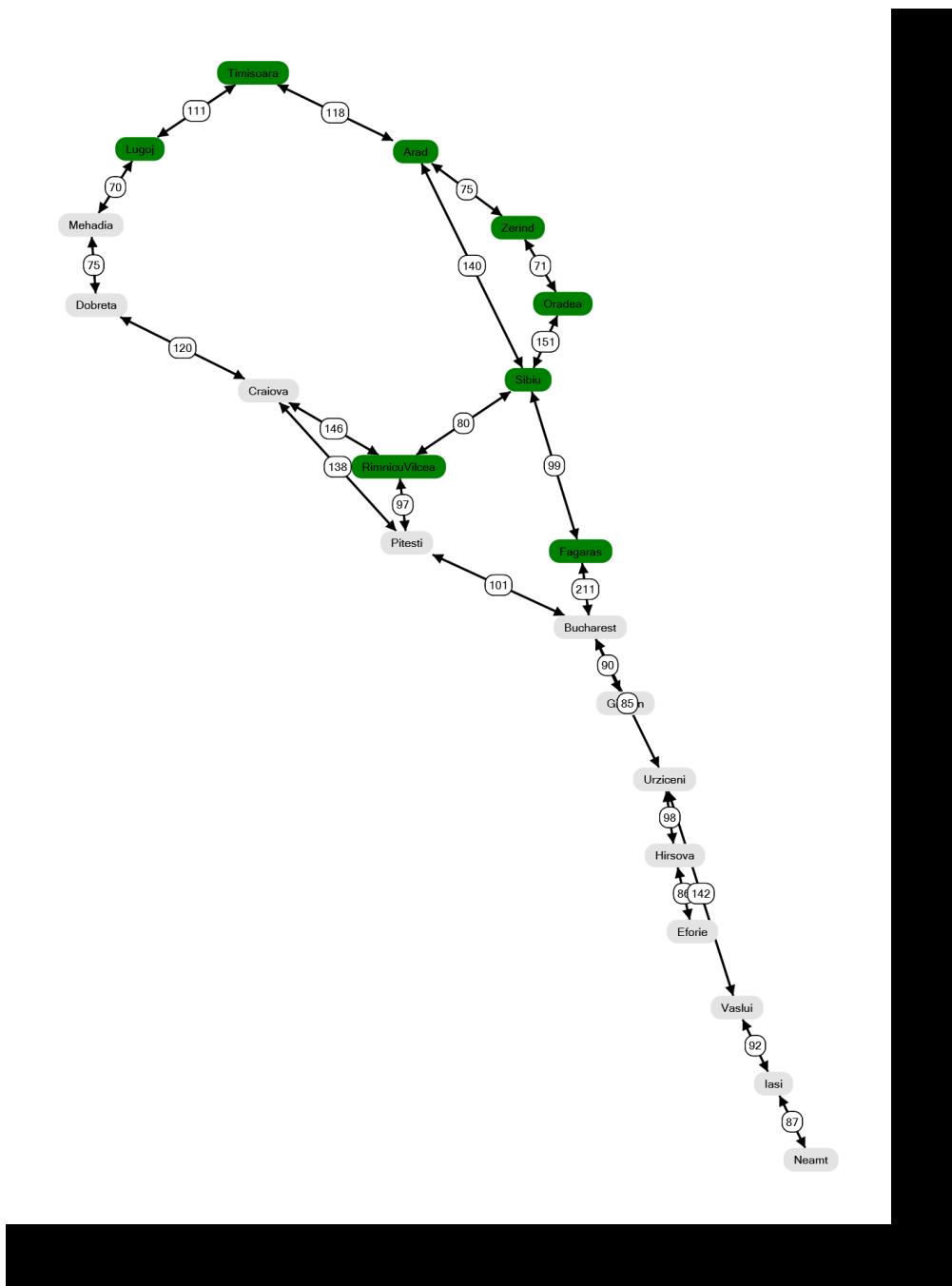
Pre-Visit

The algorithm has opened the node Oradea



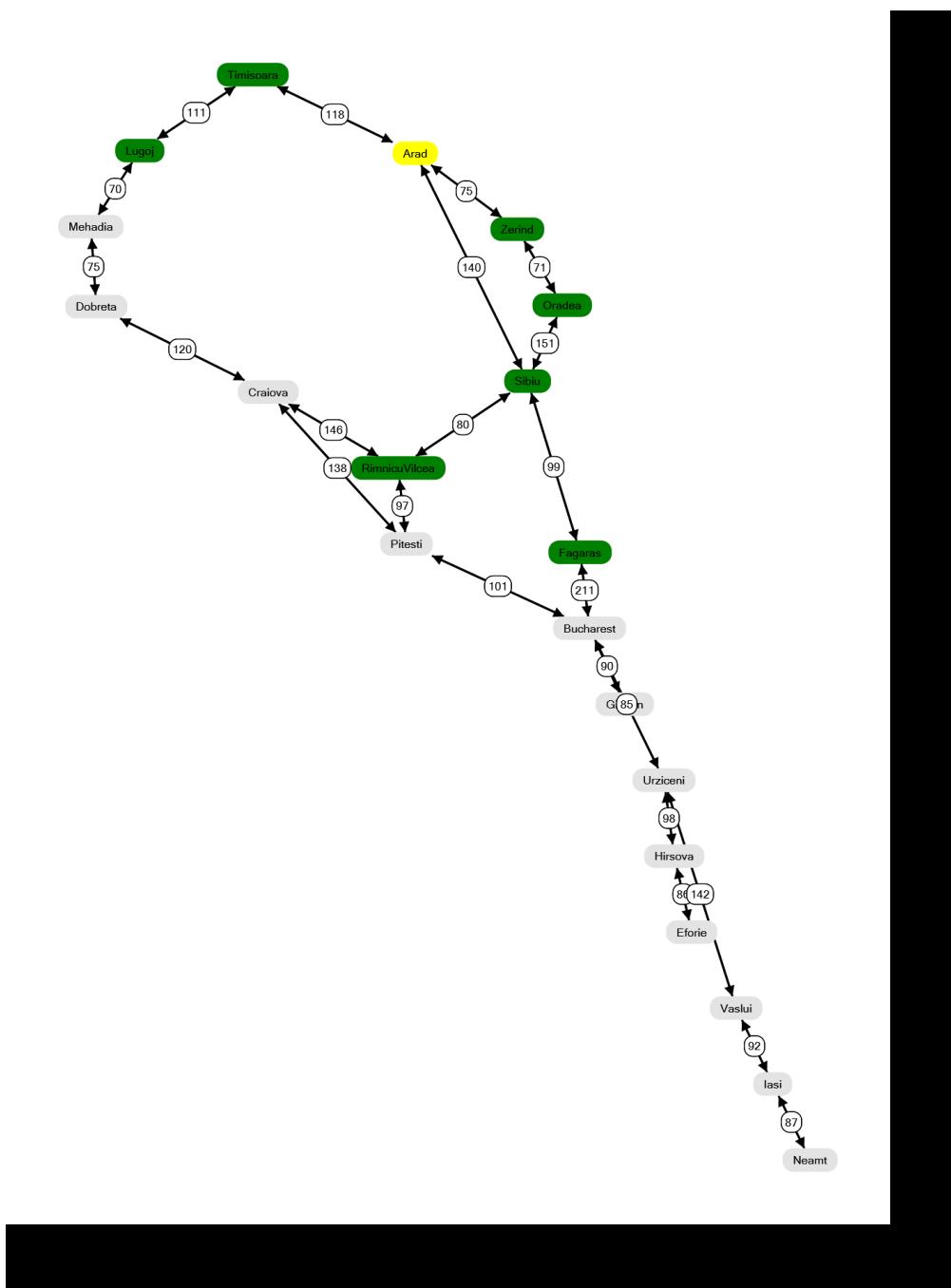
Visit

The algorithm has marked the node Oradea as visited



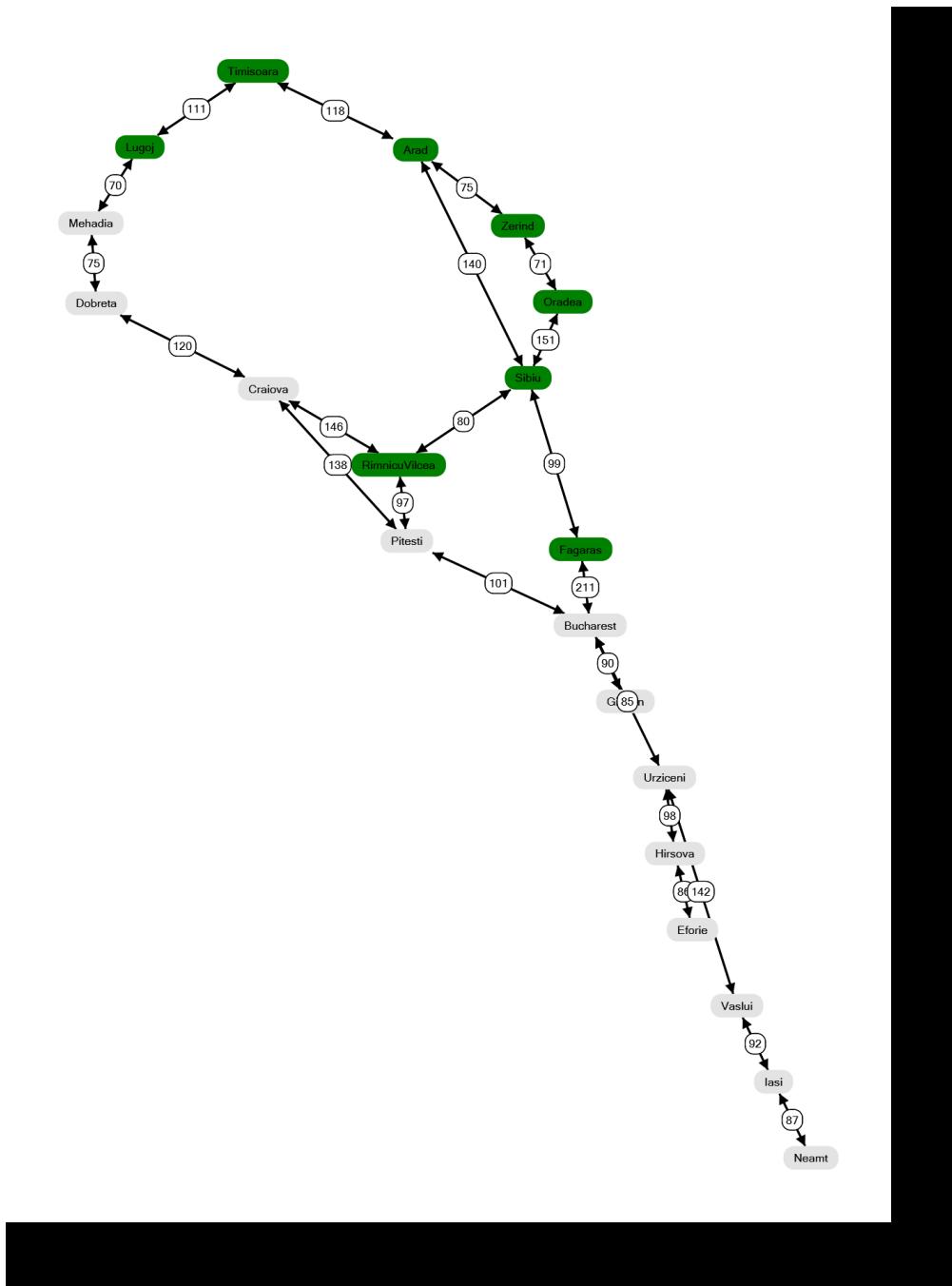
Pre-Visit

The algorithm has opened the node Arad



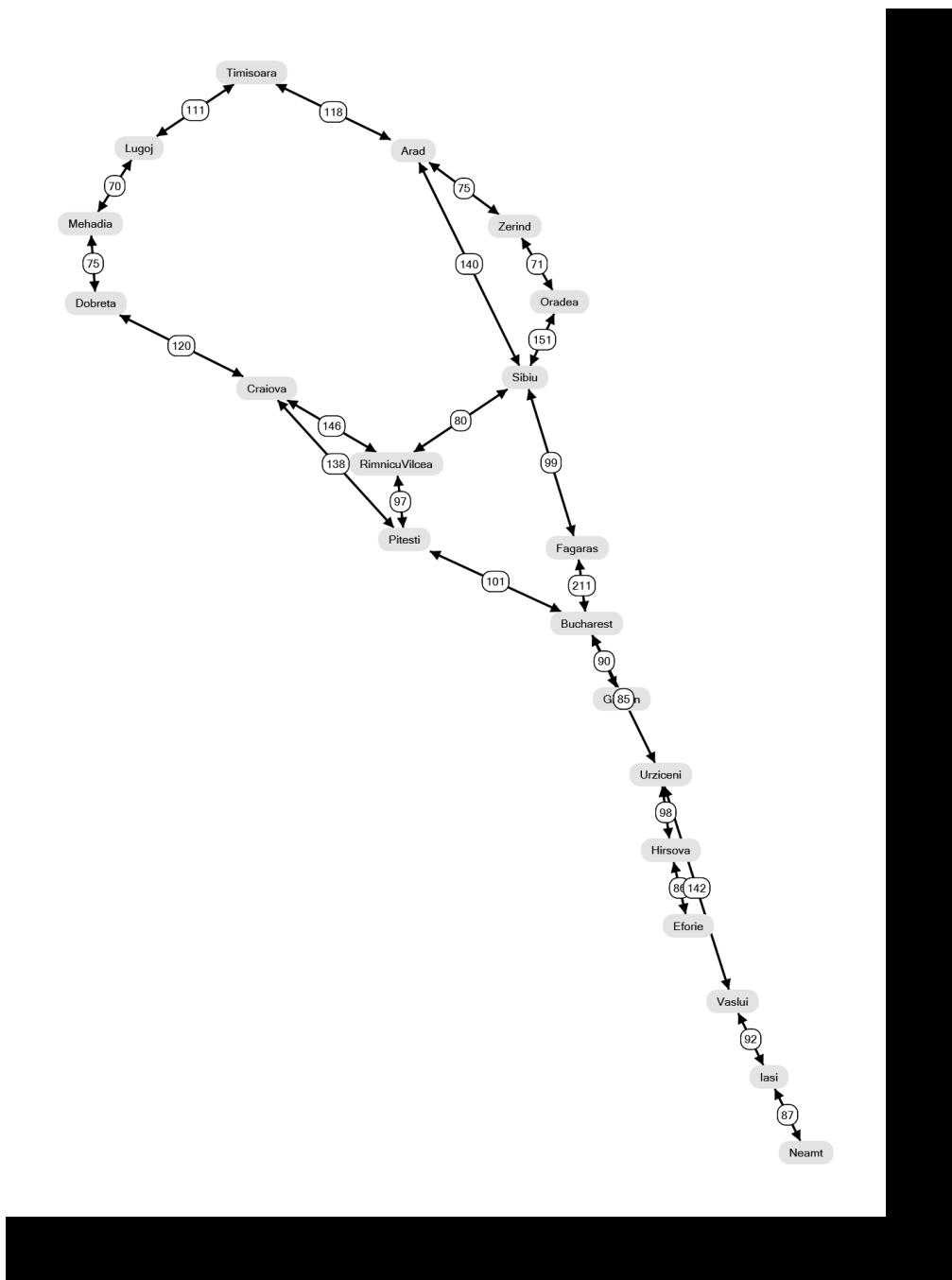
Visit

The algorithm has marked the node Arad as visited



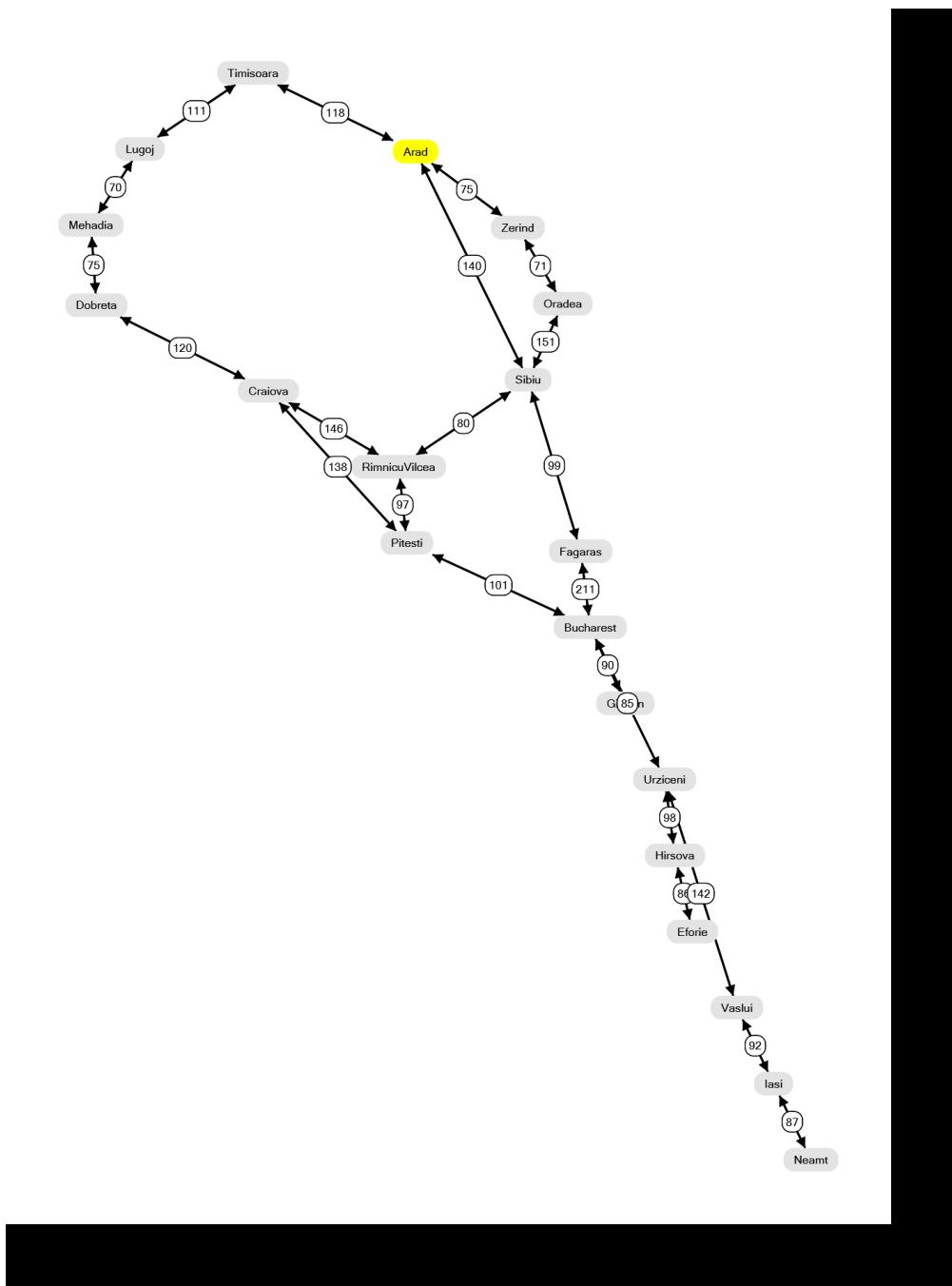
Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



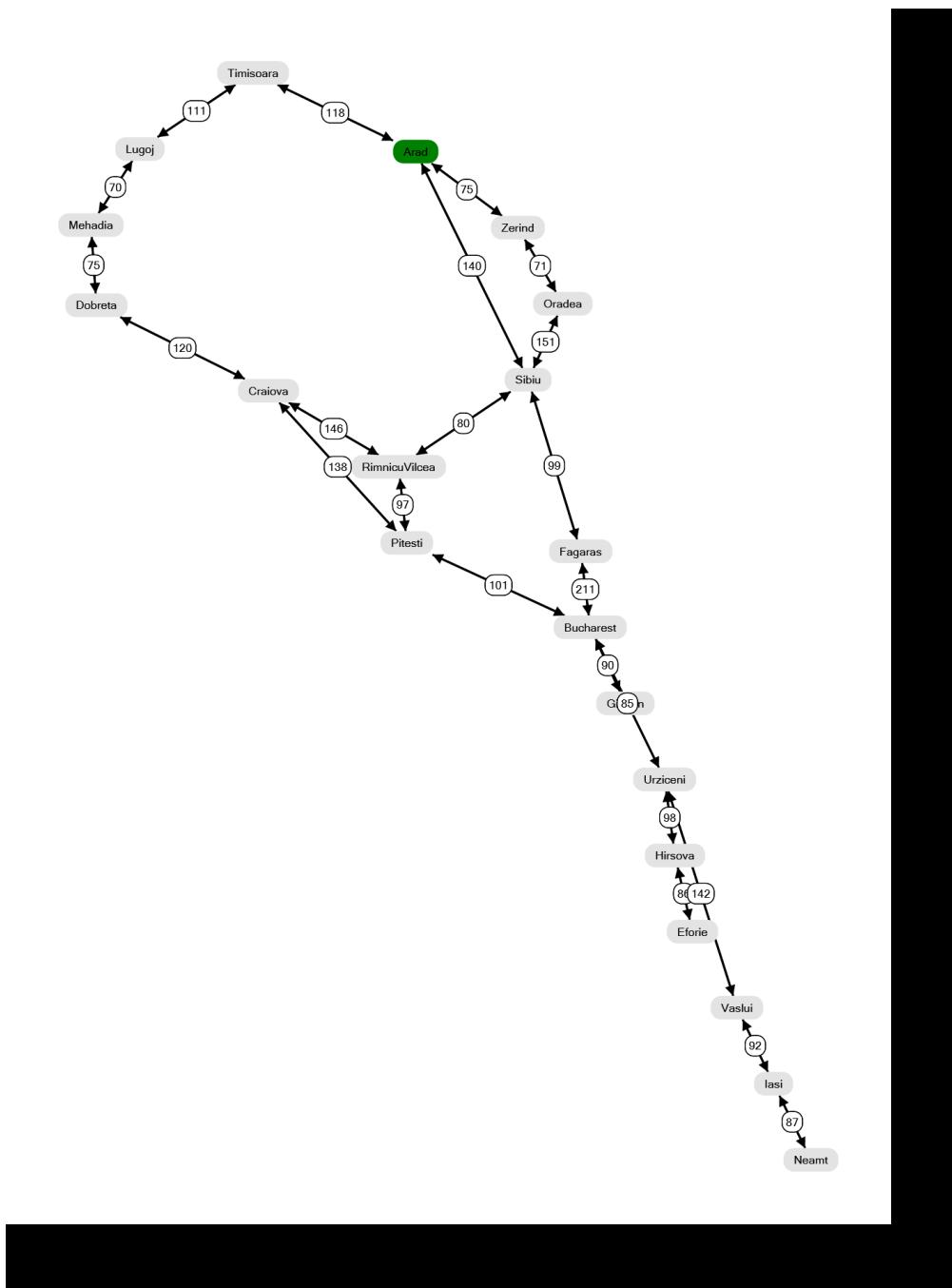
Pre-Visit

The algorithm has opened the node Arad



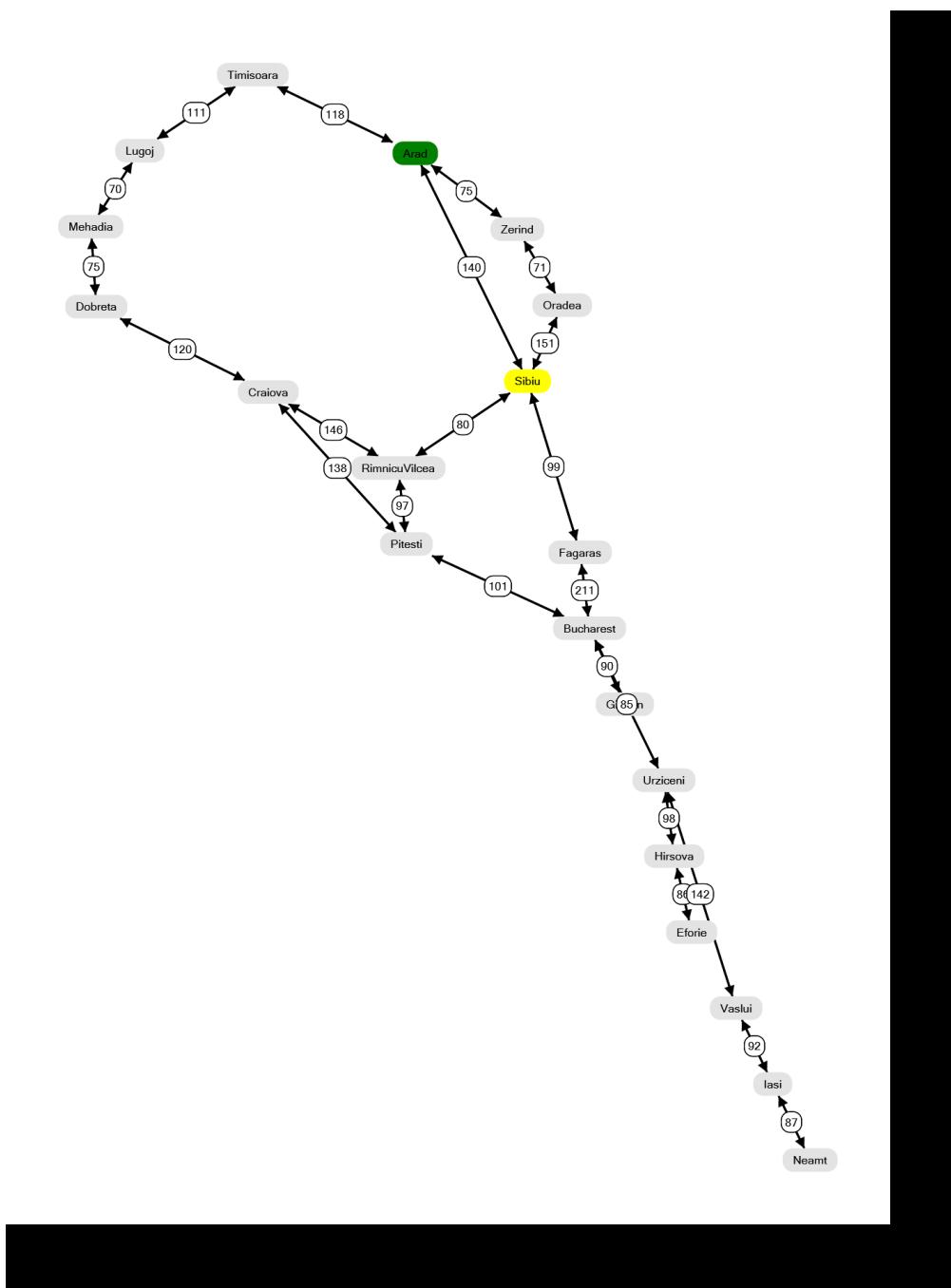
Visit

The algorithm has marked the node Arad as visited



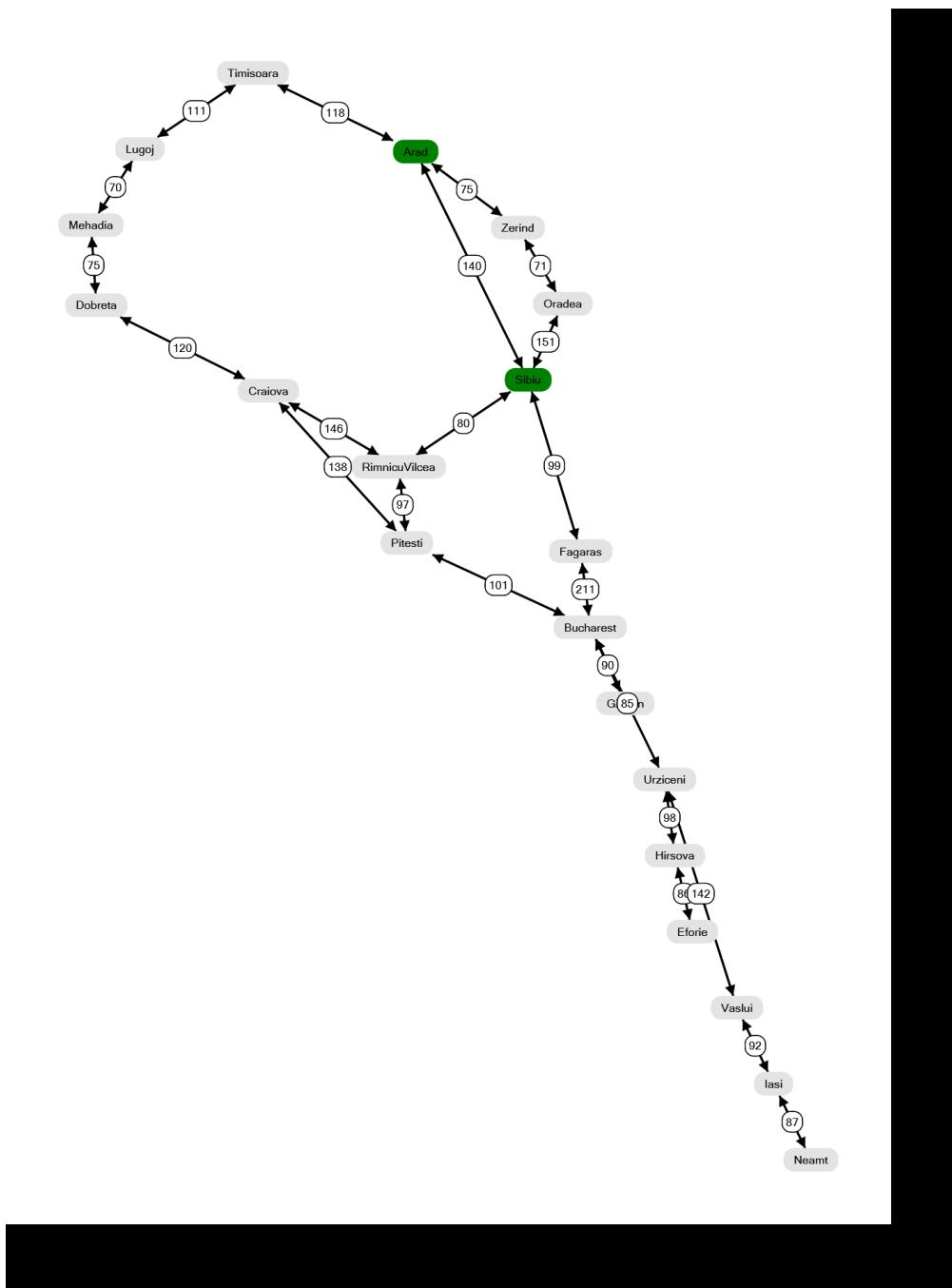
Pre-Visit

The algorithm has opened the node Sibiu



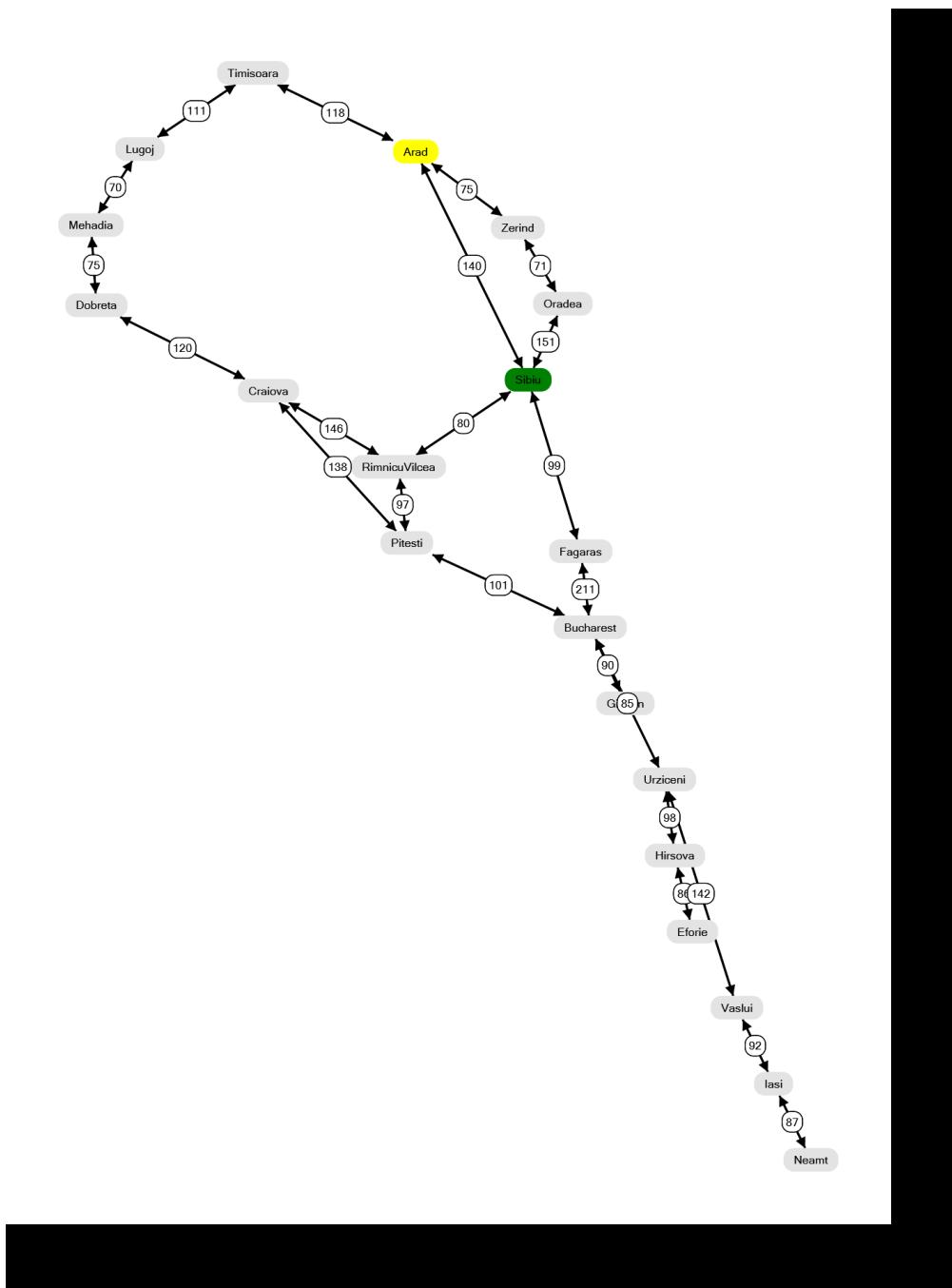
Visit

The algorithm has marked the node Sibiu as visited



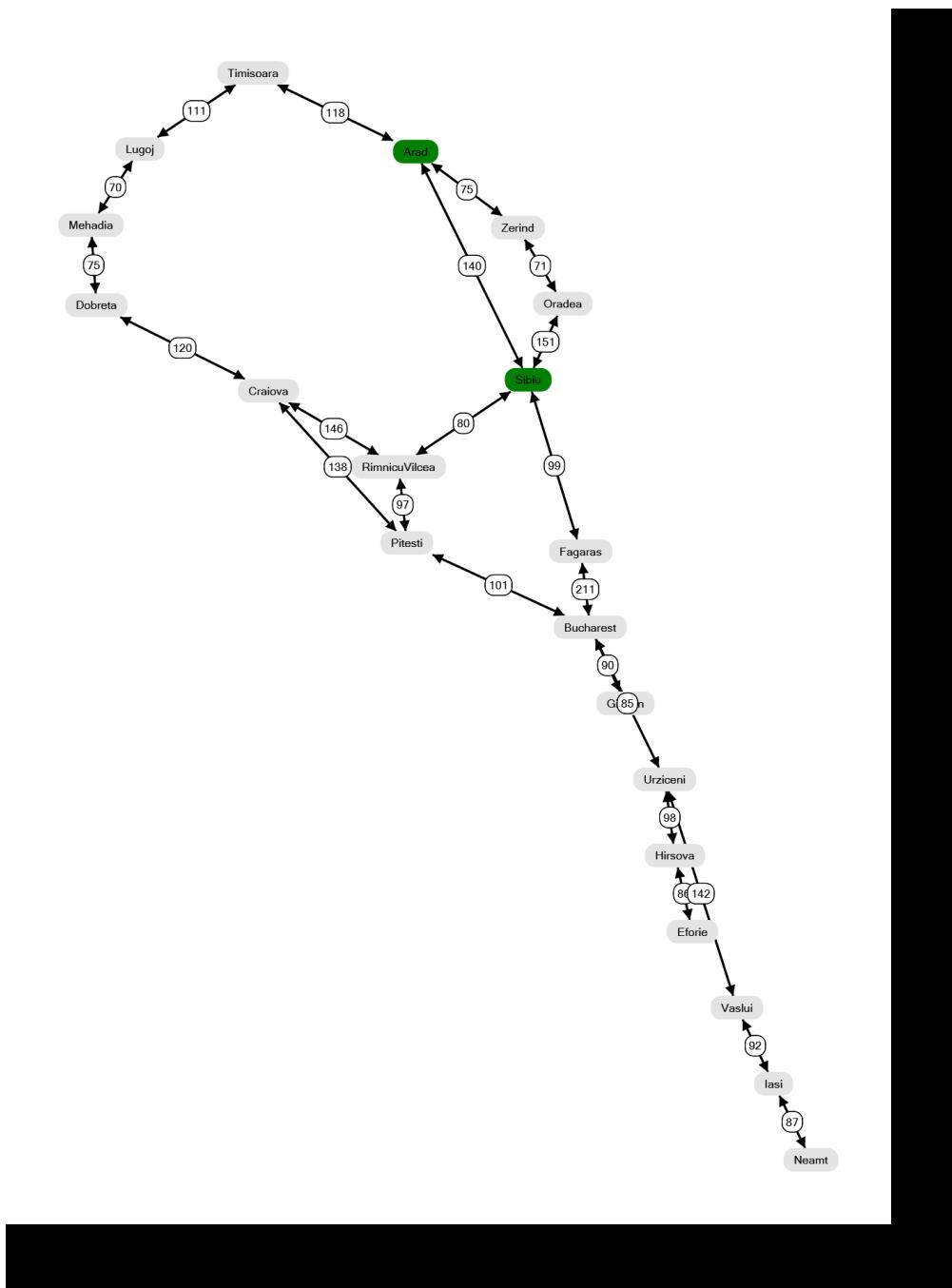
Pre-Visit

The algorithm has opened the node Arad



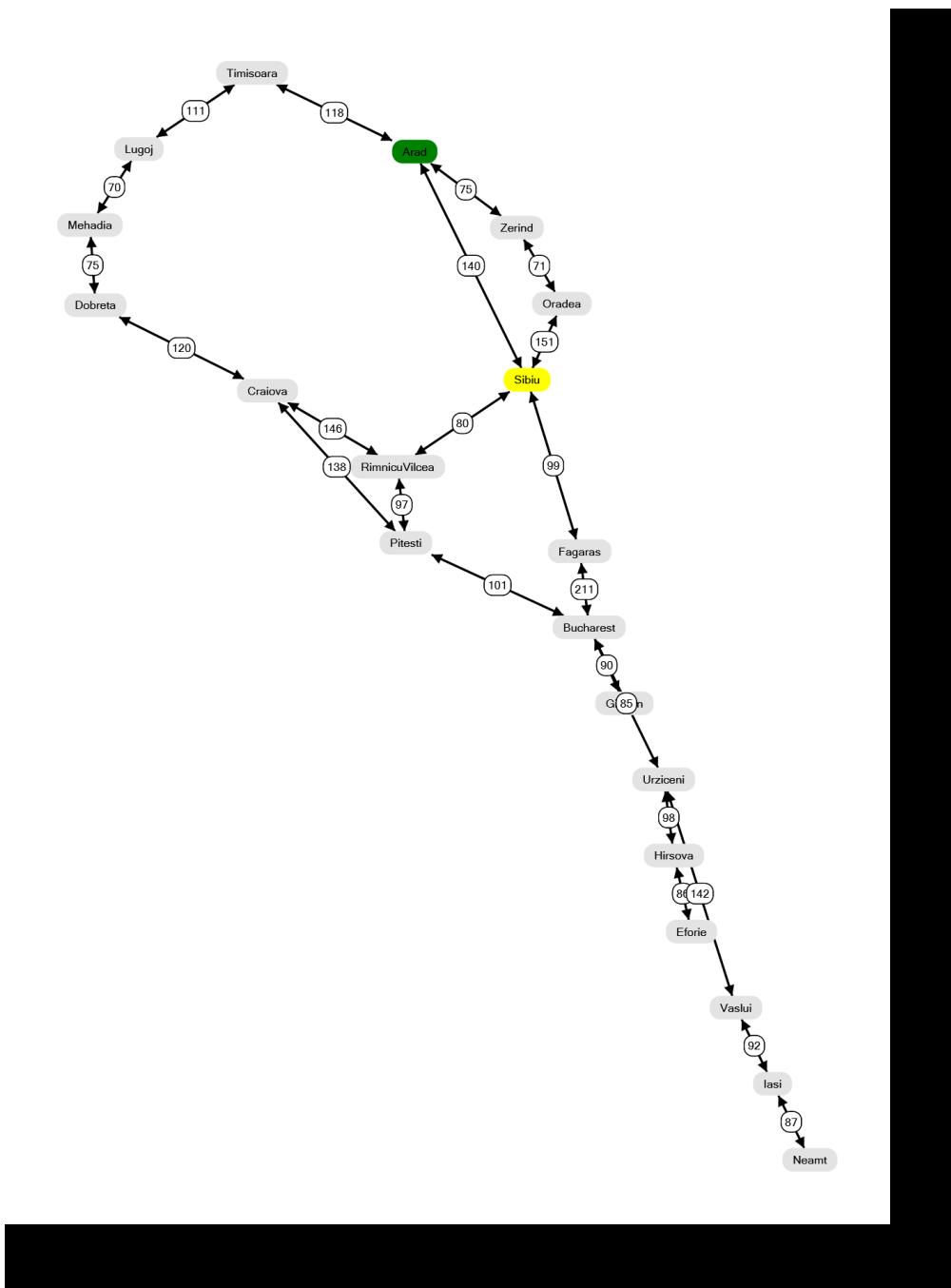
Visit

The algorithm has marked the node Arad as visited



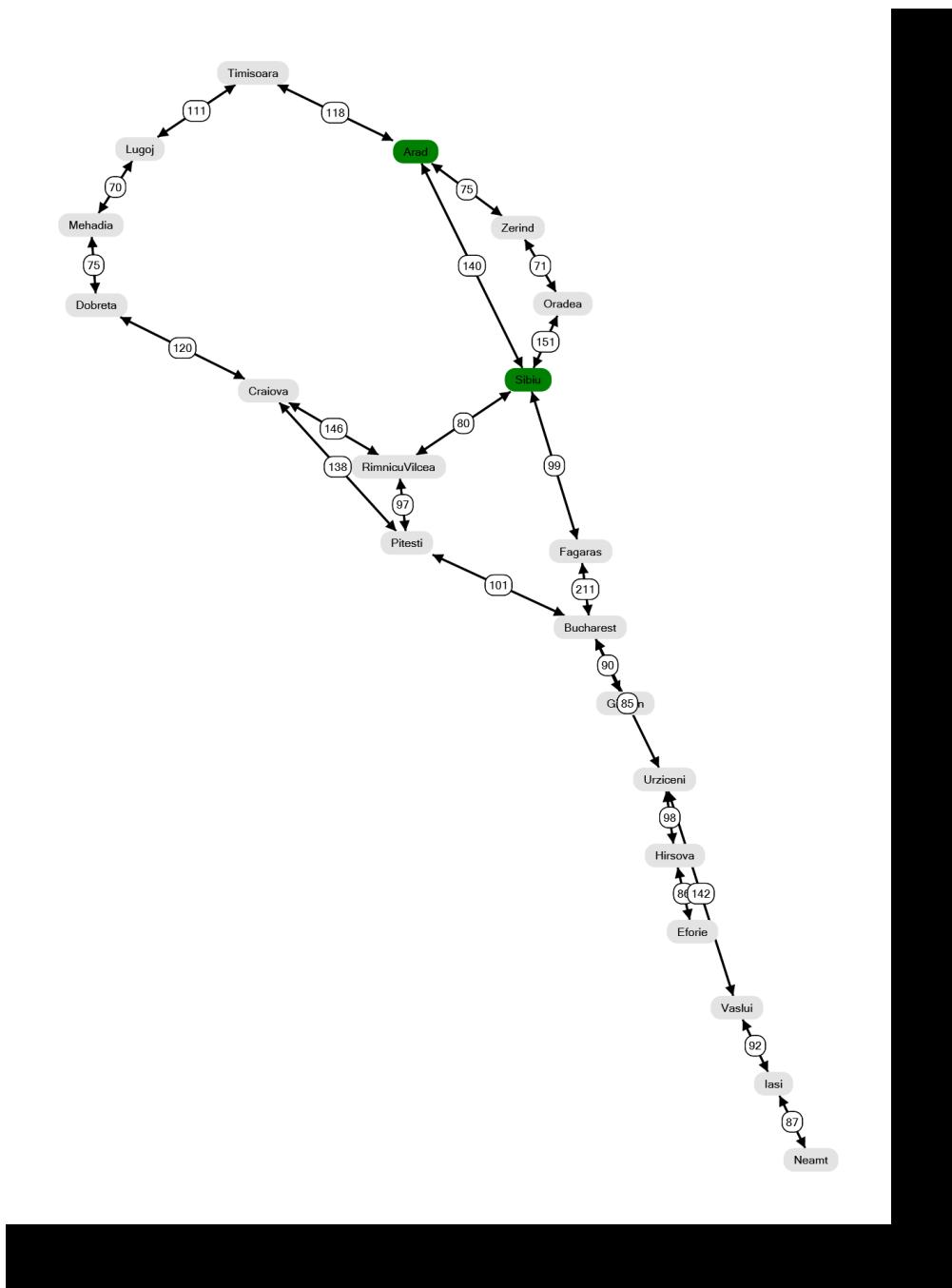
Pre-Visit

The algorithm has opened the node Sibiu



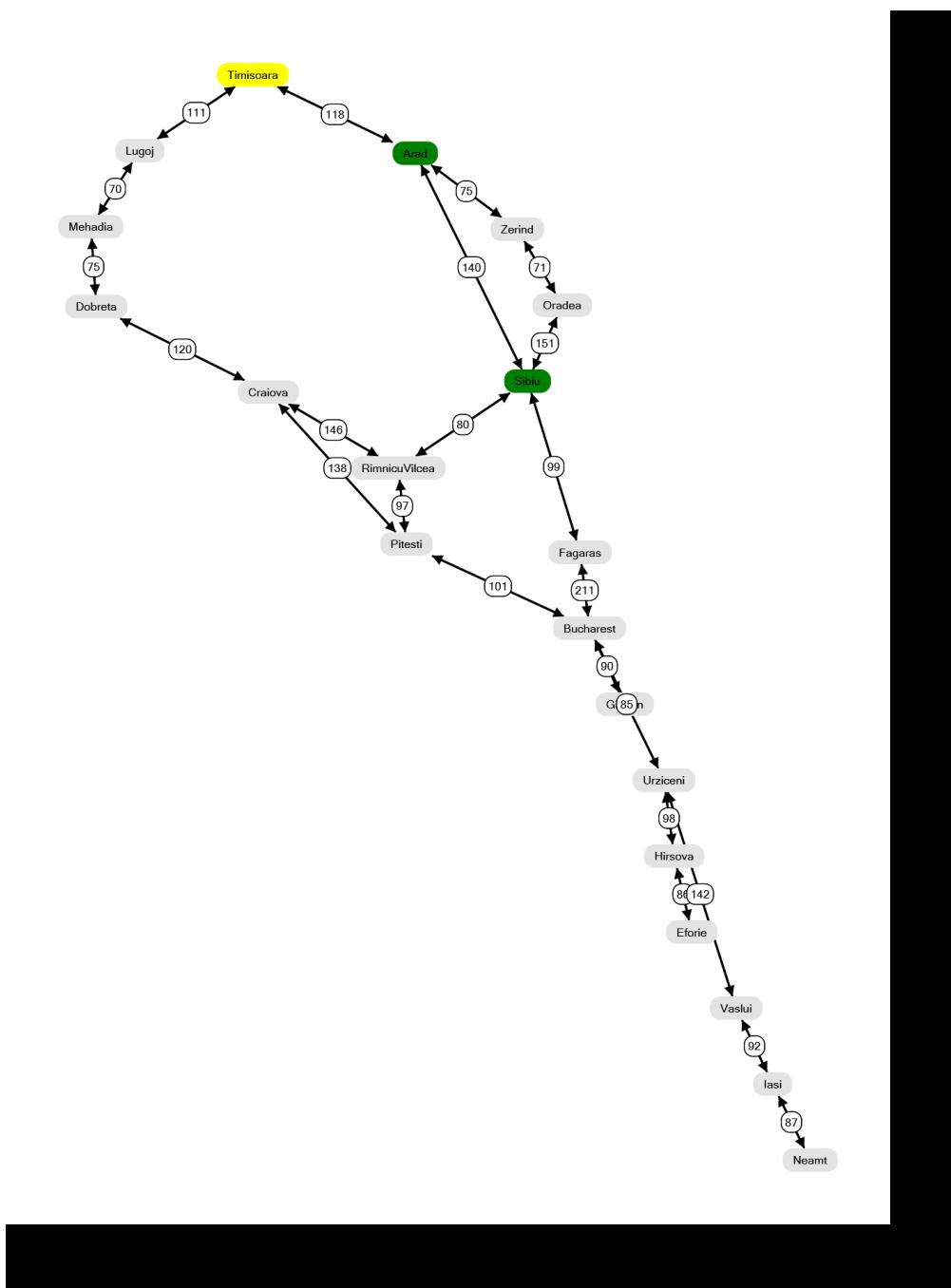
Visit

The algorithm has marked the node Sibiu as visited



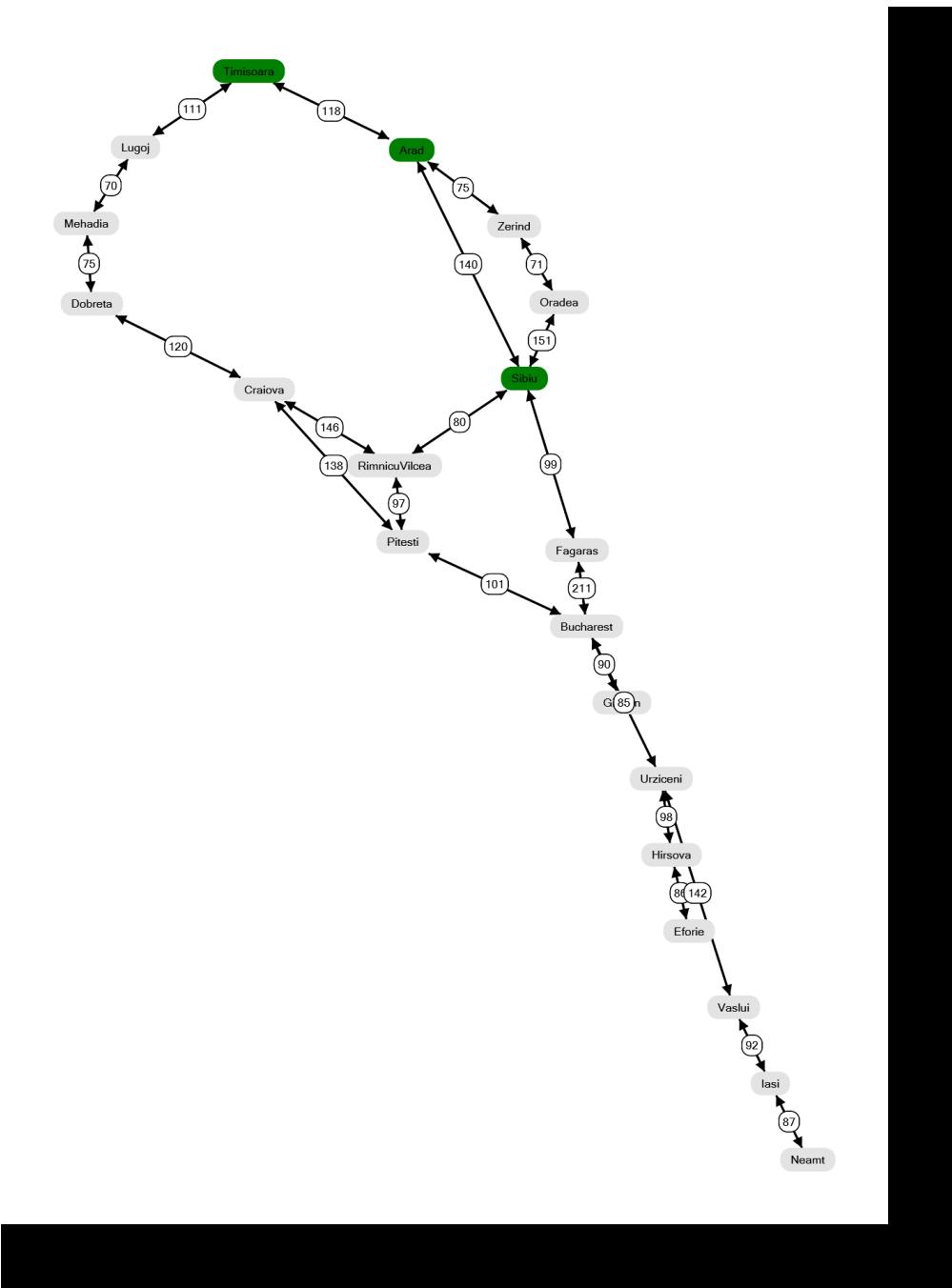
Pre-Visit

The algorithm has opened the node Timisoara



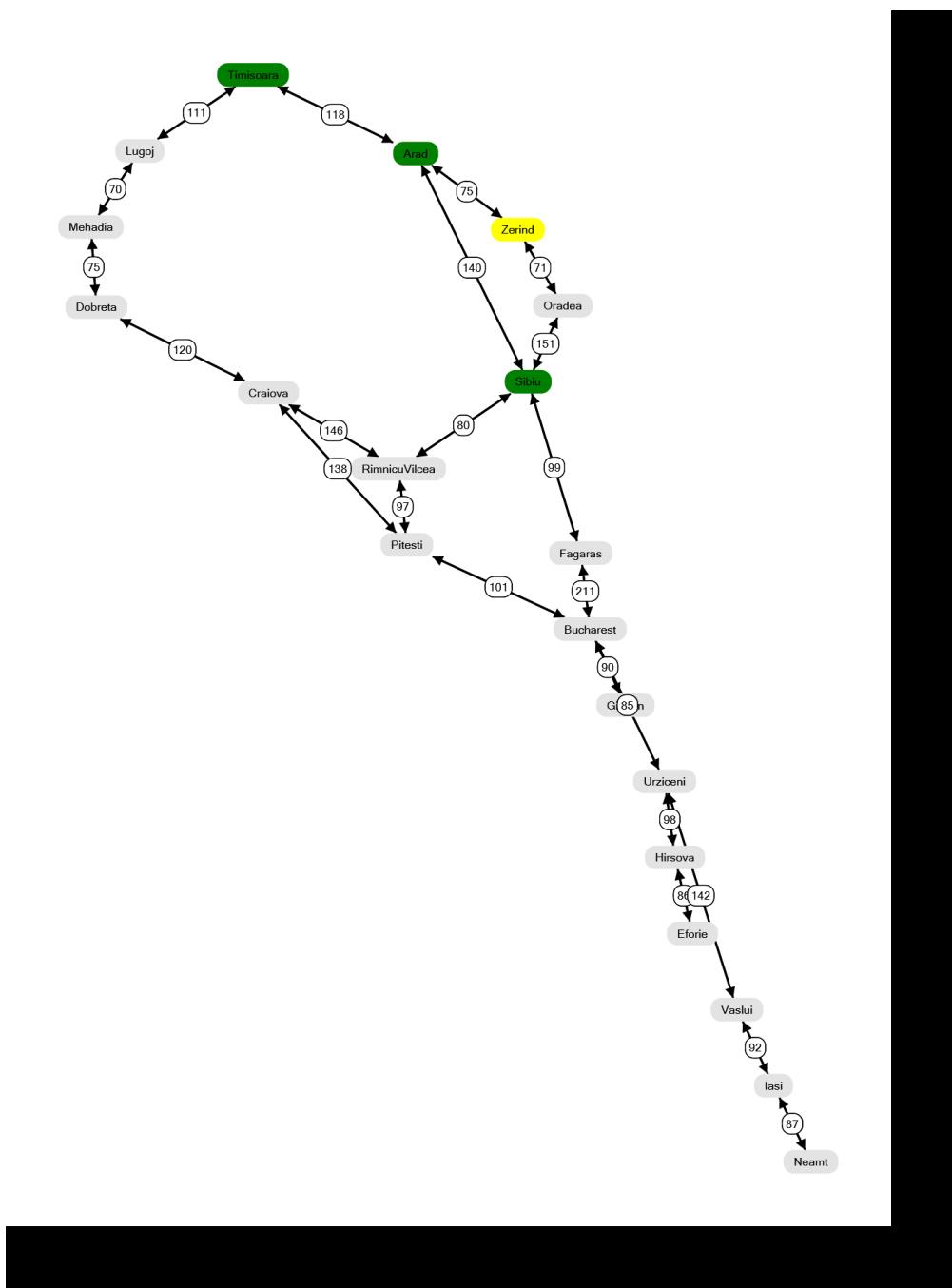
Visit

The algorithm has marked the node Timisoara as visited



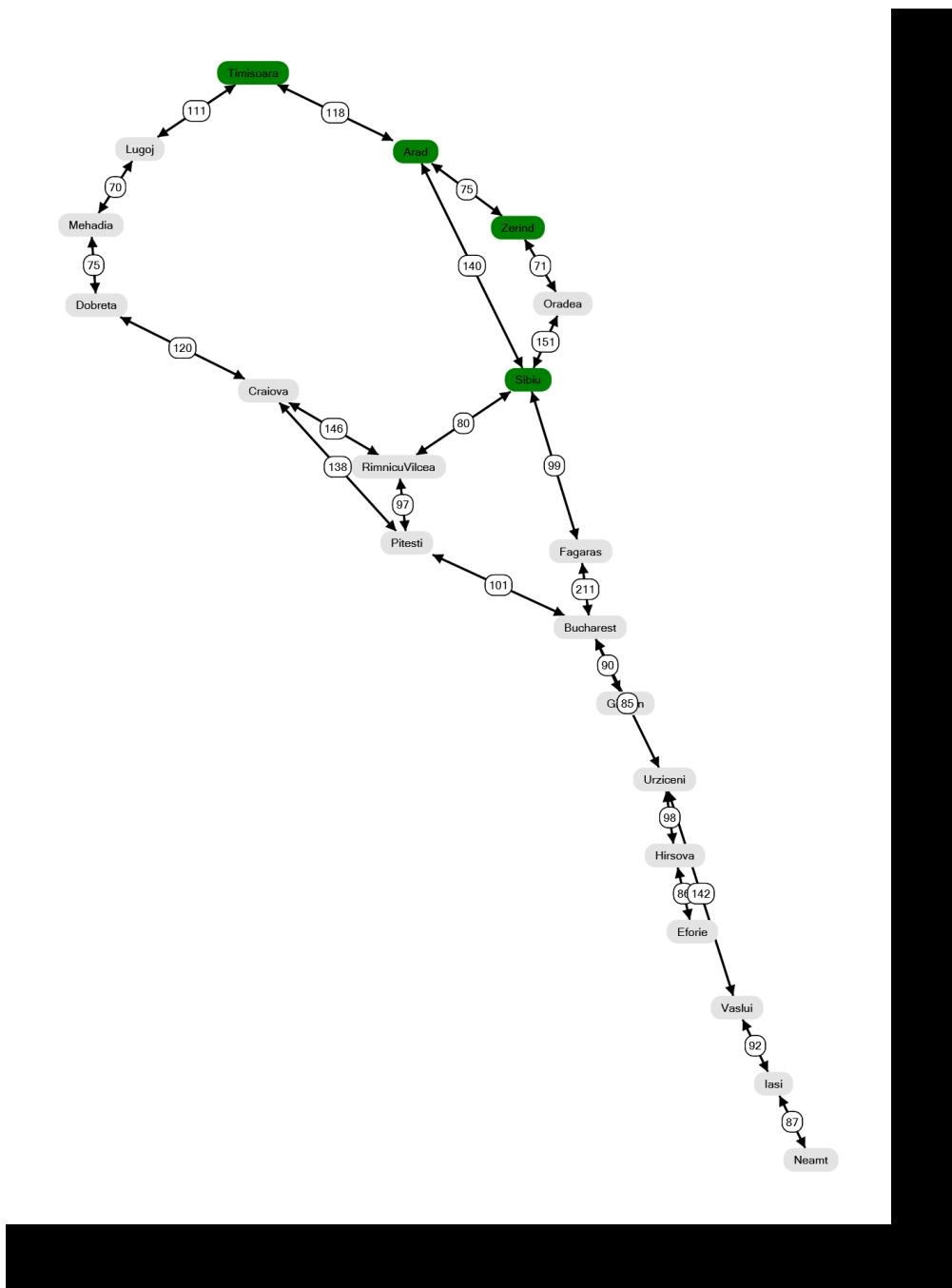
Pre-Visit

The algorithm has opened the node Zerind



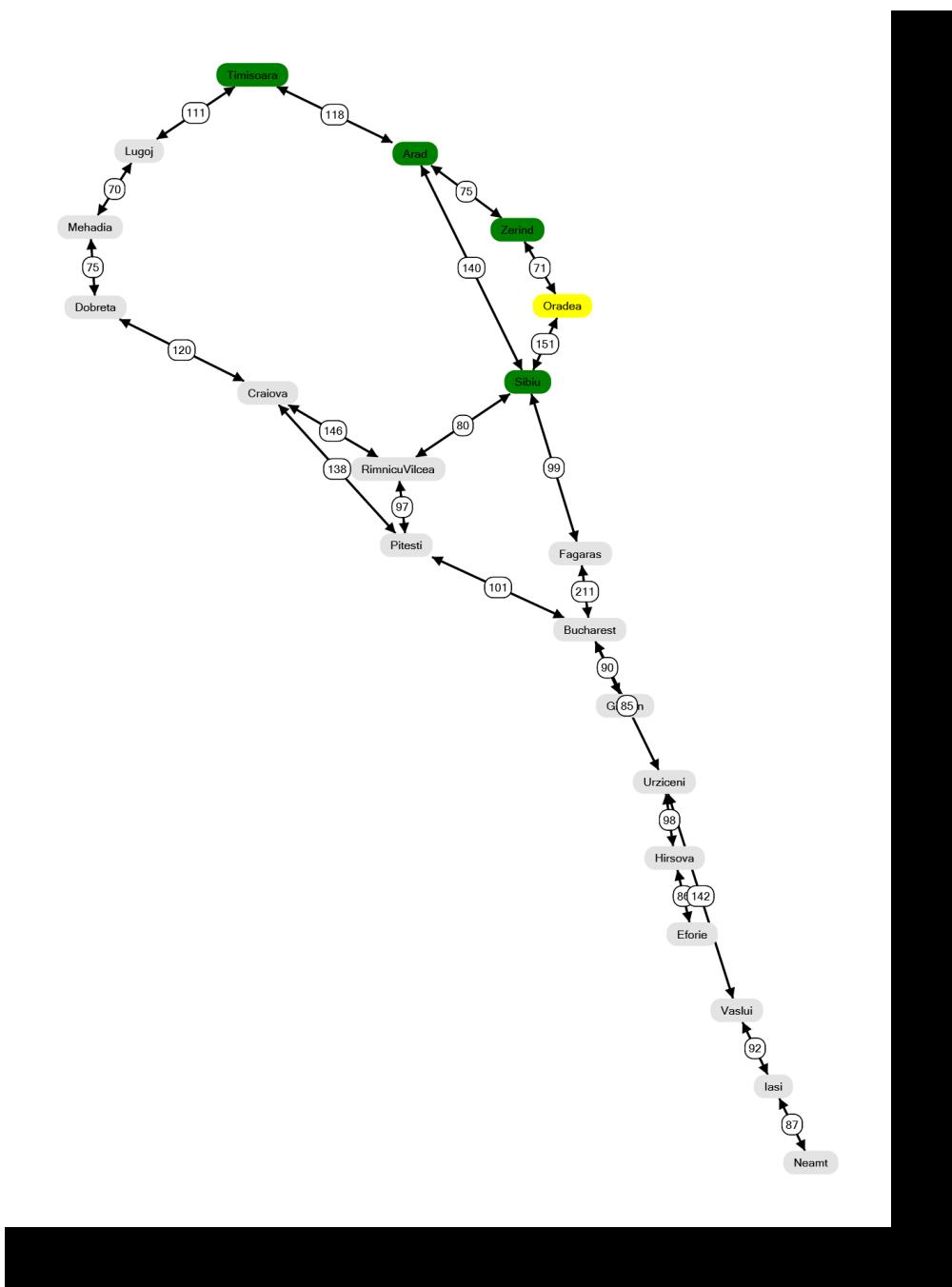
Visit

The algorithm has marked the node Zerind as visited



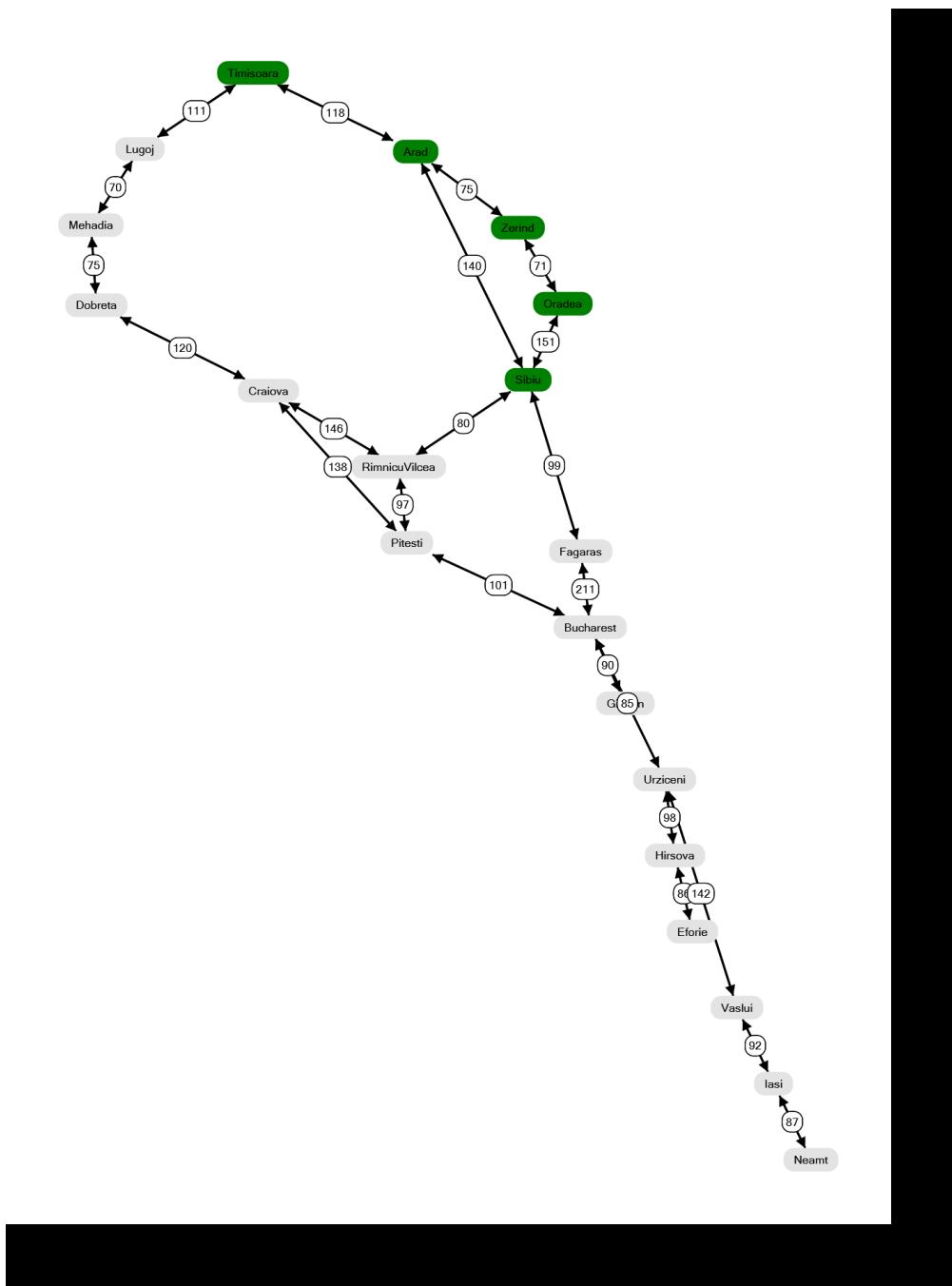
Pre-Visit

The algorithm has opened the node Oradea



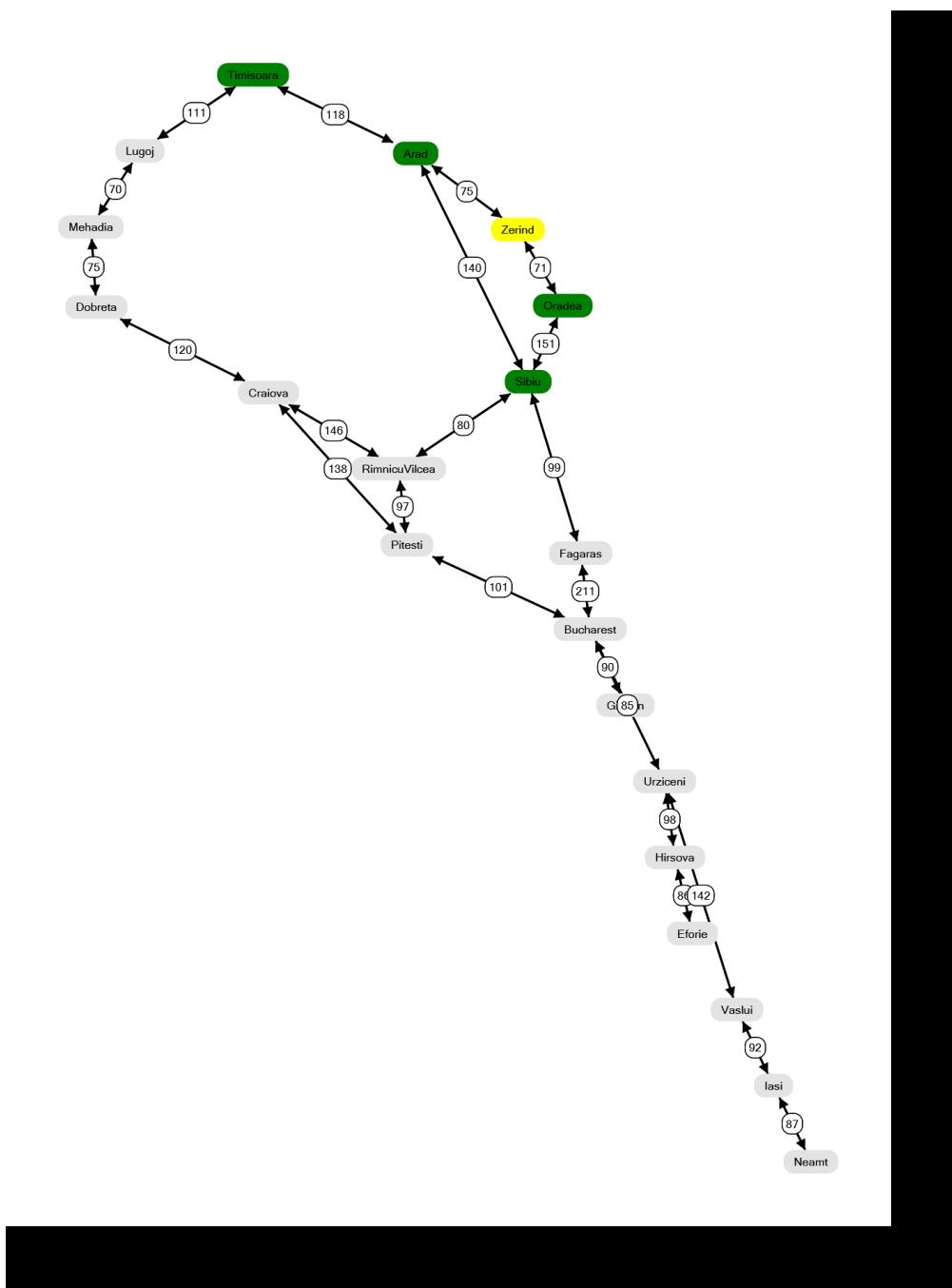
Visit

The algorithm has marked the node Oradea as visited



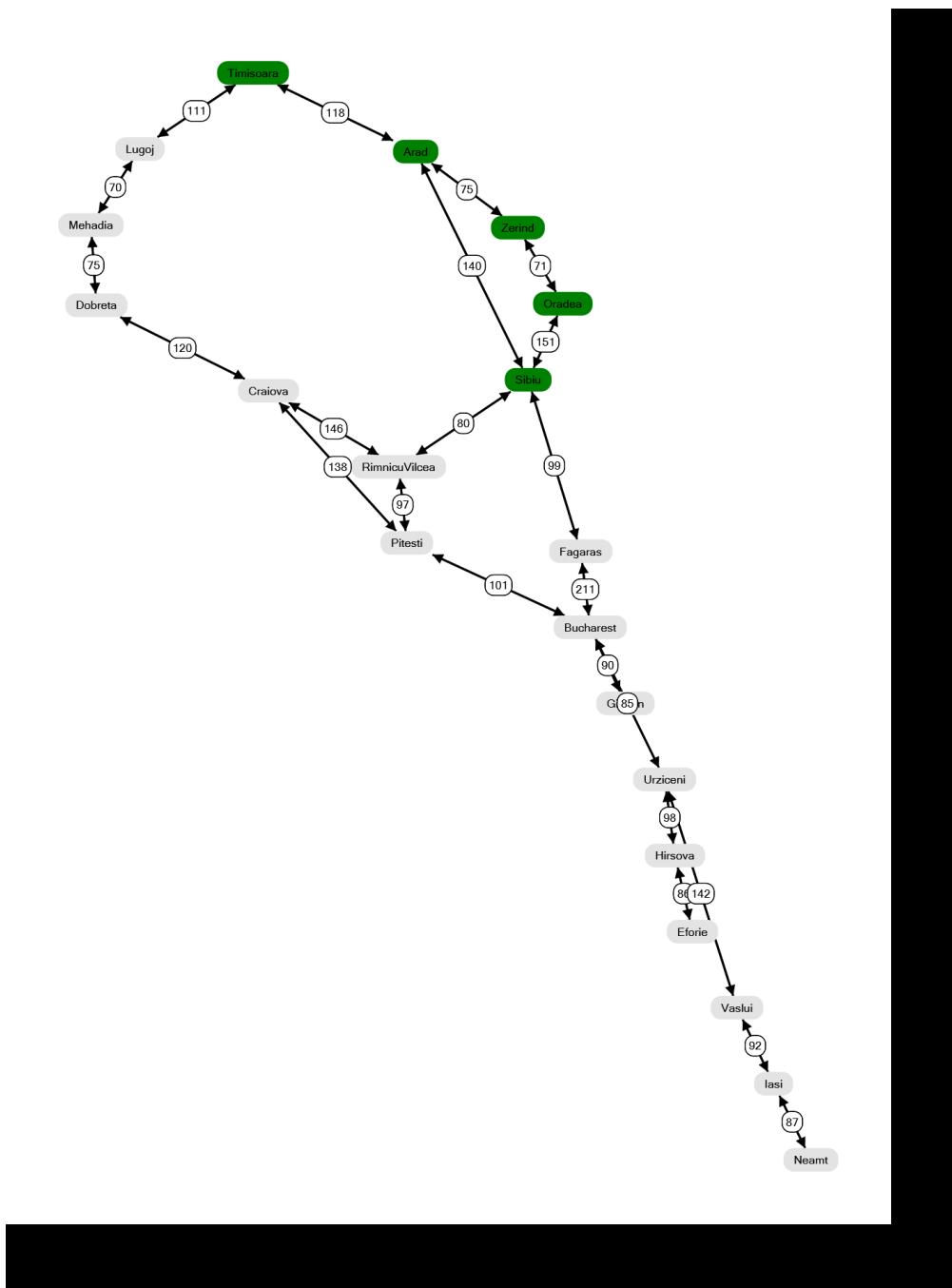
Pre-Visit

The algorithm has opened the node Zerind



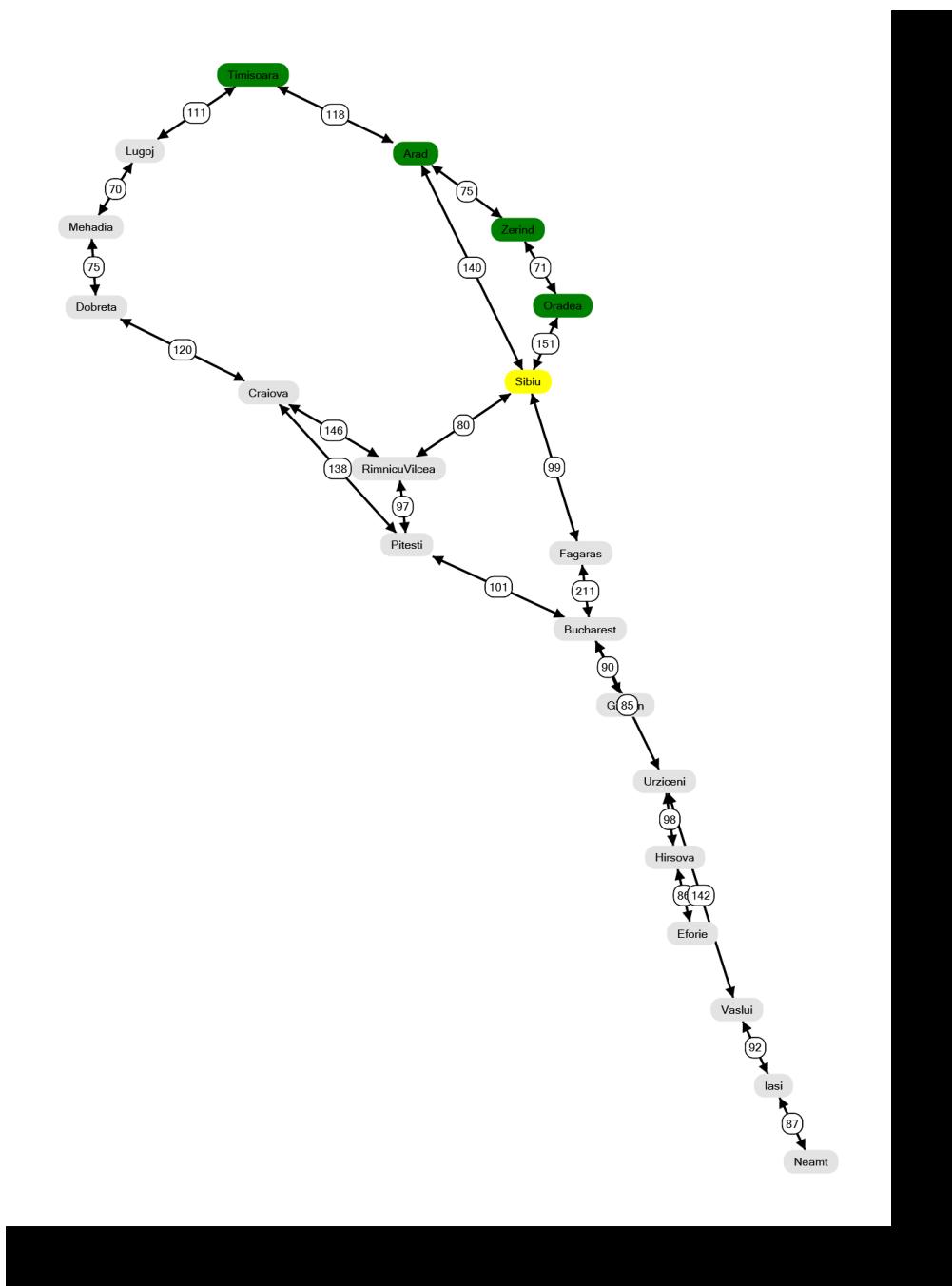
Visit

The algorithm has marked the node Zerind as visited



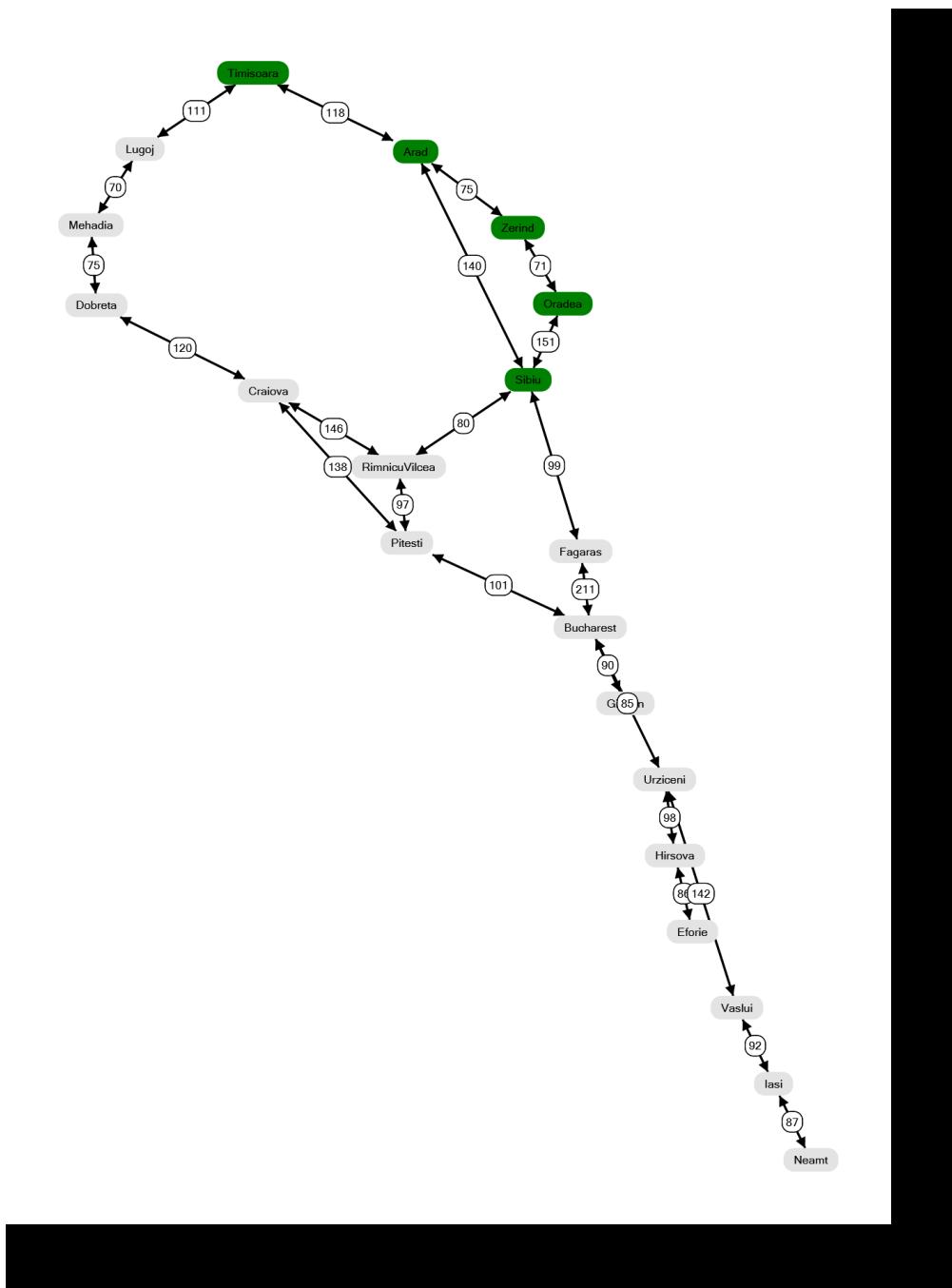
Pre-Visit

The algorithm has opened the node Sibiu



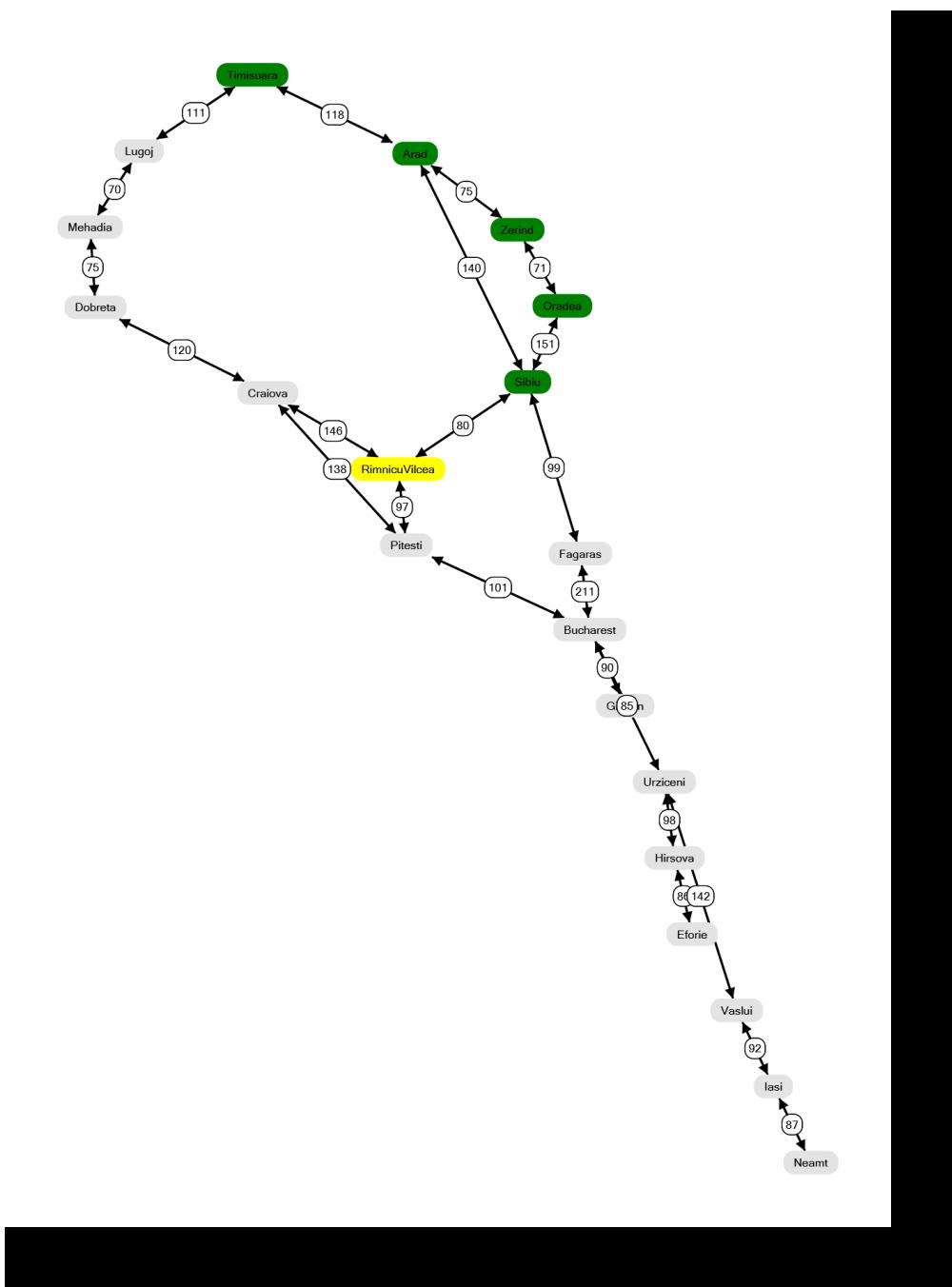
Visit

The algorithm has marked the node Sibiu as visited



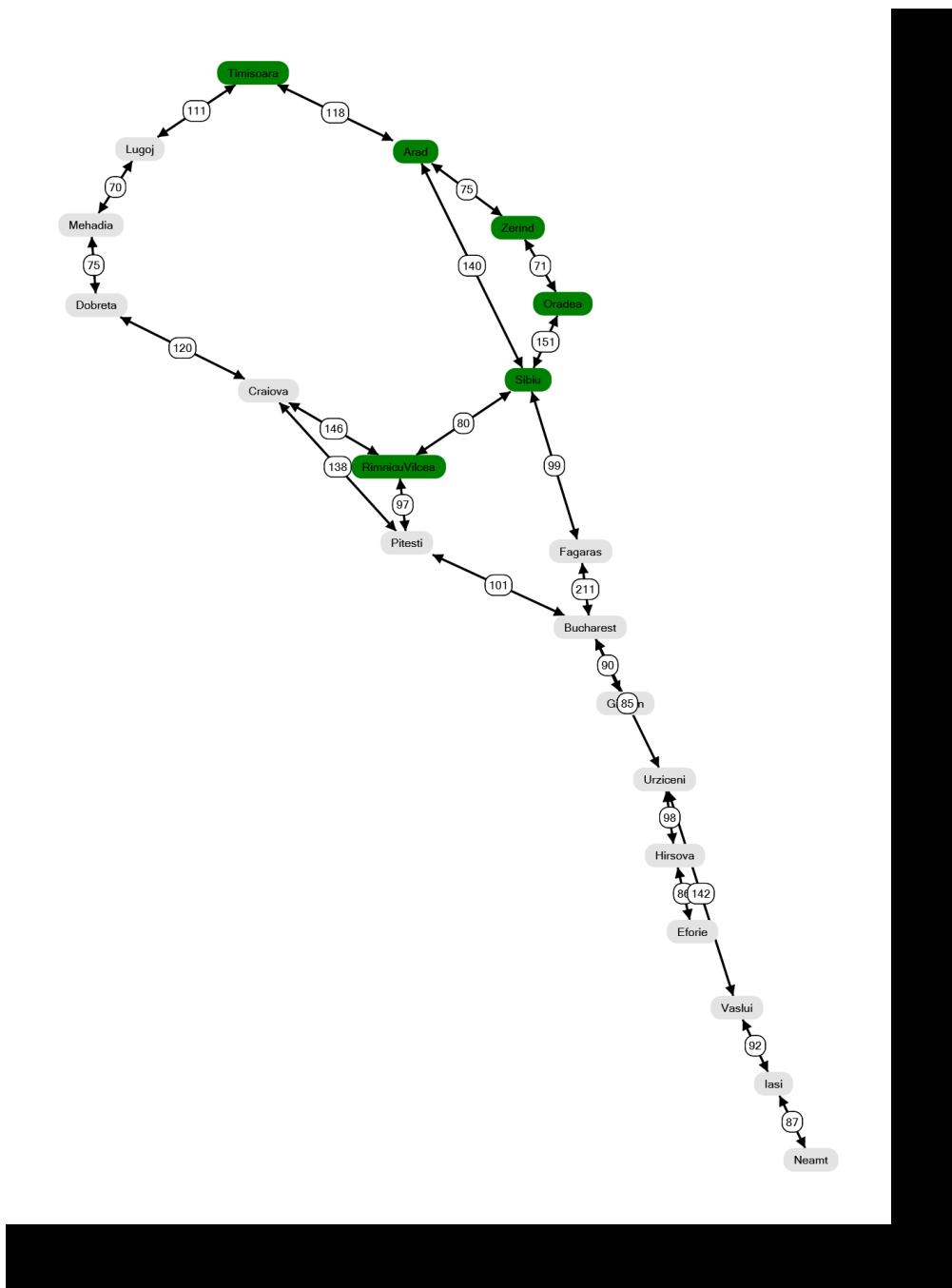
Pre-Visit

The algorithm has opened the node Rimnicu Vilcea



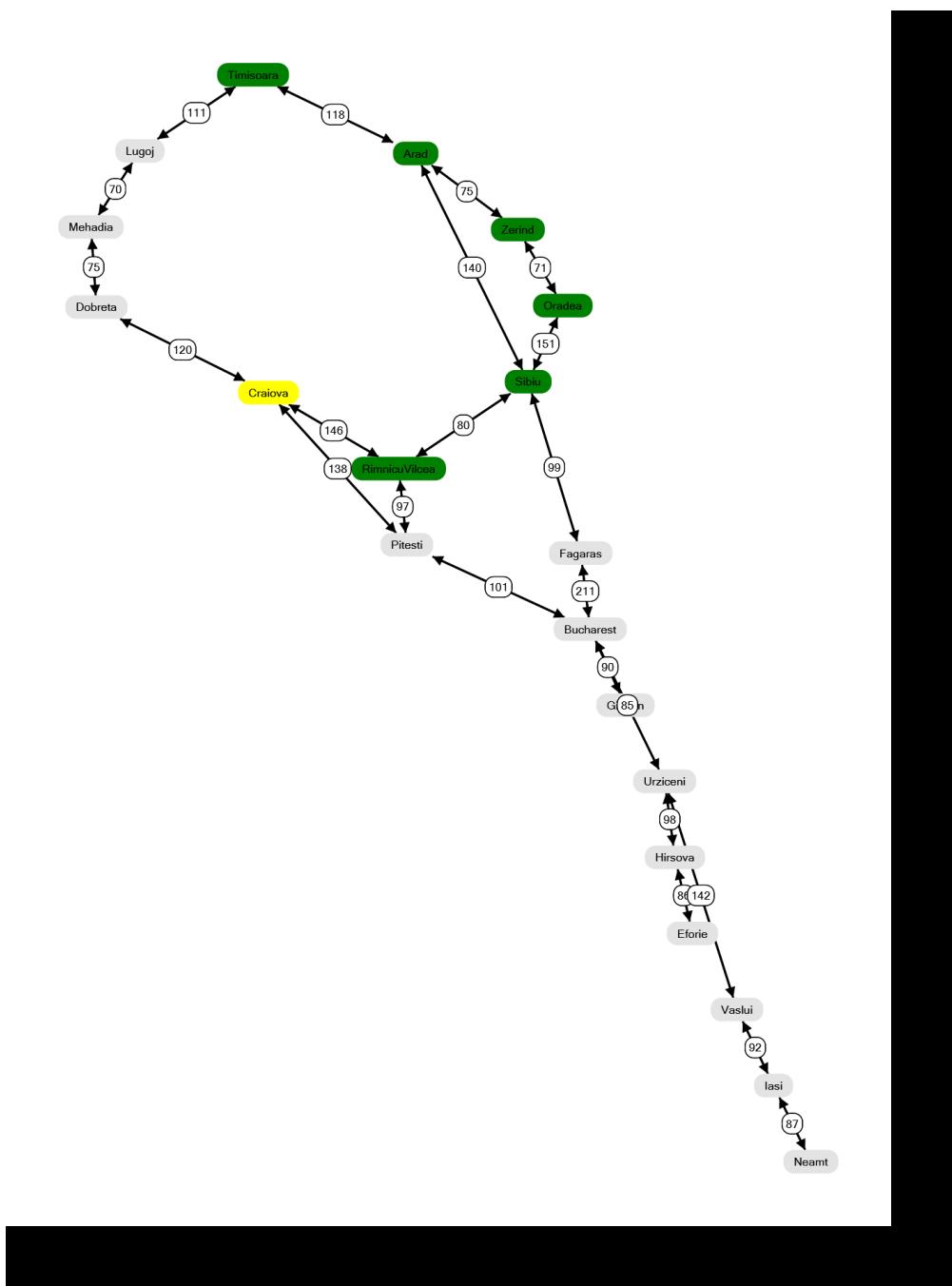
Visit

The algorithm has marked the node Rimnicu Vilcea as visited



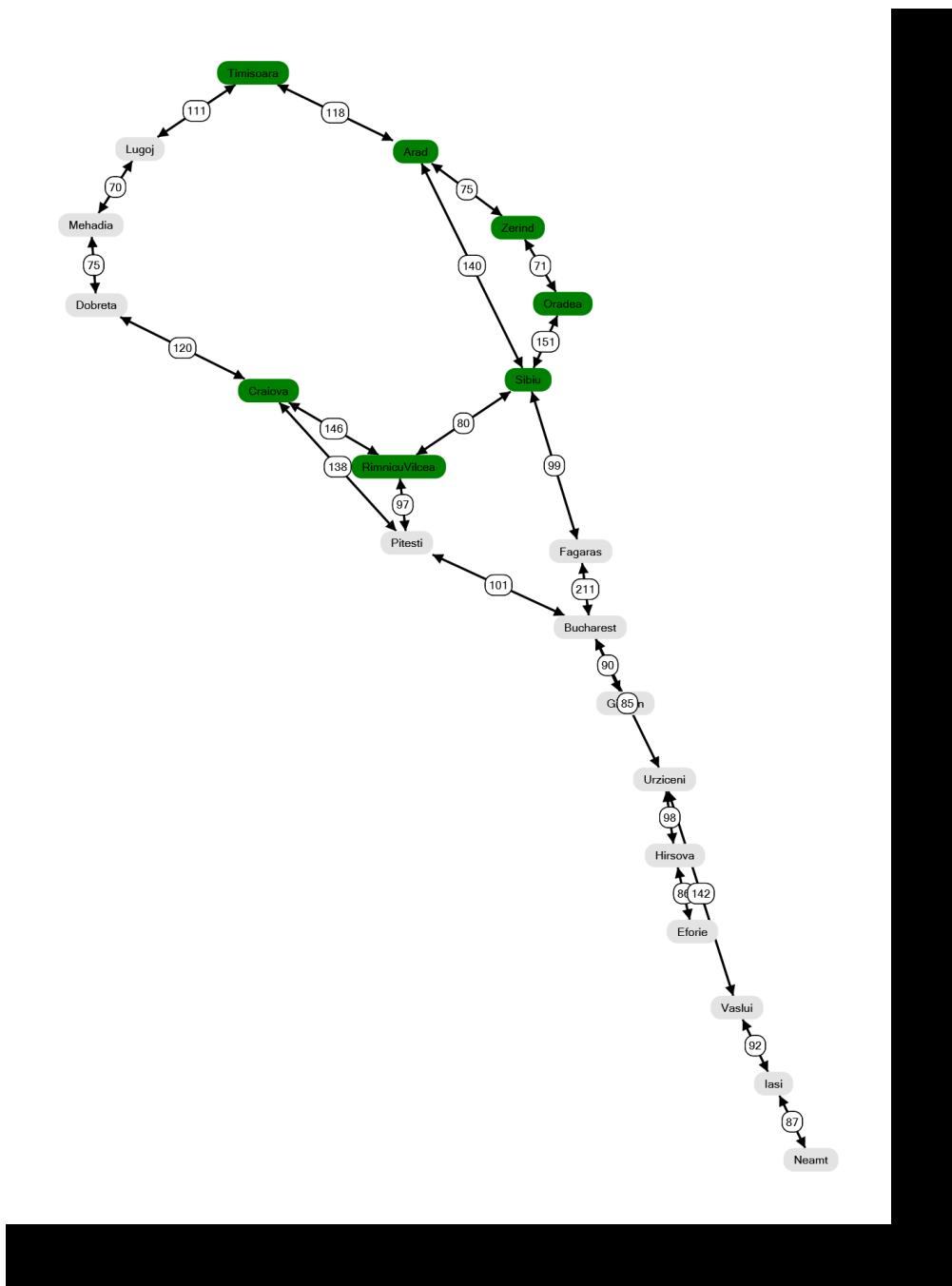
Pre-Visit

The algorithm has opened the node Craiova



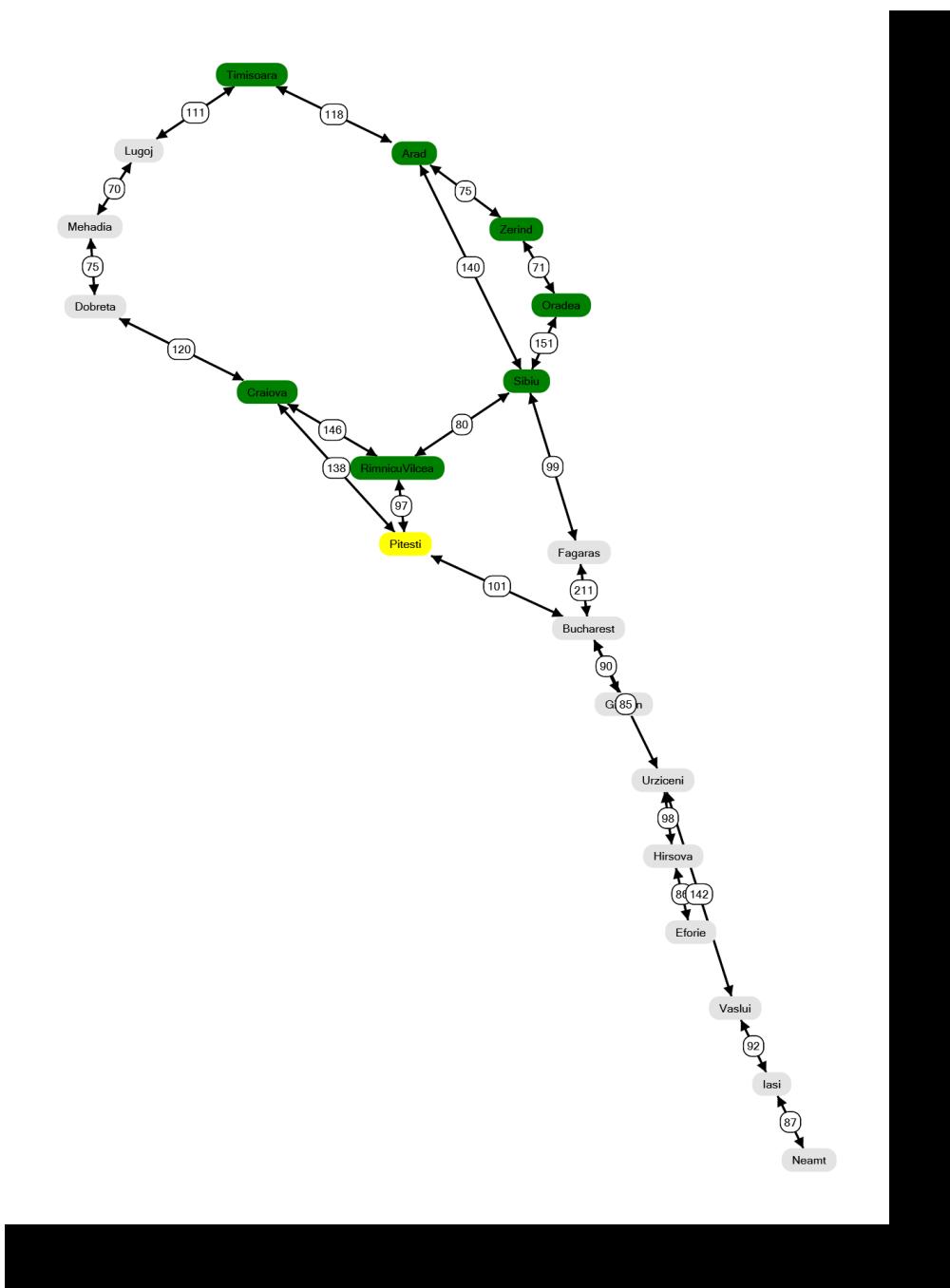
Visit

The algorithm has marked the node Craiova as visited



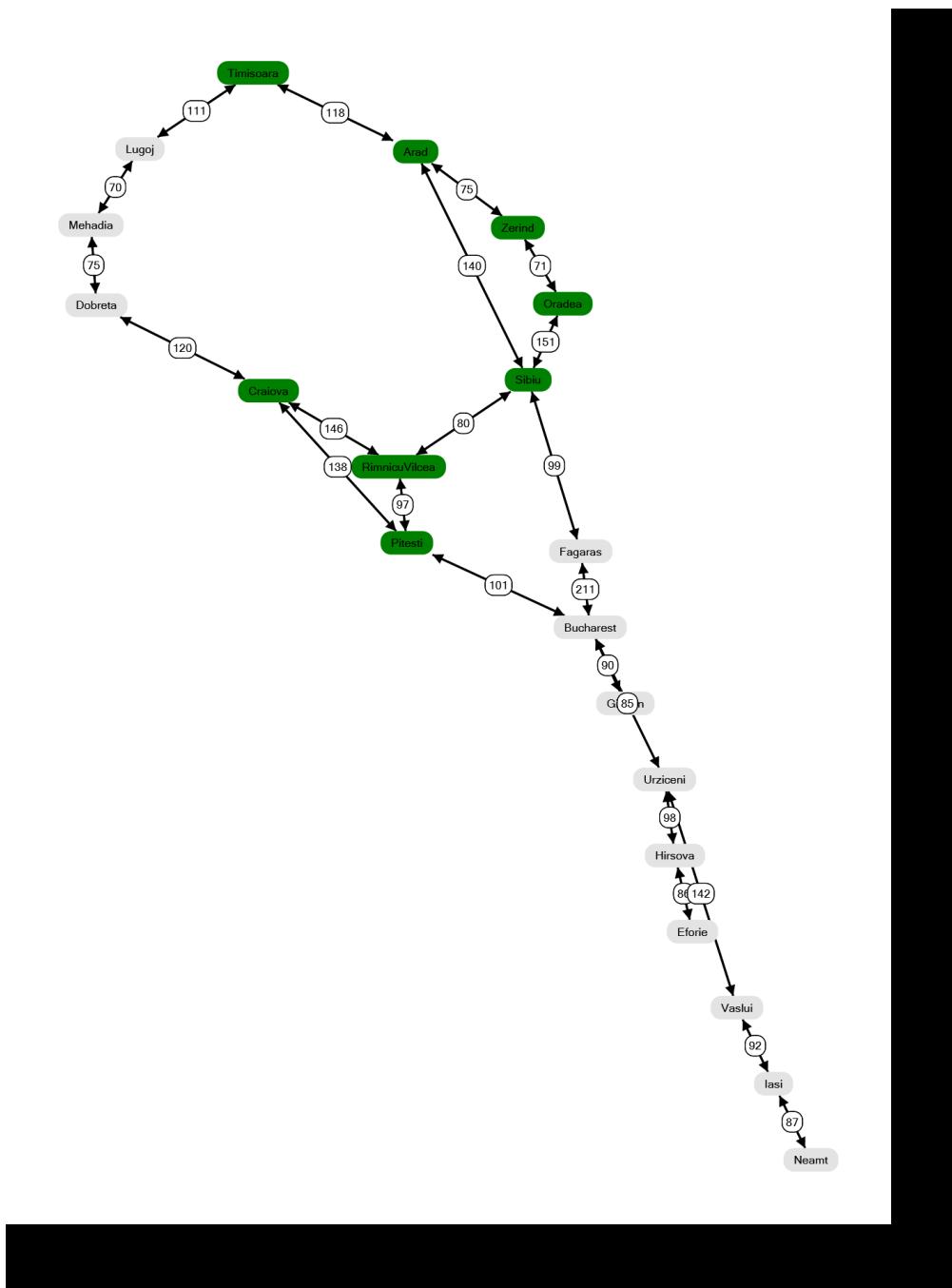
Pre-Visit

The algorithm has opened the node Pitesti



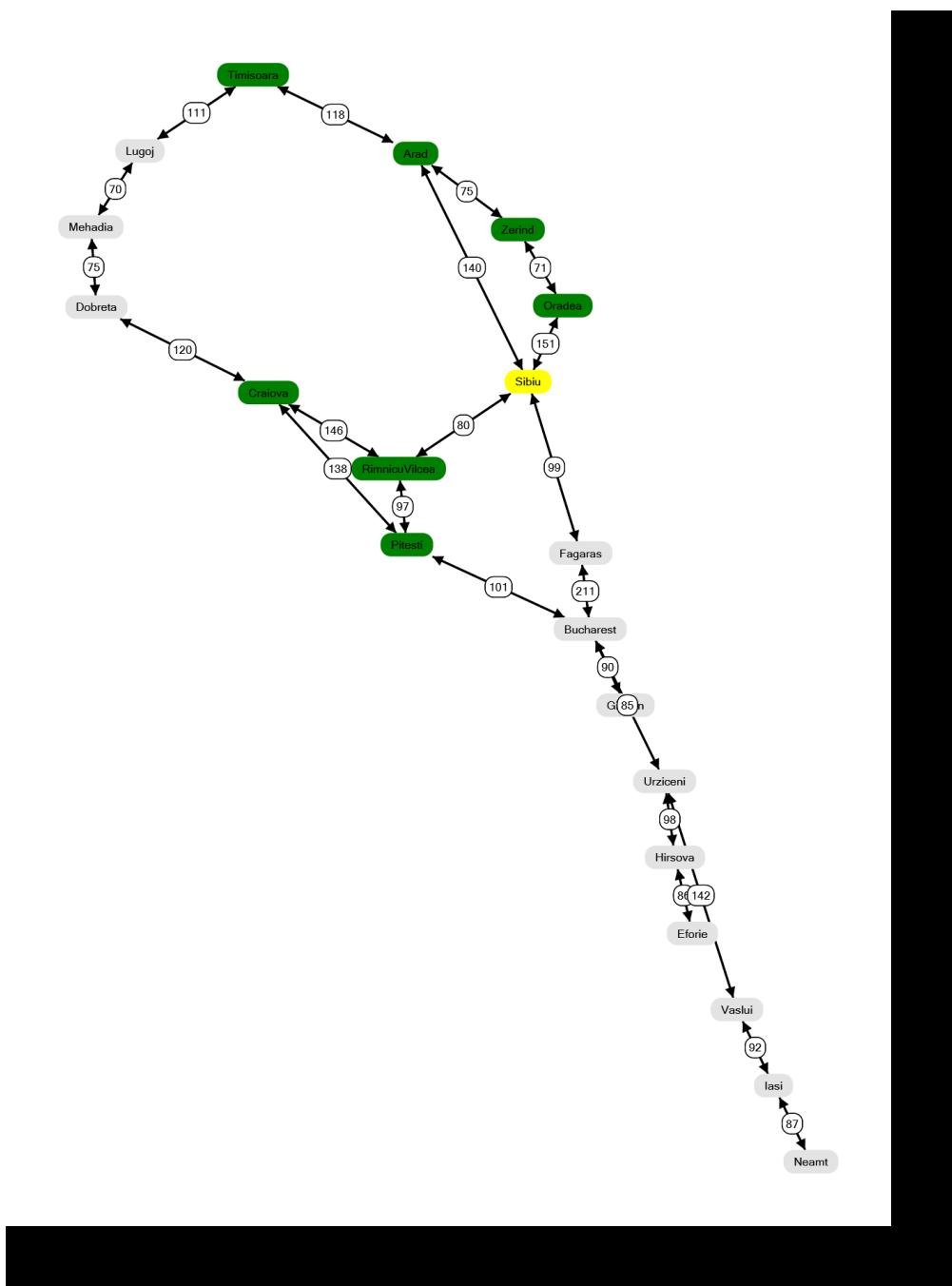
Visit

The algorithm has marked the node Pitesti as visited



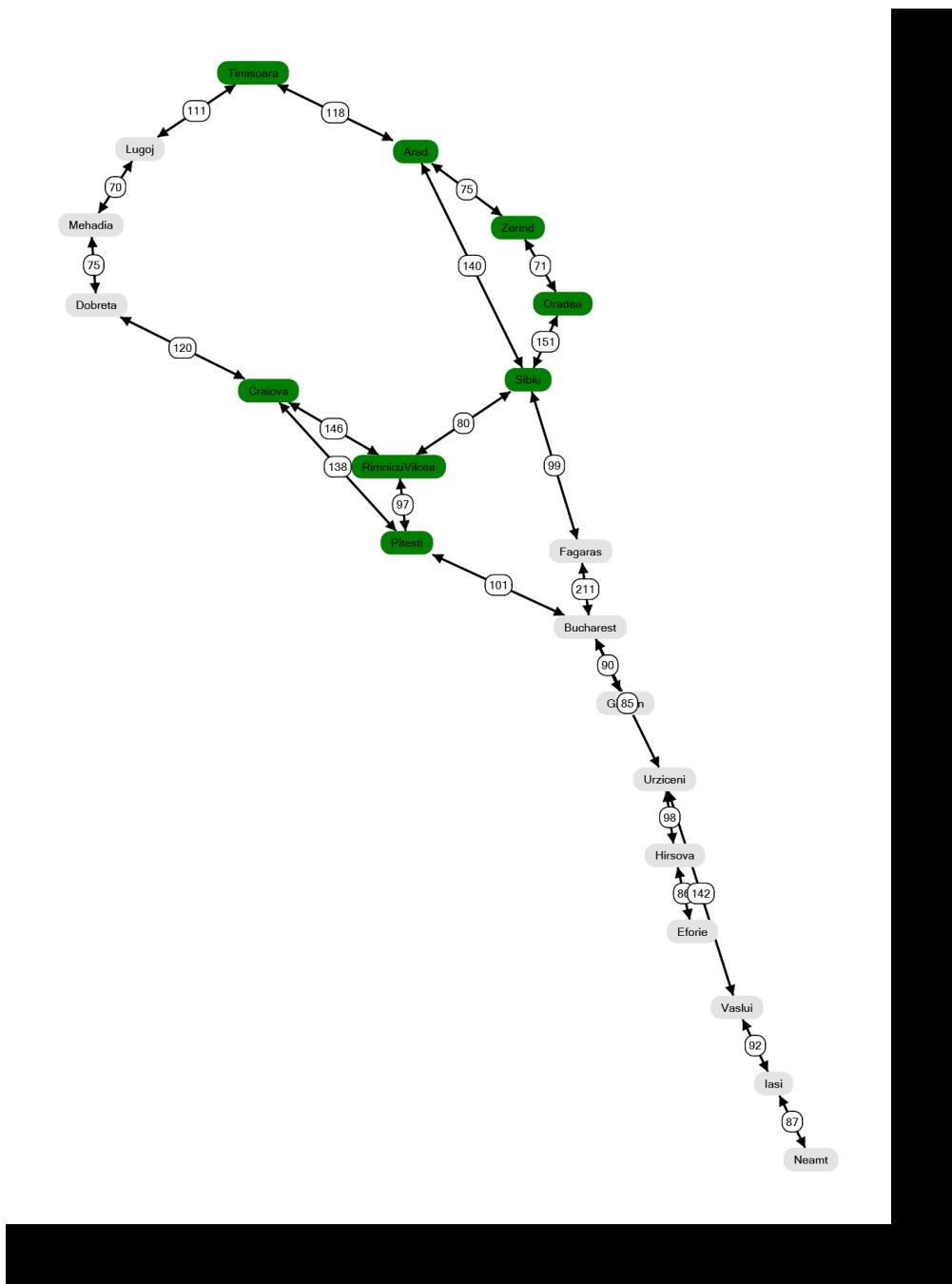
Pre-Visit

The algorithm has opened the node Sibiu



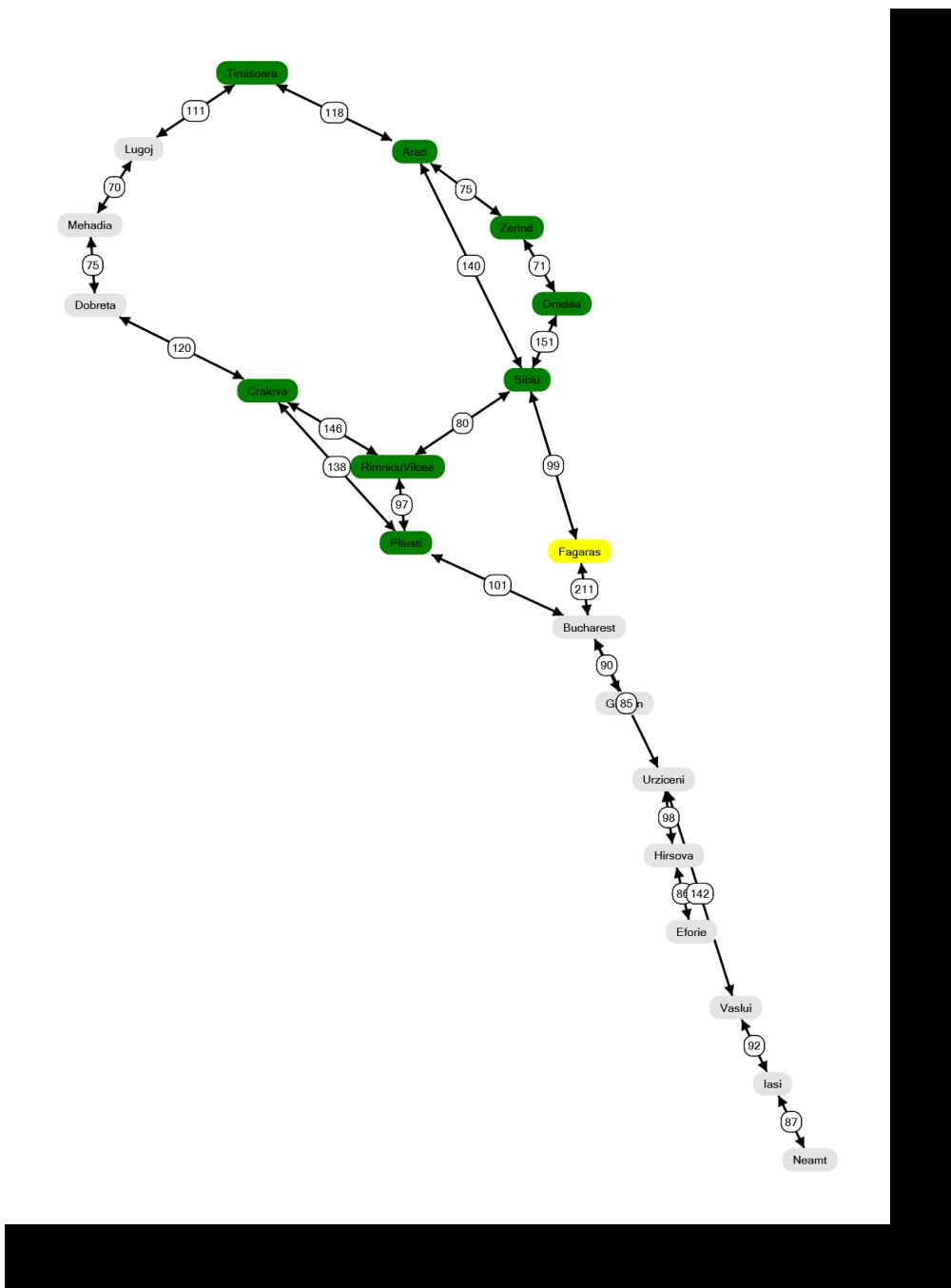
Visit

The algorithm has marked the node Sibiu as visited



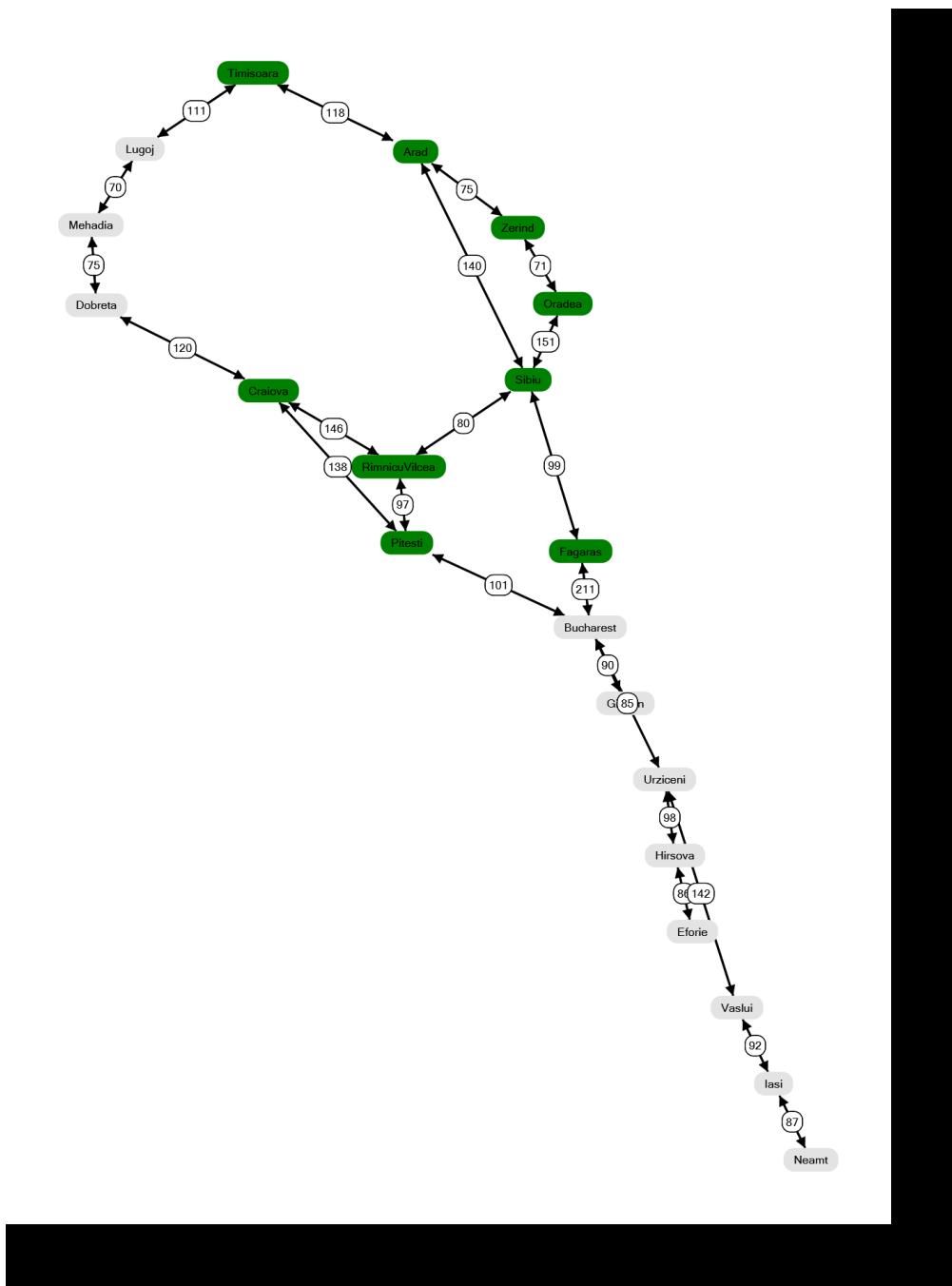
Pre-Visit

The algorithm has opened the node Fagaras



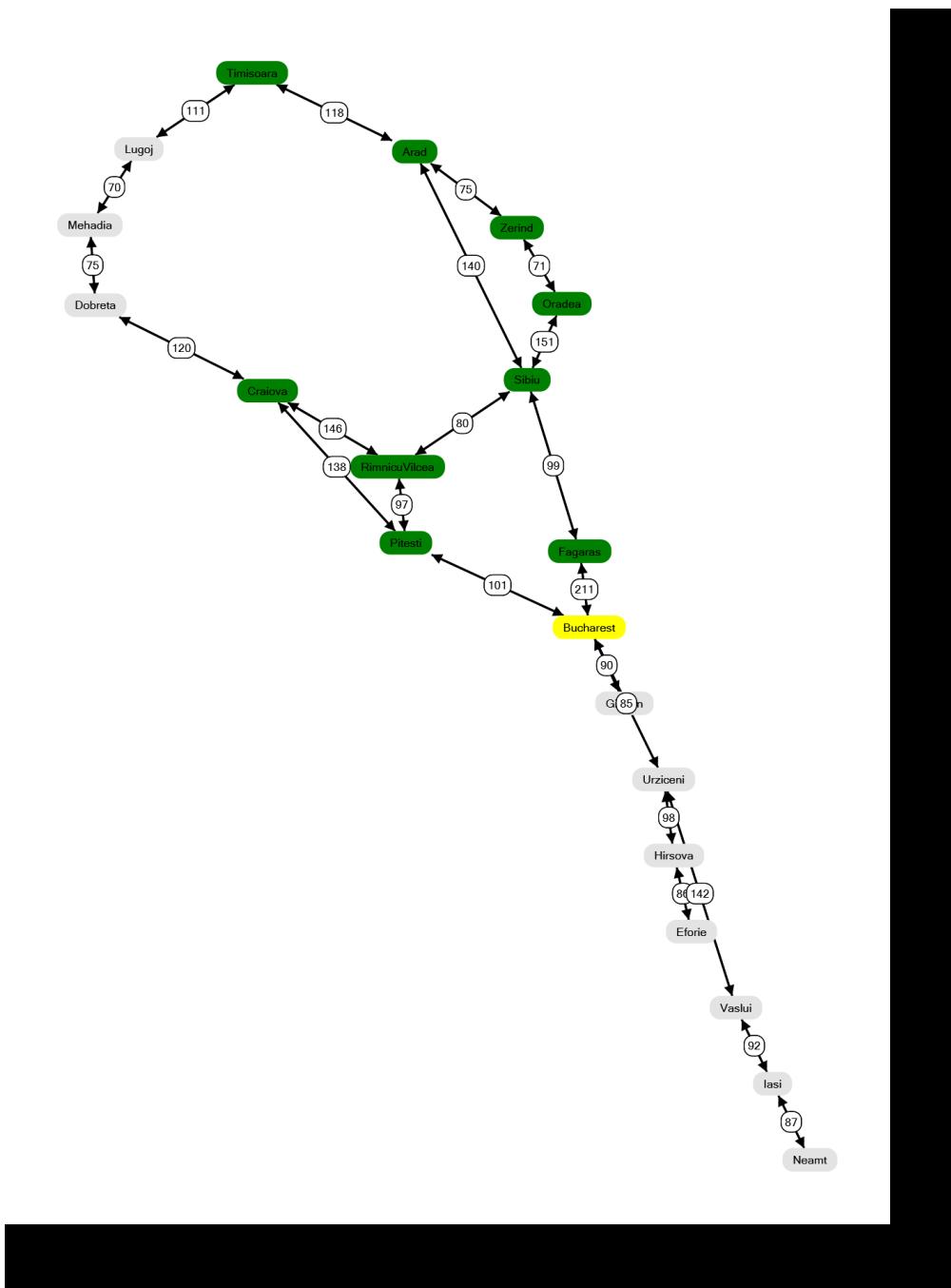
Visit

The algorithm has marked the node Fagaras as visited



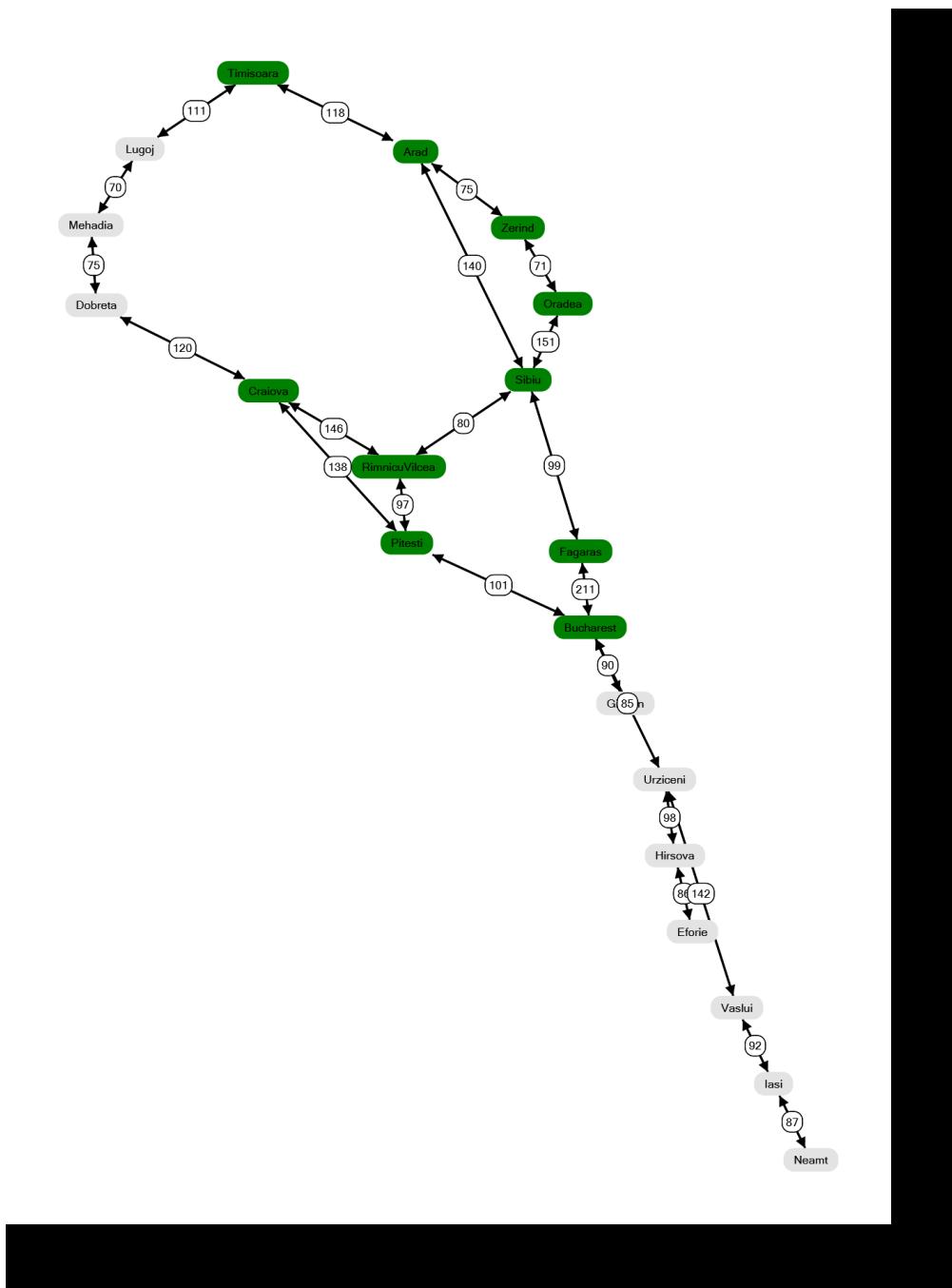
Pre-Visit

The algorithm has opened the node Bucharest



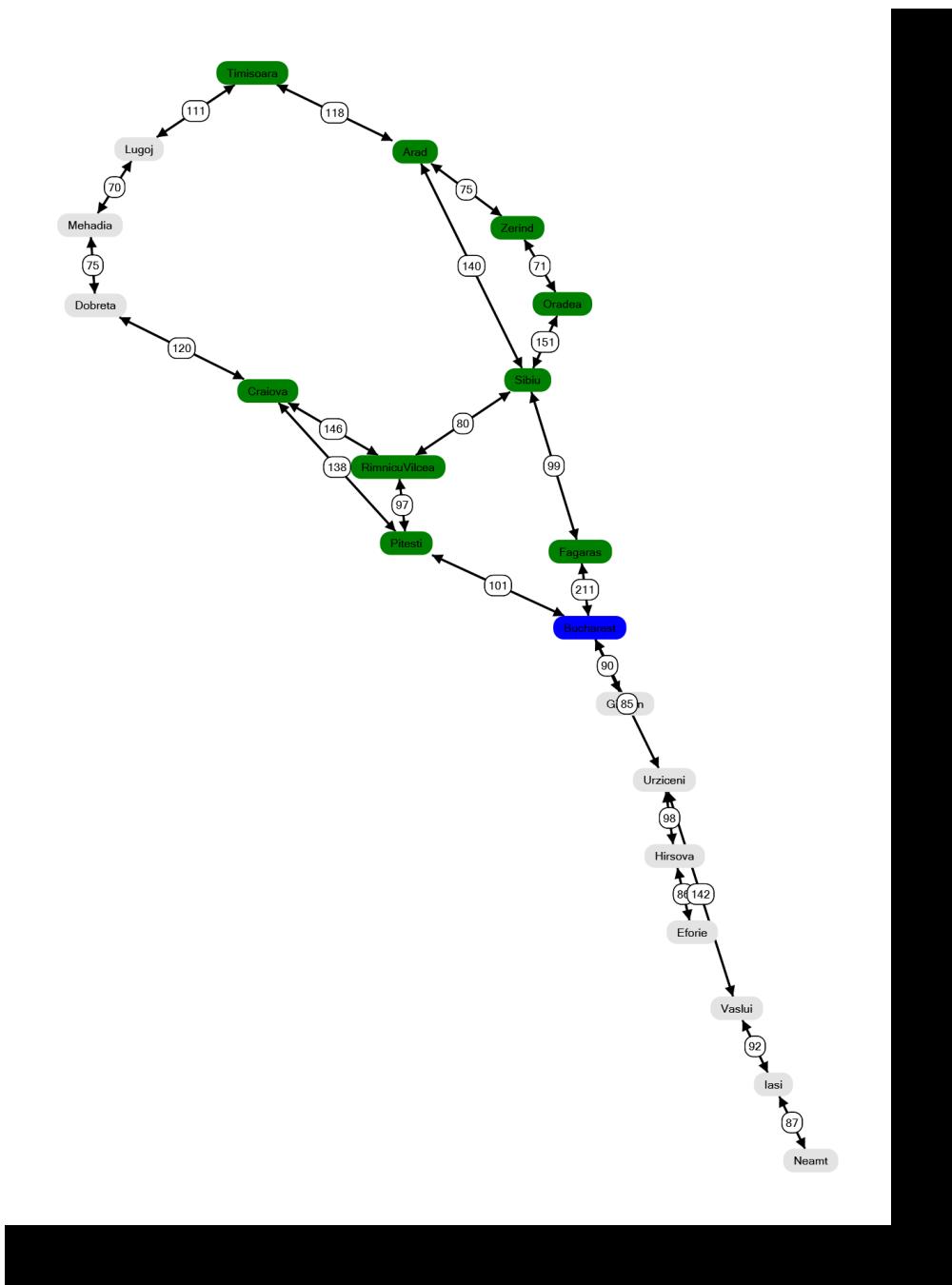
Visit

The algorithm has marked the node Bucharest as visited



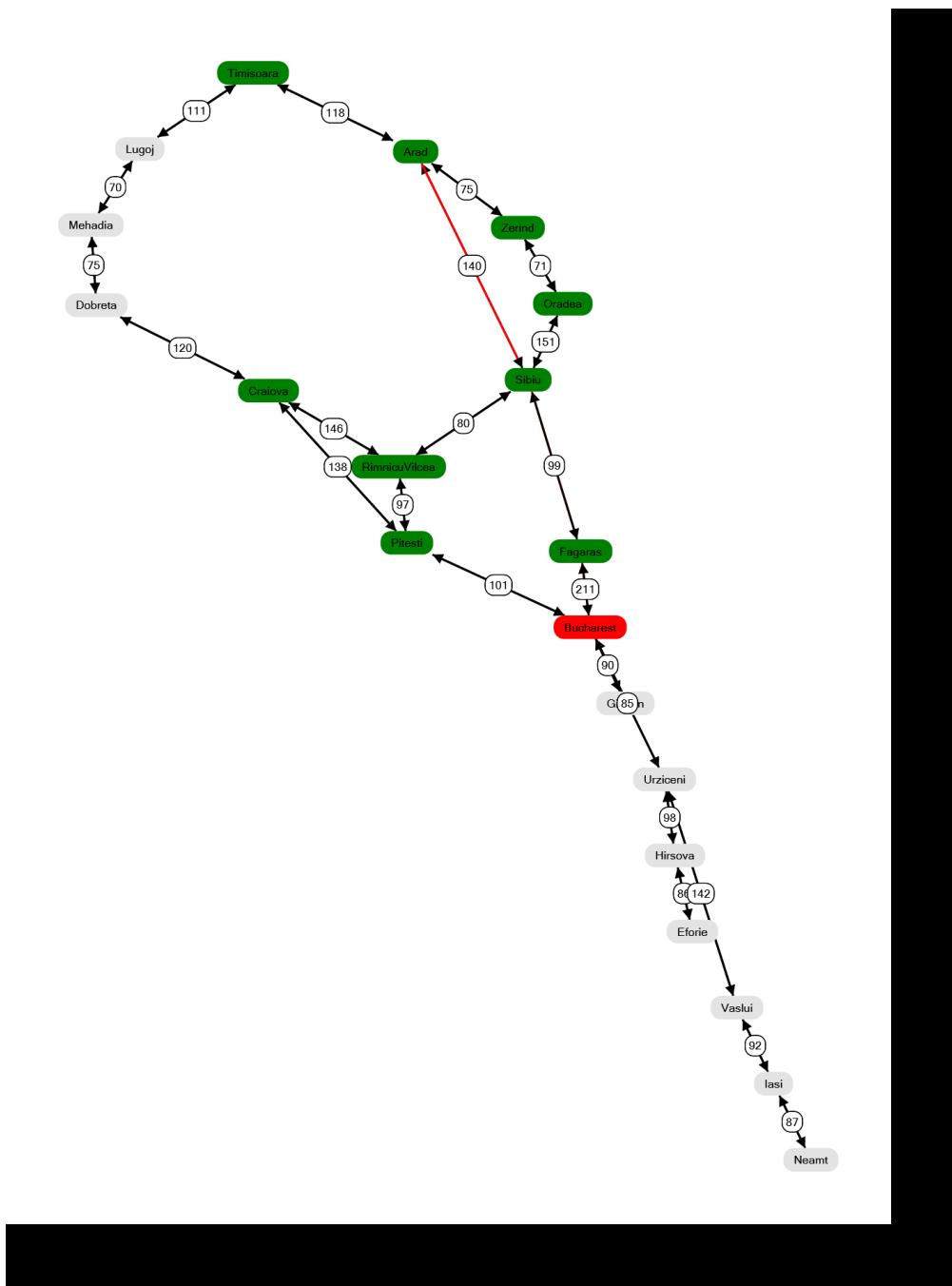
Post-Visit

The algorithm has closed the node Bucharest



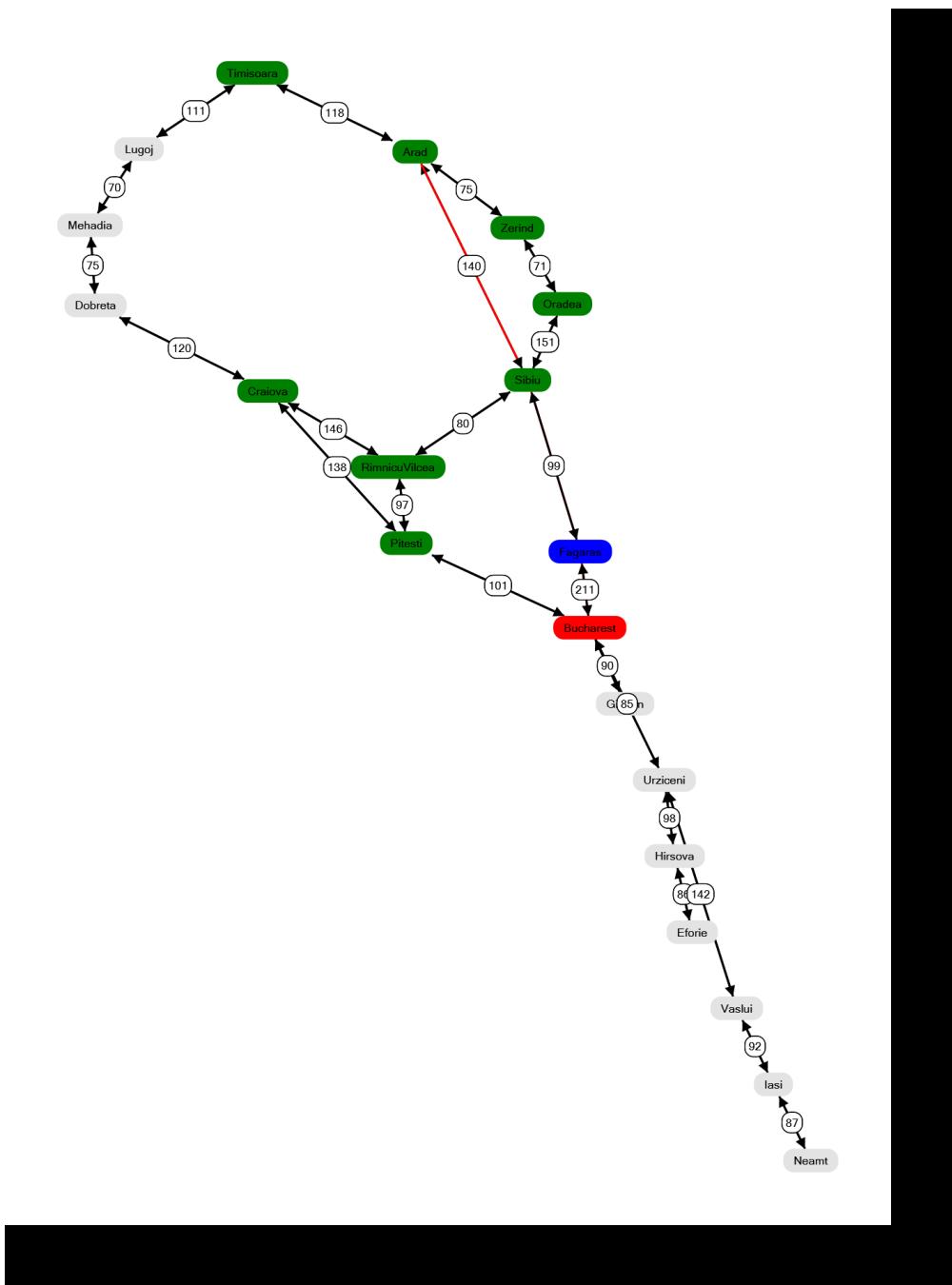
Result Found

The algorithm has found the target node Bucharest



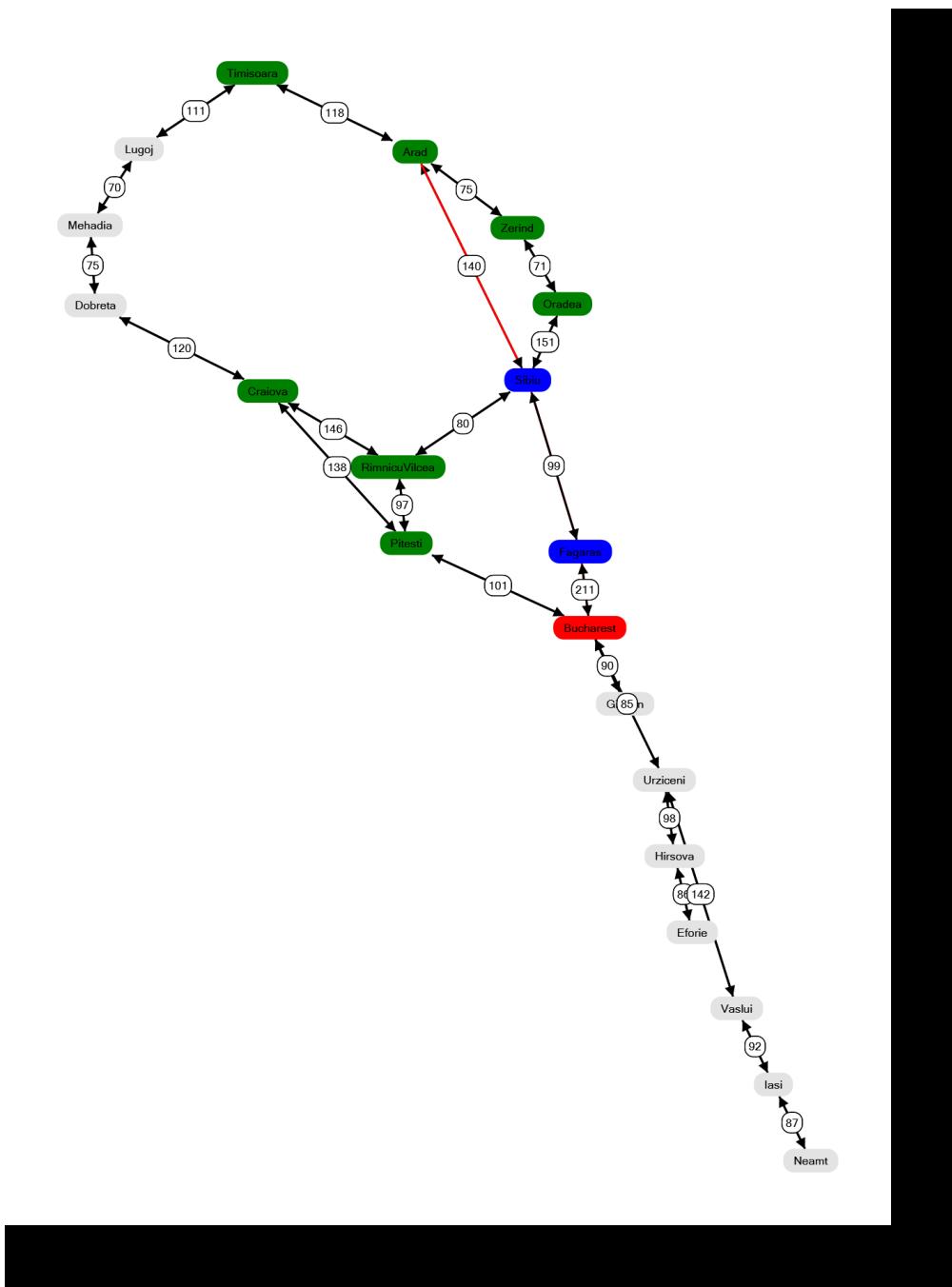
Post-Visit

The algorithm has closed the node Fagaras



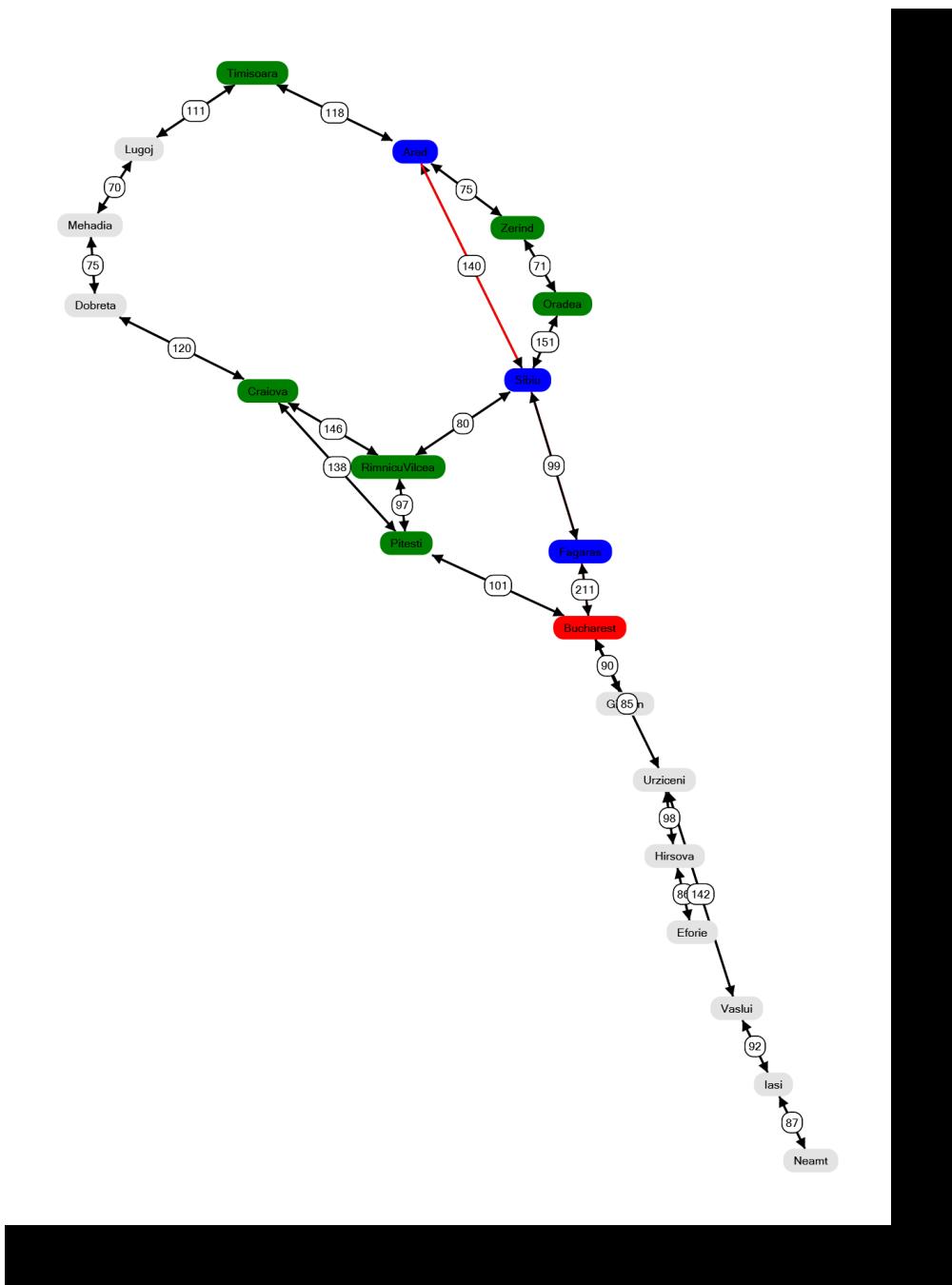
Post-Visit

The algorithm has closed the node Sibiu



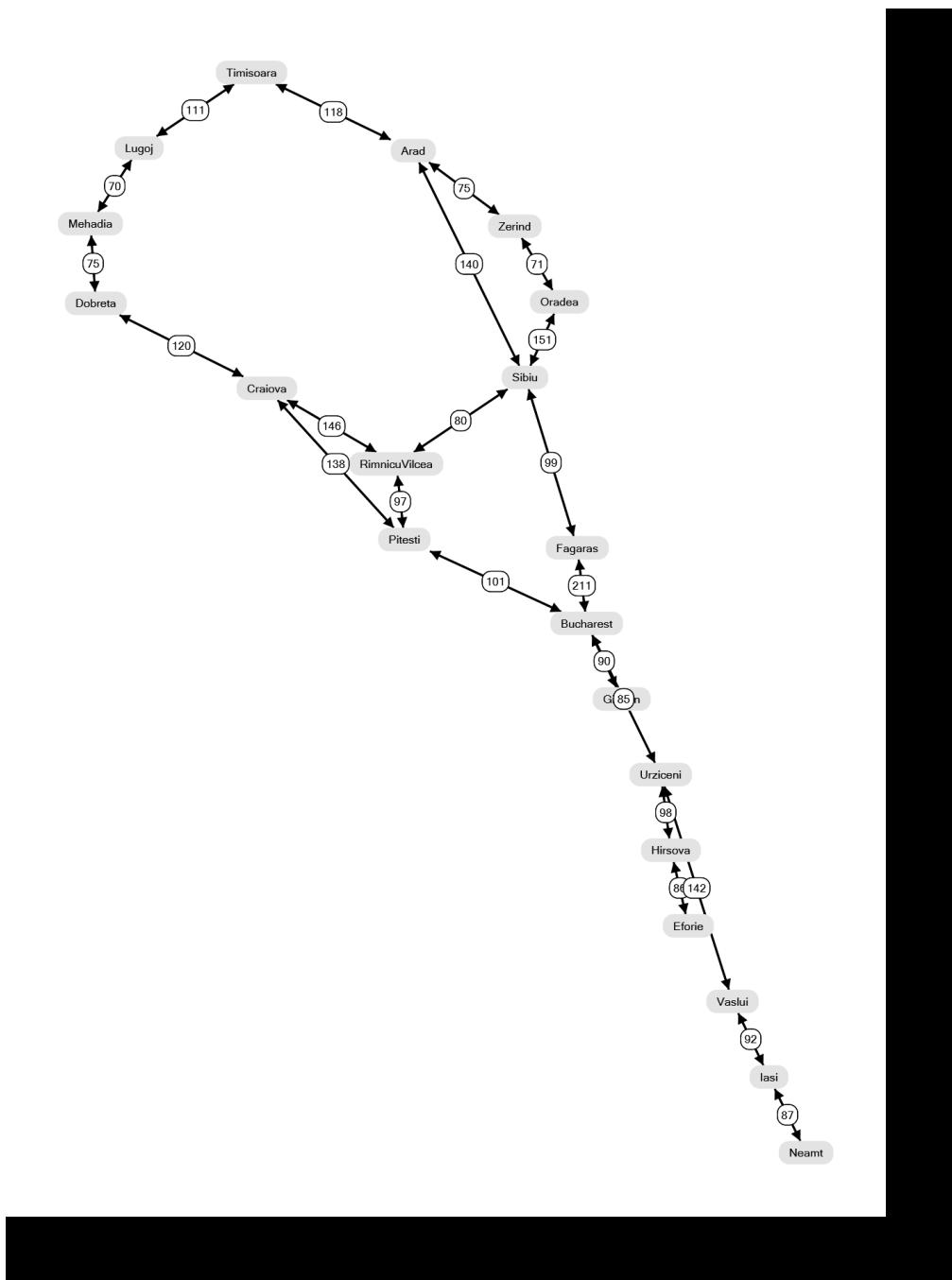
Post-Visit

The algorithm has closed the node Arad



Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



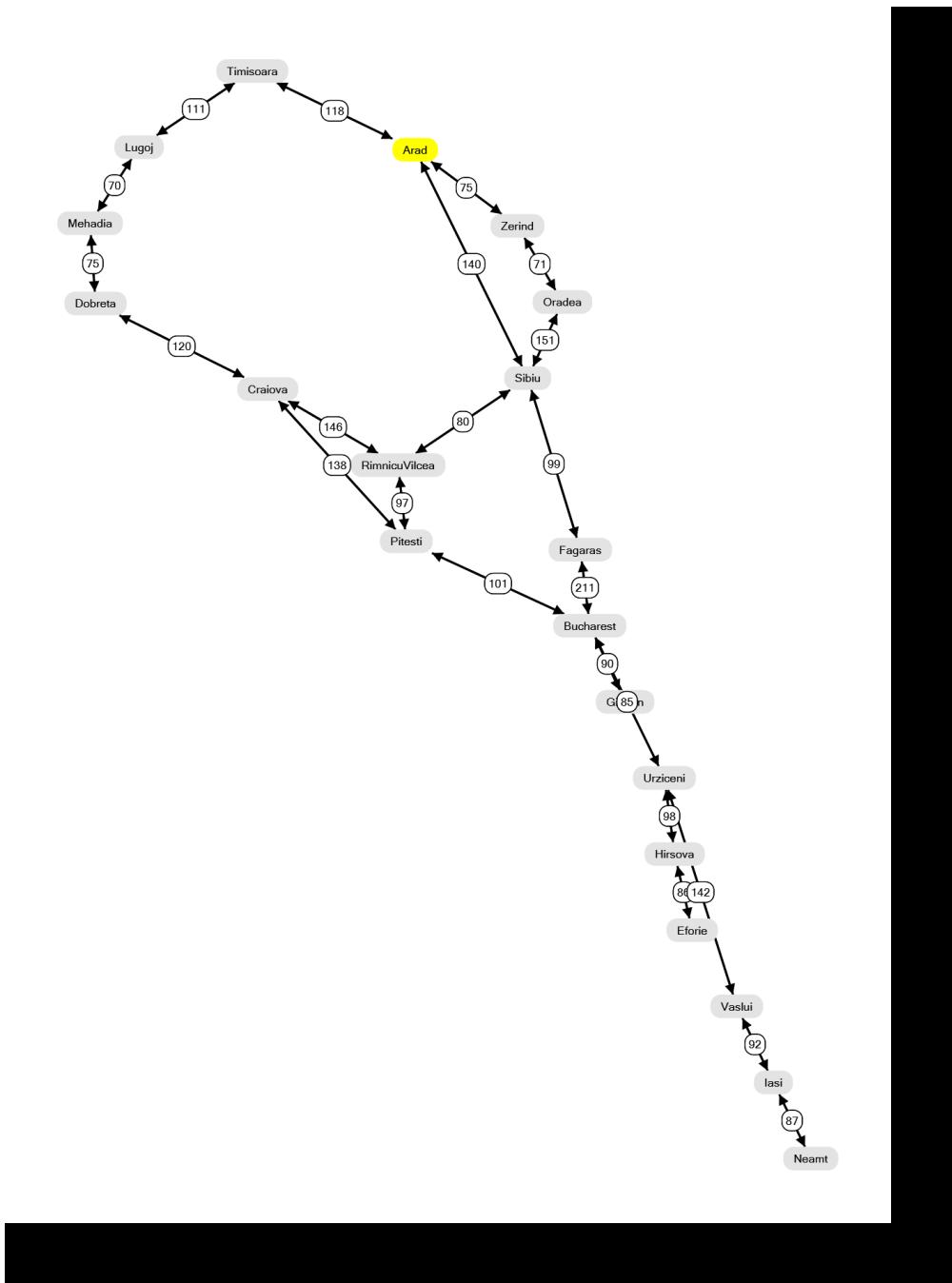
A*

In computer science, A* (pronounced as "A star") is a computer algorithm that is widely used in pathfinding and graph traversal, the process of plotting an efficiently directed path between multiple points, called nodes. It enjoys widespread use due to its performance and accuracy. However, in practical travel-routing systems, it is generally outperformed by algorithms which can pre-process the graph to attain better performance, although other work has found A* to be superior to other approaches.

The algorithm took 00:00:00.0026923 to find the goal.

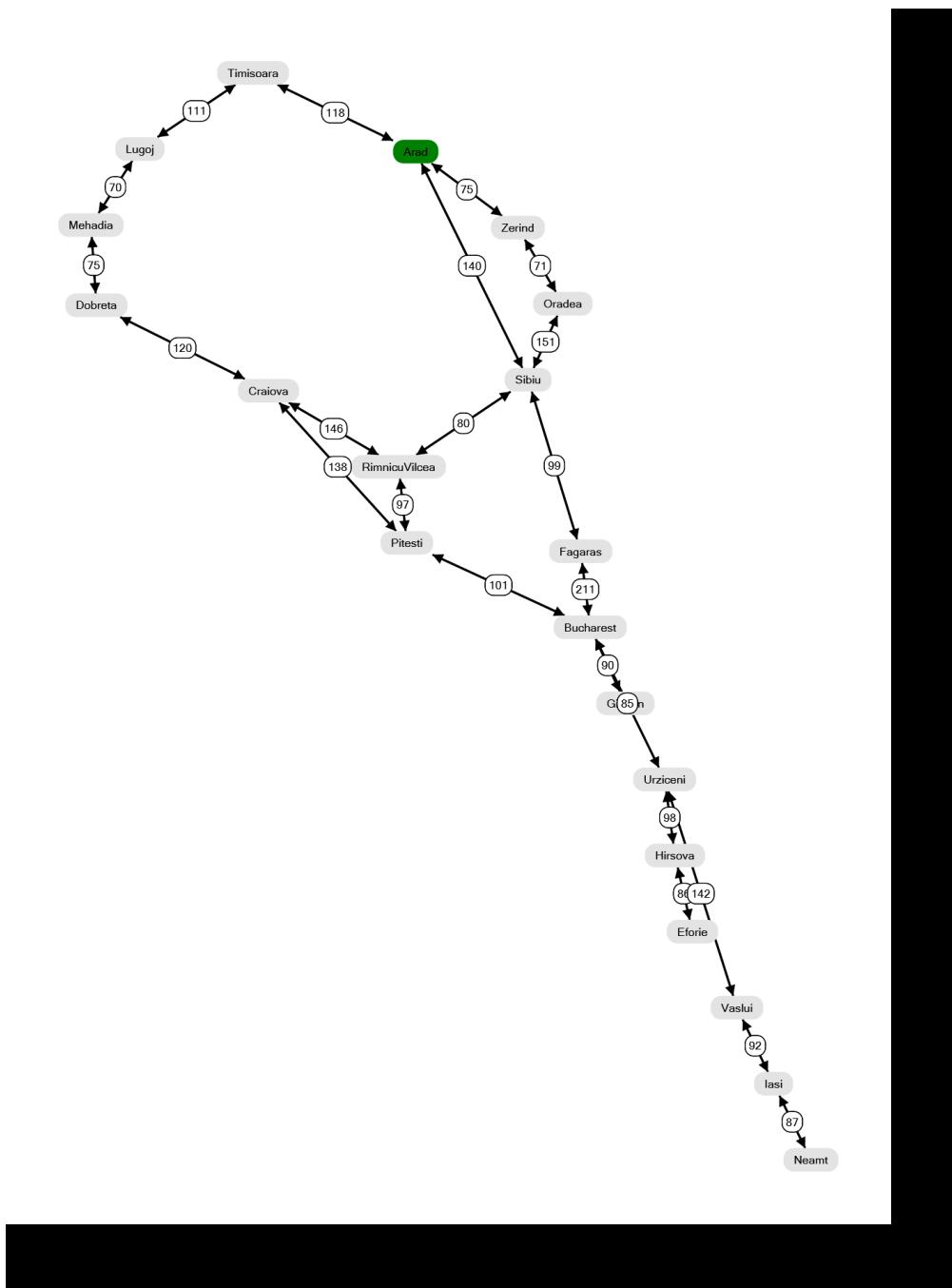
Pre-Visit

The algorithm has opened the node Arad



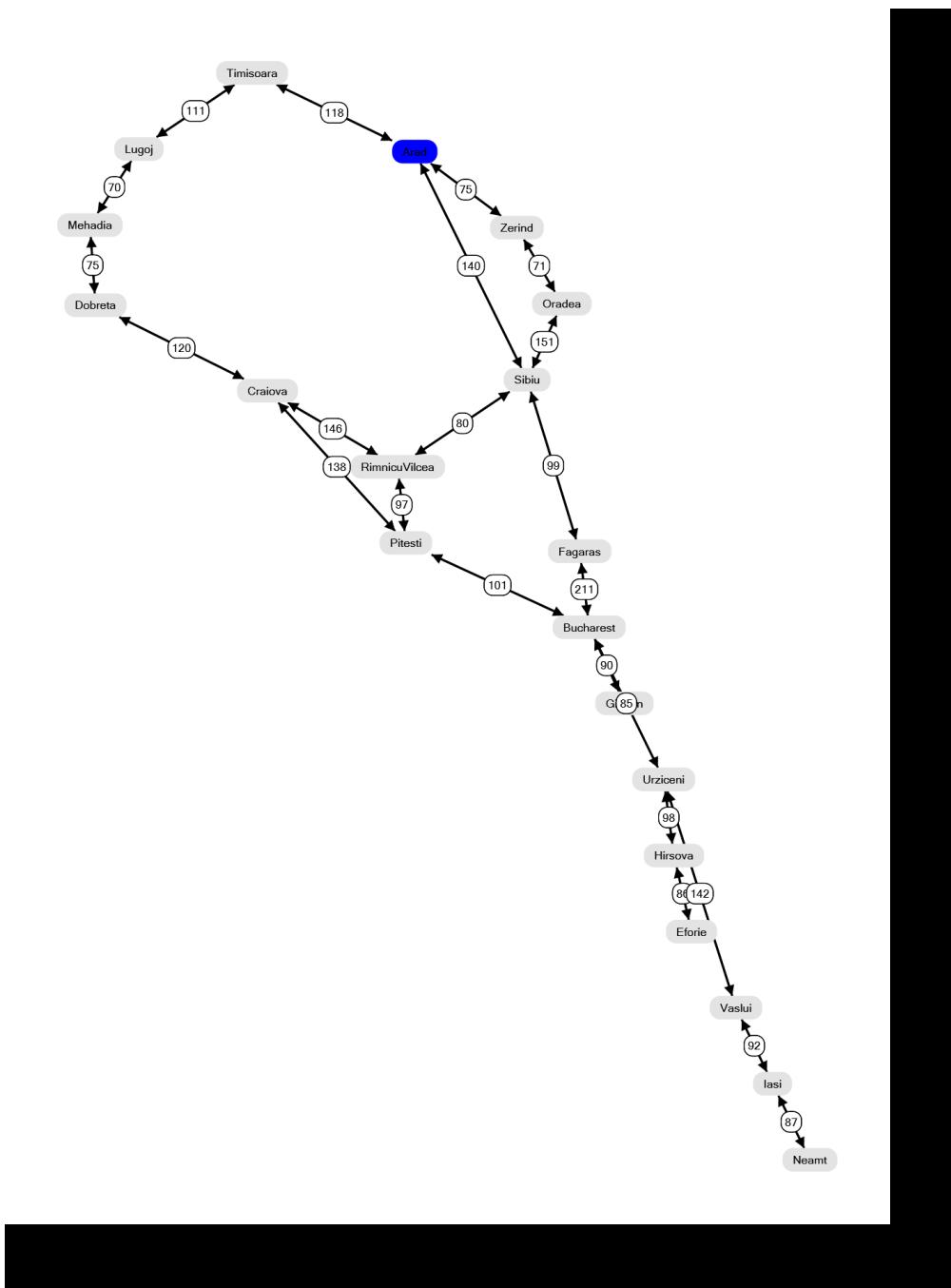
Visit

The algorithm has marked the node Arad as visited



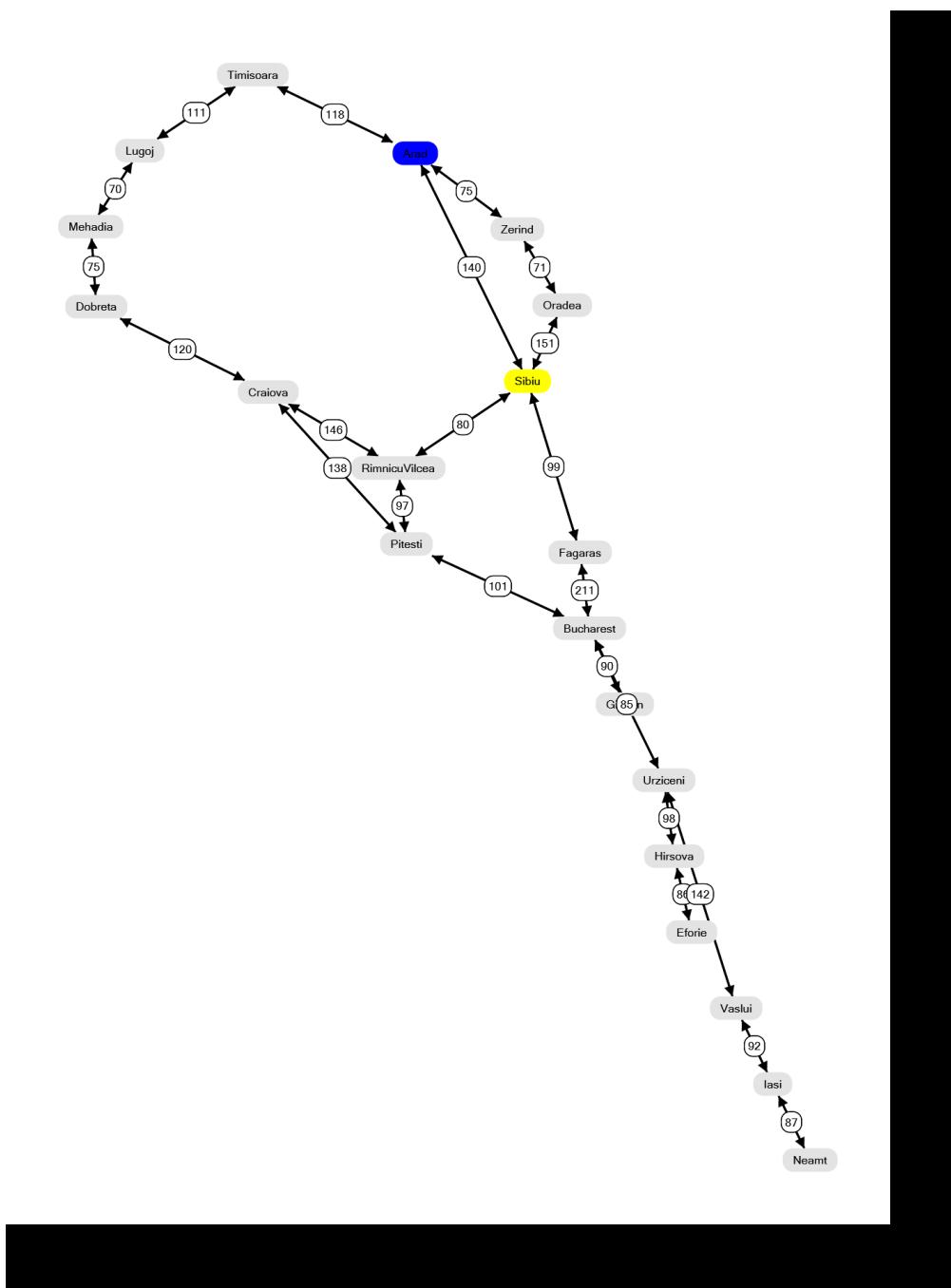
Post-Visit

The algorithm has closed the node Arad



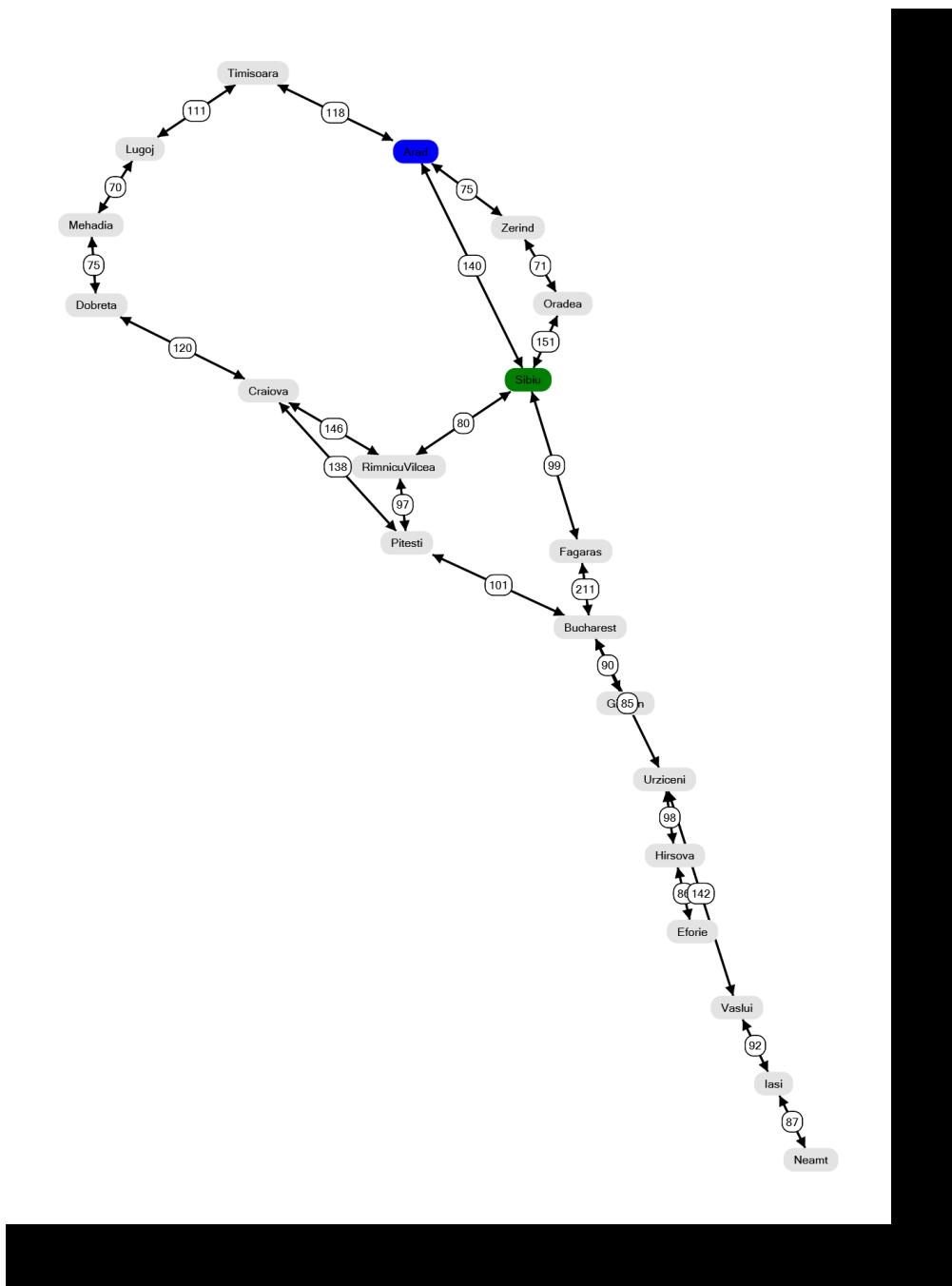
Pre-Visit

The algorithm has opened the node Sibiu



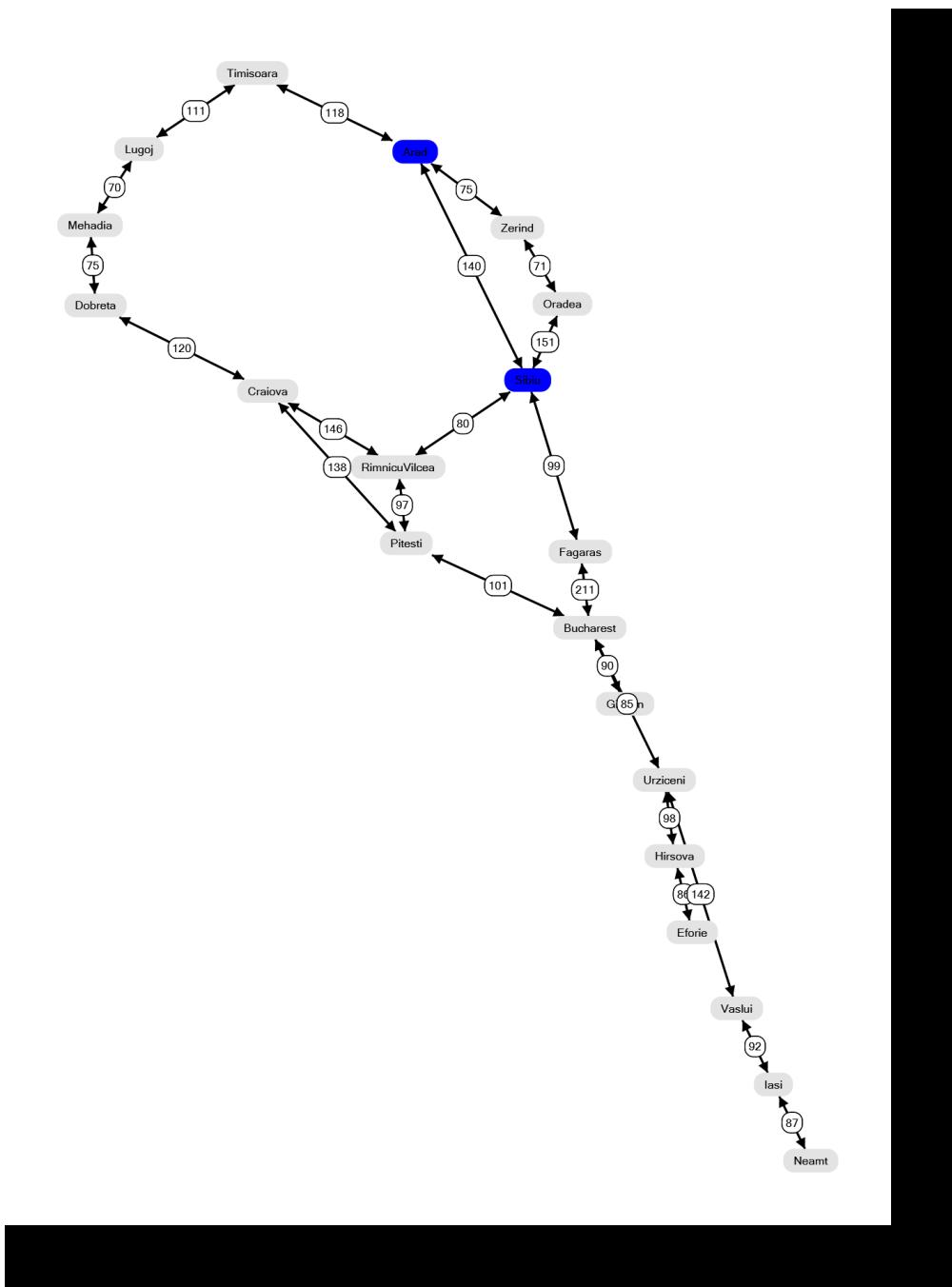
Visit

The algorithm has marked the node Sibiu as visited



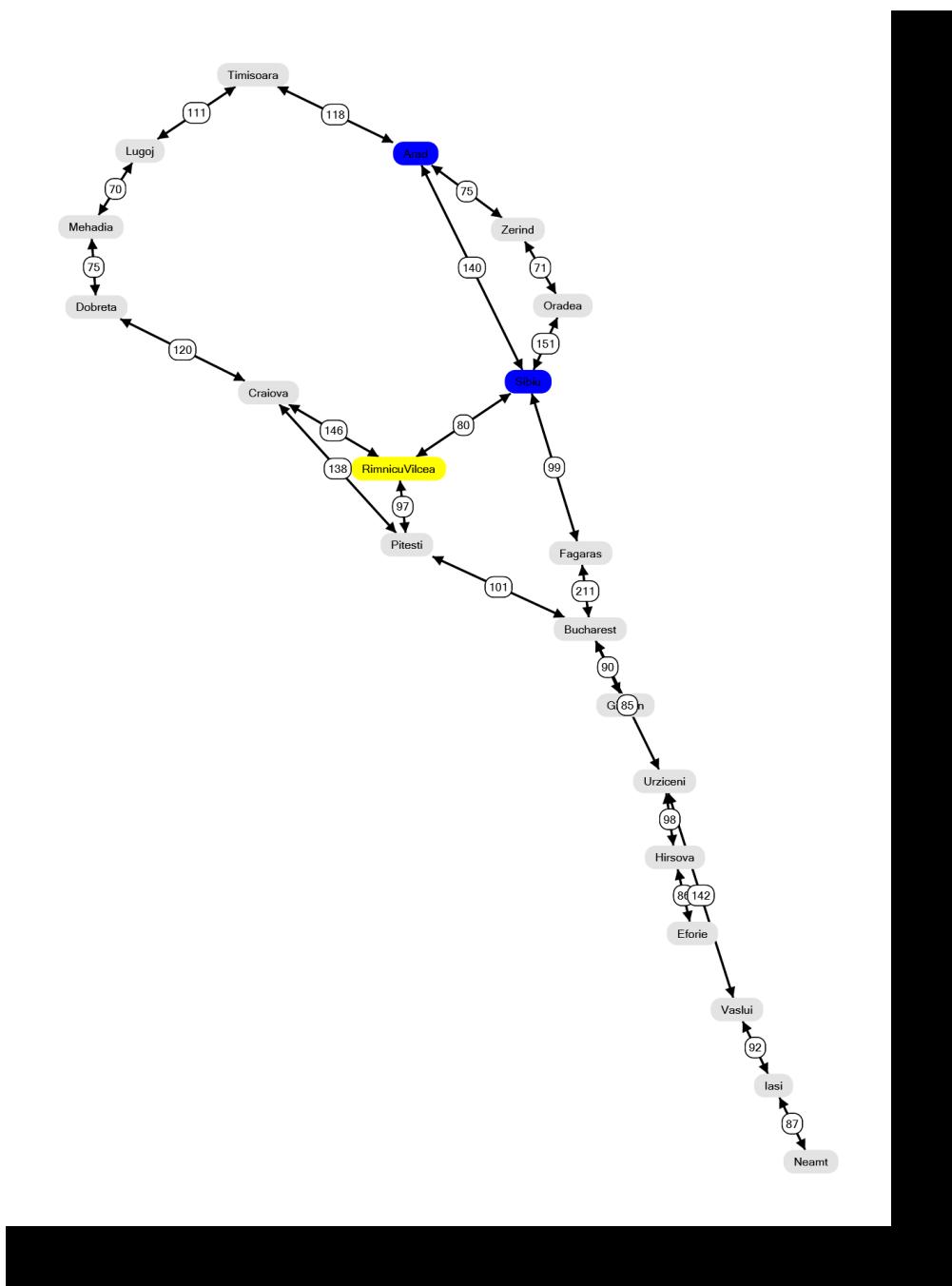
Post-Visit

The algorithm has closed the node Sibiu



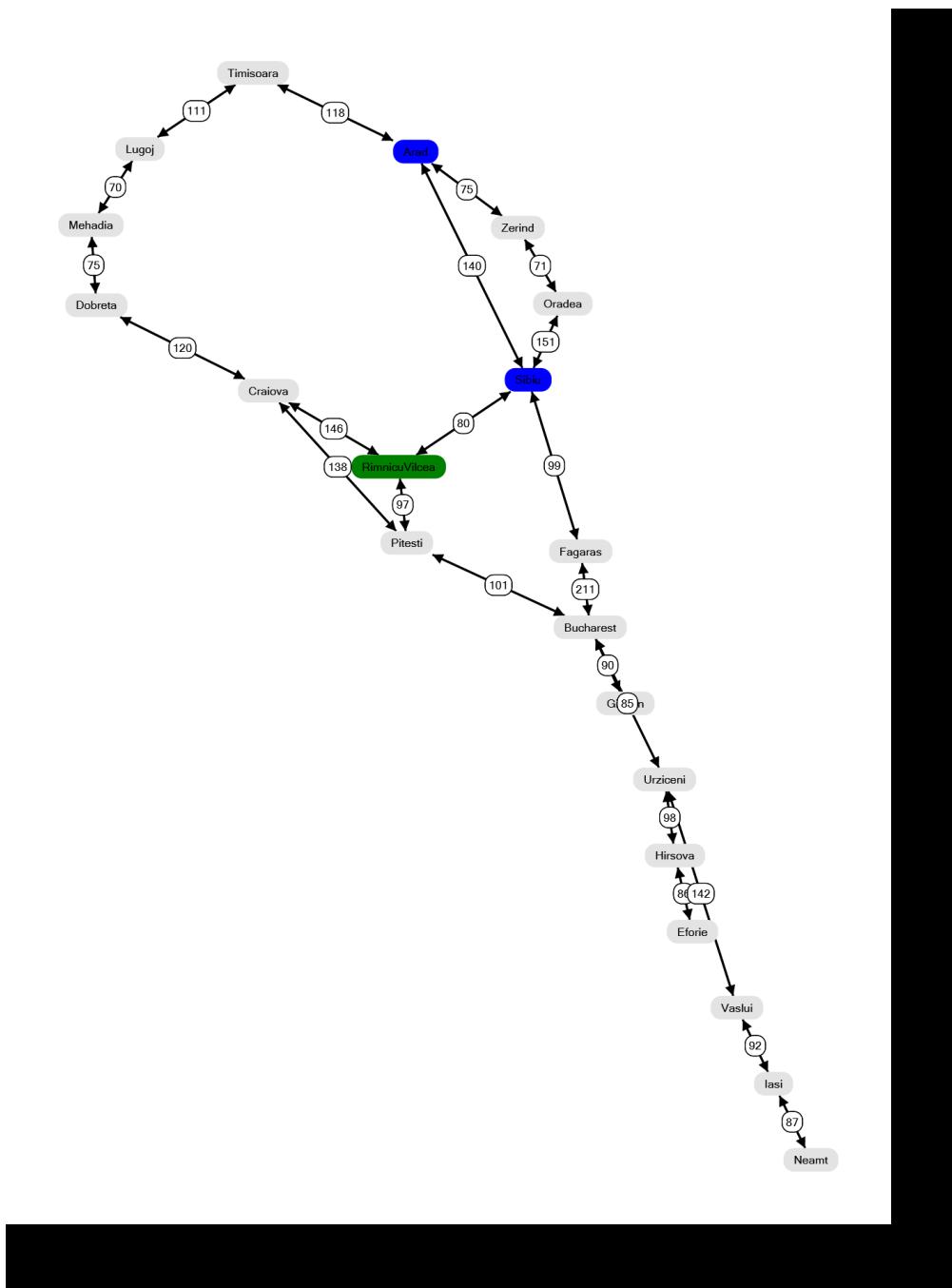
Pre-Visit

The algorithm has opened the node Rimnicu Vilcea



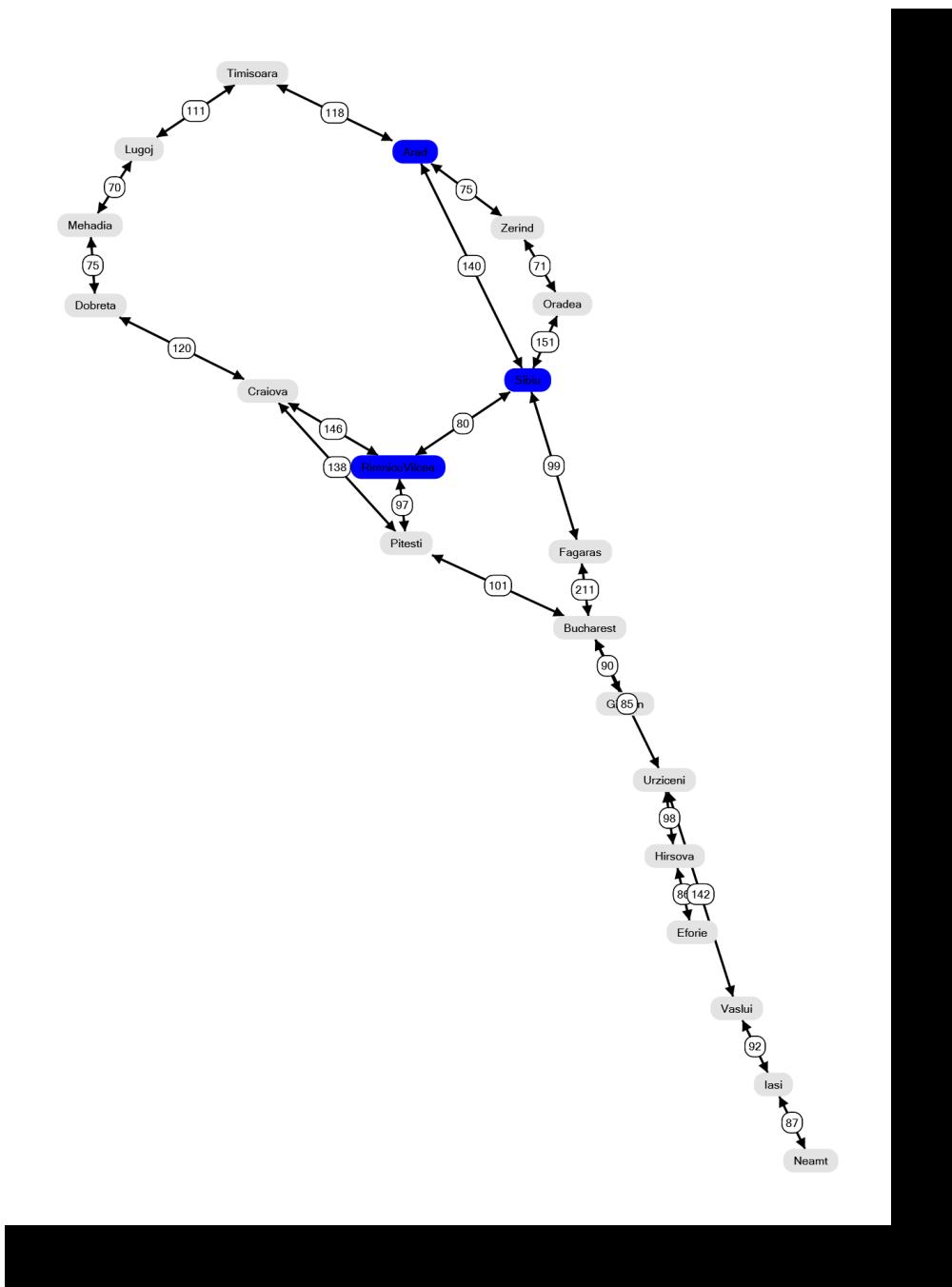
Visit

The algorithm has marked the node Rimnicu Vilcea as visited



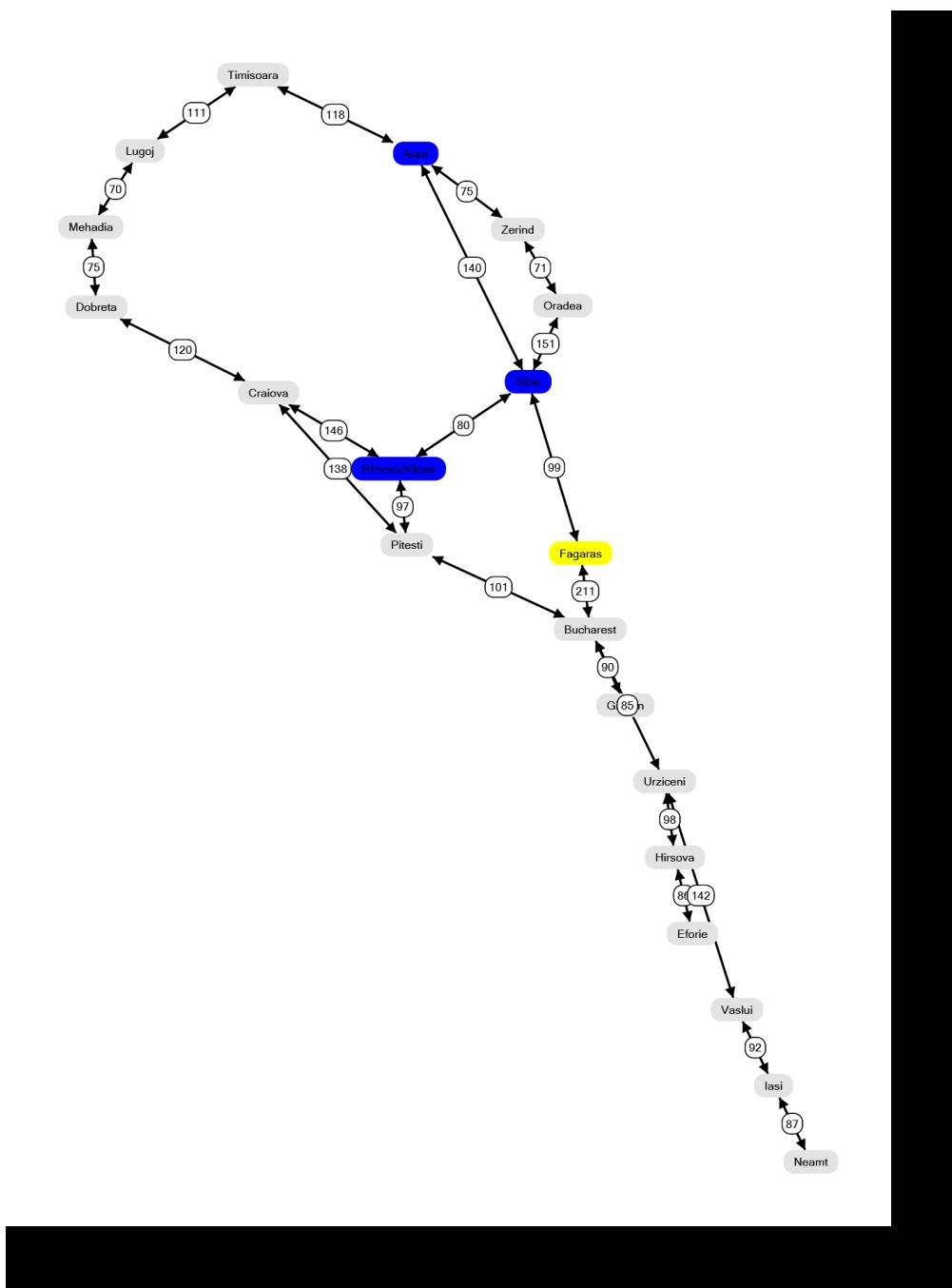
Post-Visit

The algorithm has closed the node Rimnicu Vilcea



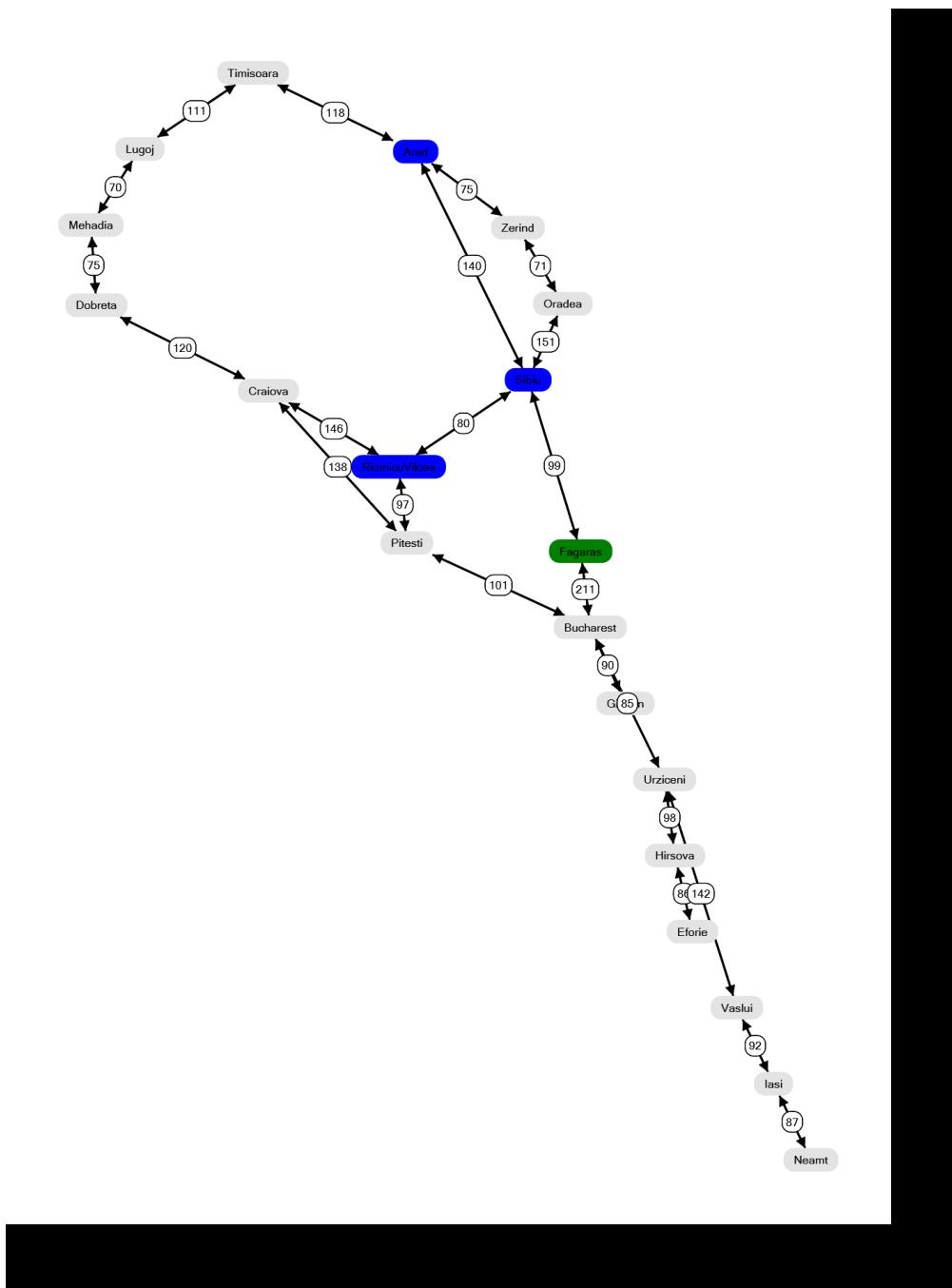
Pre-Visit

The algorithm has opened the node Fagaras



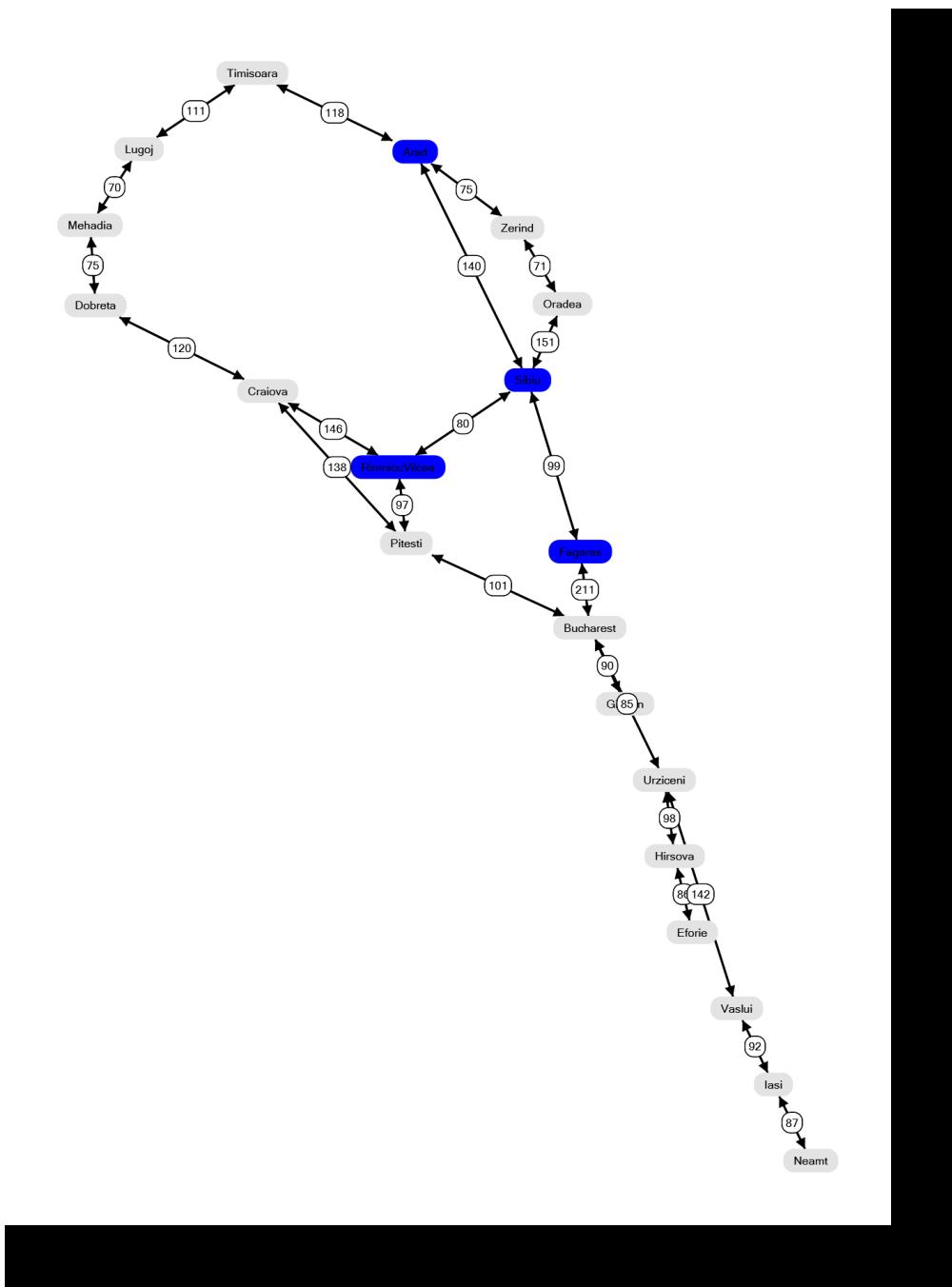
Visit

The algorithm has marked the node Fagaras as visited



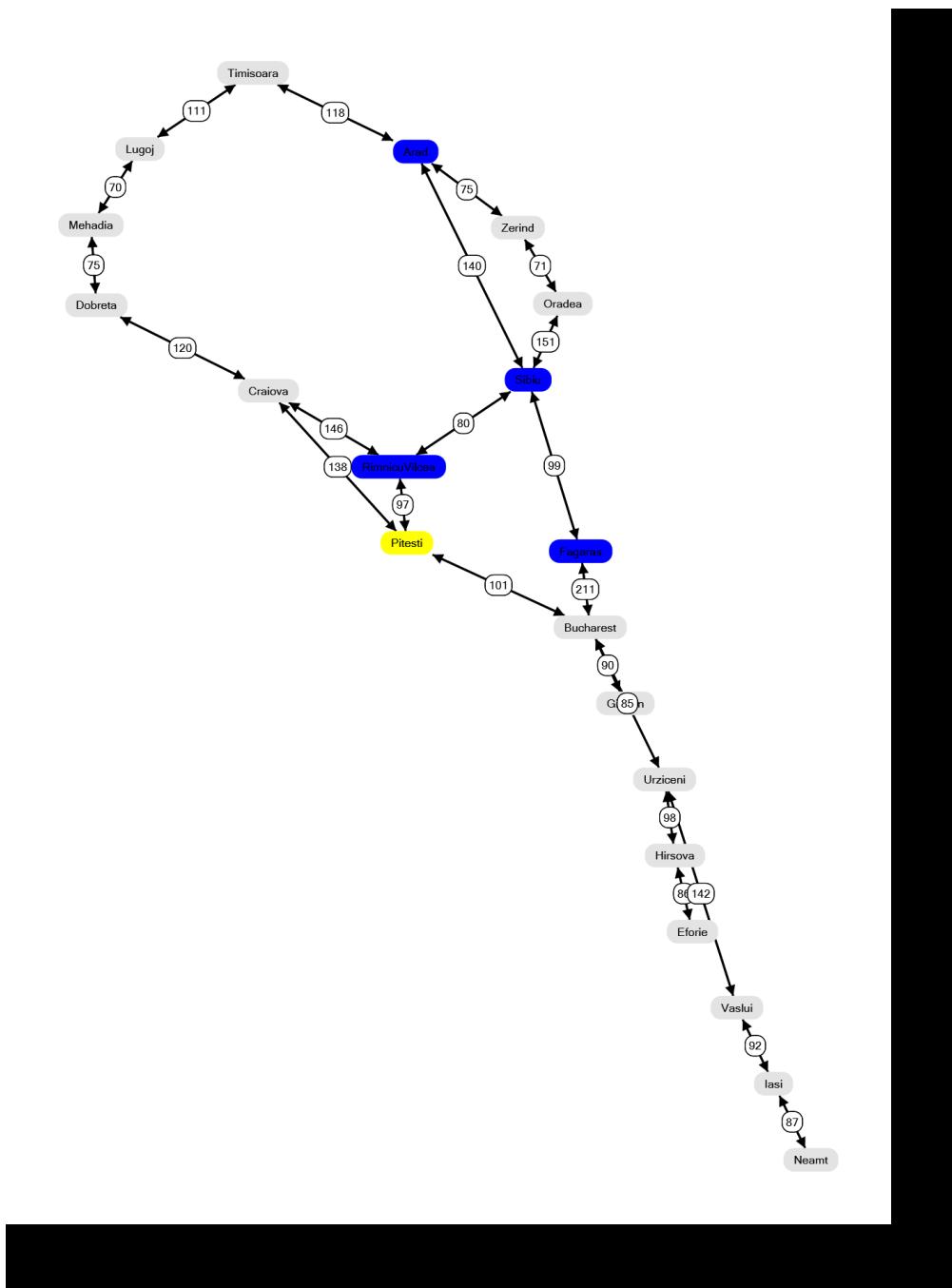
Post-Visit

The algorithm has closed the node Fagaras



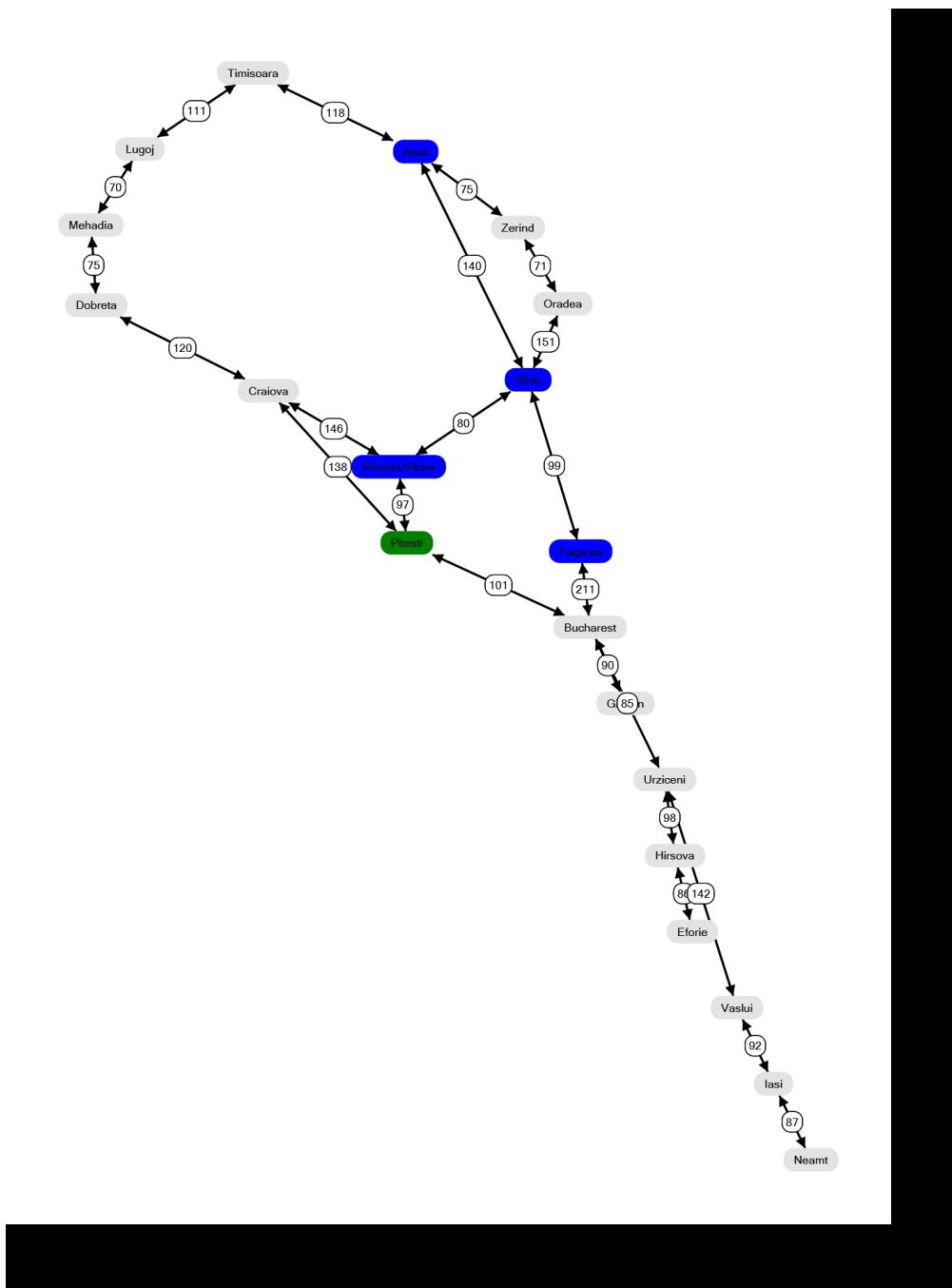
Pre-Visit

The algorithm has opened the node Pitesti



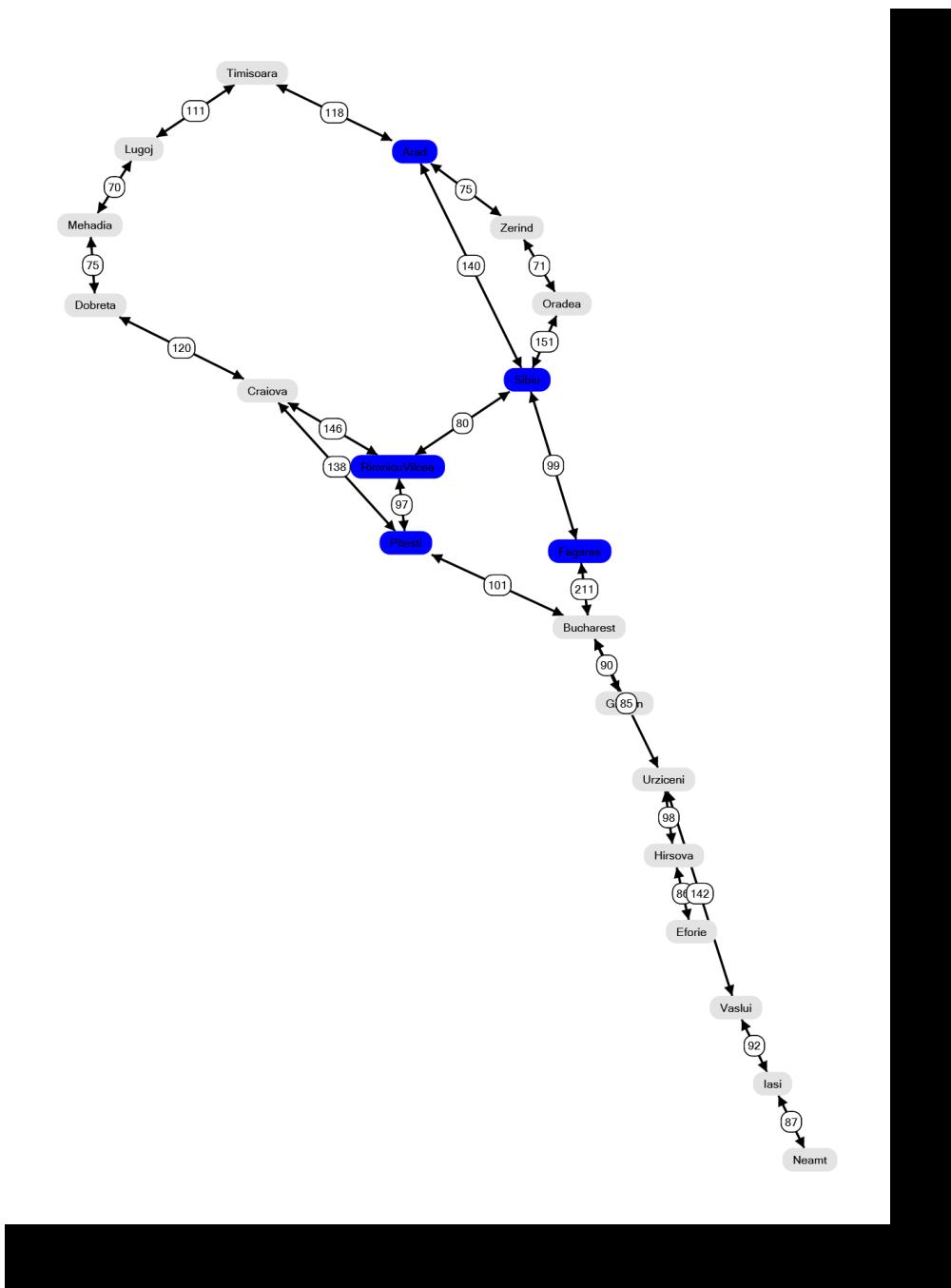
Visit

The algorithm has marked the node Pitesti as visited



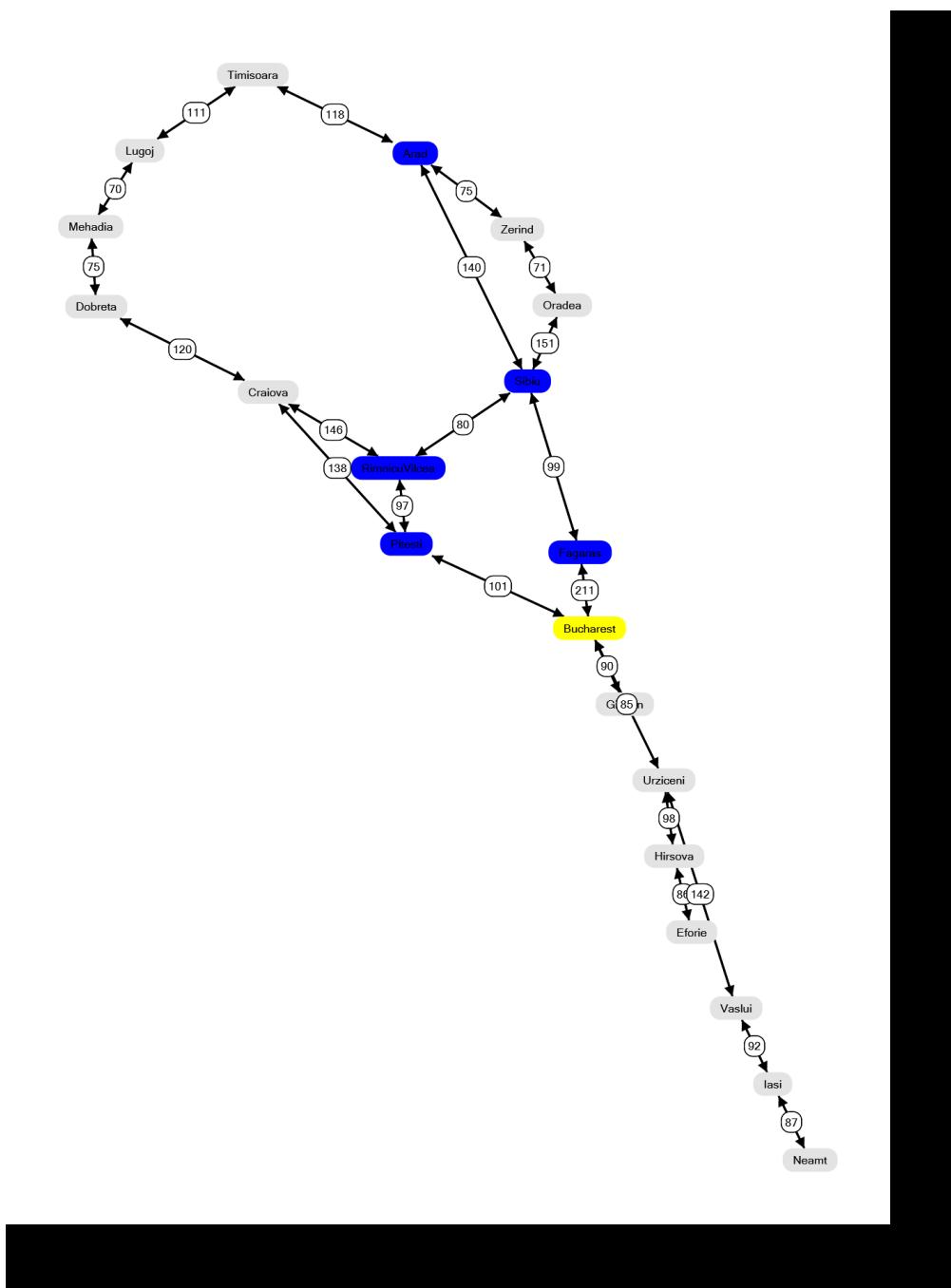
Post-Visit

The algorithm has closed the node Pitesti



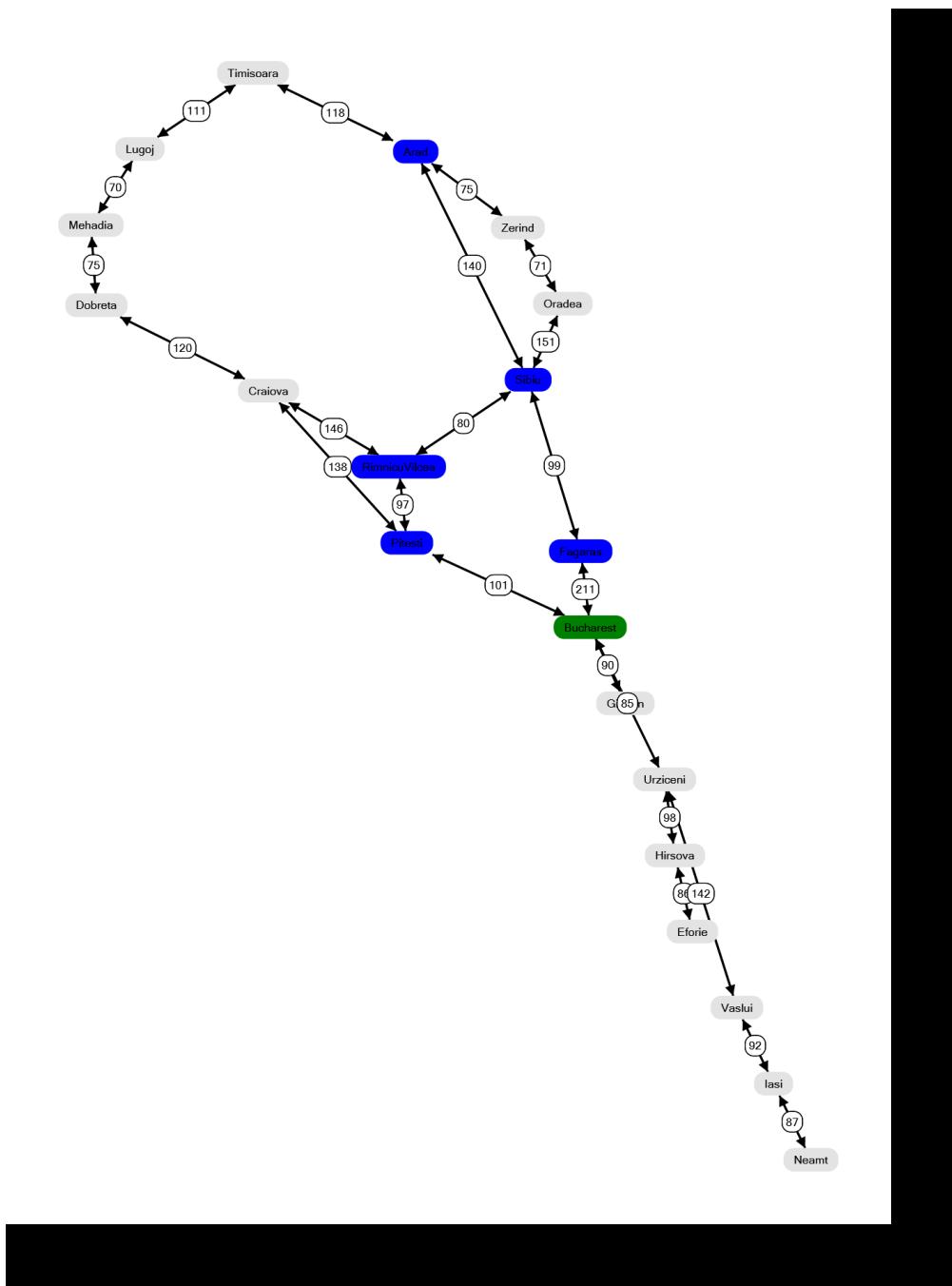
Pre-Visit

The algorithm has opened the node Bucharest



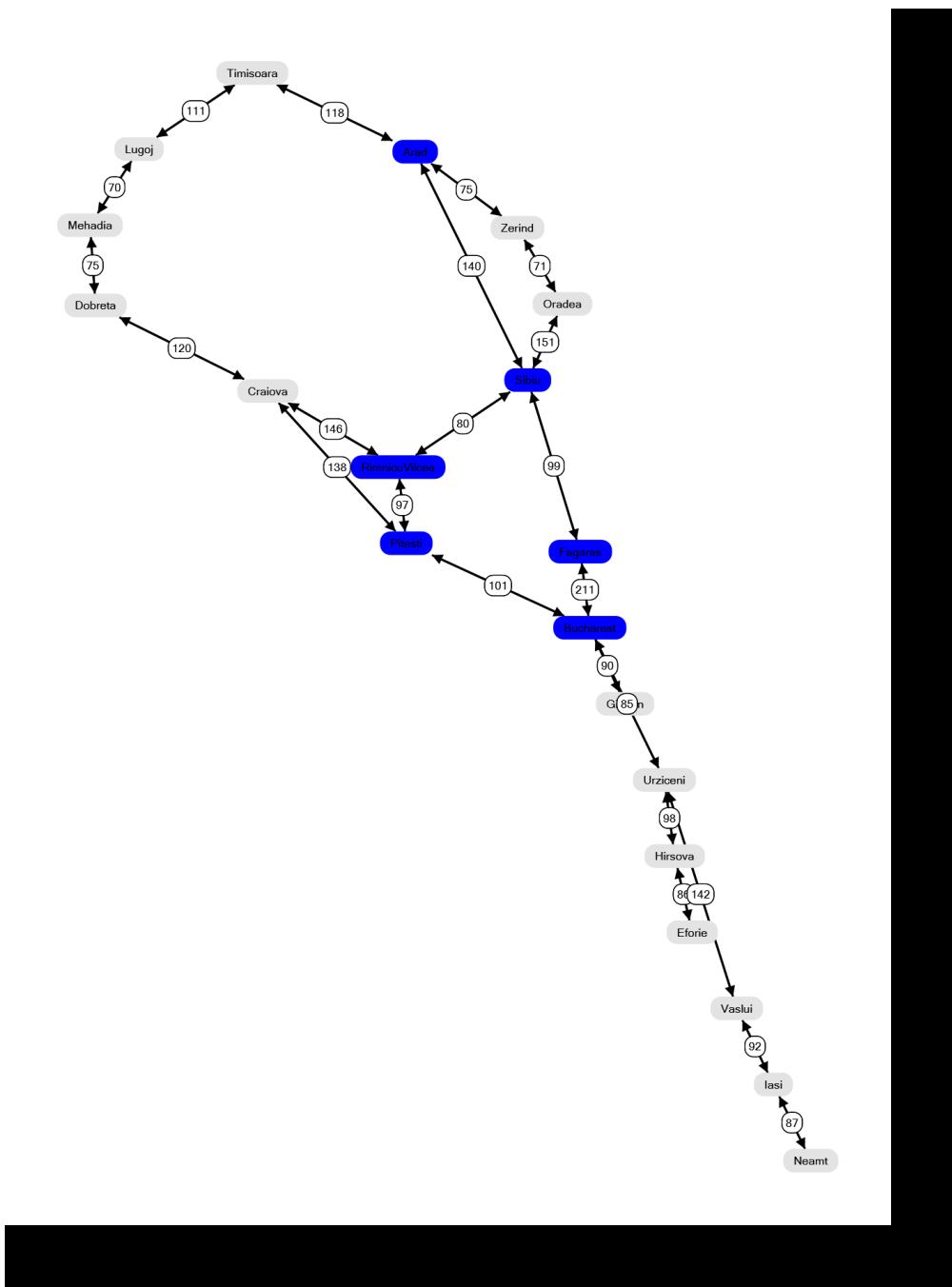
Visit

The algorithm has marked the node Bucharest as visited



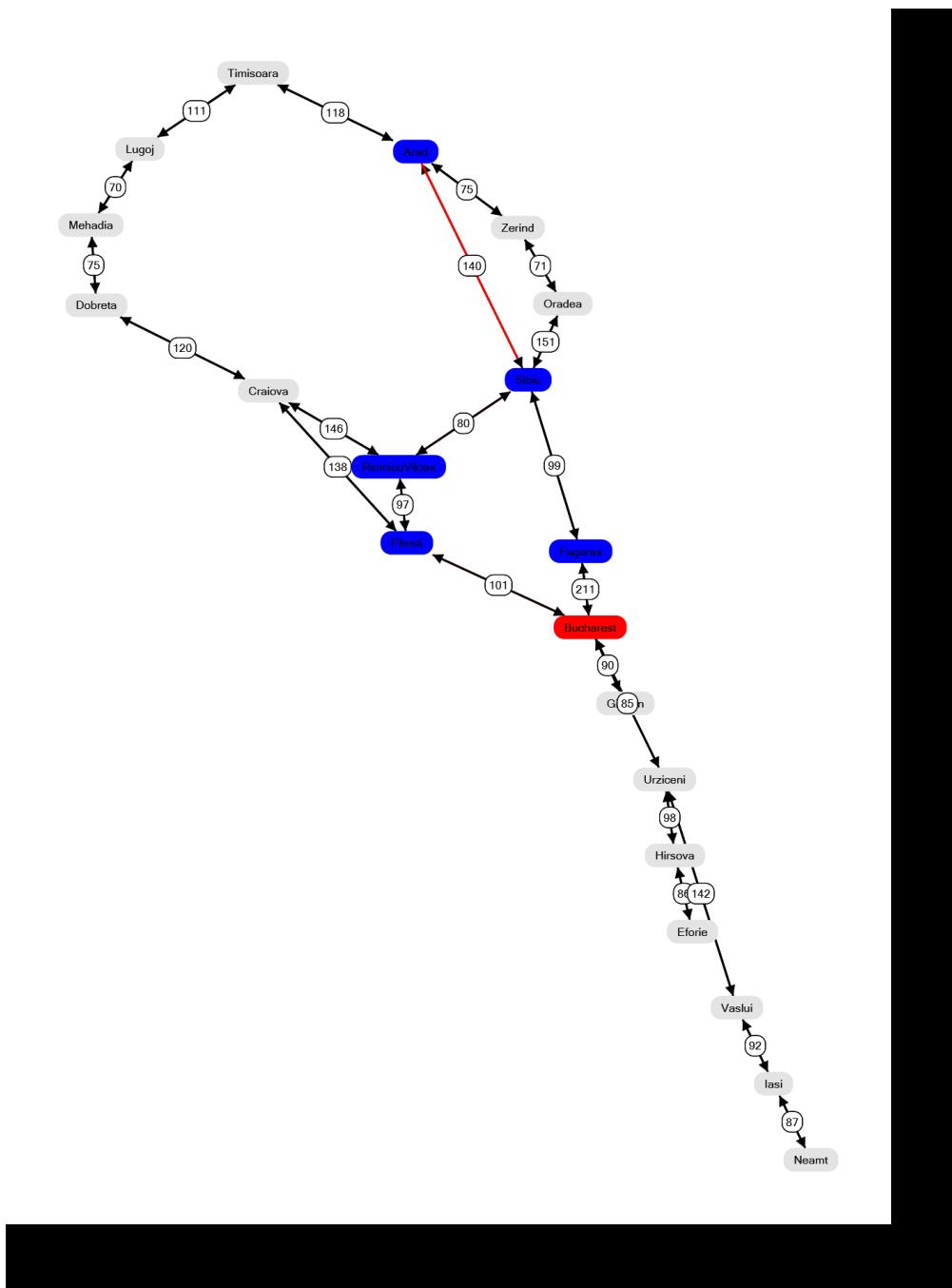
Post-Visit

The algorithm has closed the node Bucharest



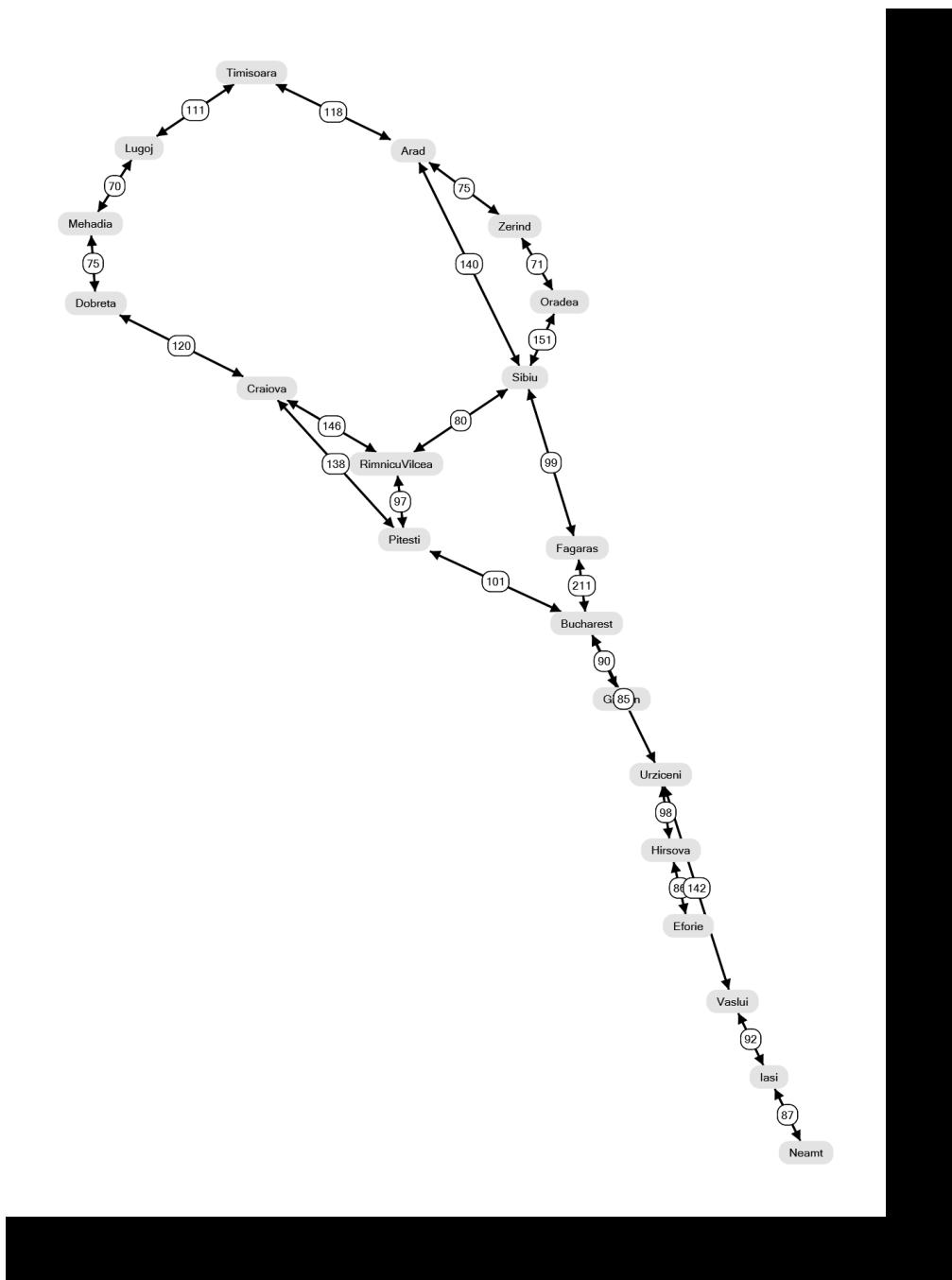
Result Found

The algorithm has found the target node Bucharest



Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



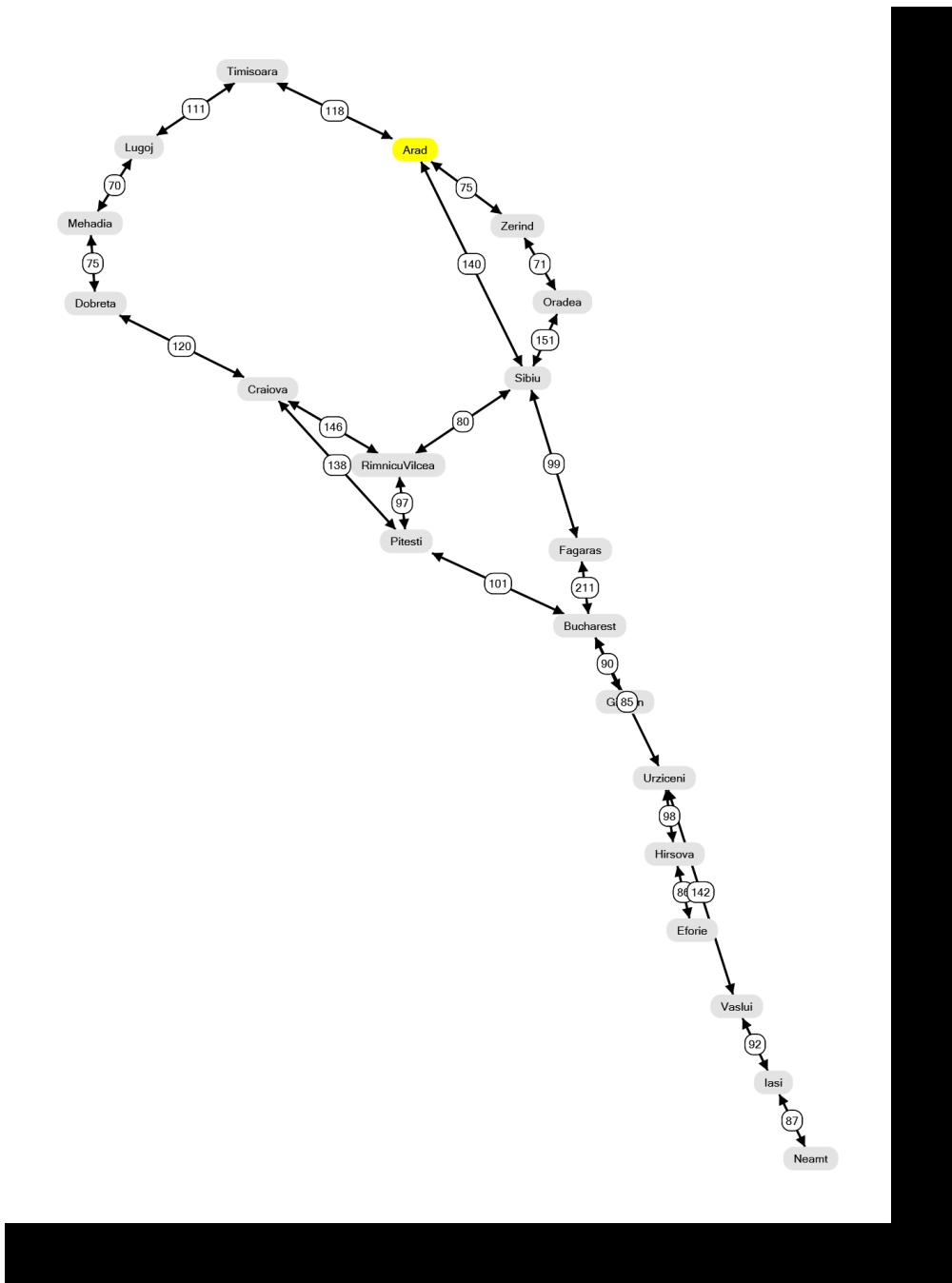
Greedy Best-first search

Best-first search is a search algorithm which explores a graph by expanding the most promising node chosen according to a specified rule.

The algorithm took 00:00:00.0021669 to find the goal.

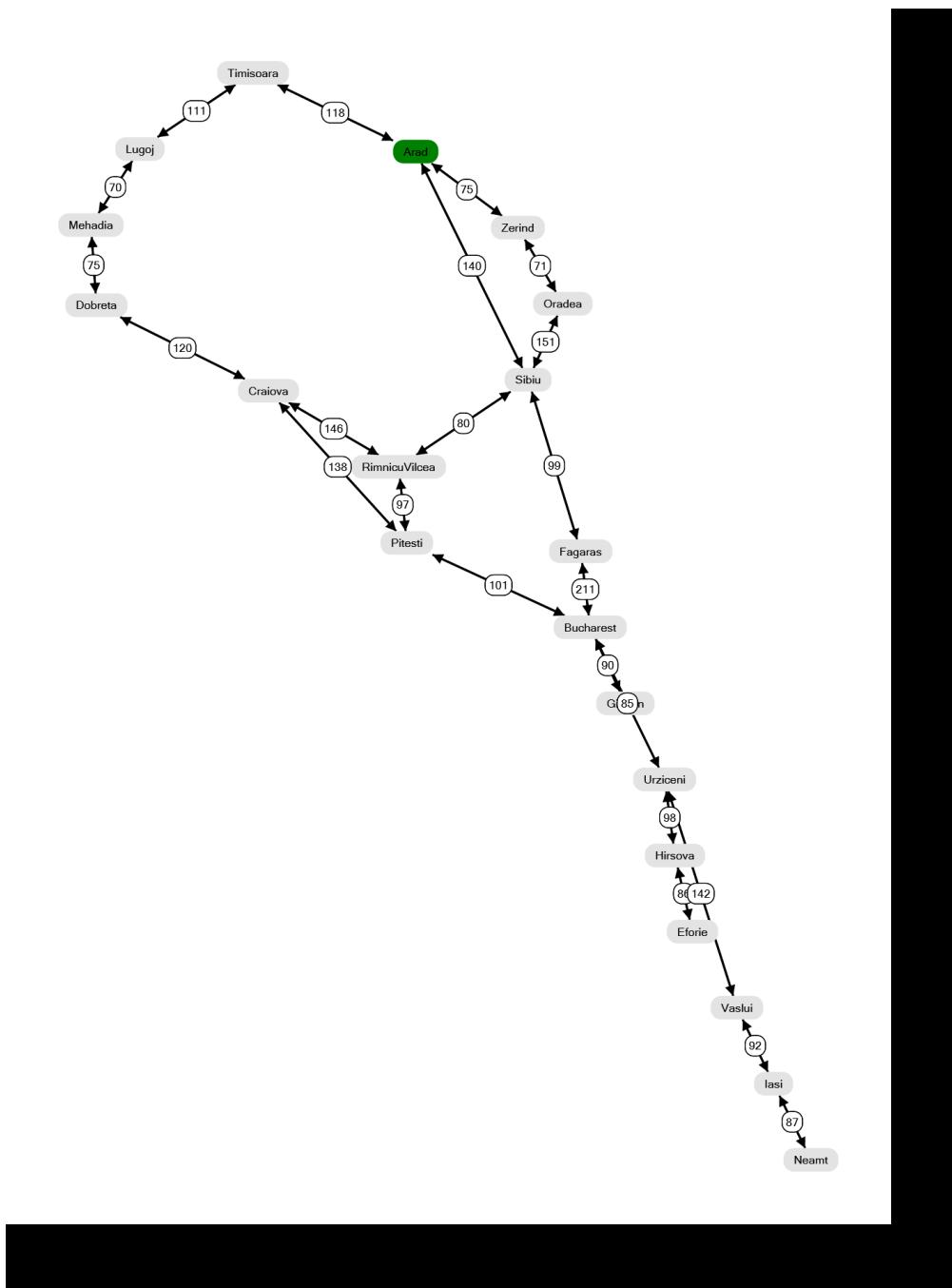
Pre-Visit

The algorithm has opened the node Arad



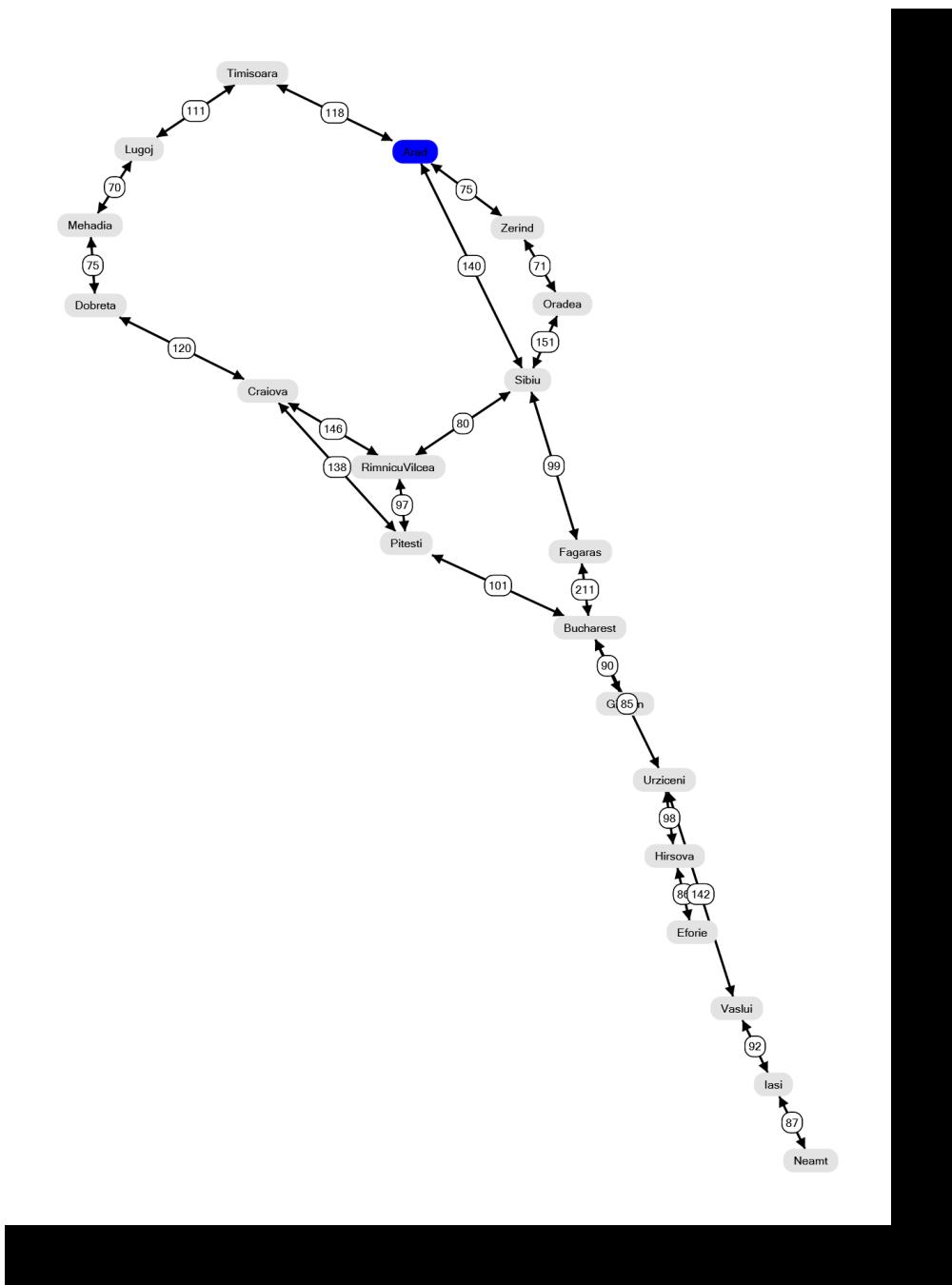
Visit

The algorithm has marked the node Arad as visited



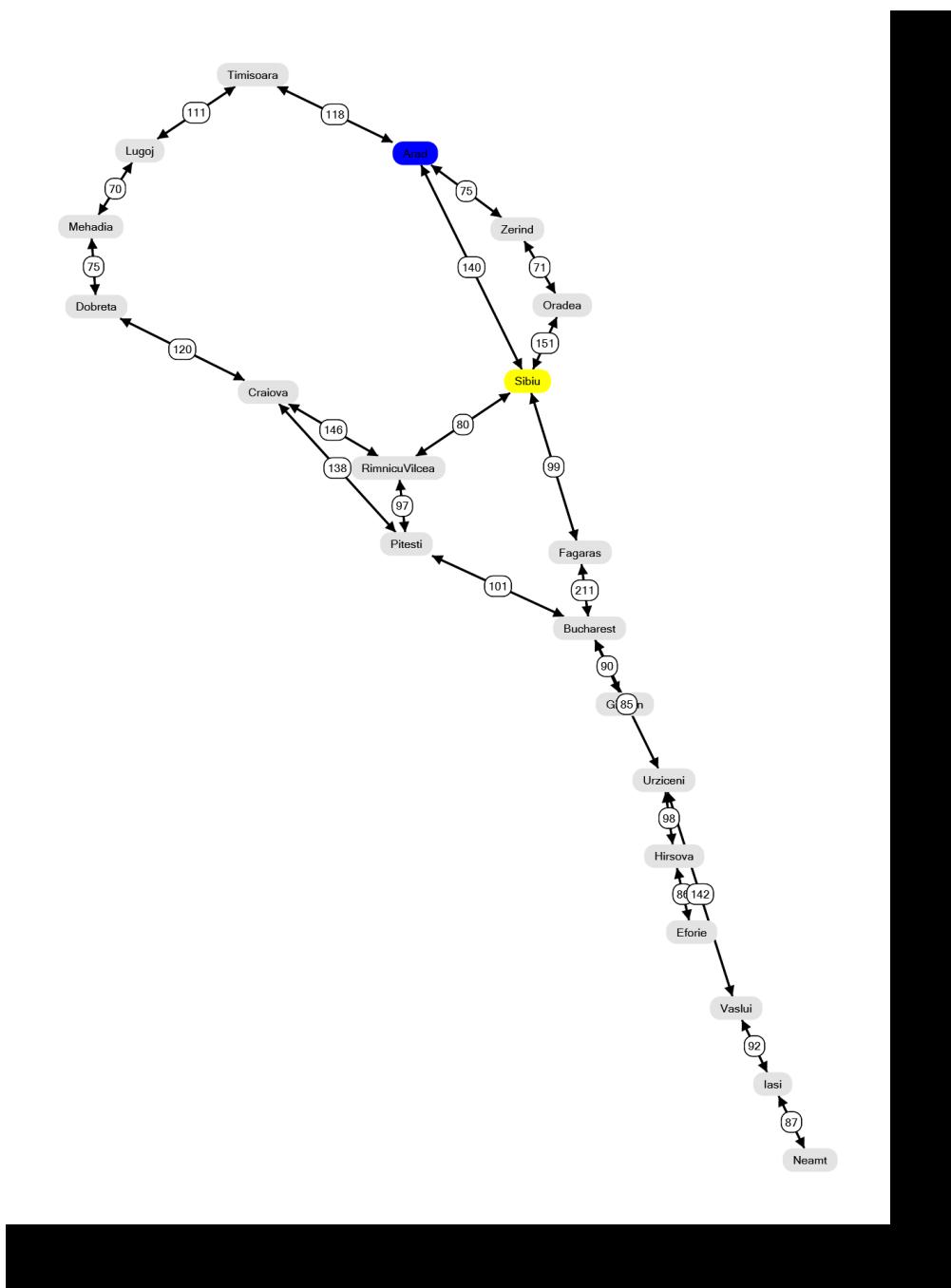
Post-Visit

The algorithm has closed the node Arad



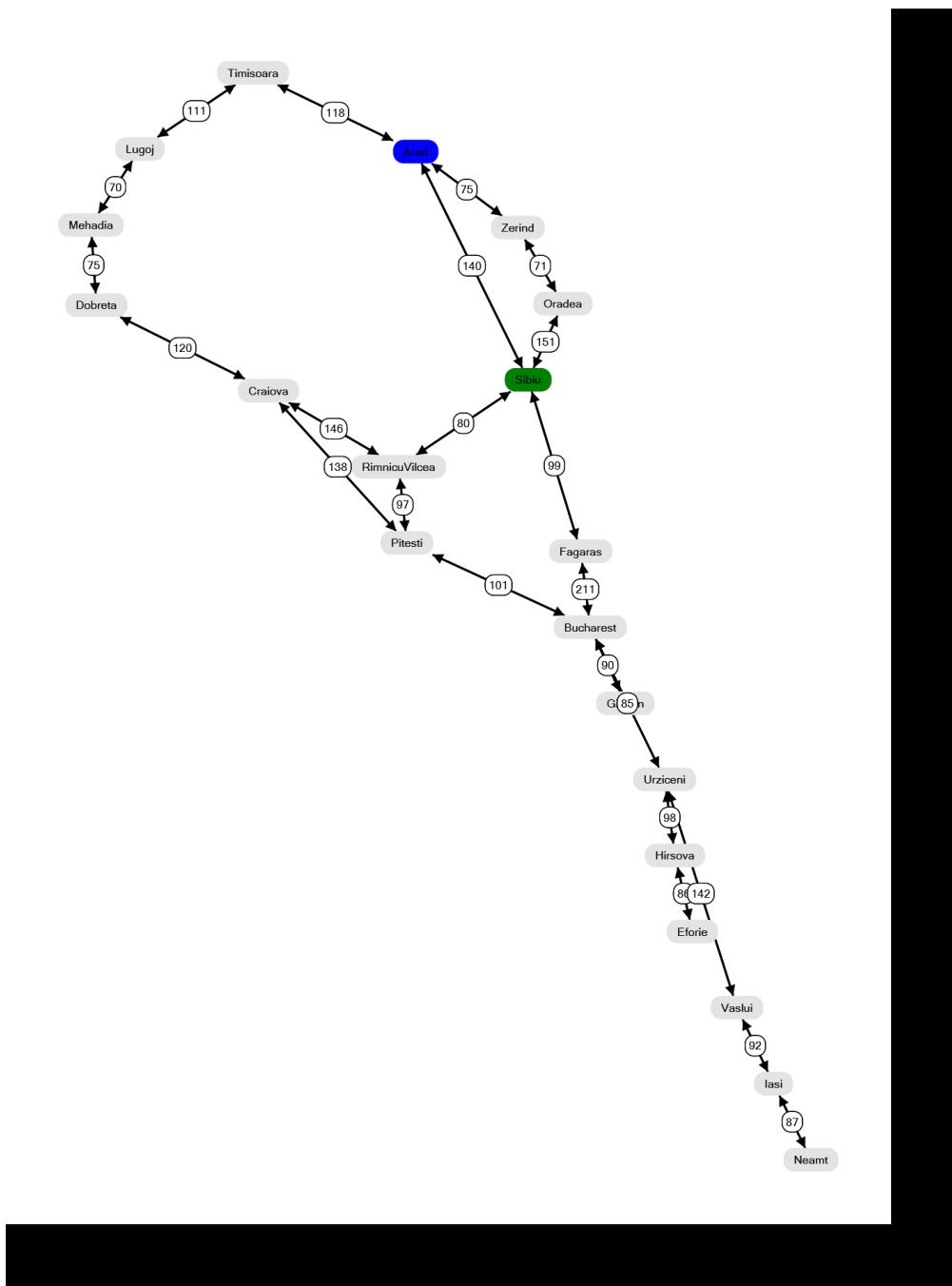
Pre-Visit

The algorithm has opened the node Sibiu



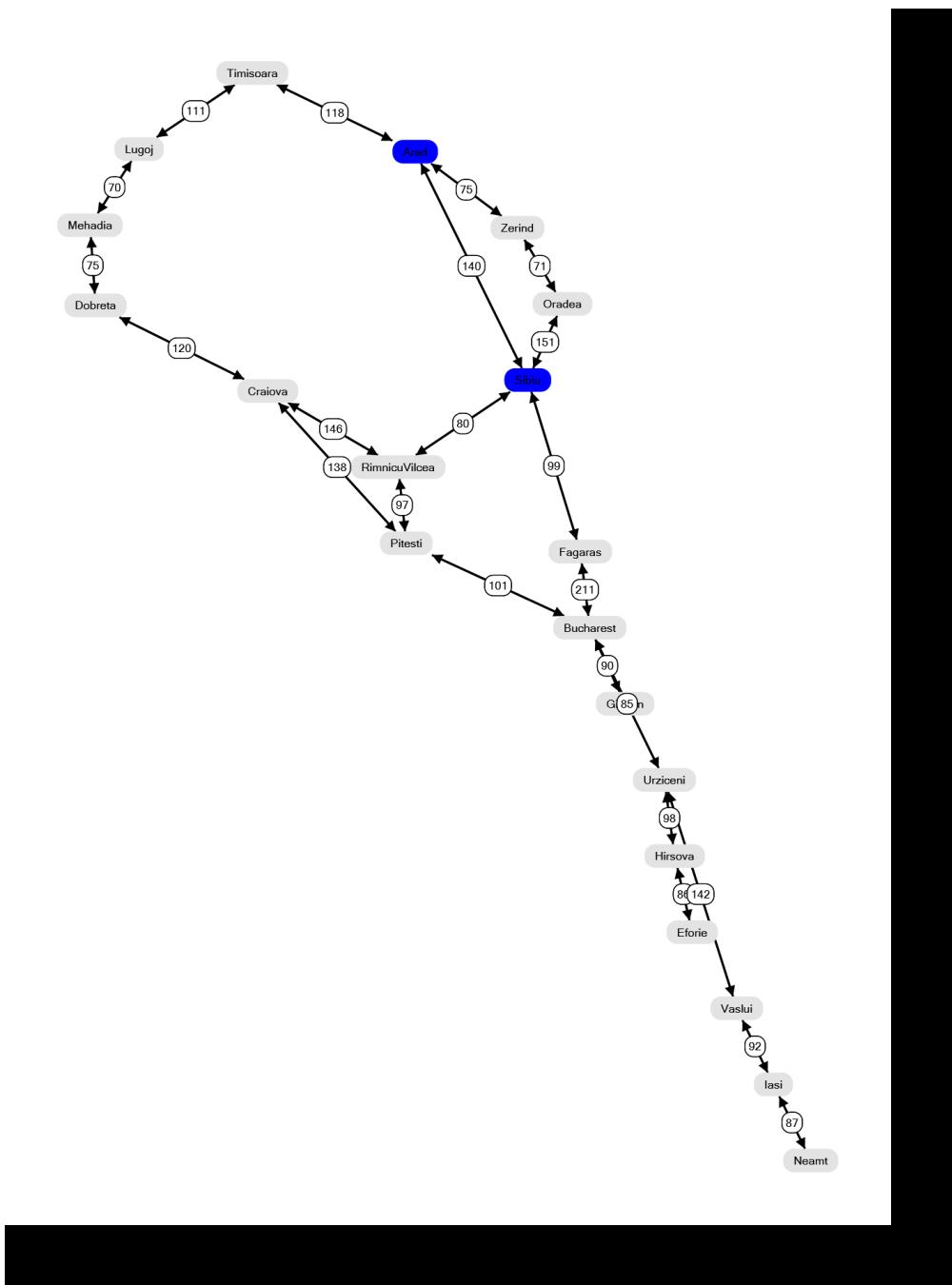
Visit

The algorithm has marked the node Sibiu as visited



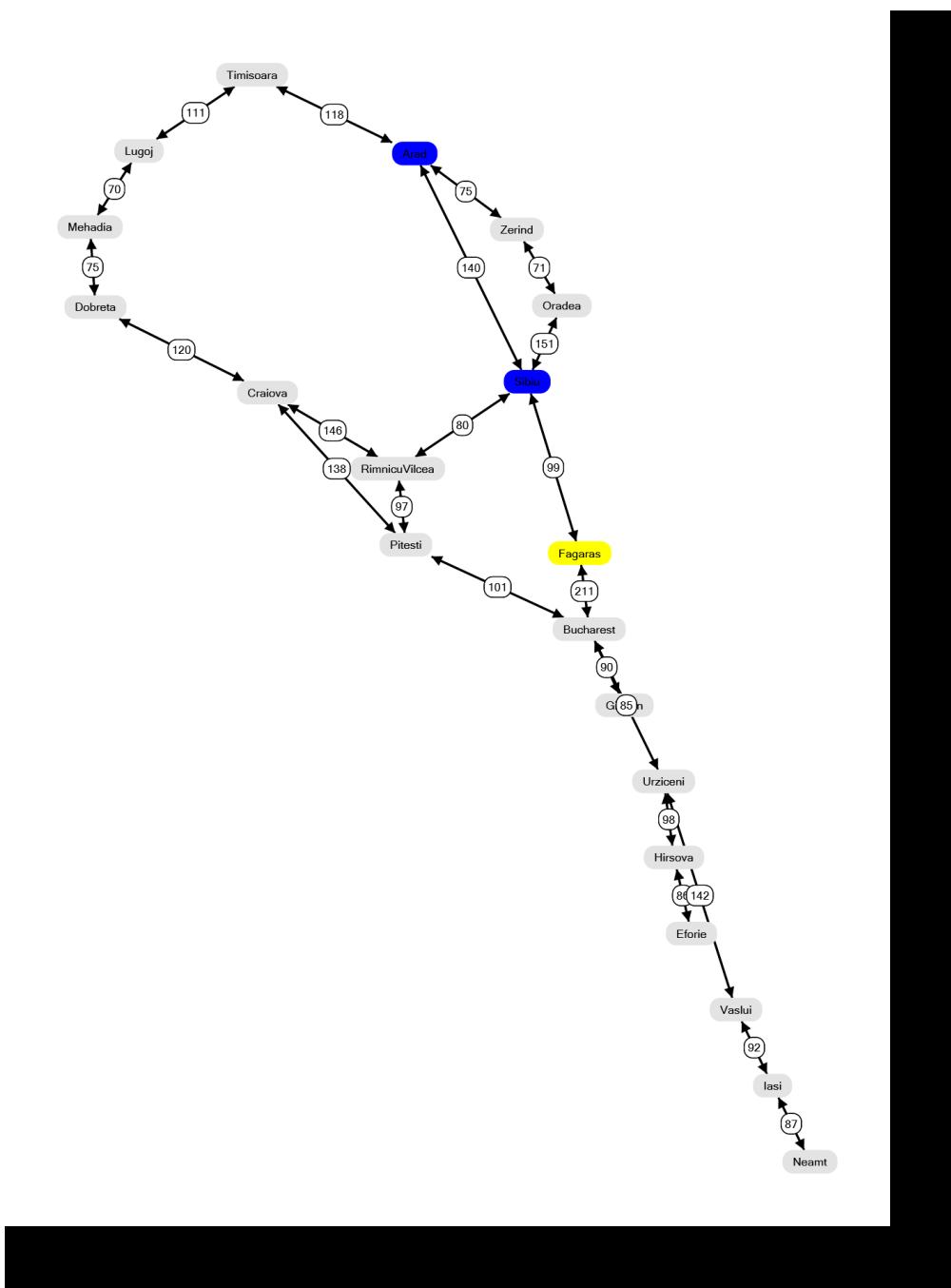
Post-Visit

The algorithm has closed the node Sibiu



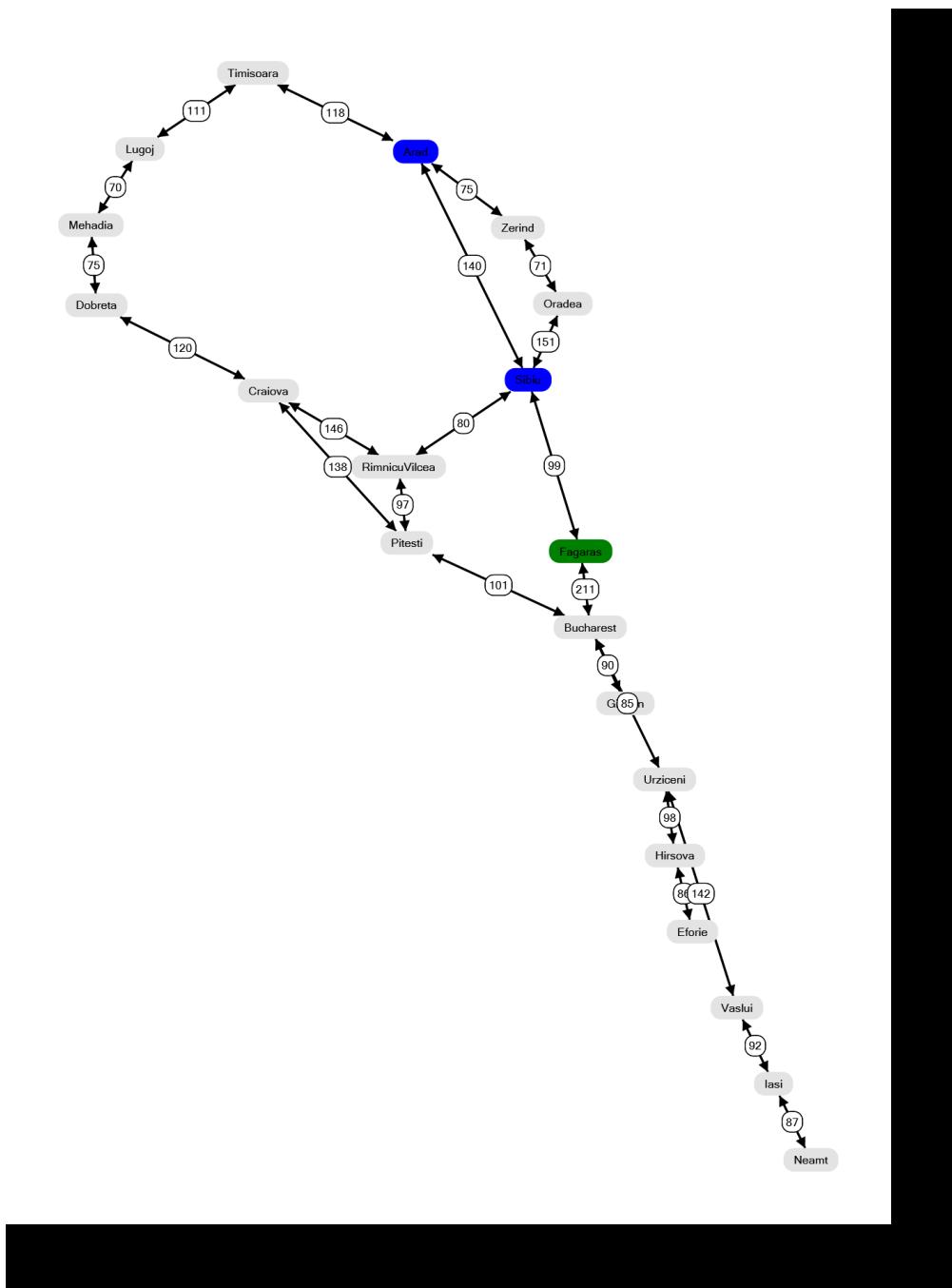
Pre-Visit

The algorithm has opened the node Fagaras



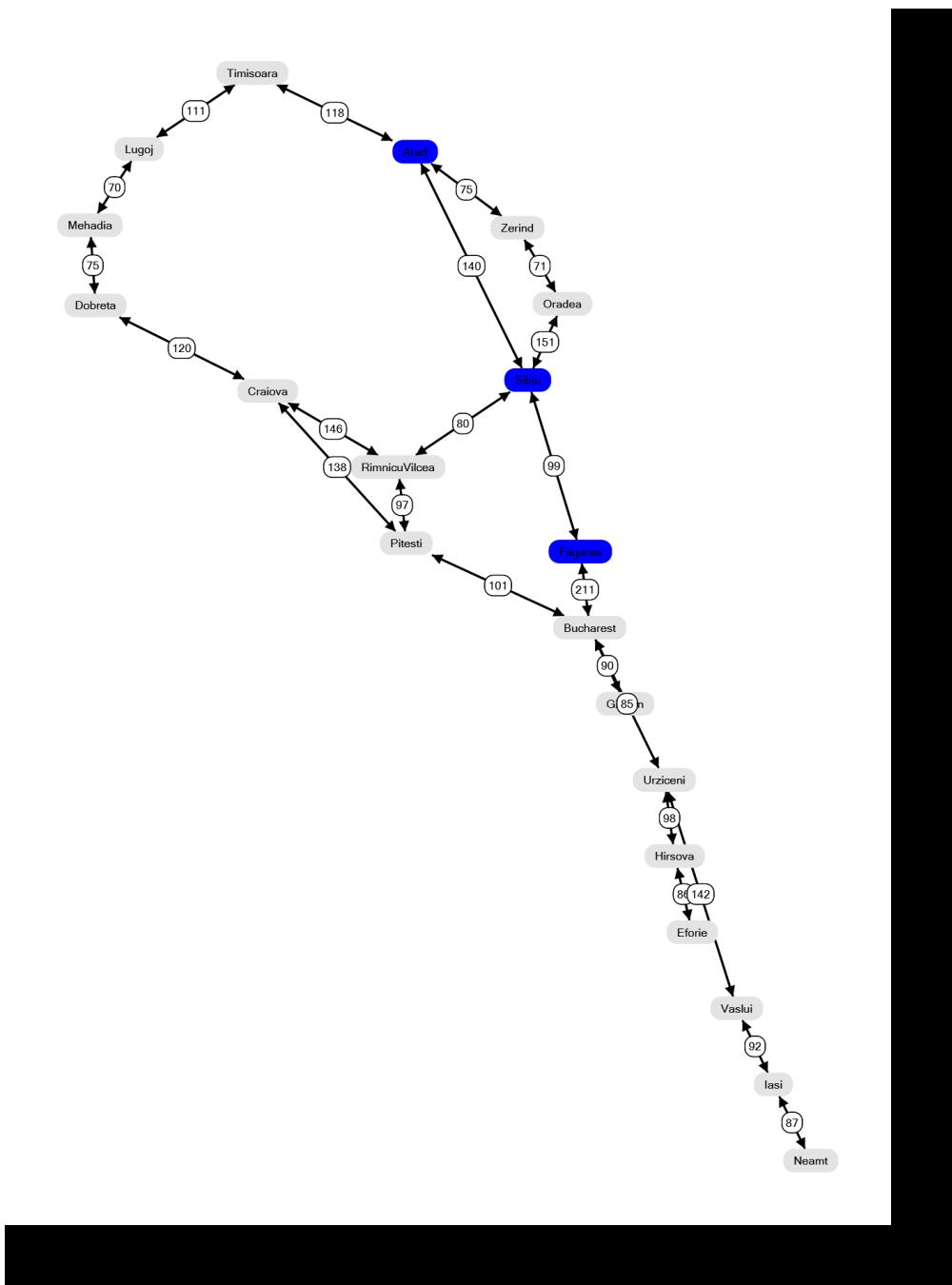
Visit

The algorithm has marked the node Fagaras as visited



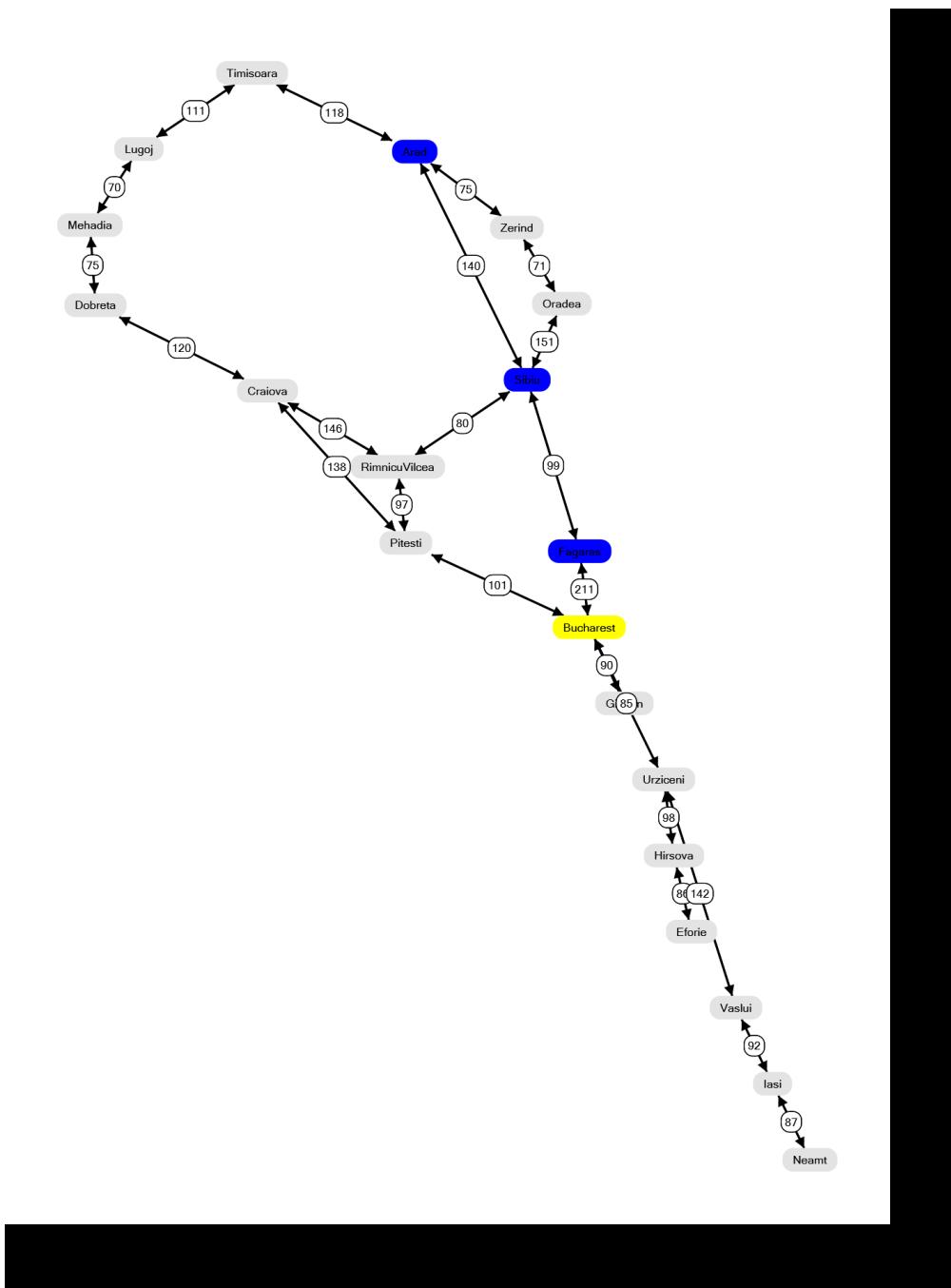
Post-Visit

The algorithm has closed the node Fagaras



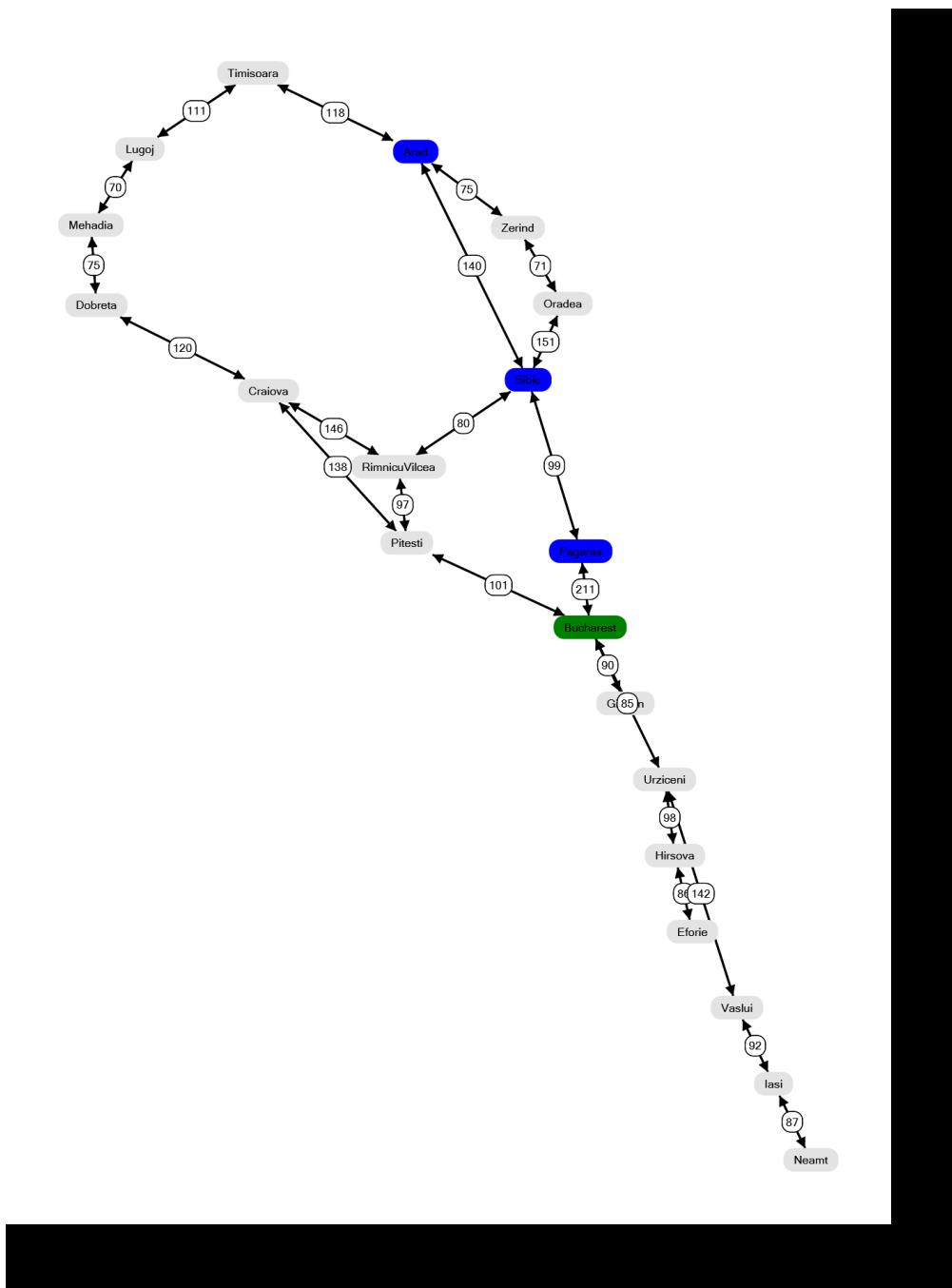
Pre-Visit

The algorithm has opened the node Bucharest



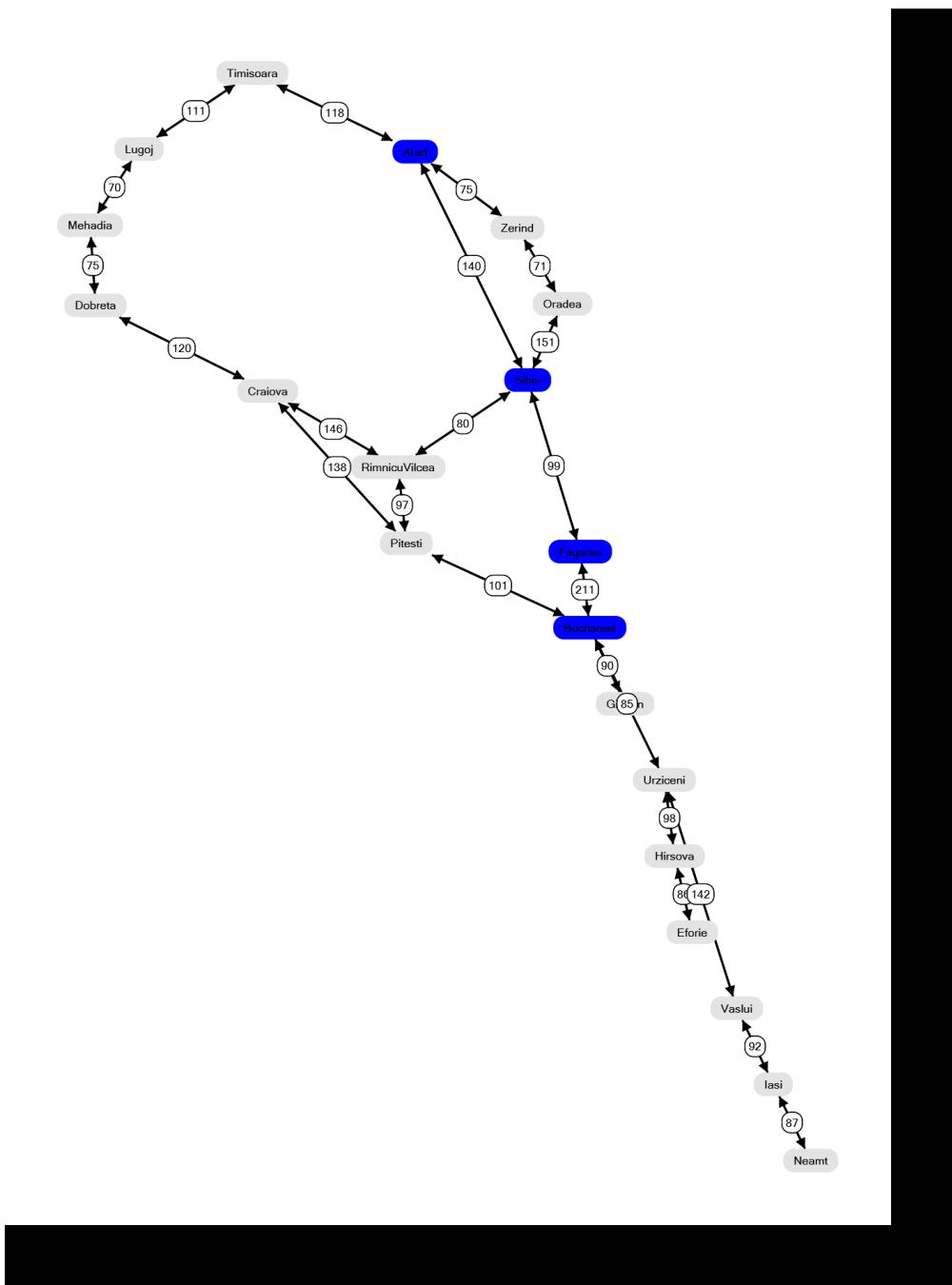
Visit

The algorithm has marked the node Bucharest as visited



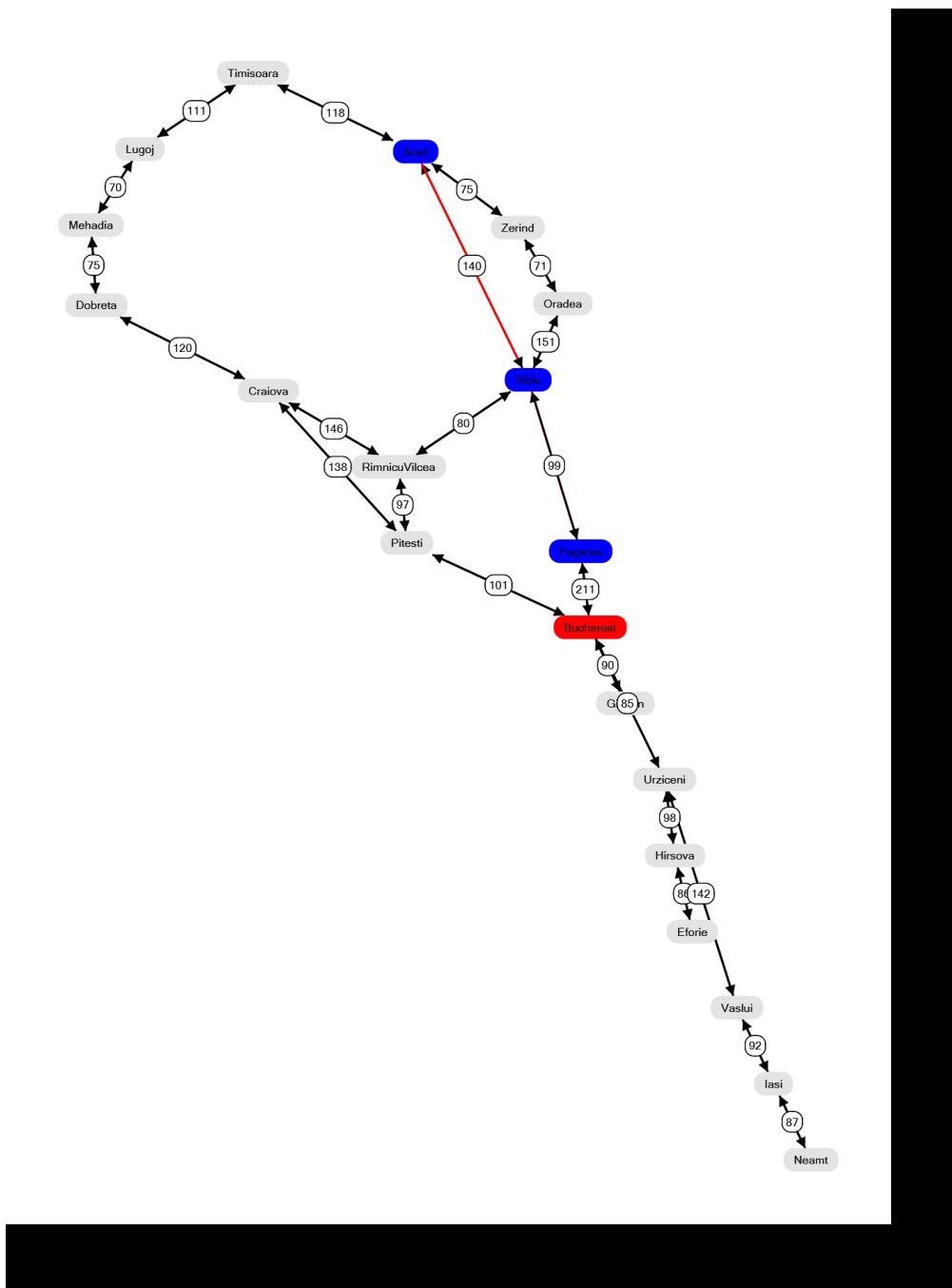
Post-Visit

The algorithm has closed the node Bucharest



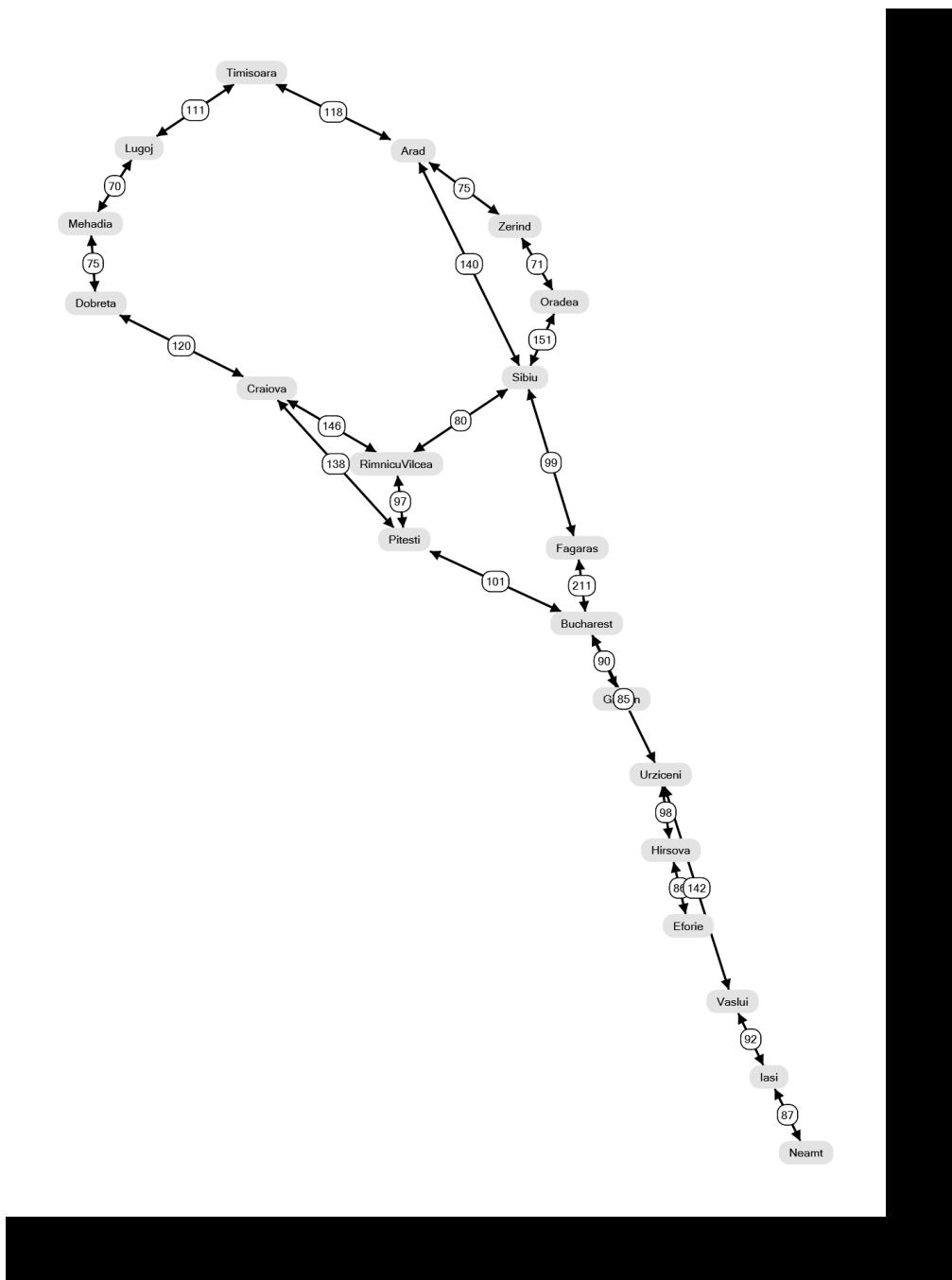
Result Found

The algorithm has found the target node Bucharest



Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



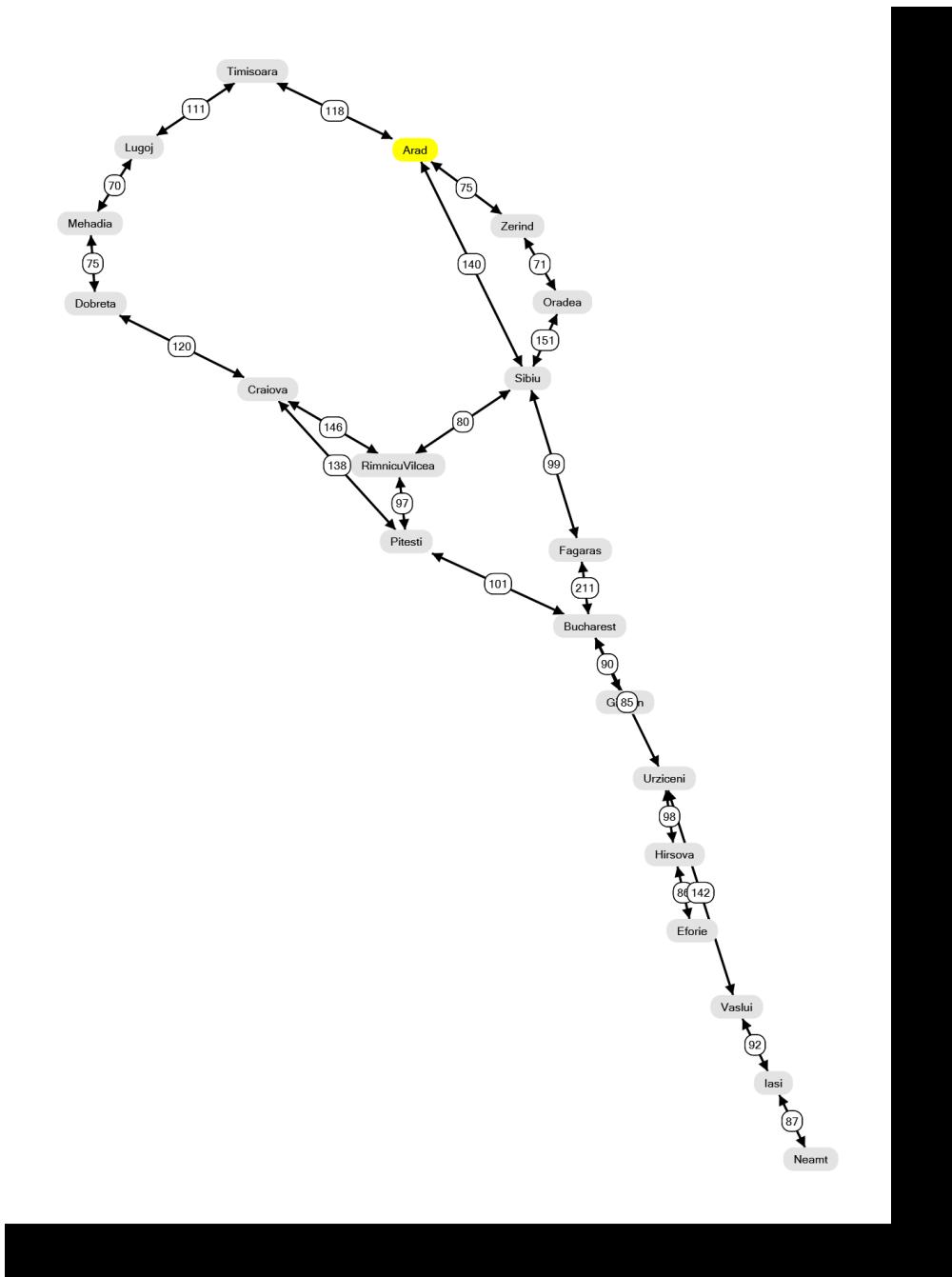
Iterative deepening A*

Iterative deepening A* (IDA*) is a graph traversal and path search algorithm that can find the shortest path between a designated start node and any member of a set of goal nodes in a weighted graph. It is a variant of iterative deepening depth-first search that borrows the idea to use a heuristic function to evaluate the remaining cost to get to the goal from the A* search algorithm. Since it is a depth-first search algorithm, its memory usage is lower than in A*, but unlike ordinary iterative deepening search, it concentrates on exploring the most promising nodes and thus does not go to the same depth everywhere in the search tree. Unlike A*, IDA* does not utilize dynamic programming and therefore often ends up exploring the same nodes many times.

The algorithm took 00:00:00.0032667 to find the goal.

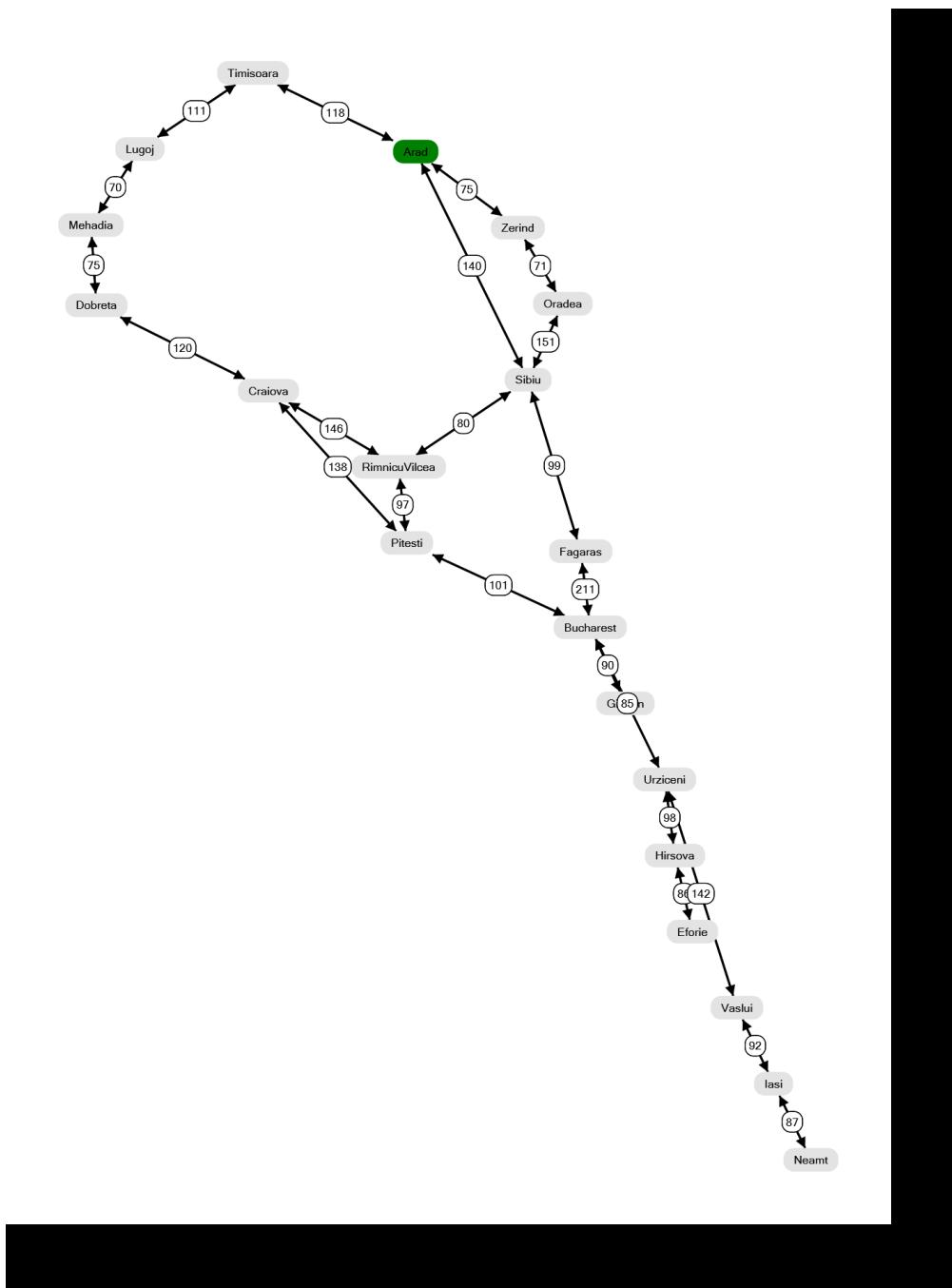
Pre-Visit

The algorithm has opened the node Arad



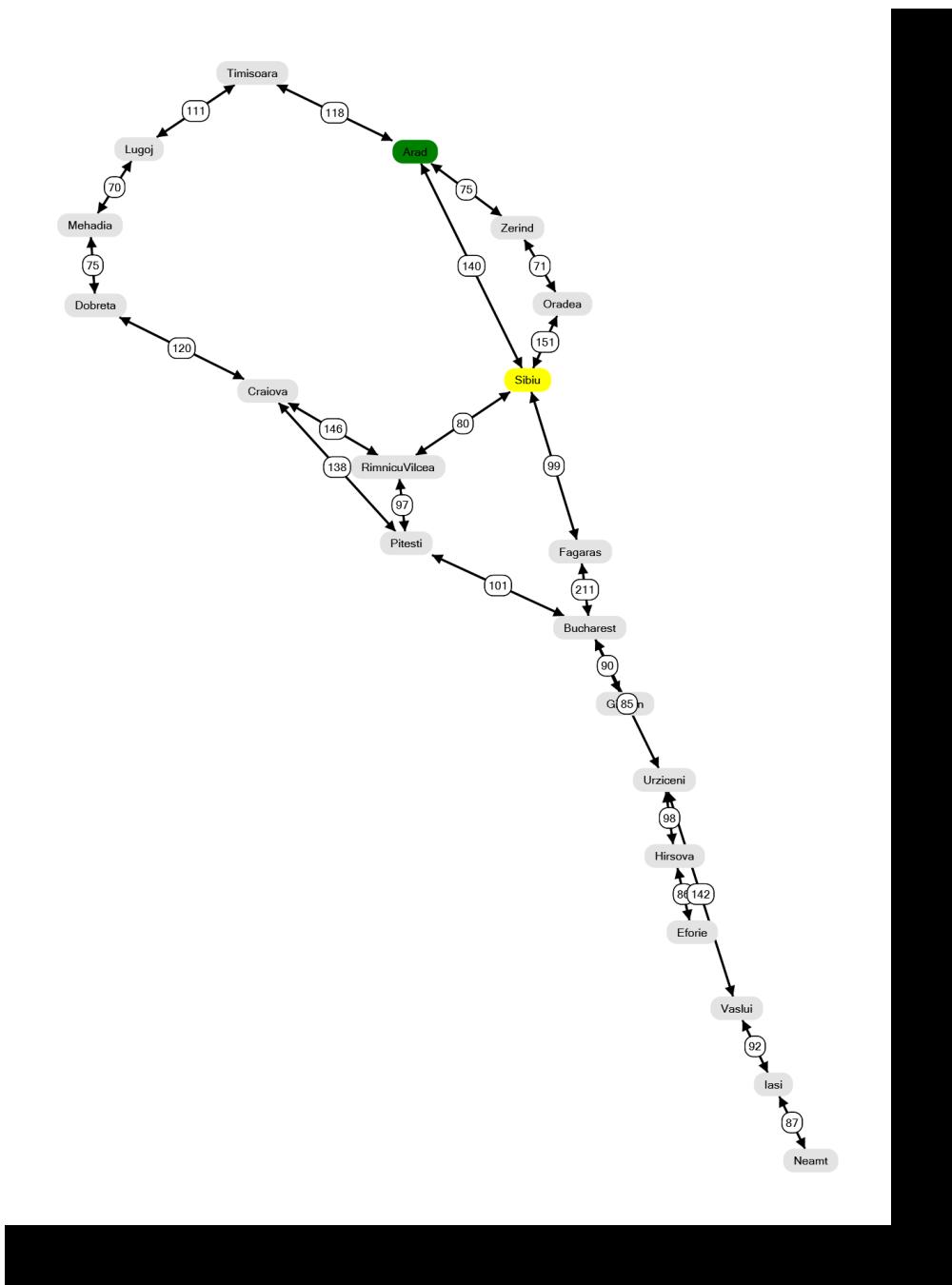
Visit

The algorithm has marked the node Arad as visited



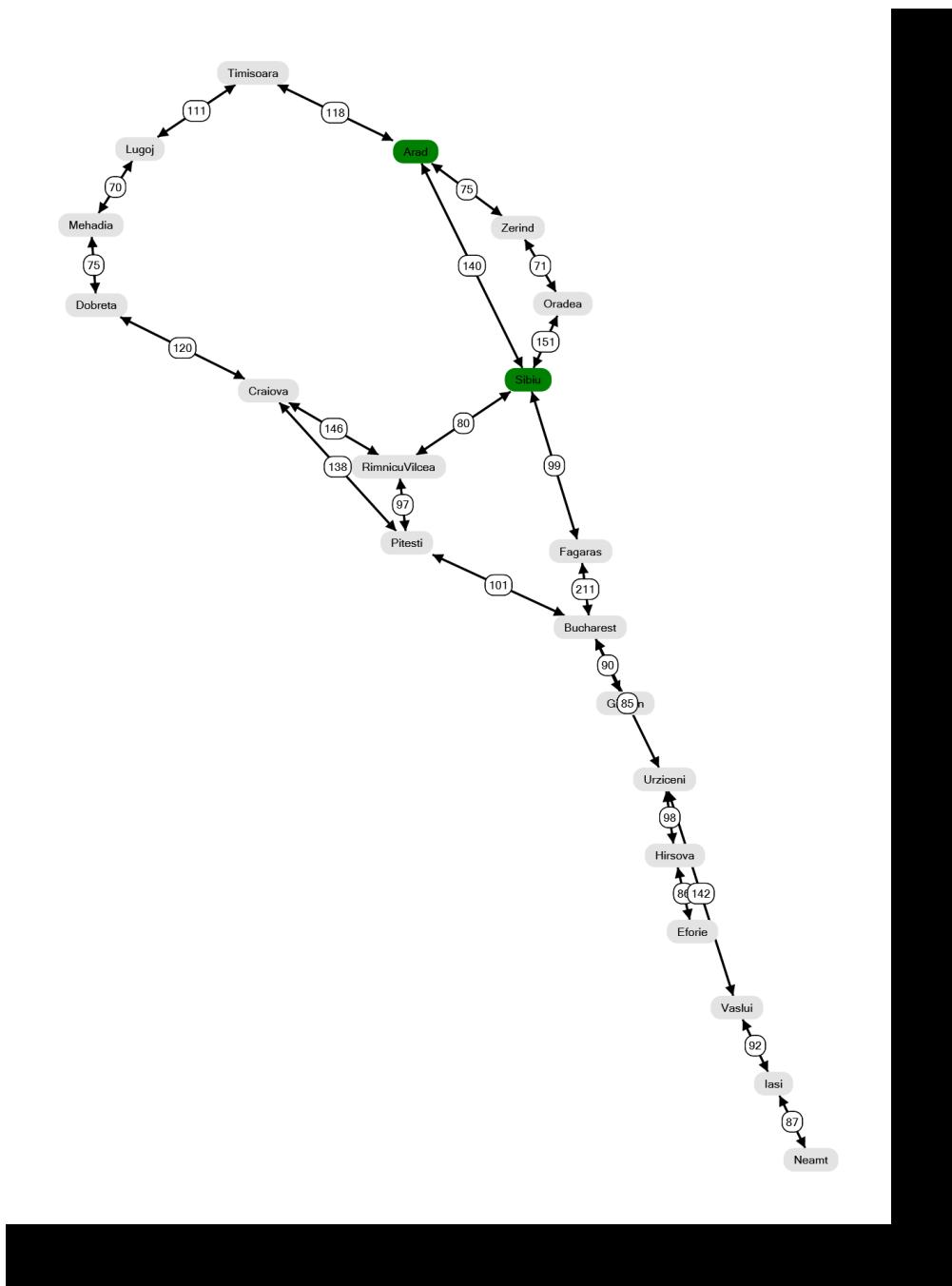
Pre-Visit

The algorithm has opened the node Sibiu



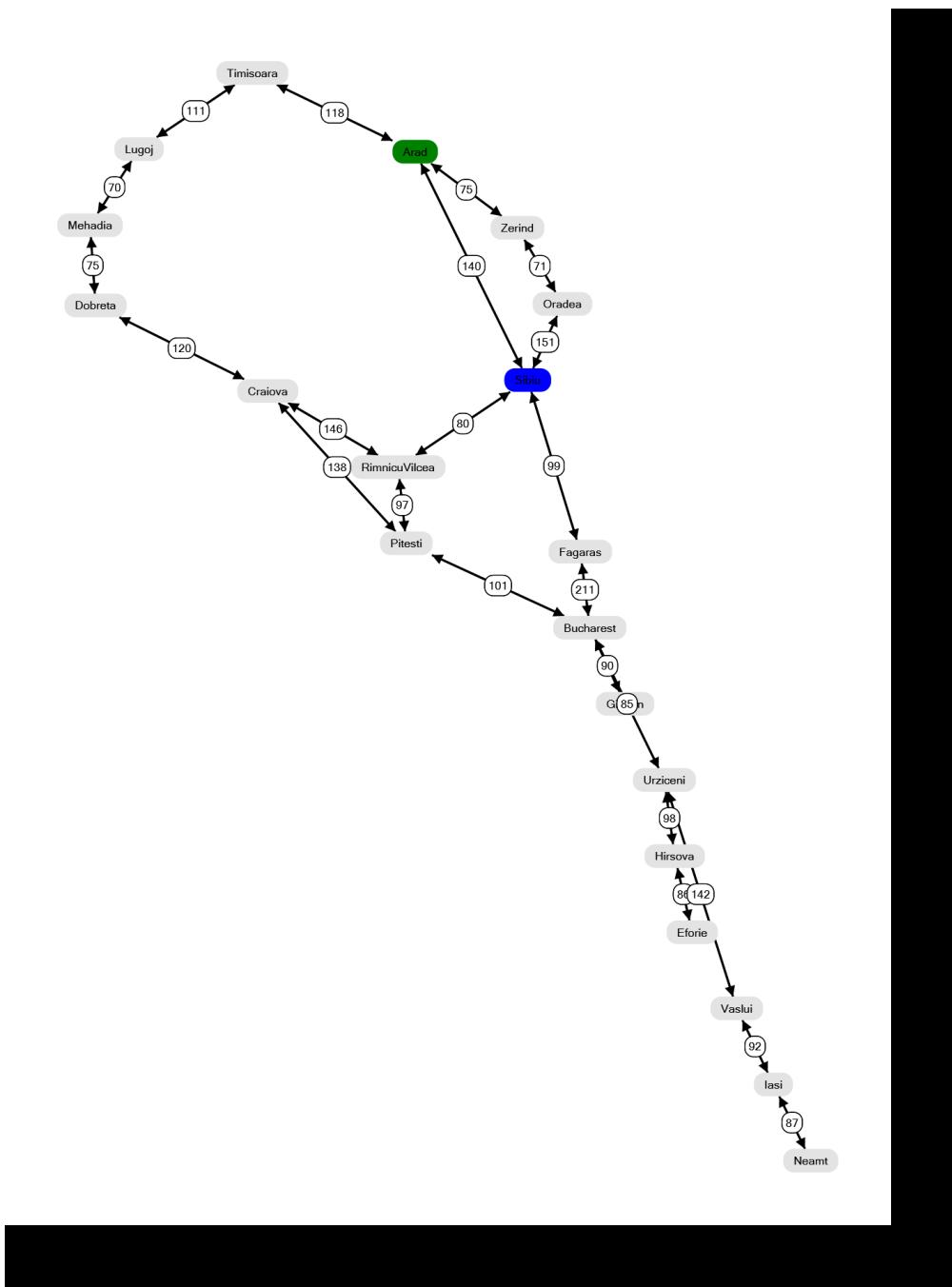
Visit

The algorithm has marked the node Sibiu as visited



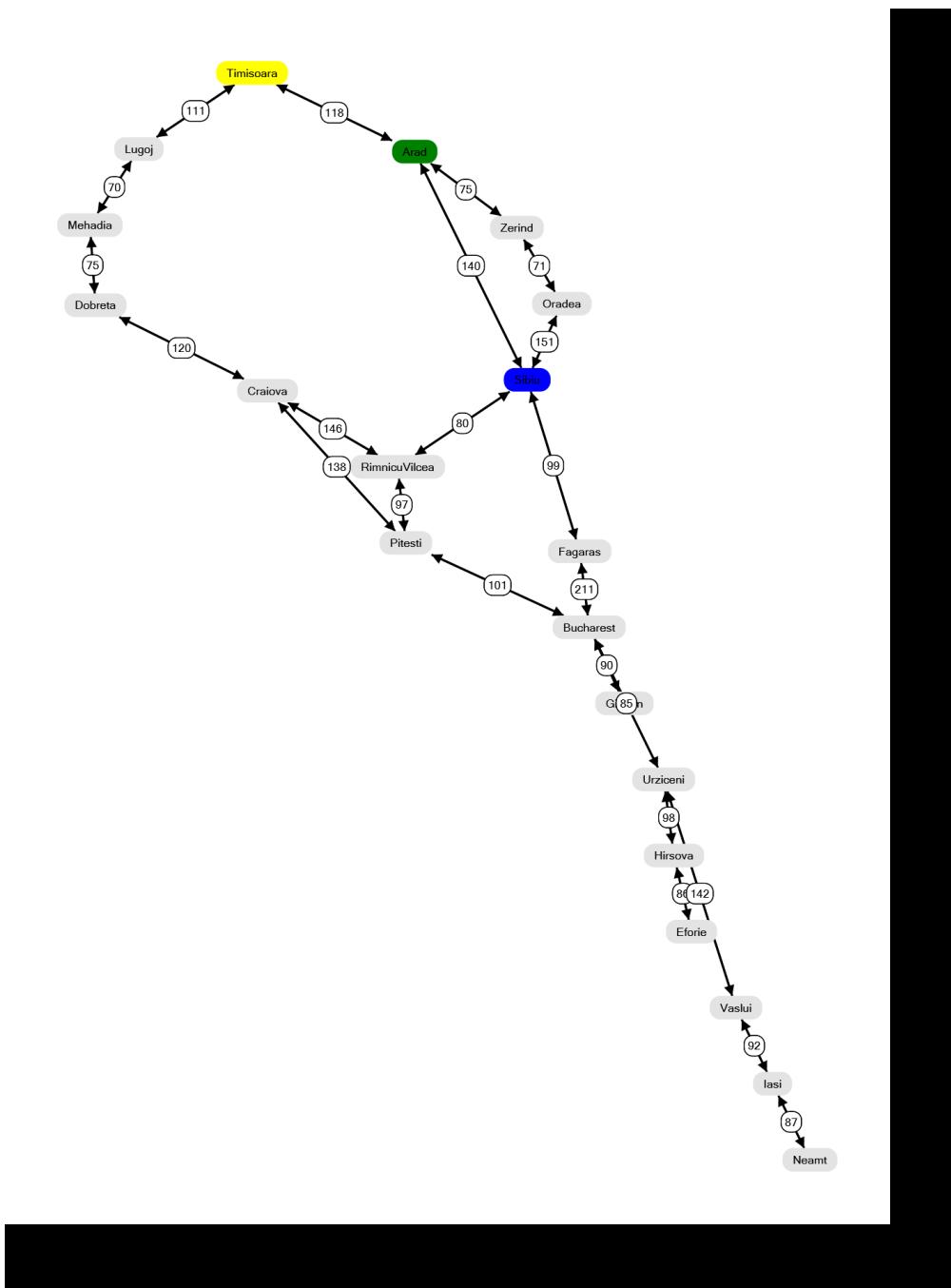
Post-Visit

The algorithm has closed the node Sibiu



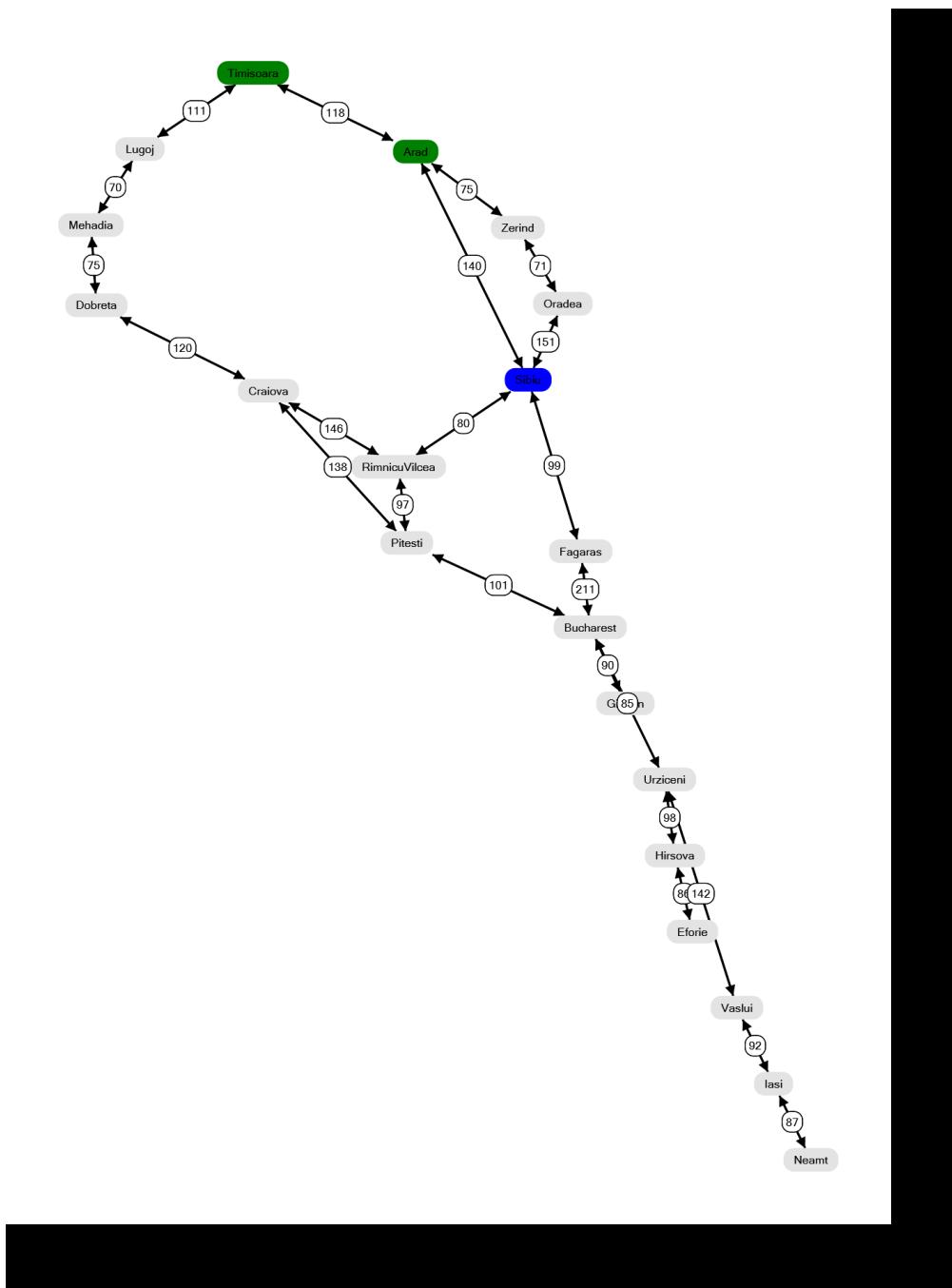
Pre-Visit

The algorithm has opened the node Timisoara



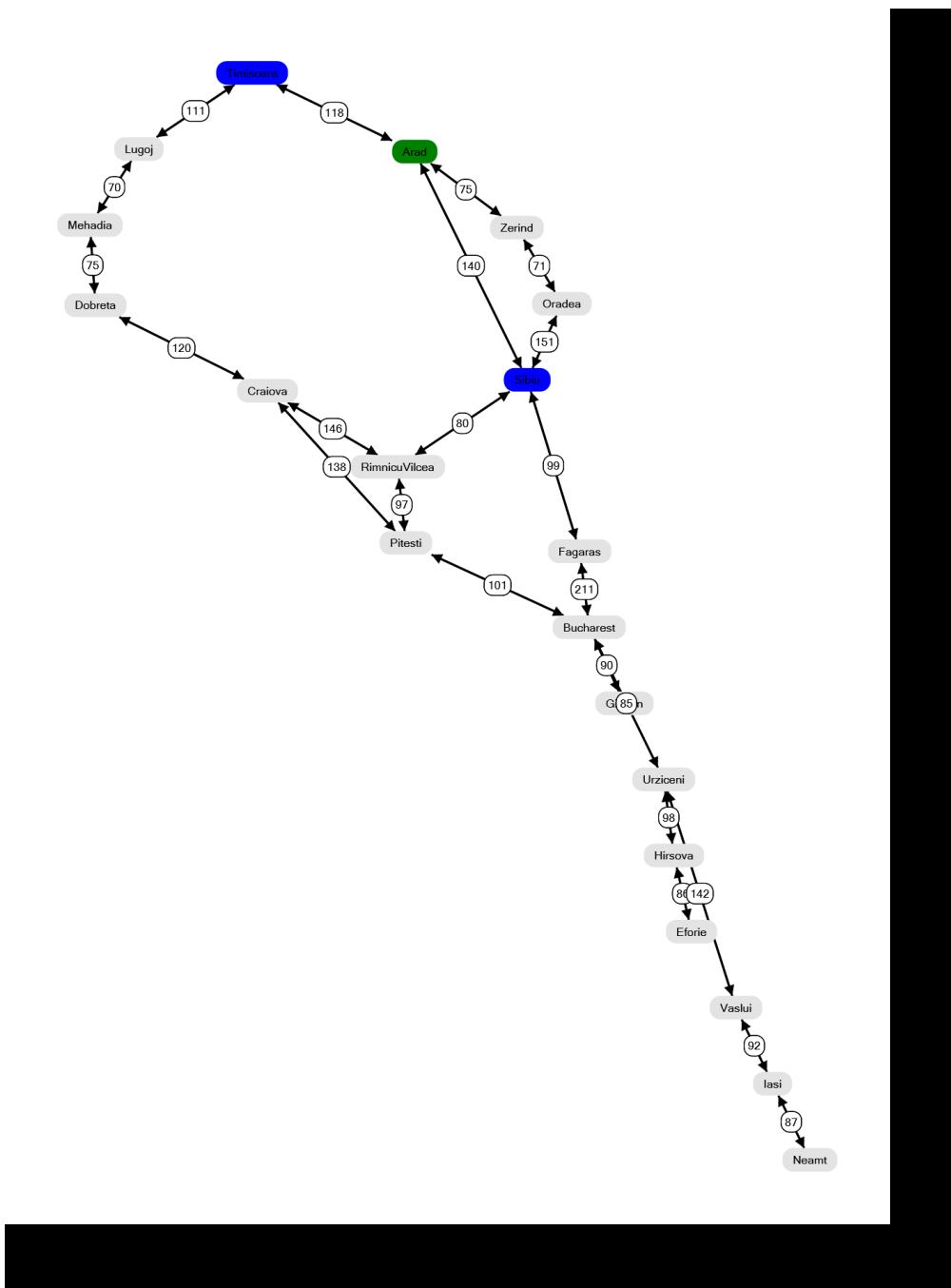
Visit

The algorithm has marked the node Timisoara as visited



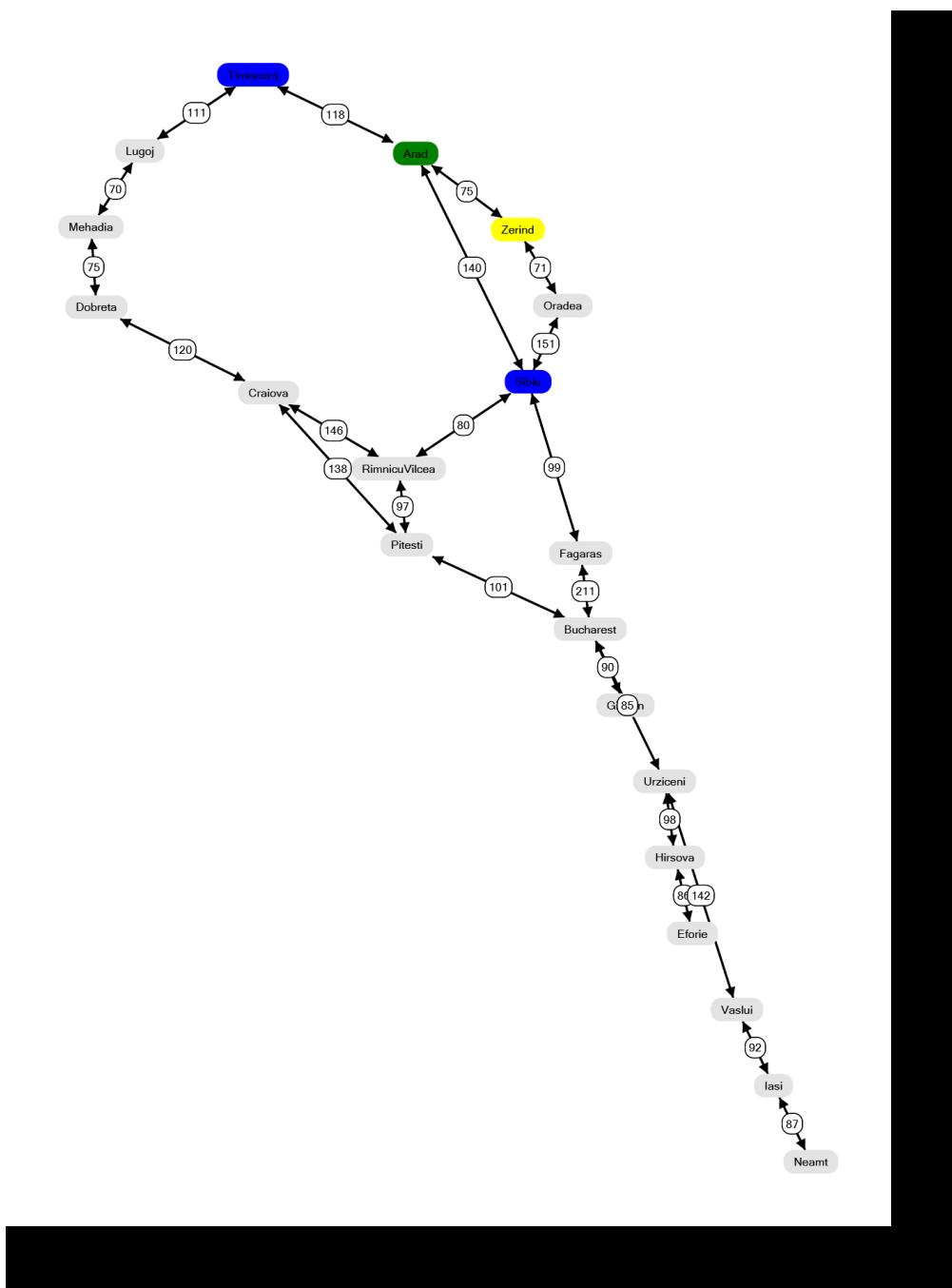
Post-Visit

The algorithm has closed the node Timisoara



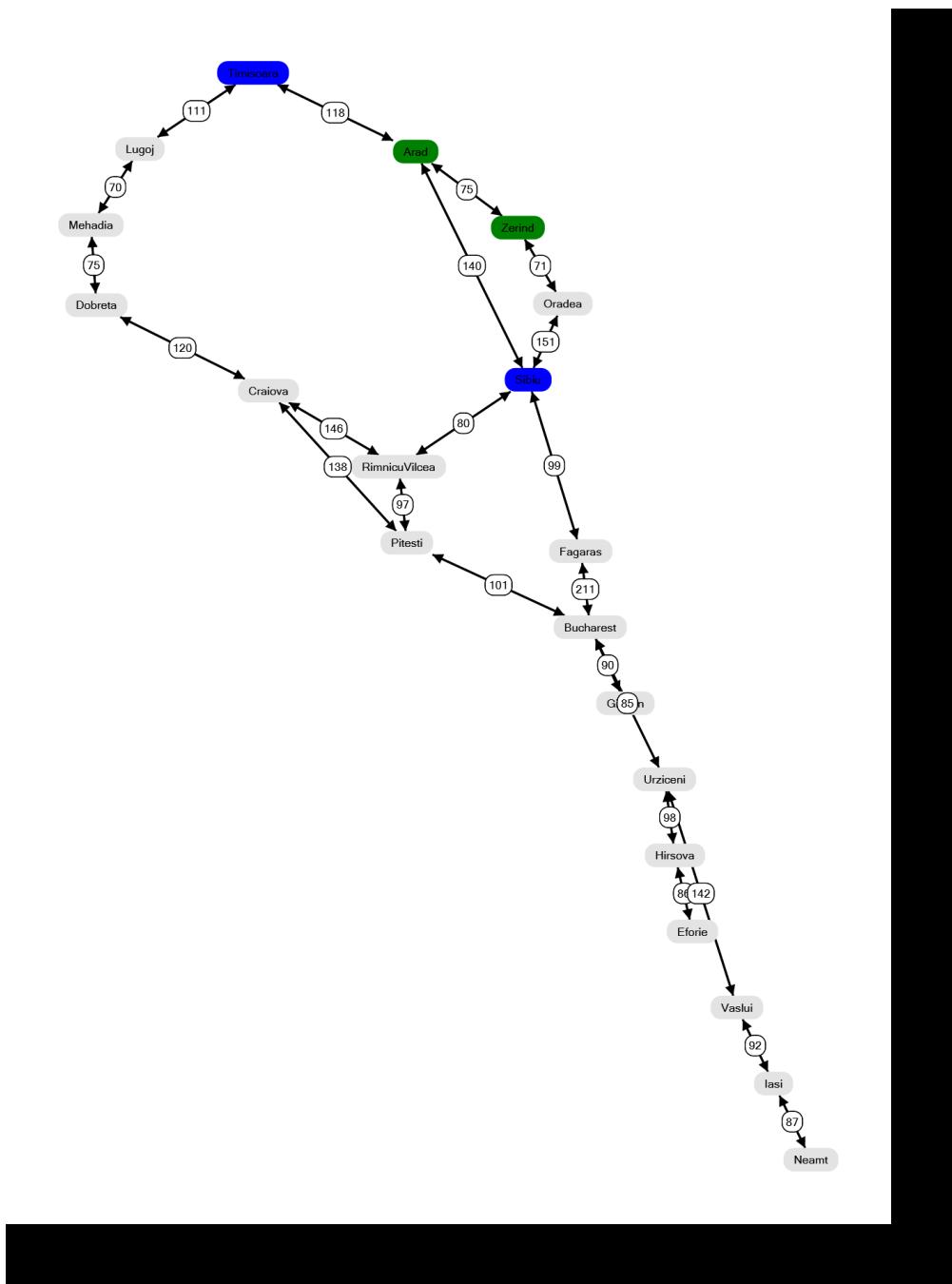
Pre-Visit

The algorithm has opened the node Zerind



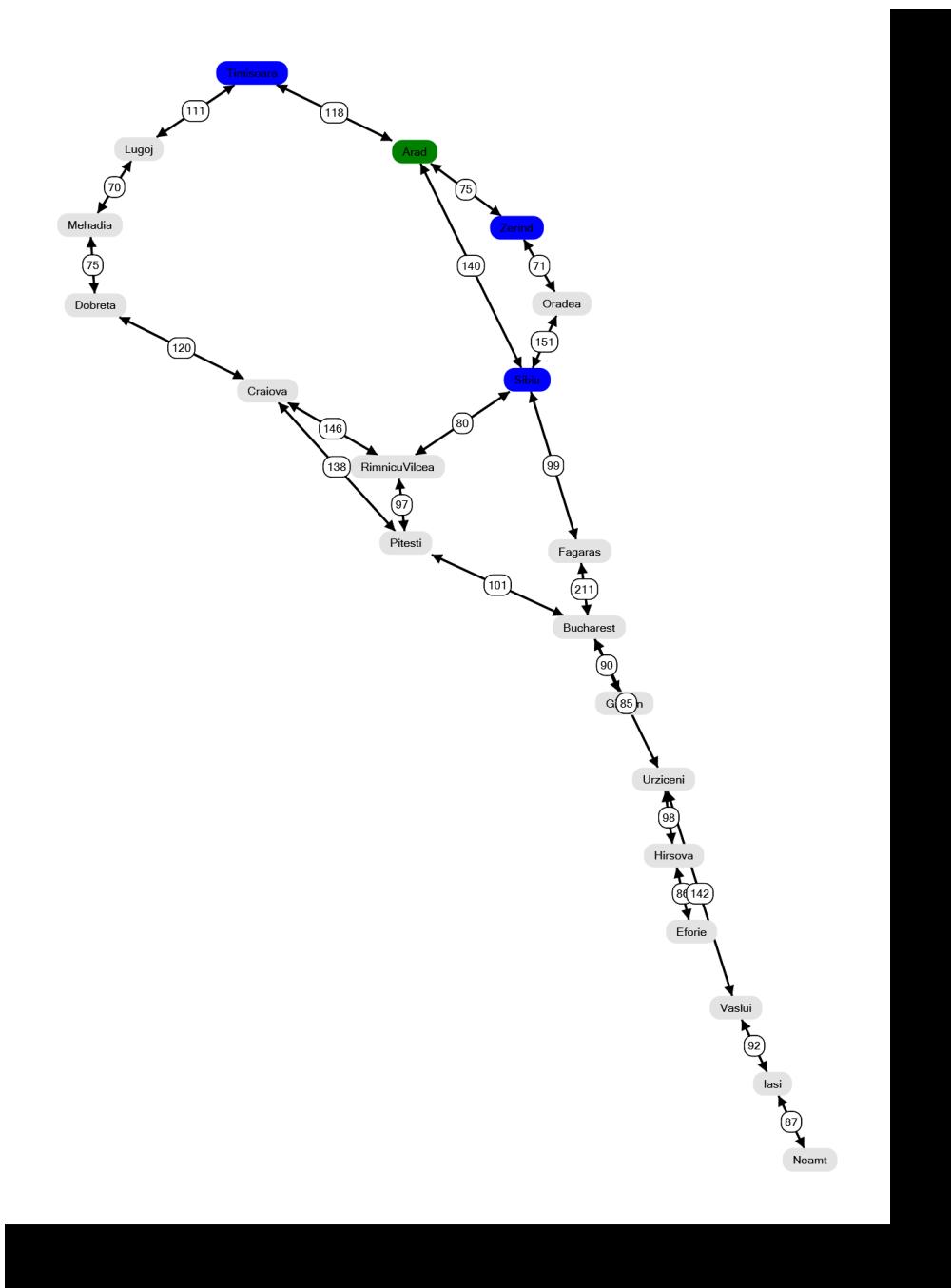
Visit

The algorithm has marked the node Zerind as visited



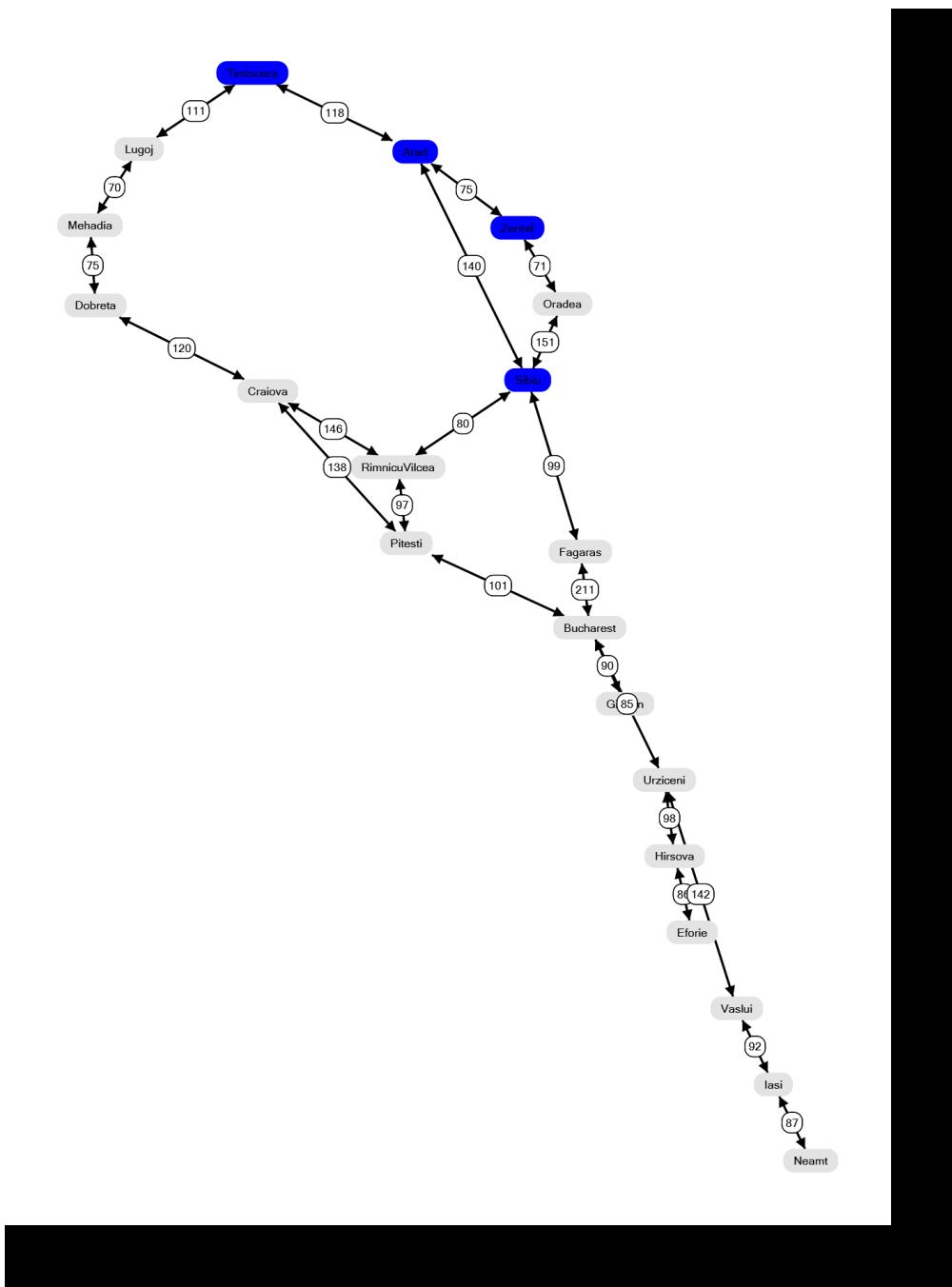
Post-Visit

The algorithm has closed the node Zerind



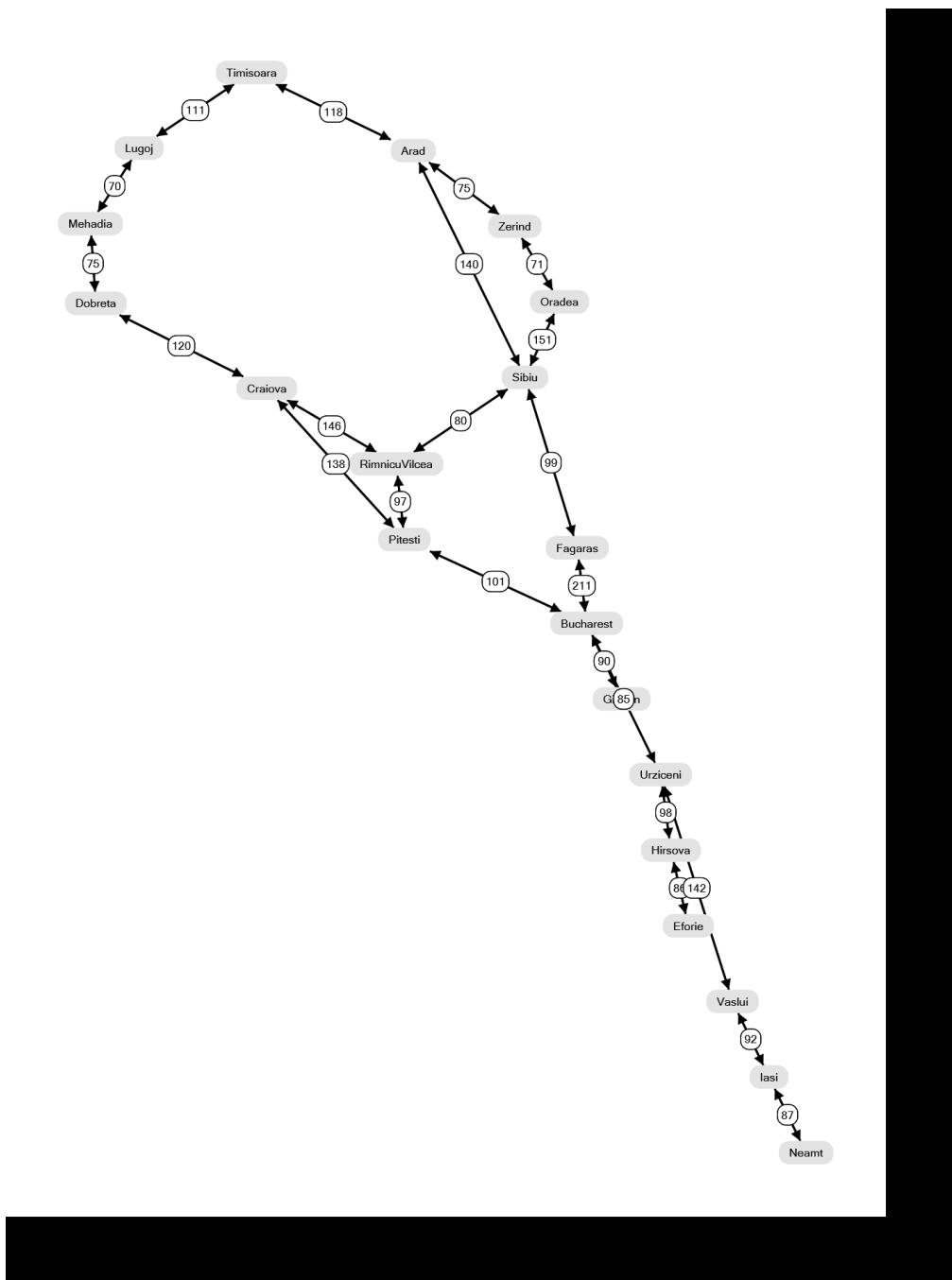
Post-Visit

The algorithm has closed the node Arad



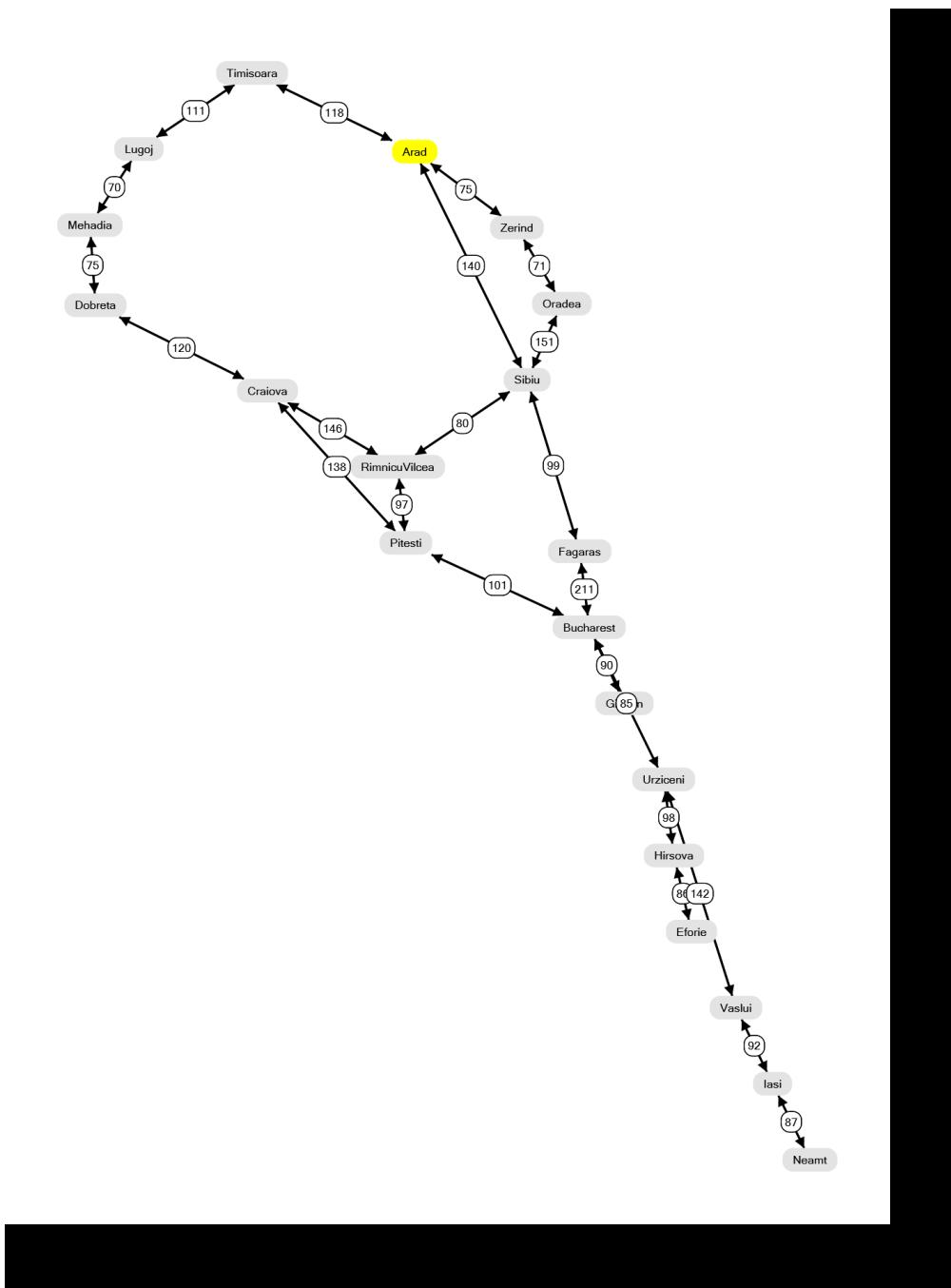
Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



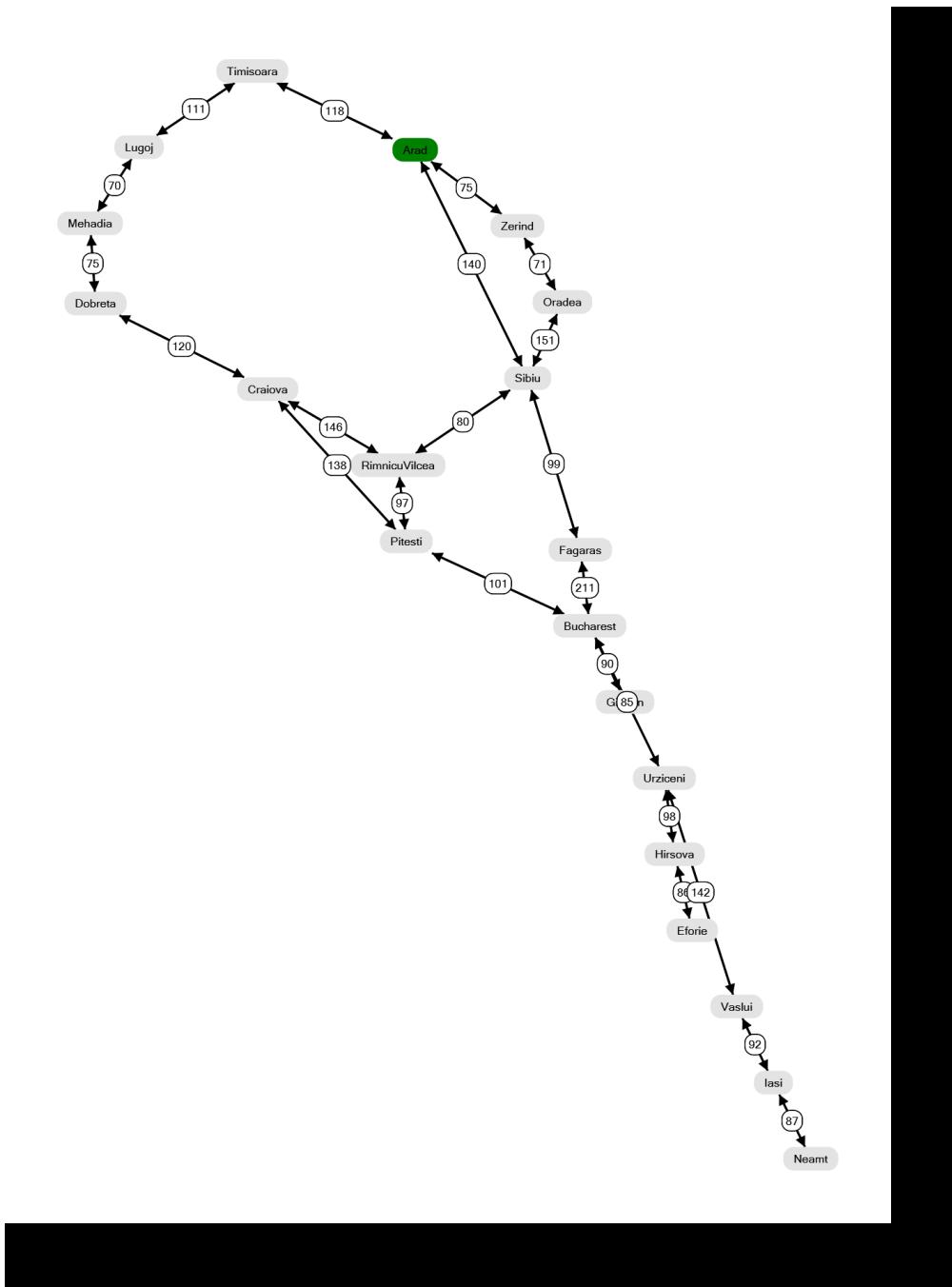
Pre-Visit

The algorithm has opened the node Arad



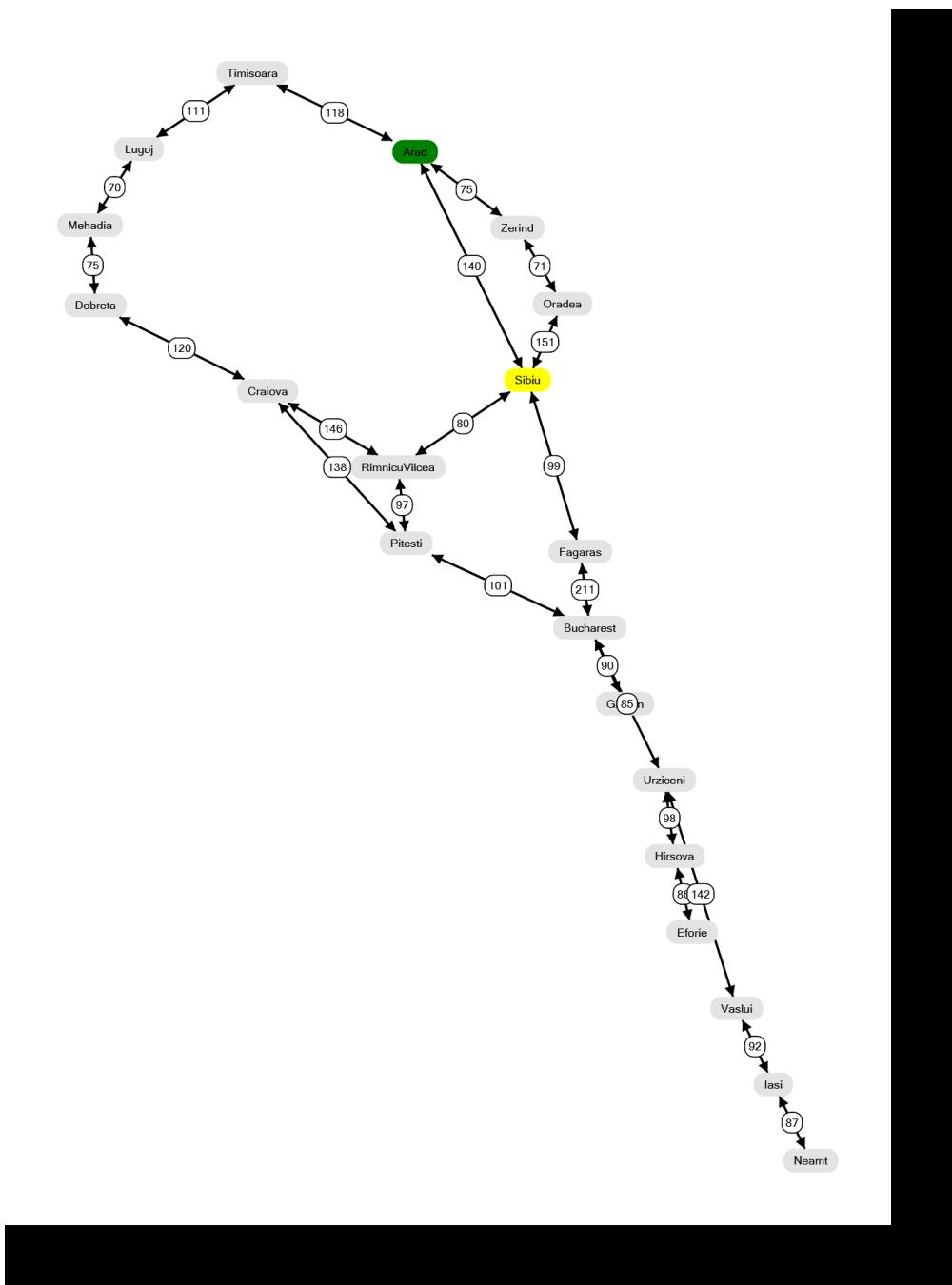
Visit

The algorithm has marked the node Arad as visited



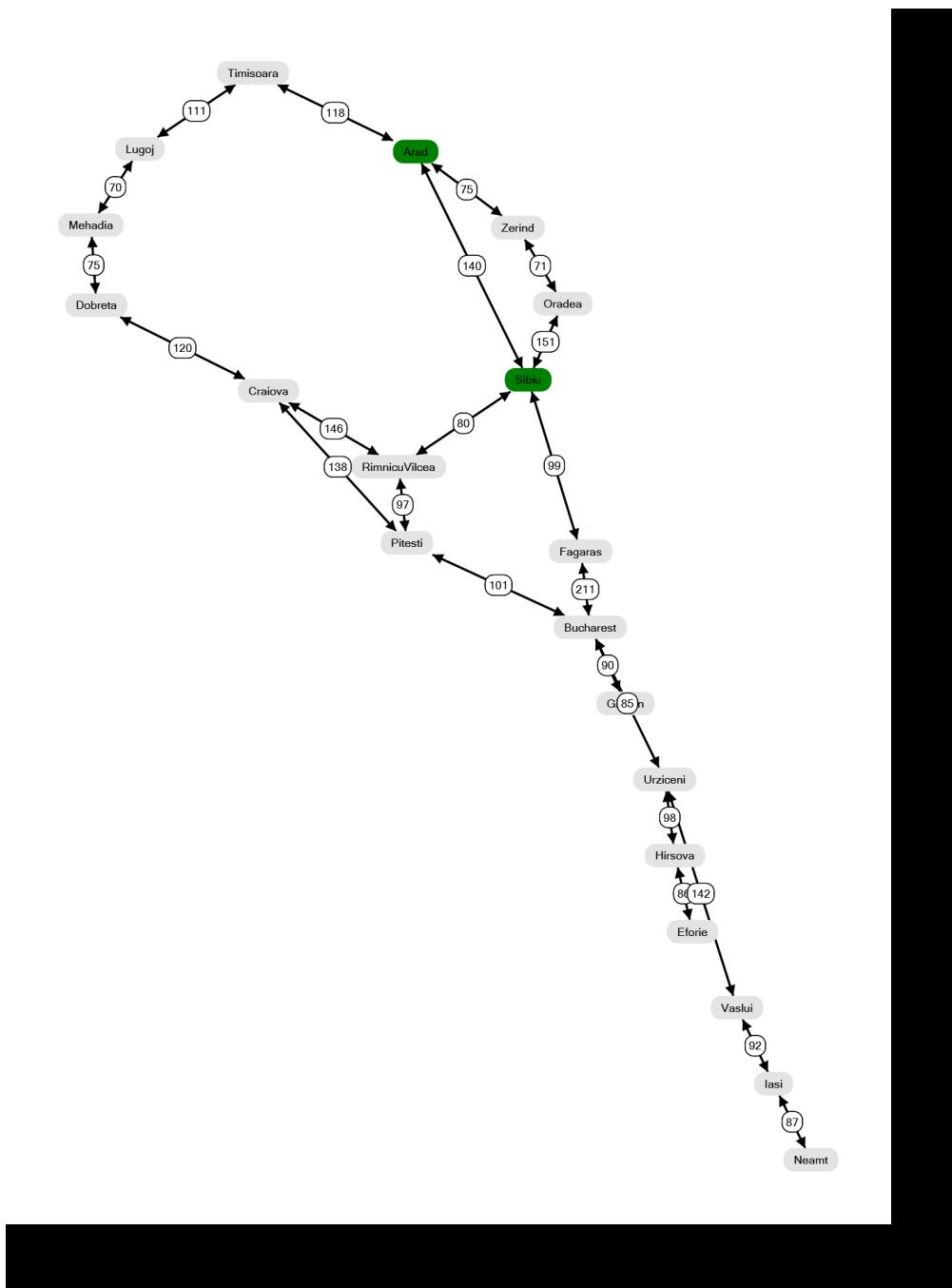
Pre-Visit

The algorithm has opened the node Sibiu



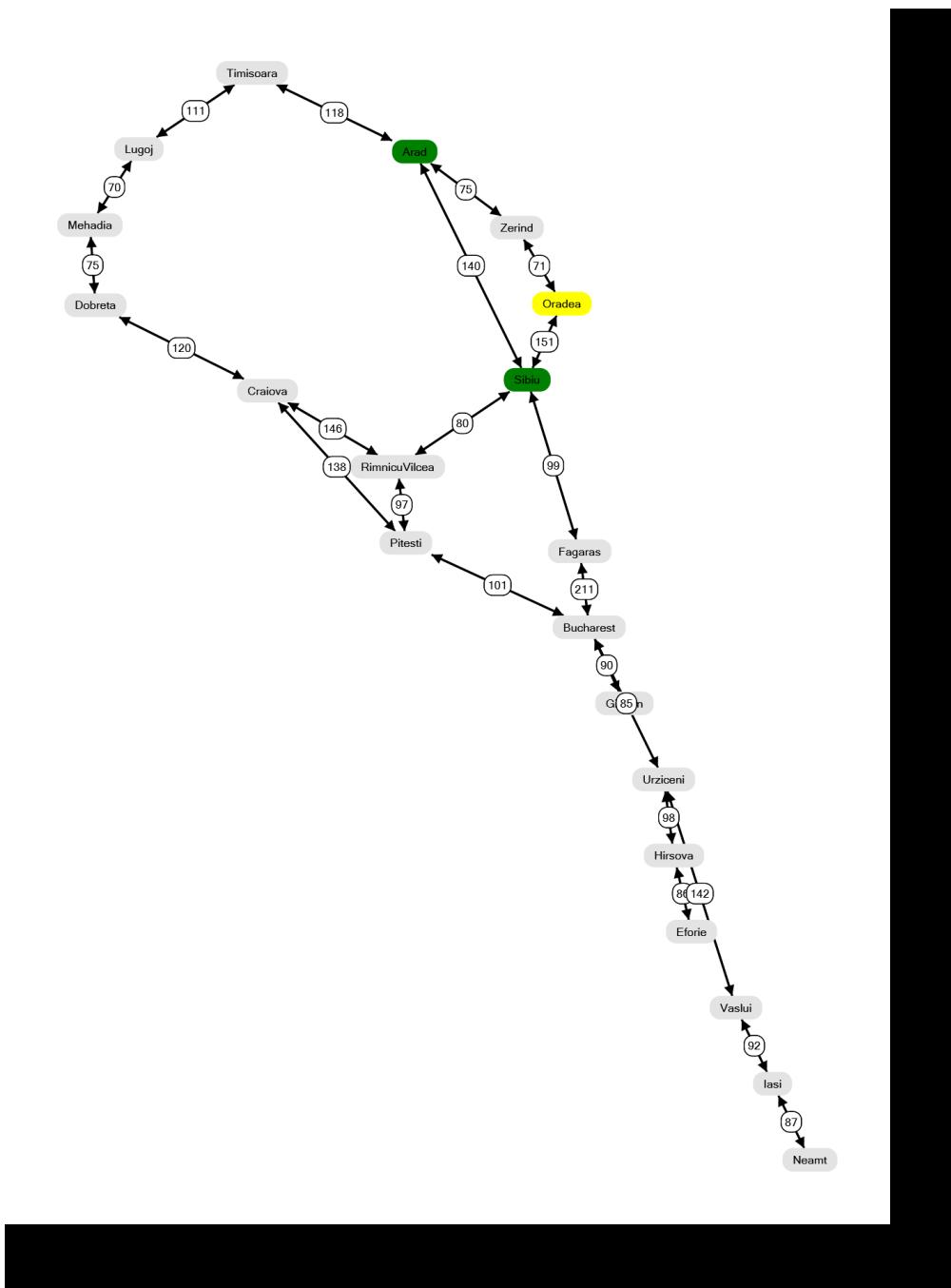
Visit

The algorithm has marked the node Sibiu as visited



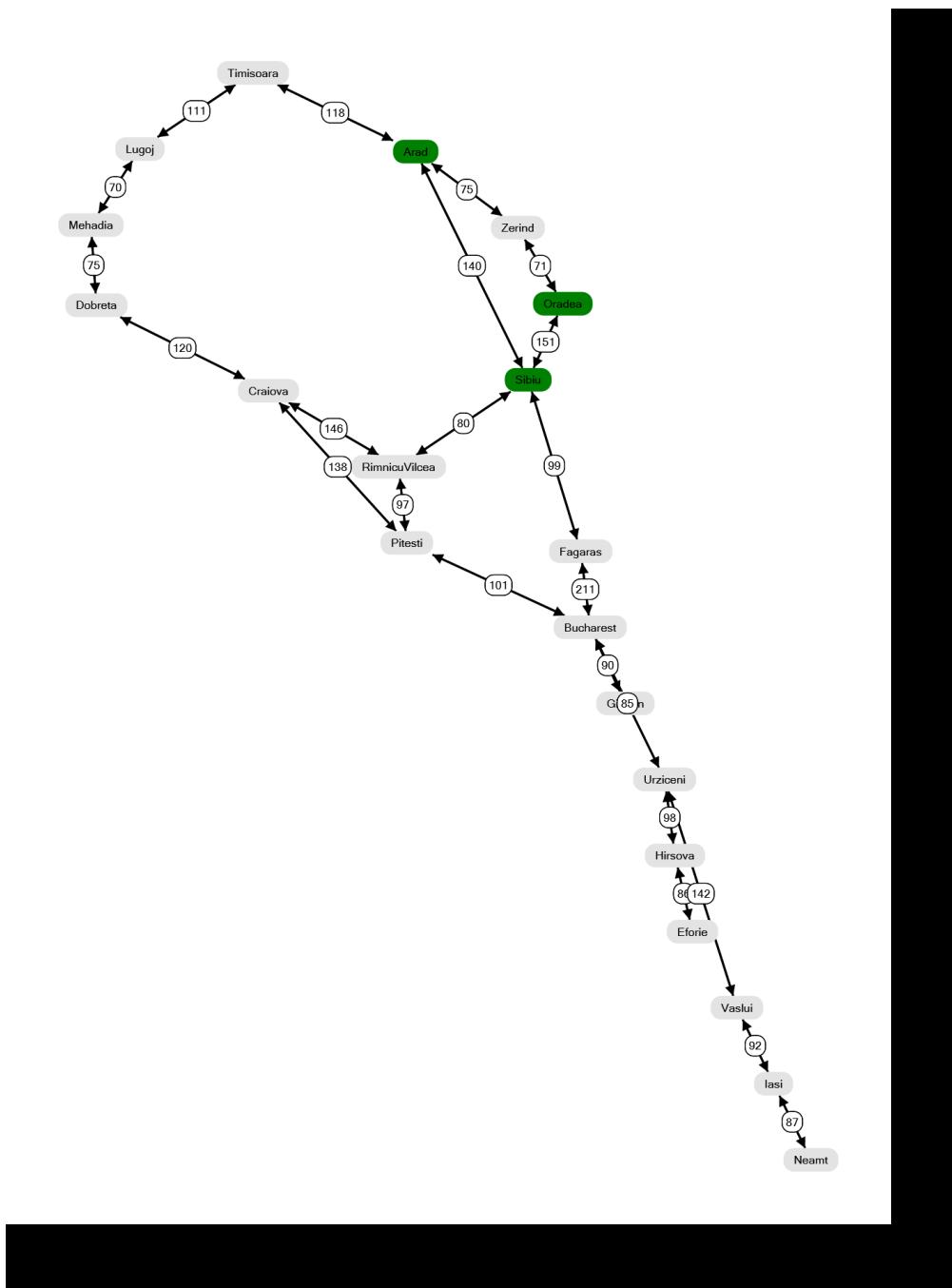
Pre-Visit

The algorithm has opened the node Oradea



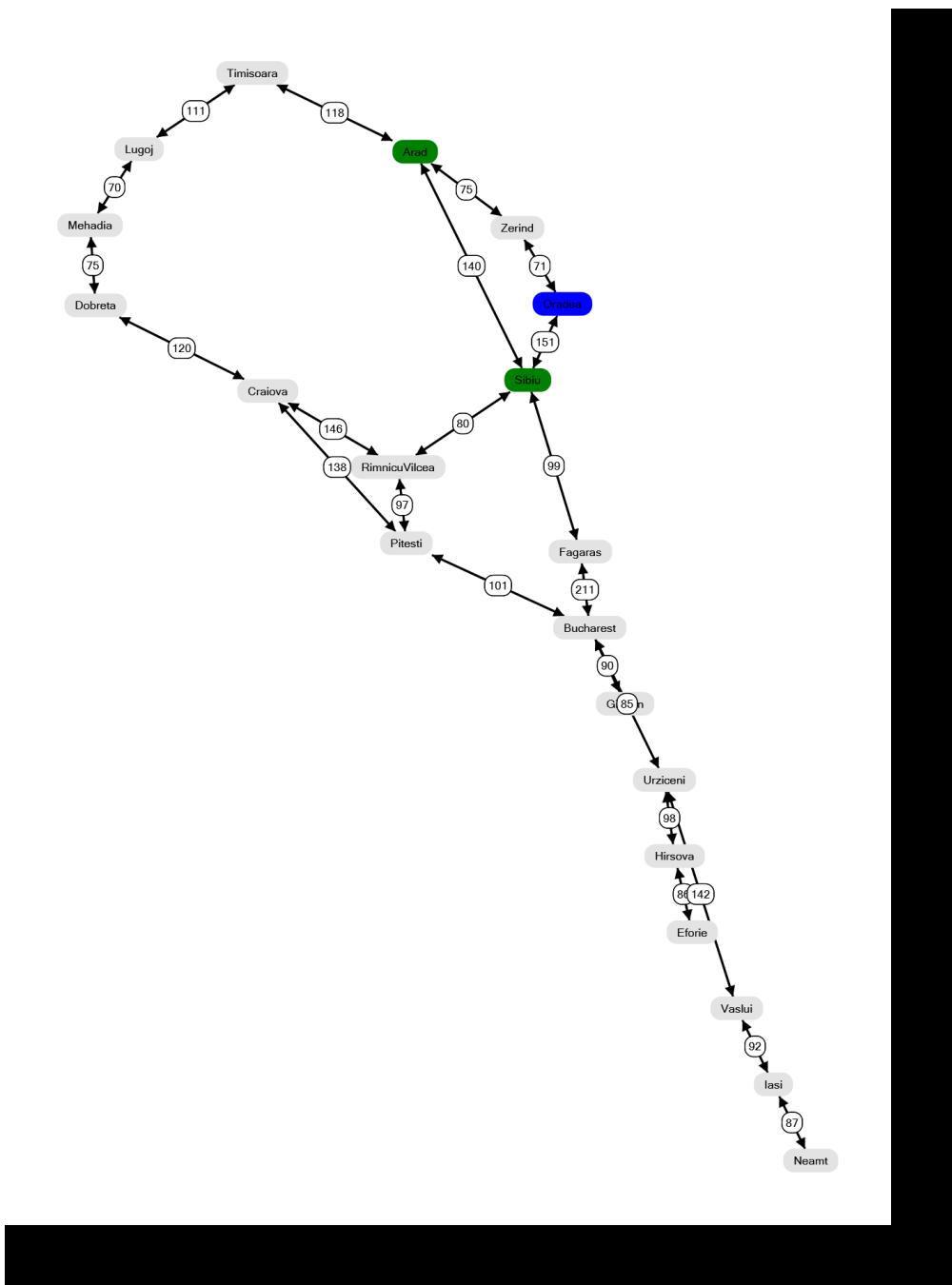
Visit

The algorithm has marked the node Oradea as visited



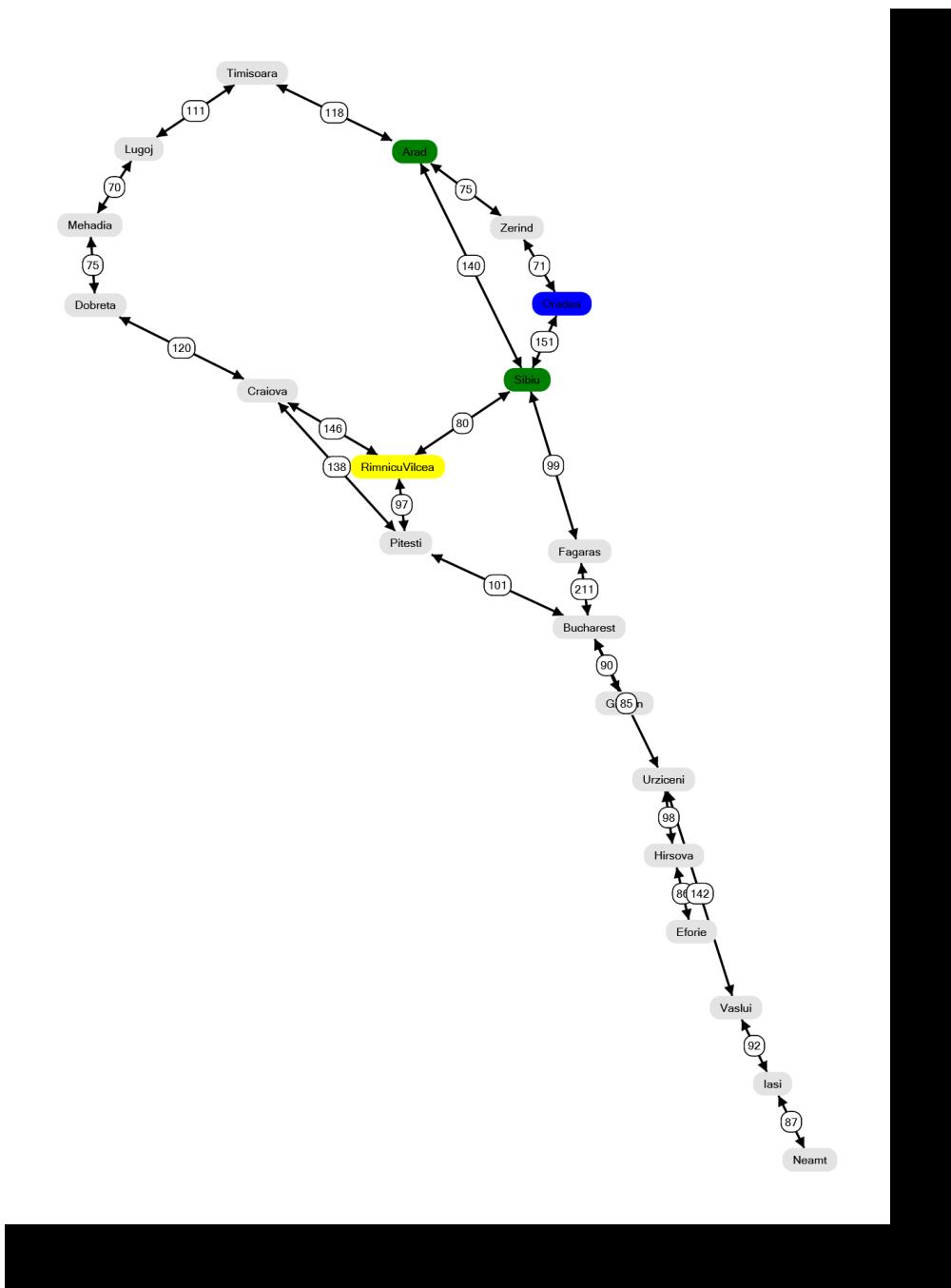
Post-Visit

The algorithm has closed the node Oradea



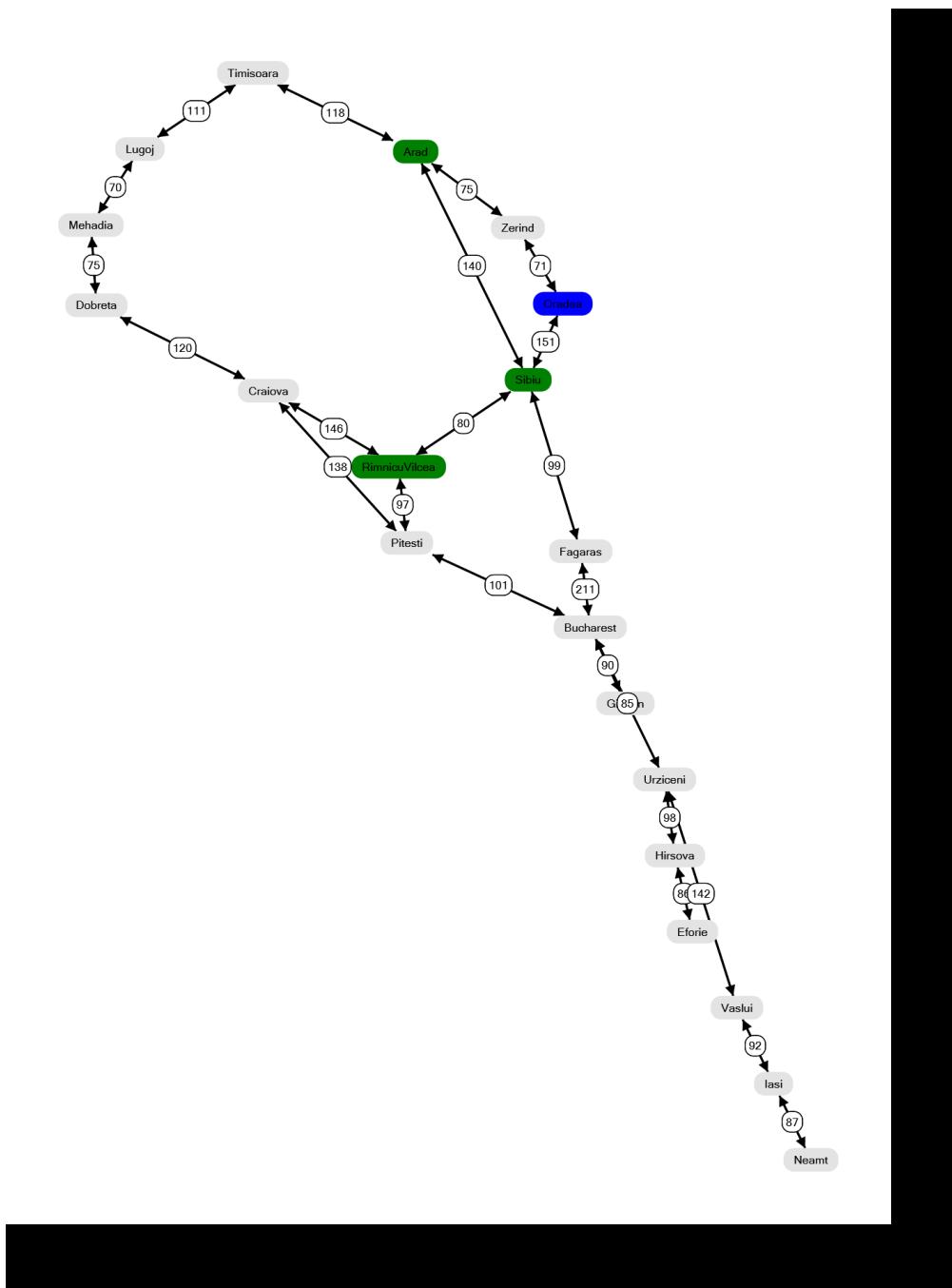
Pre-Visit

The algorithm has opened the node Rimnicu Vilcea



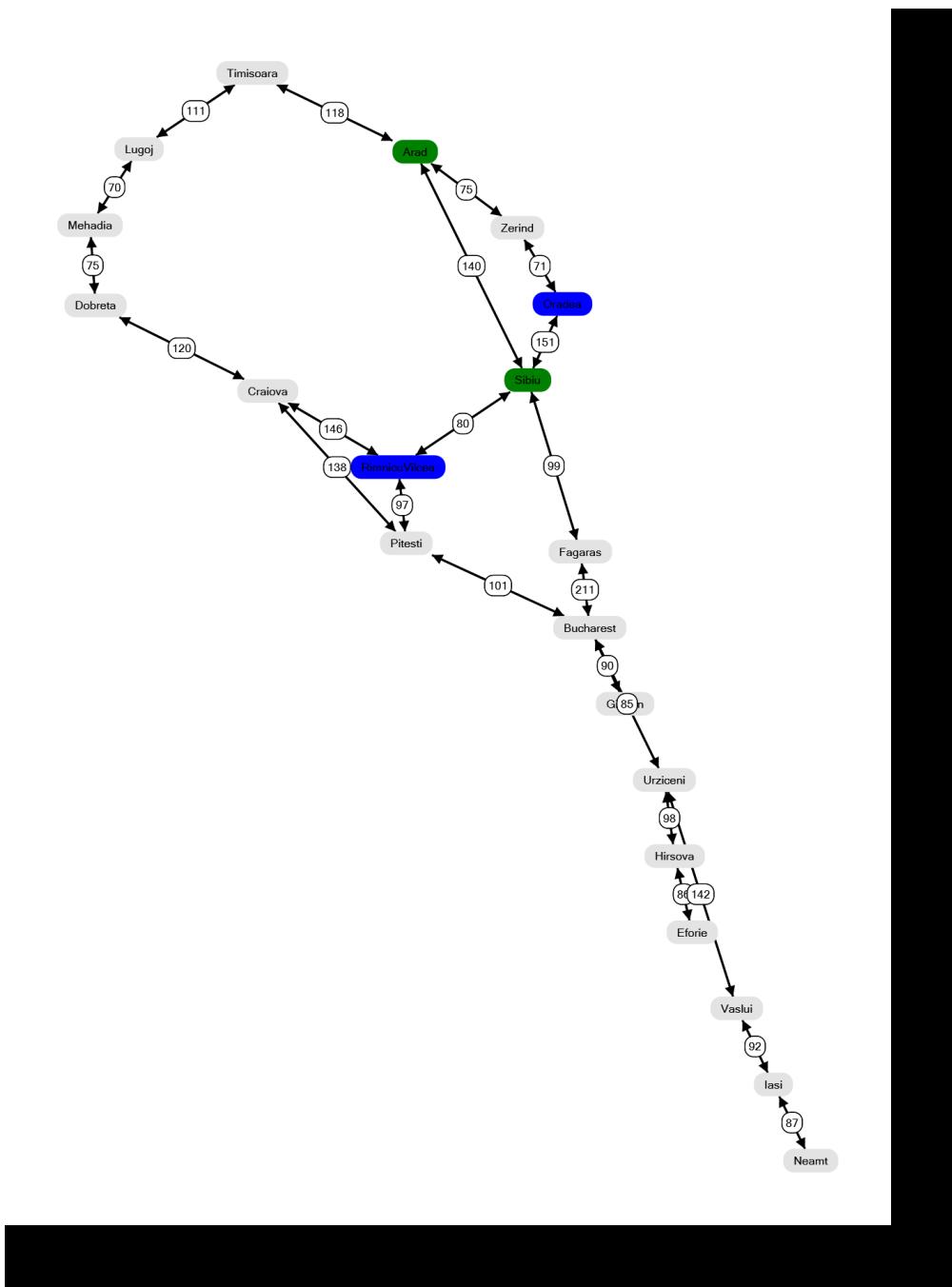
Visit

The algorithm has marked the node Rimnicu Vilcea as visited



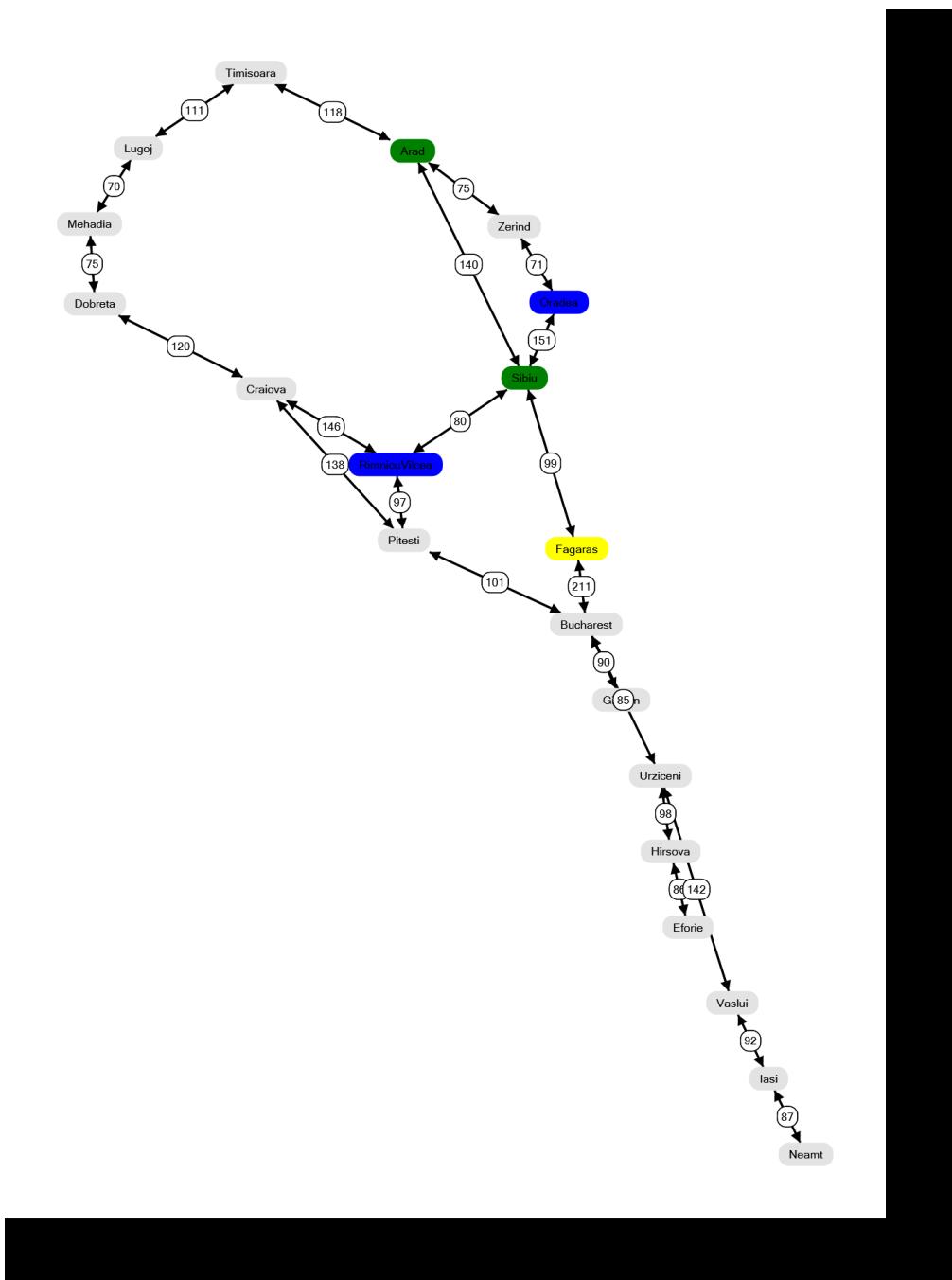
Post-Visit

The algorithm has closed the node Rimnicu Vilcea



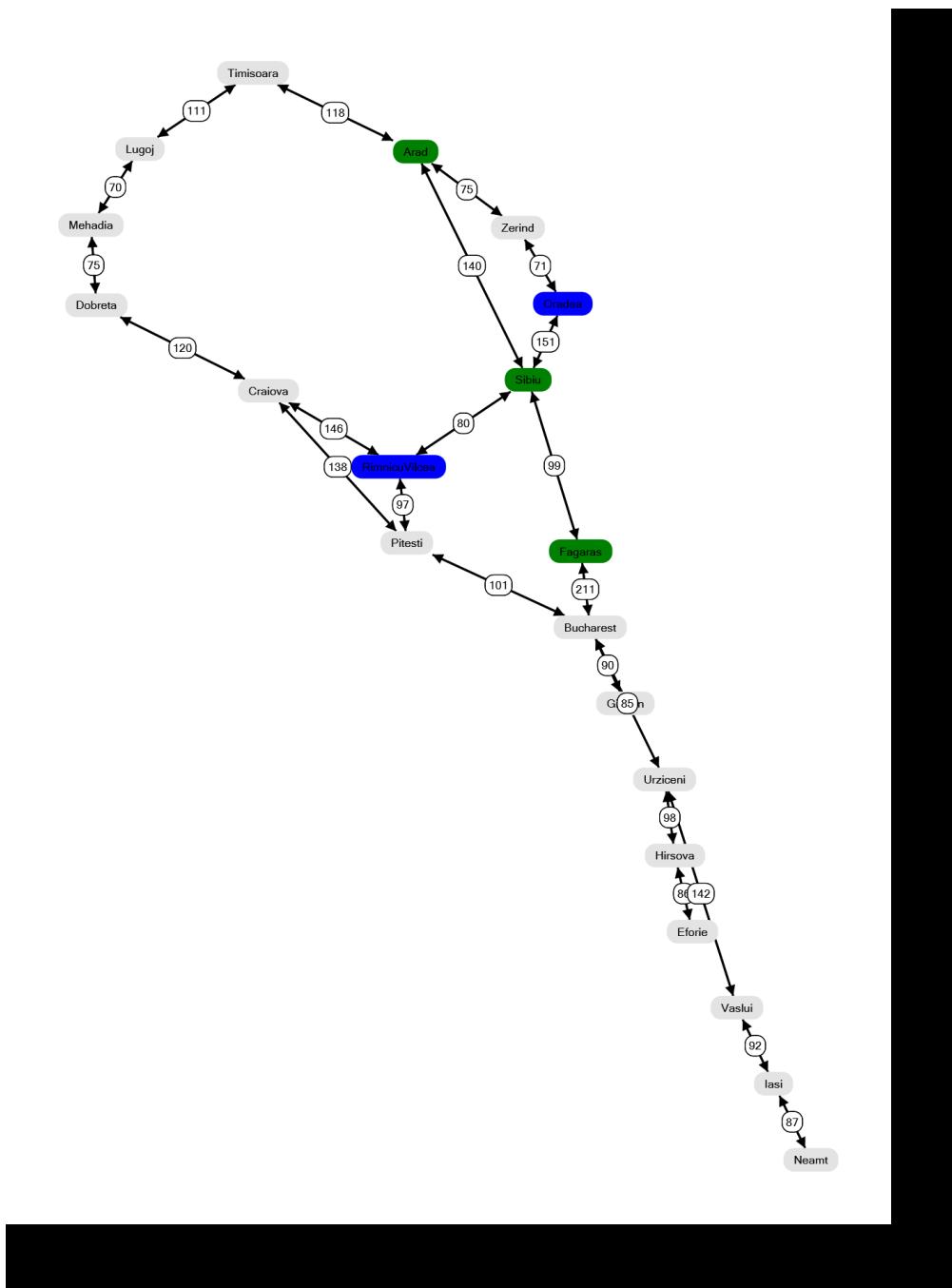
Pre-Visit

The algorithm has opened the node Fagaras



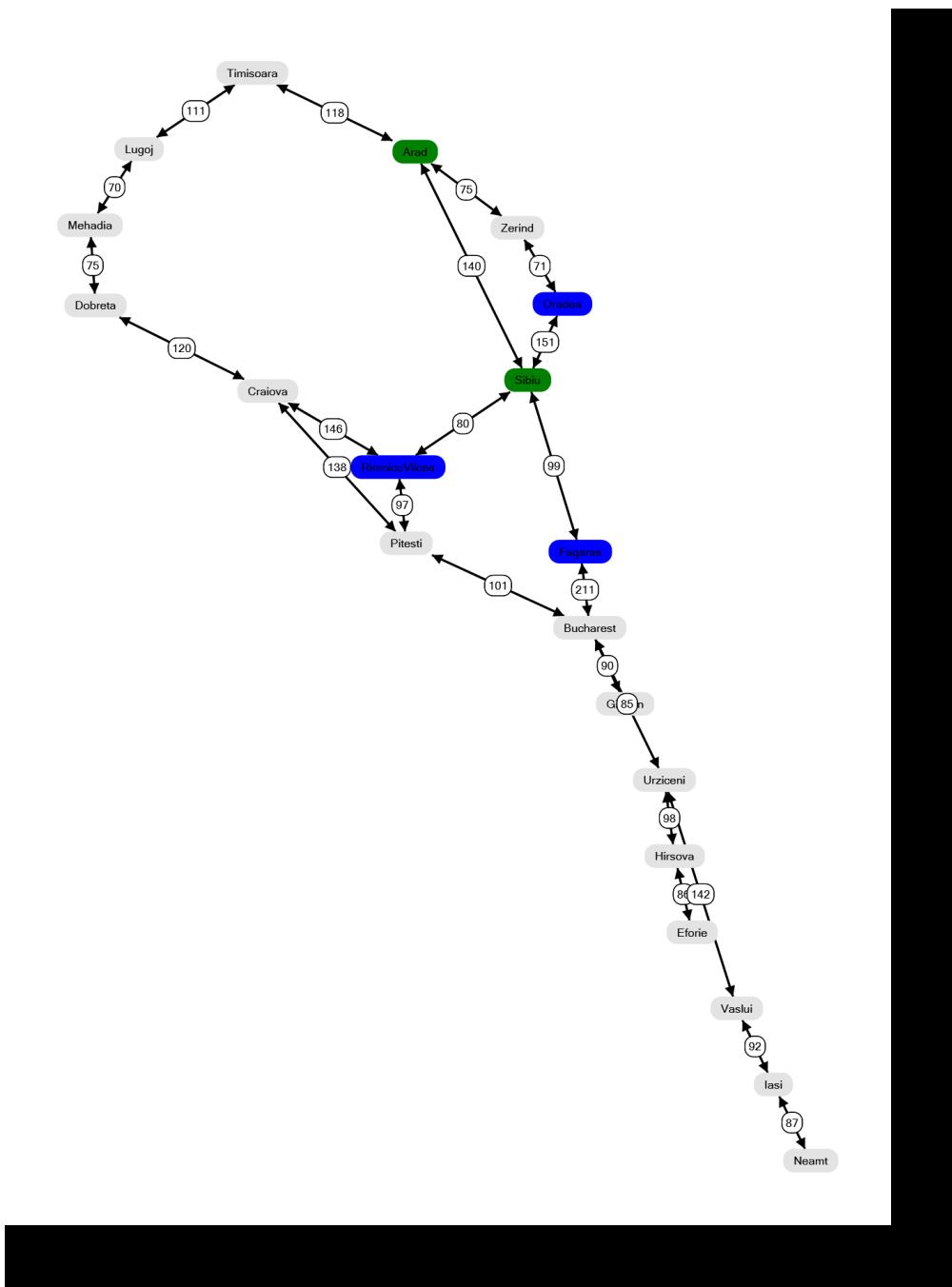
Visit

The algorithm has marked the node Fagaras as visited



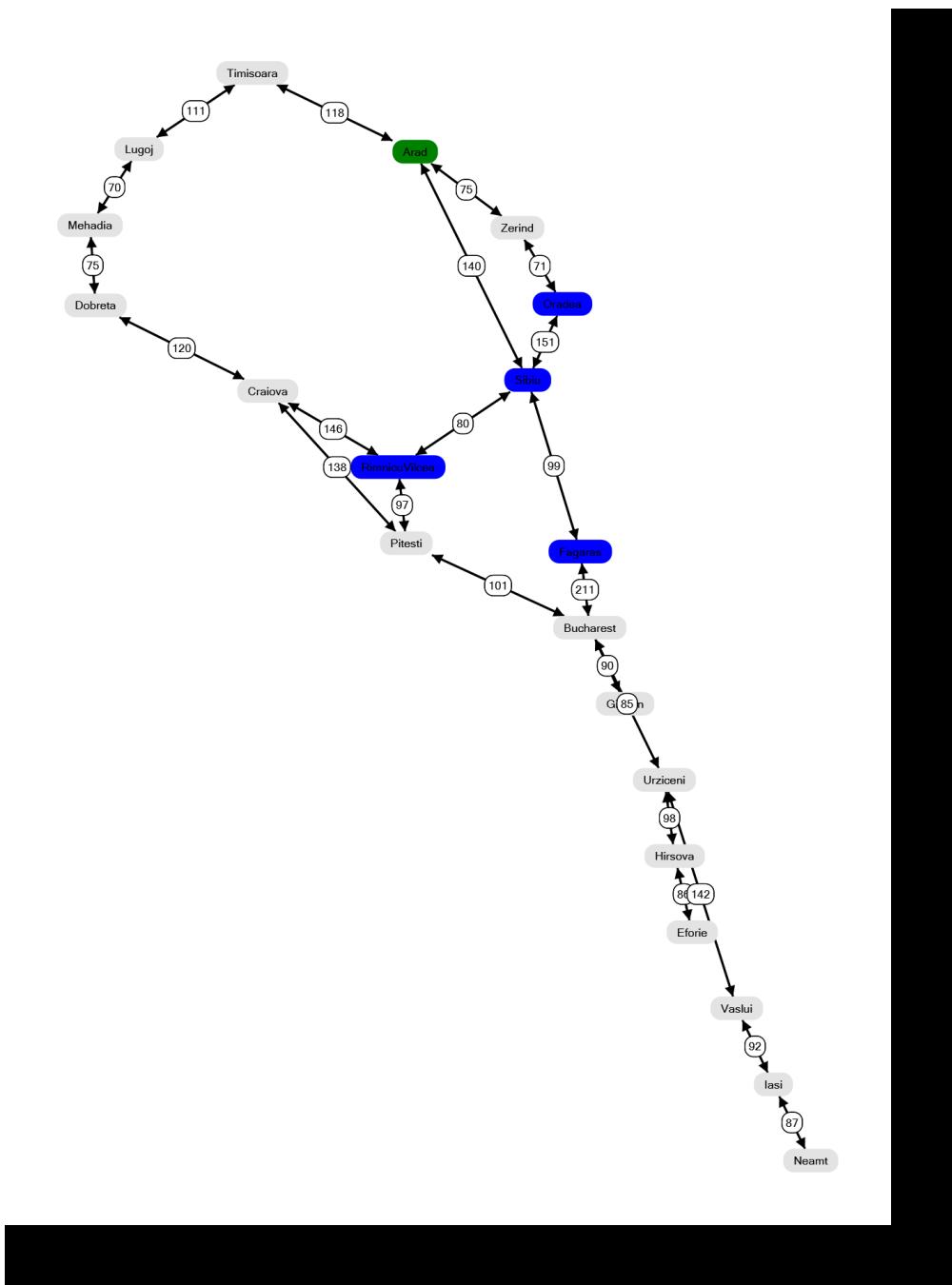
Post-Visit

The algorithm has closed the node Fagaras



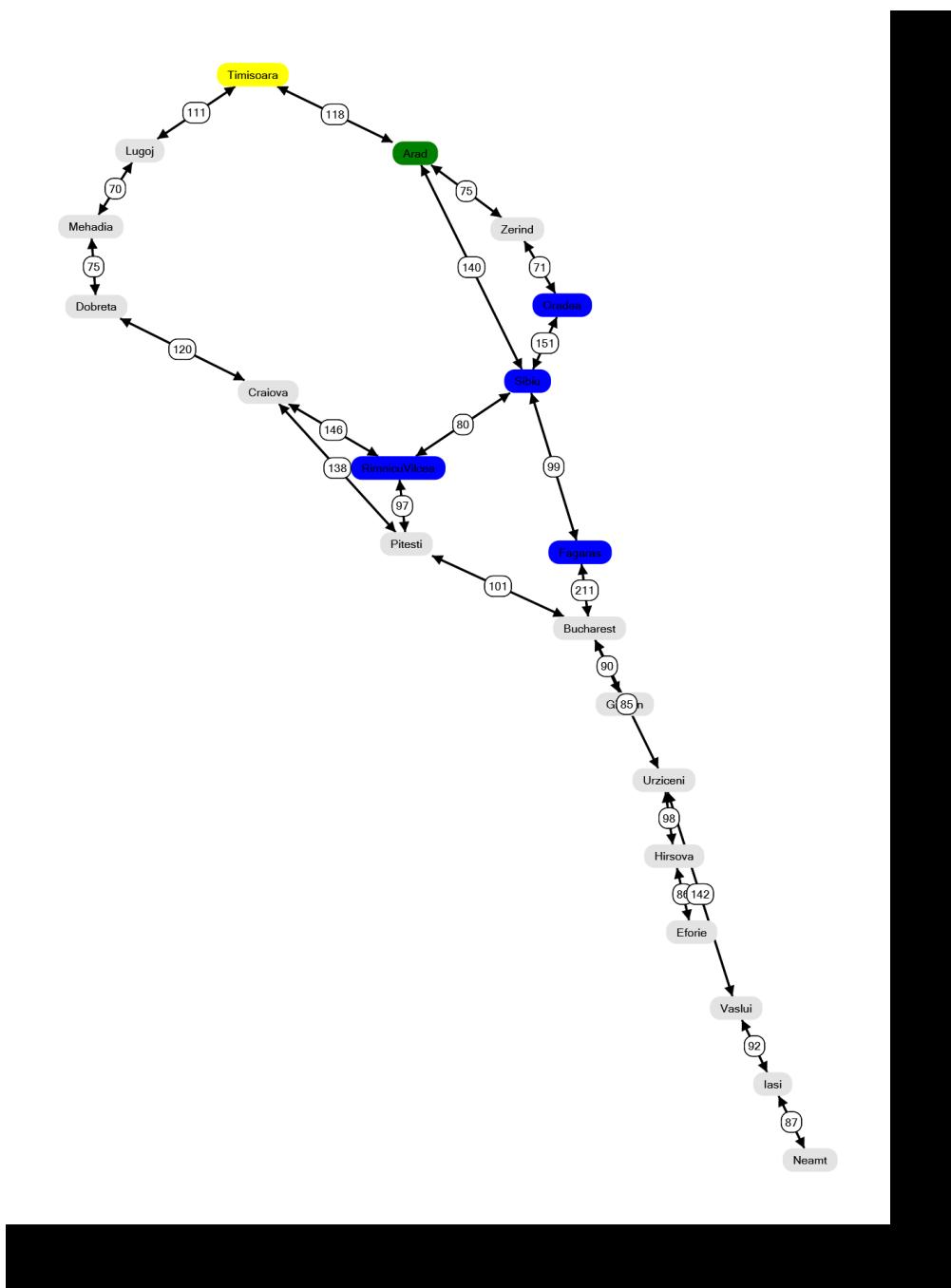
Post-Visit

The algorithm has closed the node Sibiu



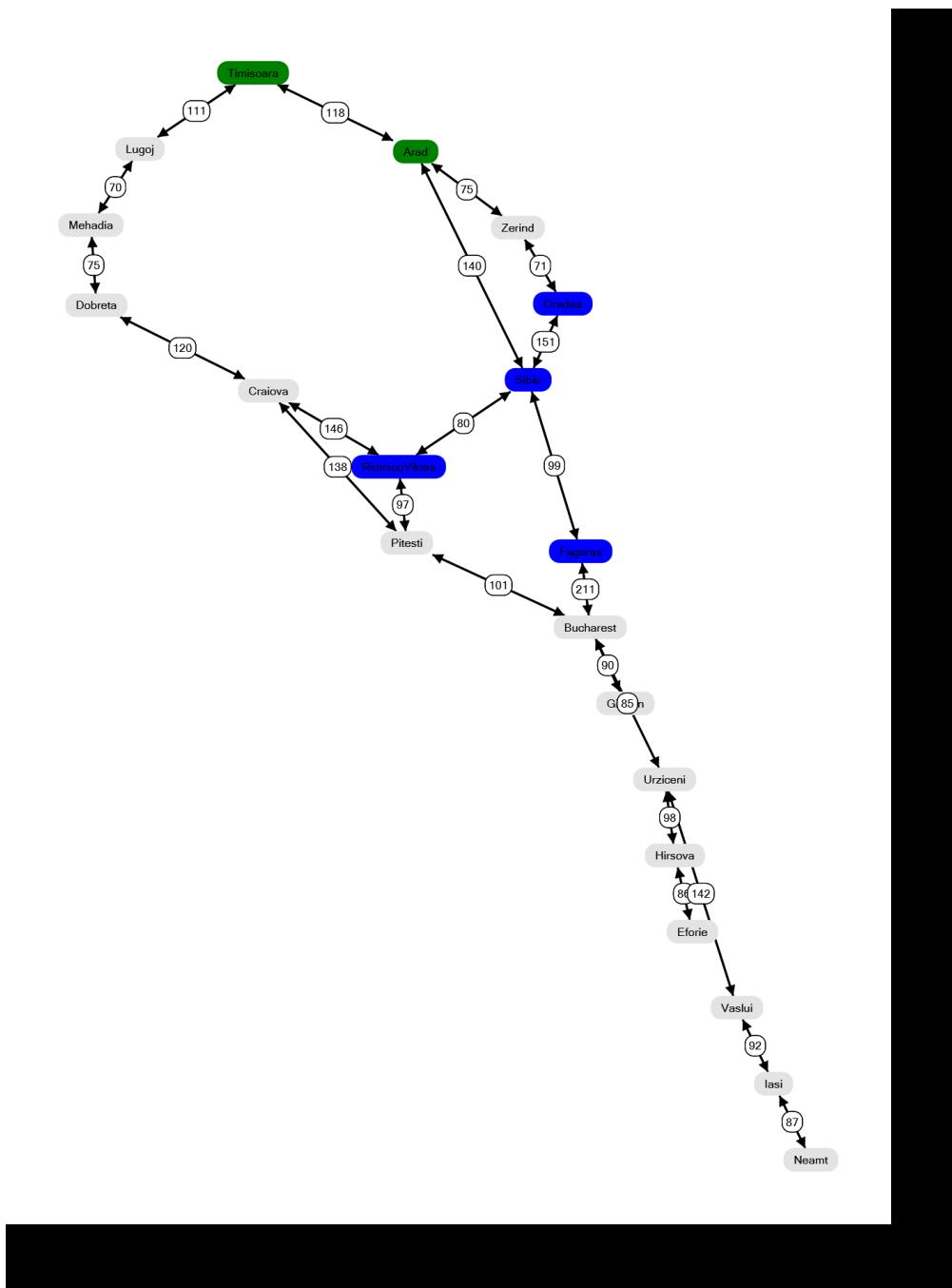
Pre-Visit

The algorithm has opened the node Timisoara



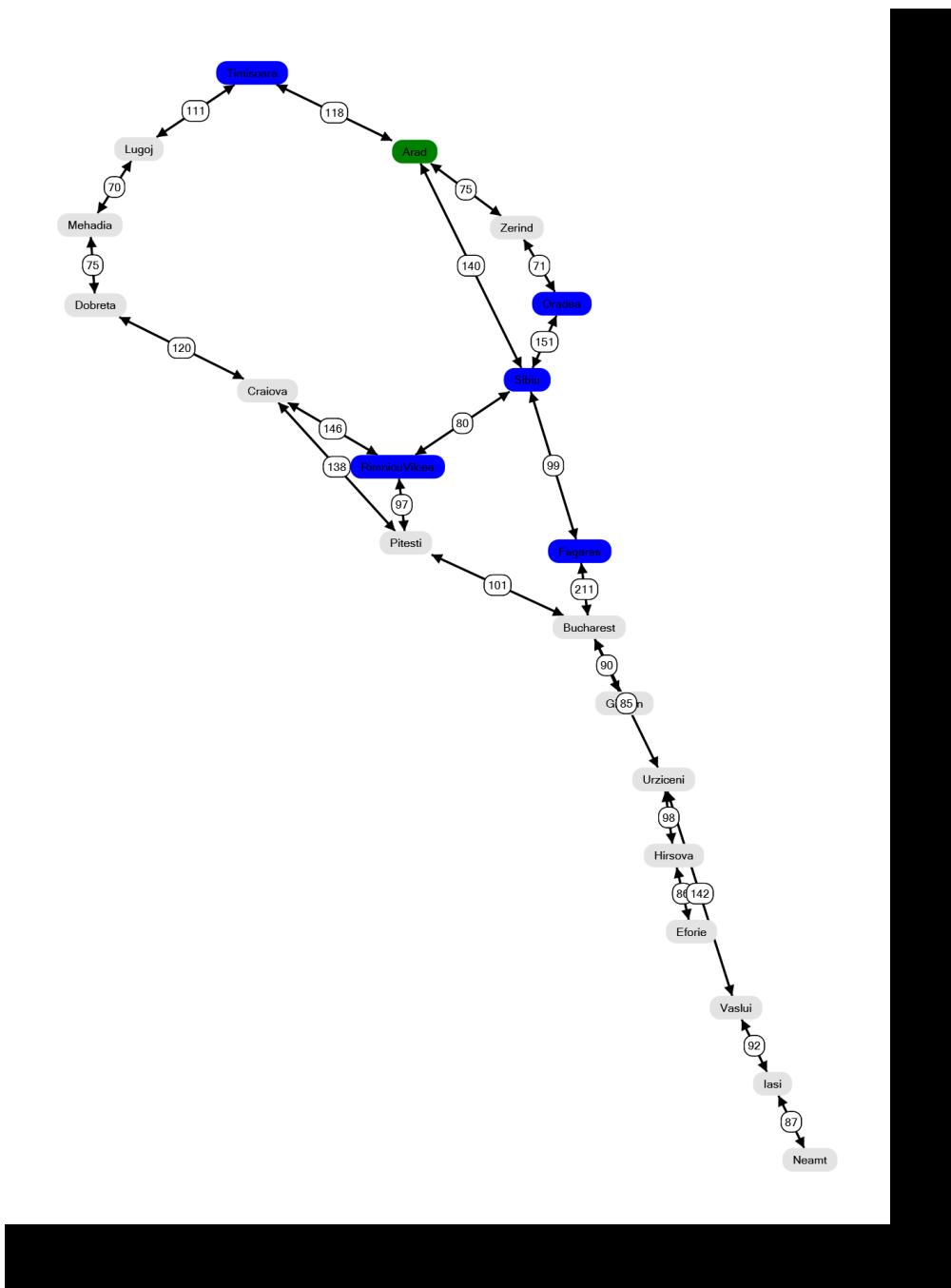
Visit

The algorithm has marked the node Timisoara as visited



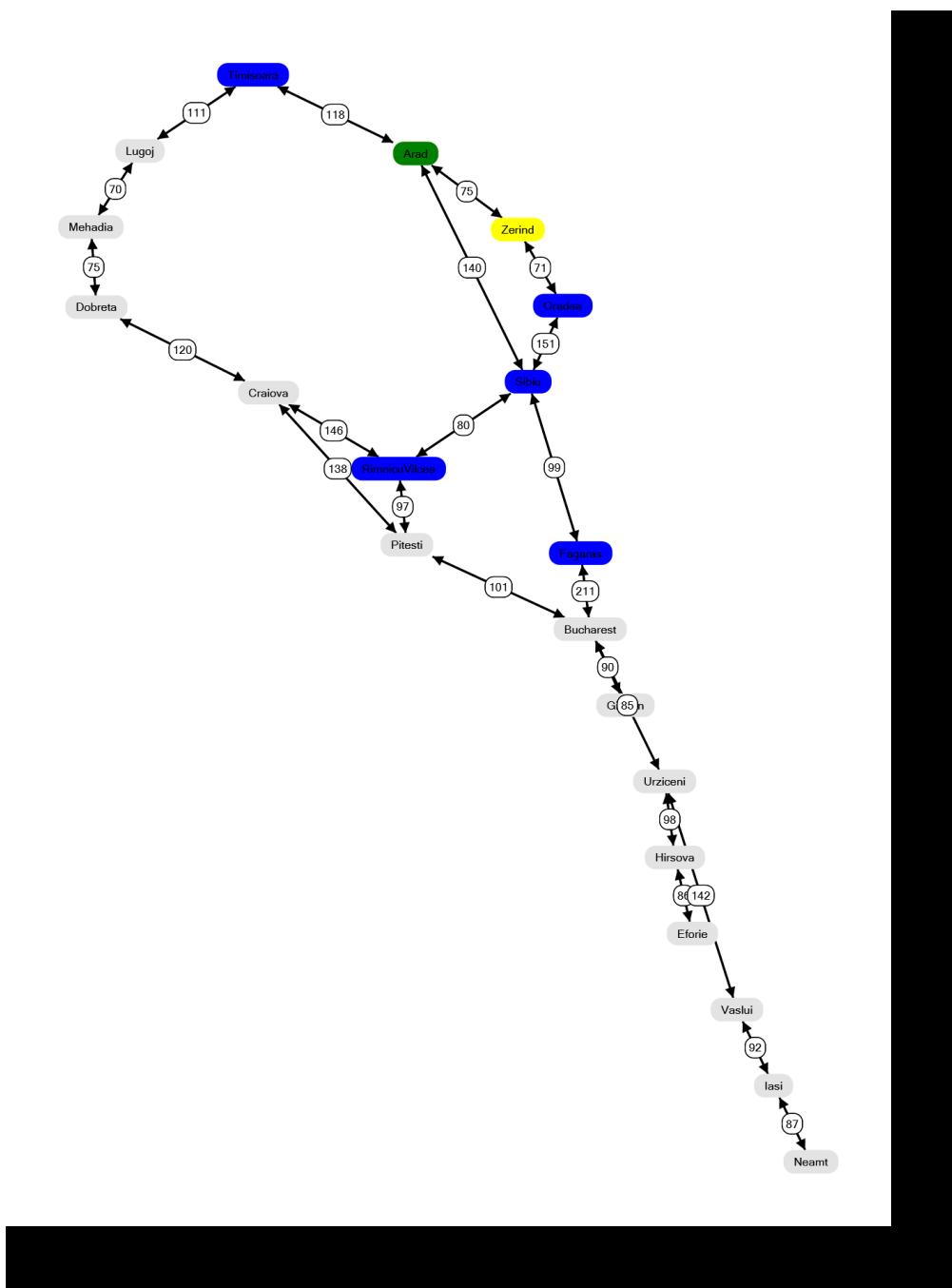
Post-Visit

The algorithm has closed the node Timisoara



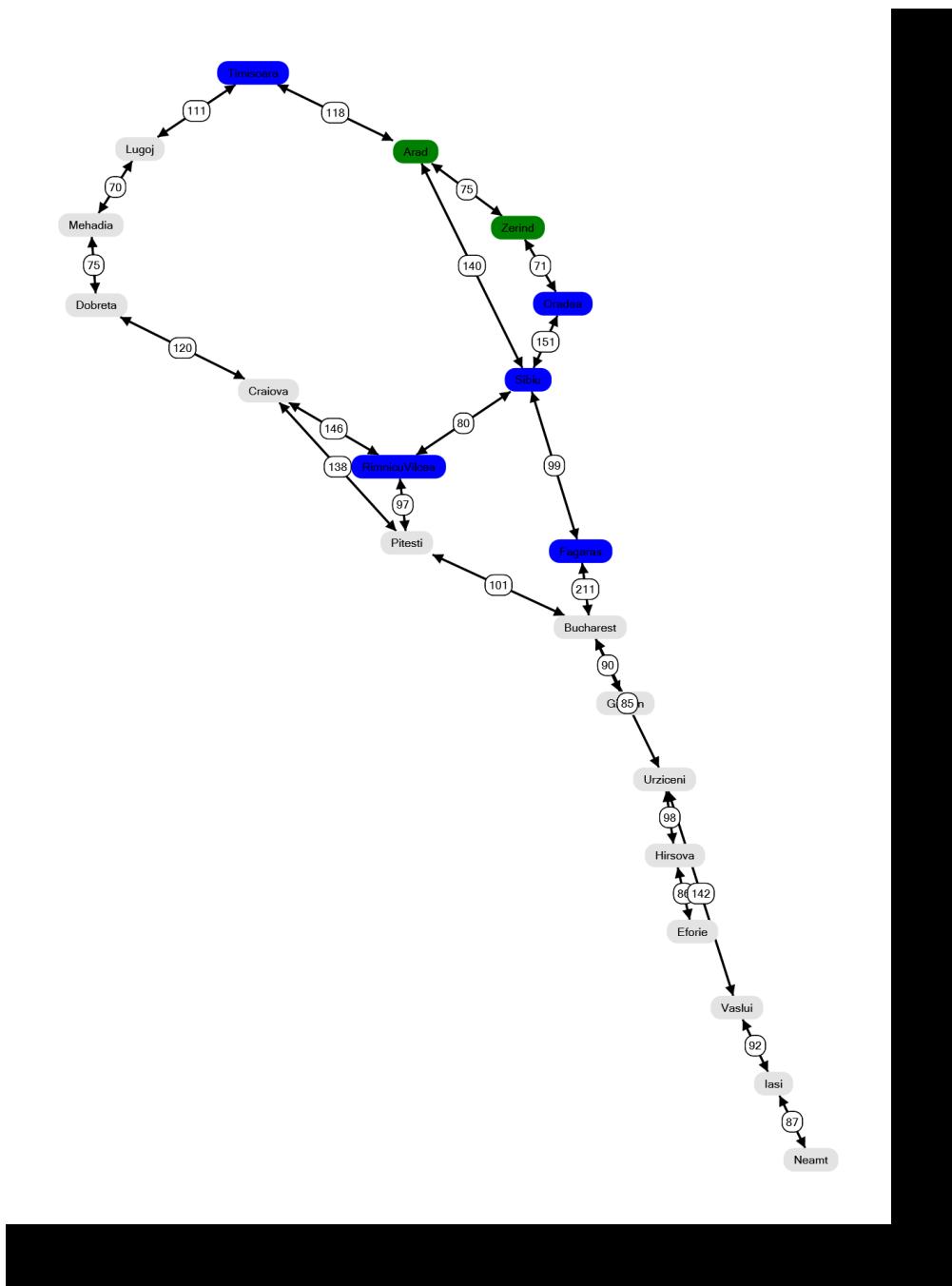
Pre-Visit

The algorithm has opened the node Zerind



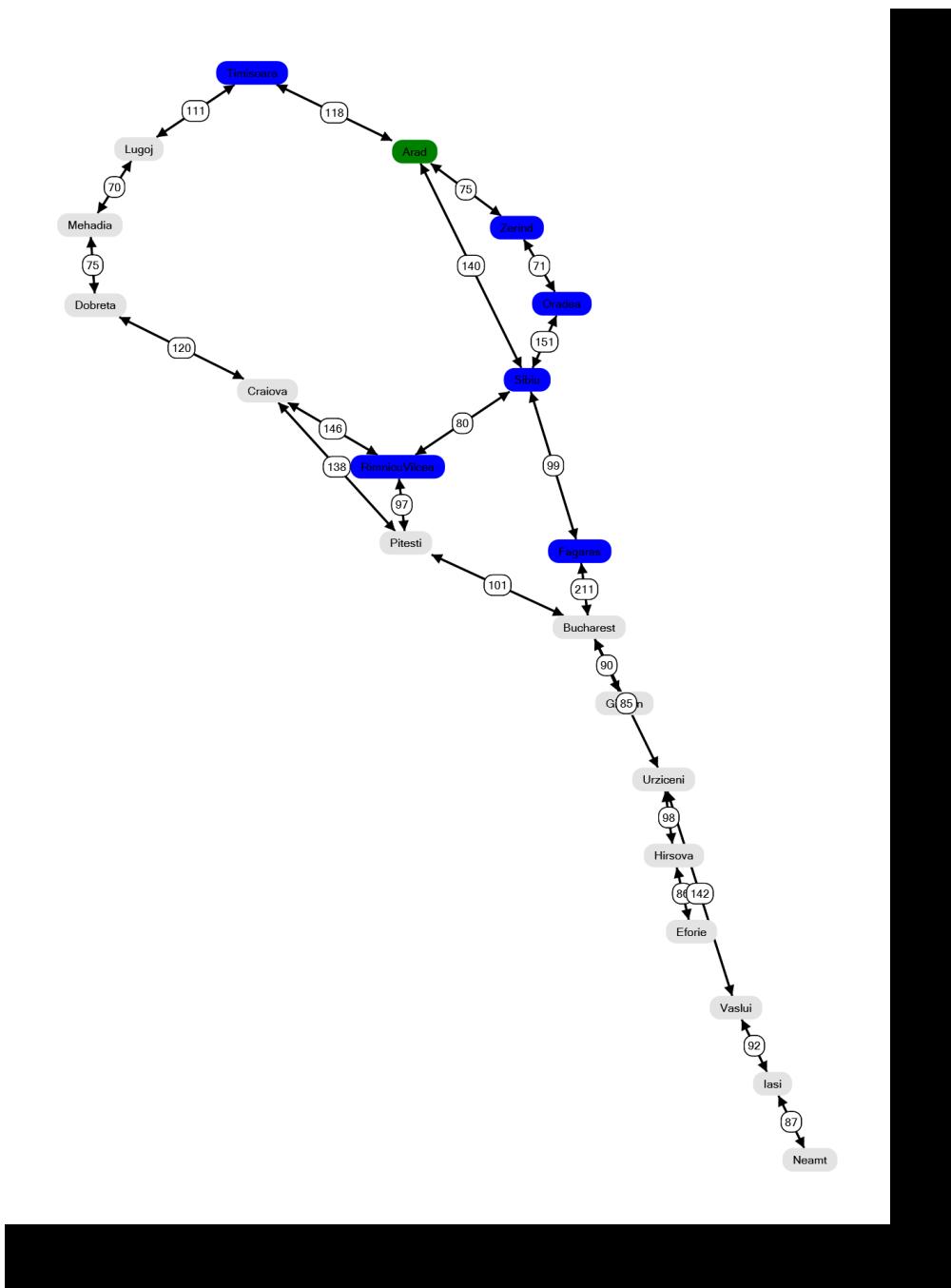
Visit

The algorithm has marked the node Zerind as visited



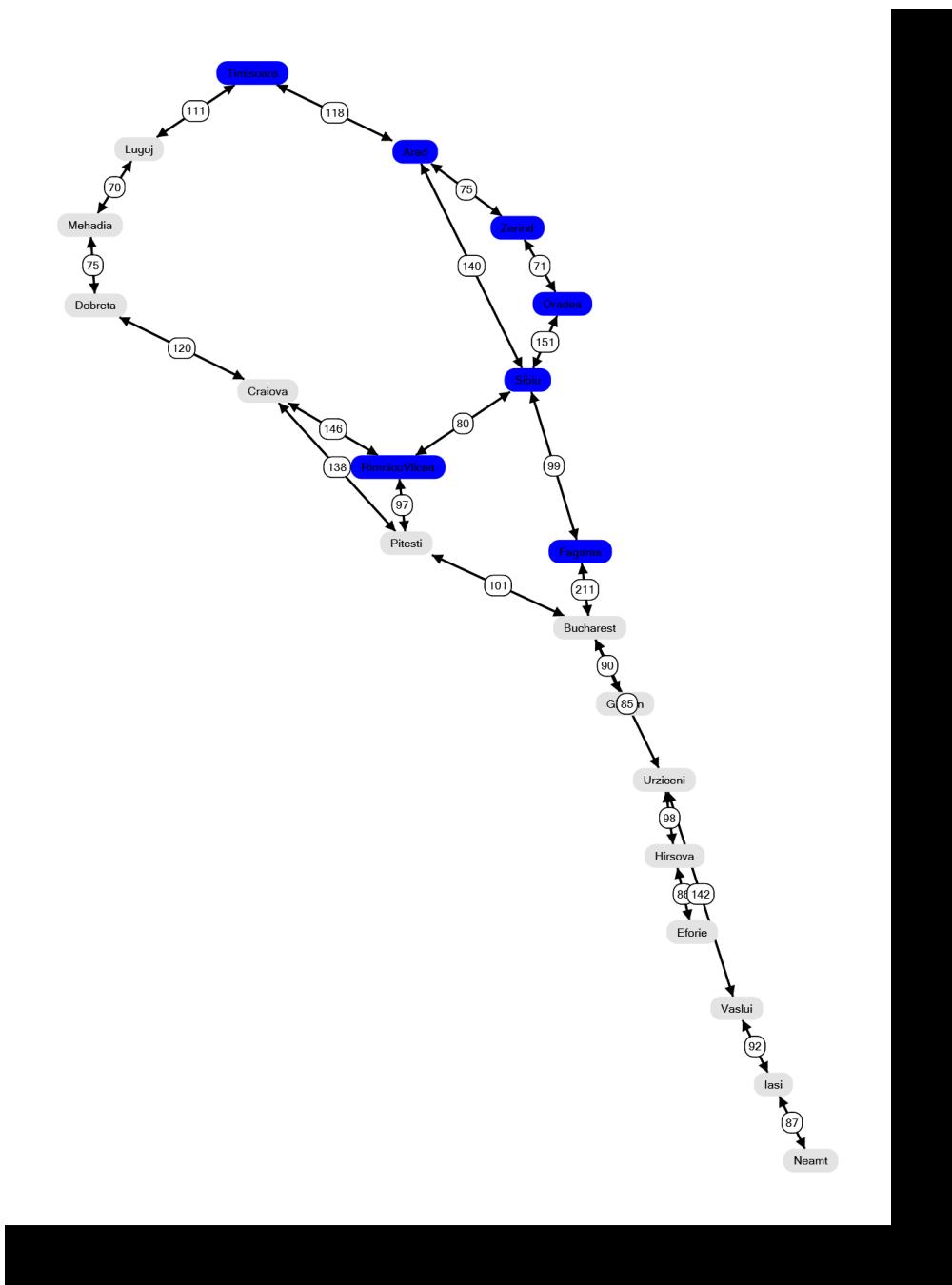
Post-Visit

The algorithm has closed the node Zerind



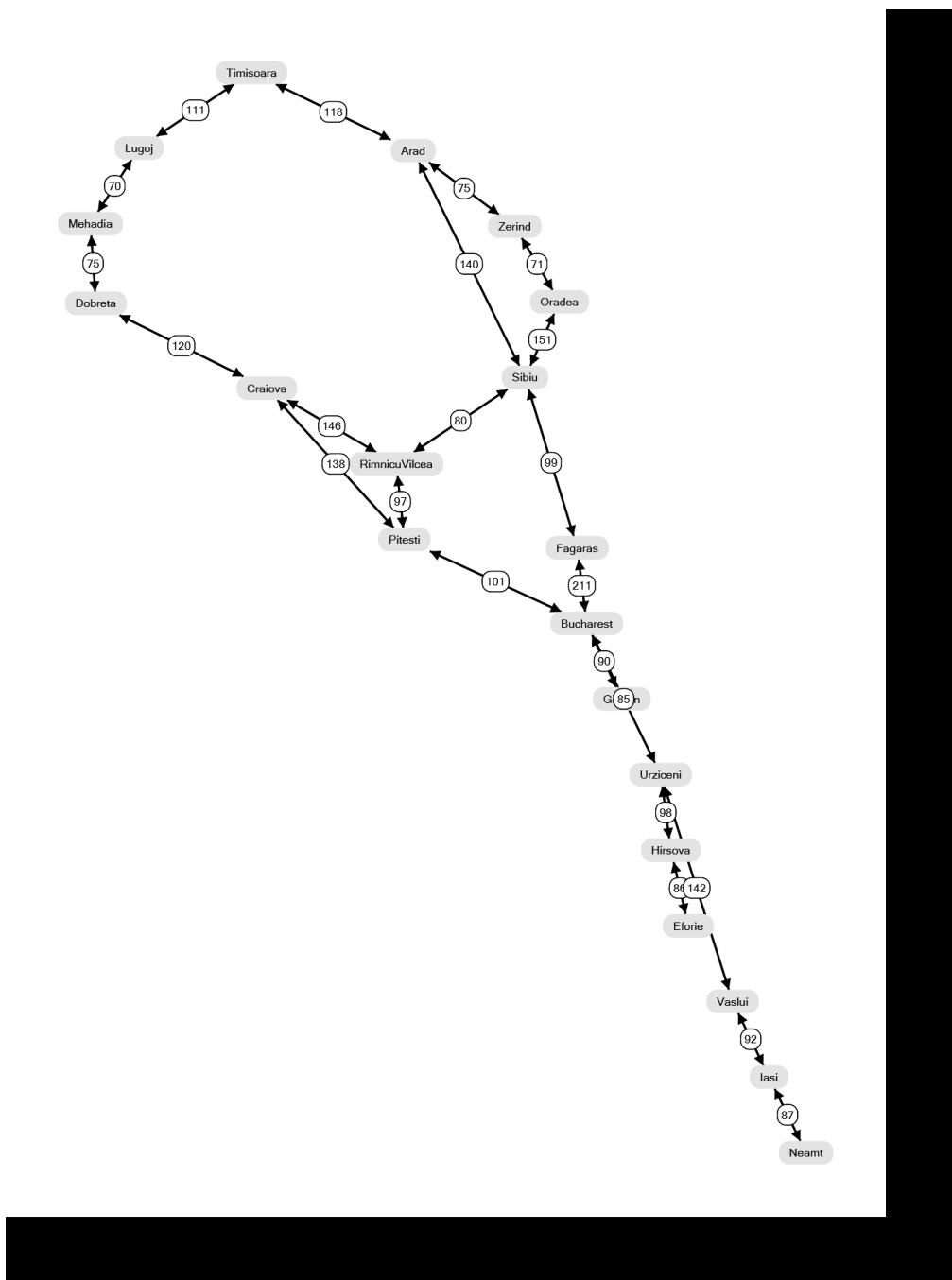
Post-Visit

The algorithm has closed the node Arad



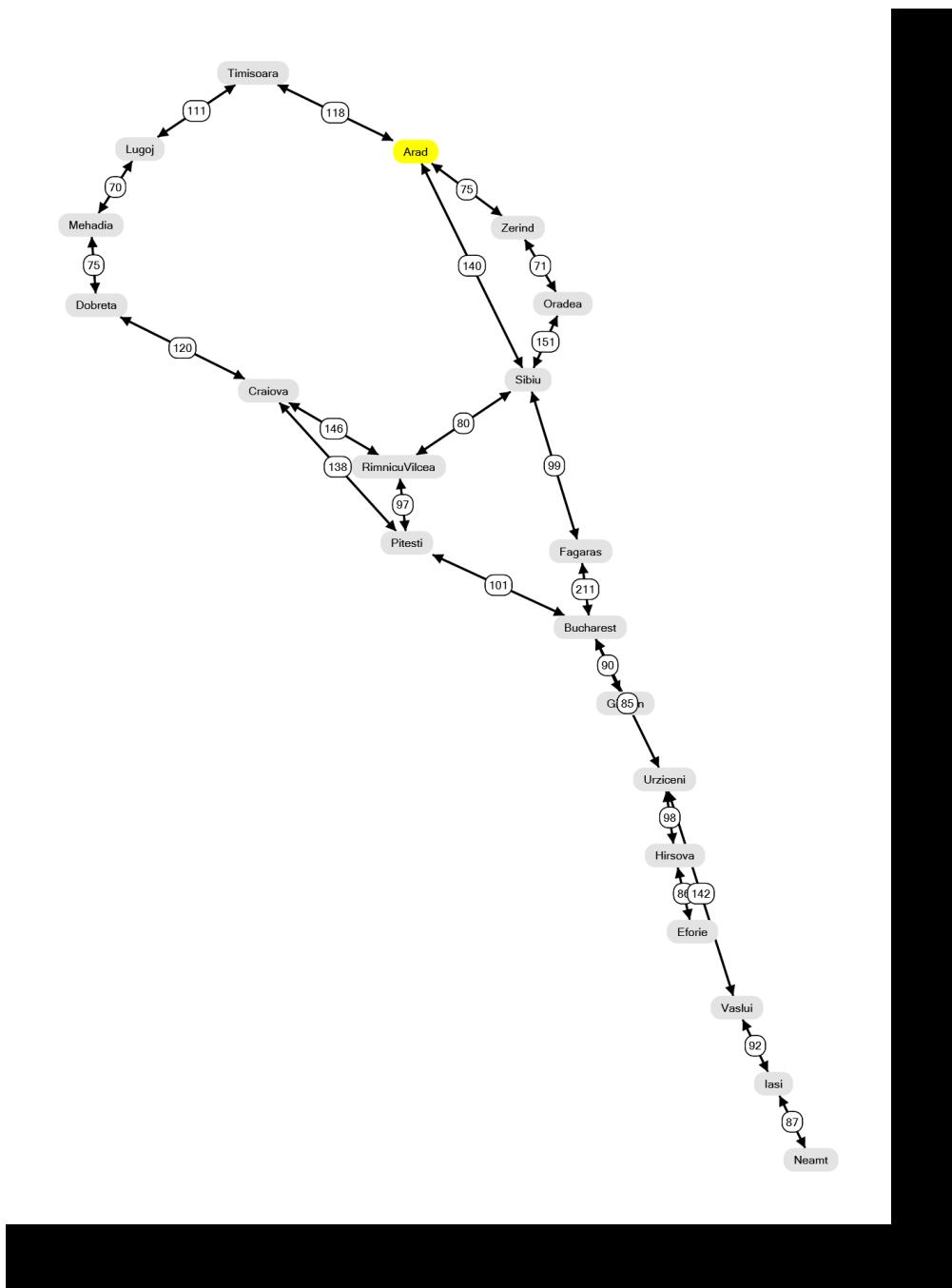
Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



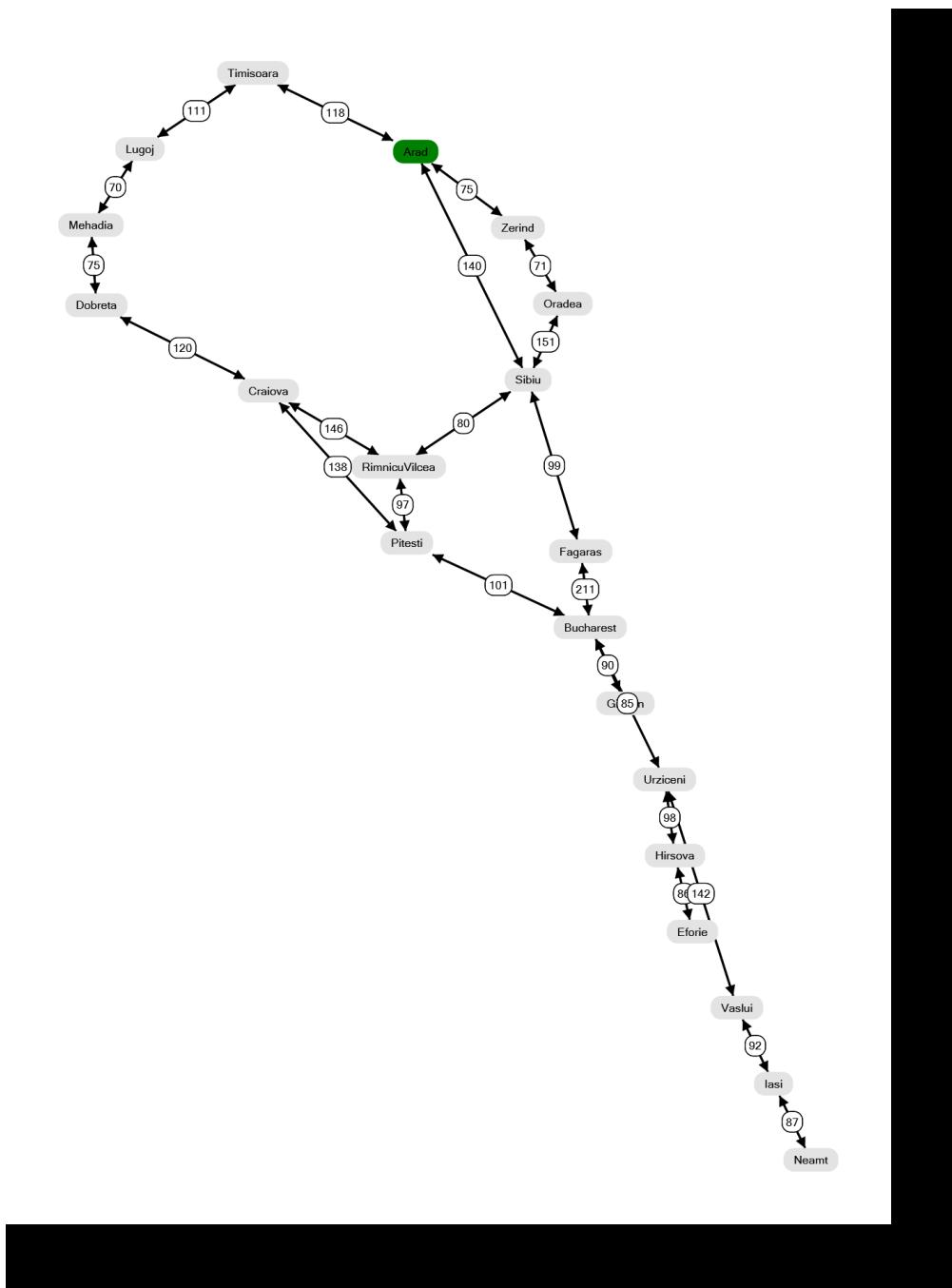
Pre-Visit

The algorithm has opened the node Arad



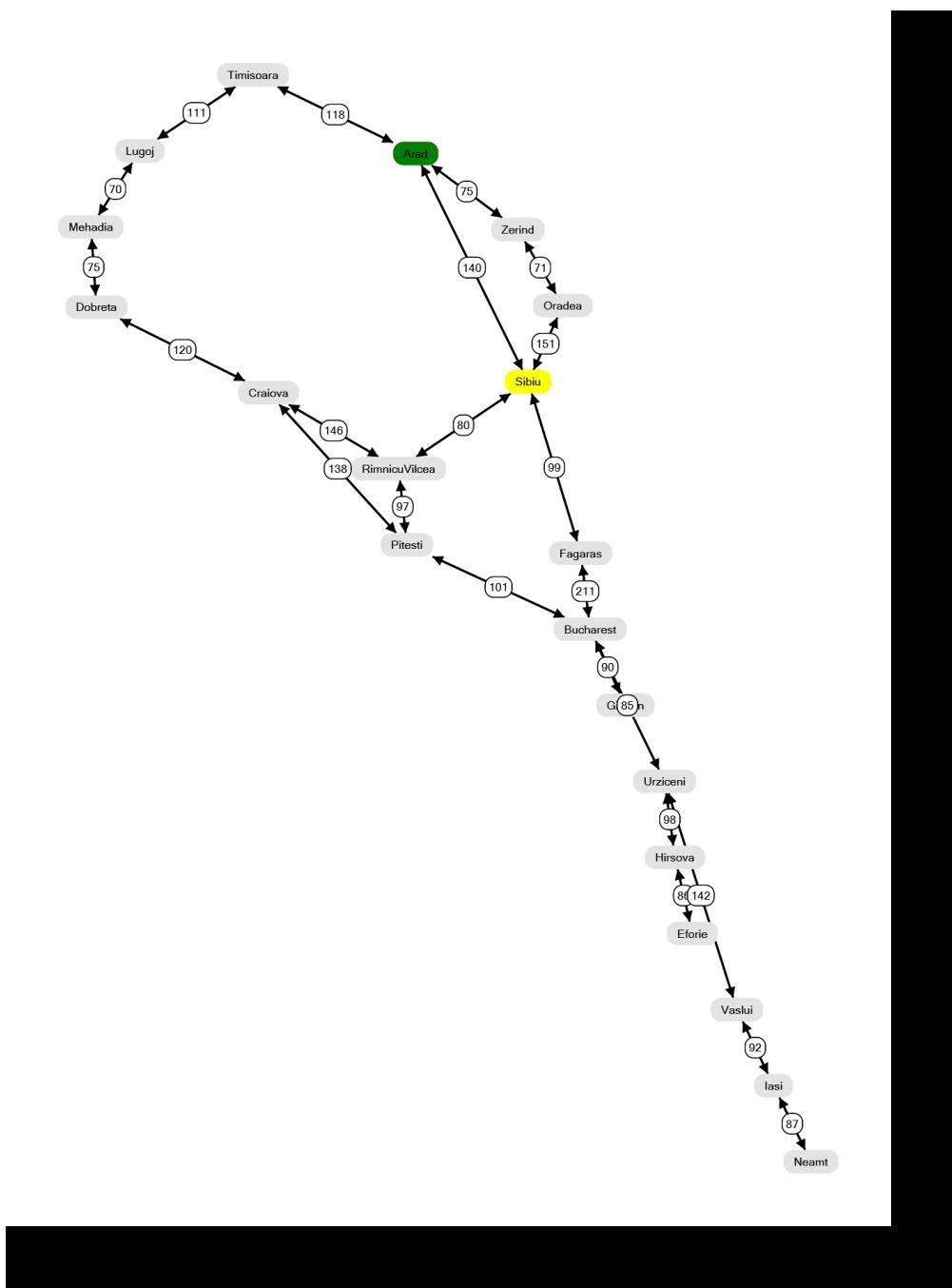
Visit

The algorithm has marked the node Arad as visited



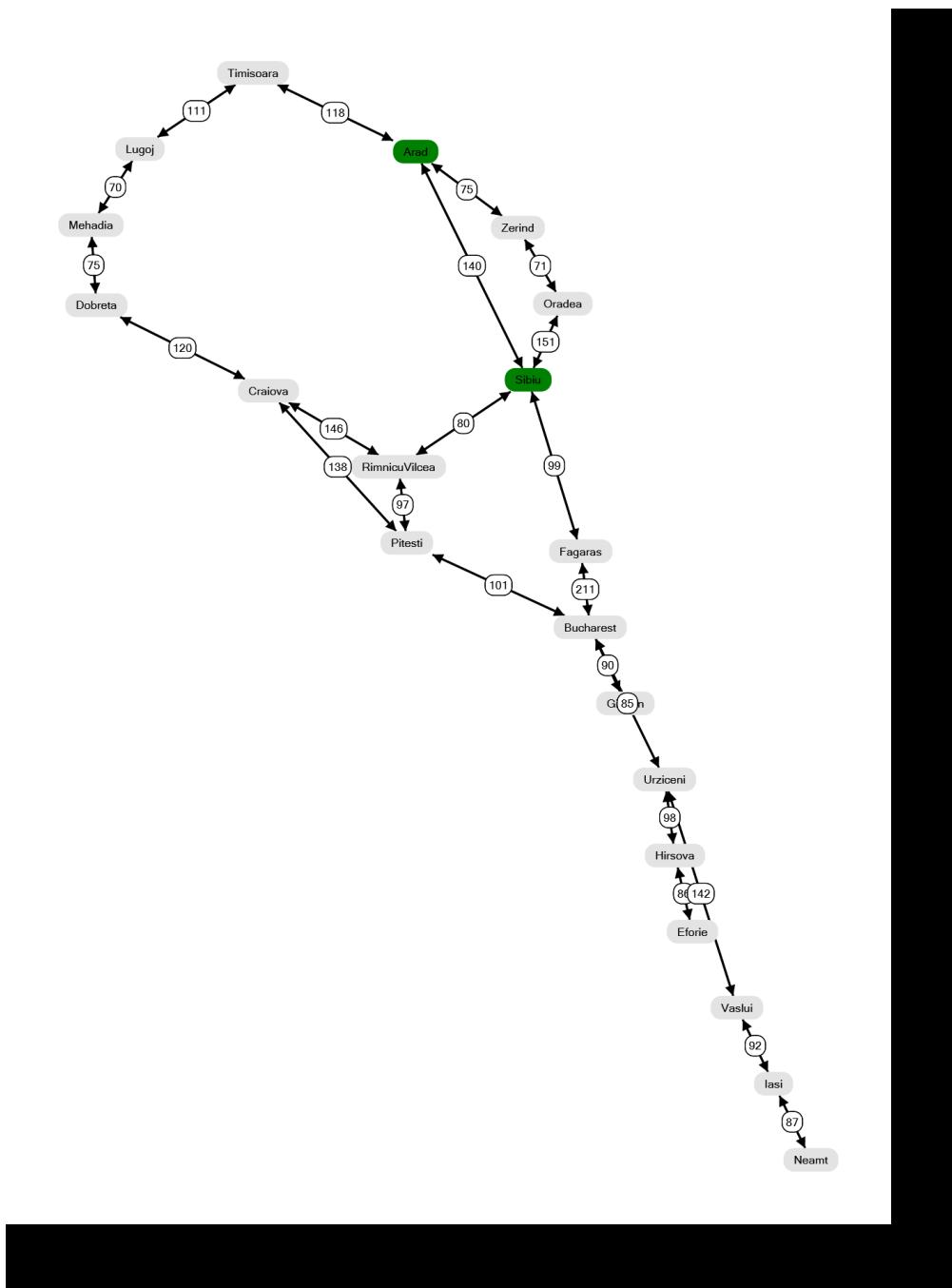
Pre-Visit

The algorithm has opened the node Sibiu



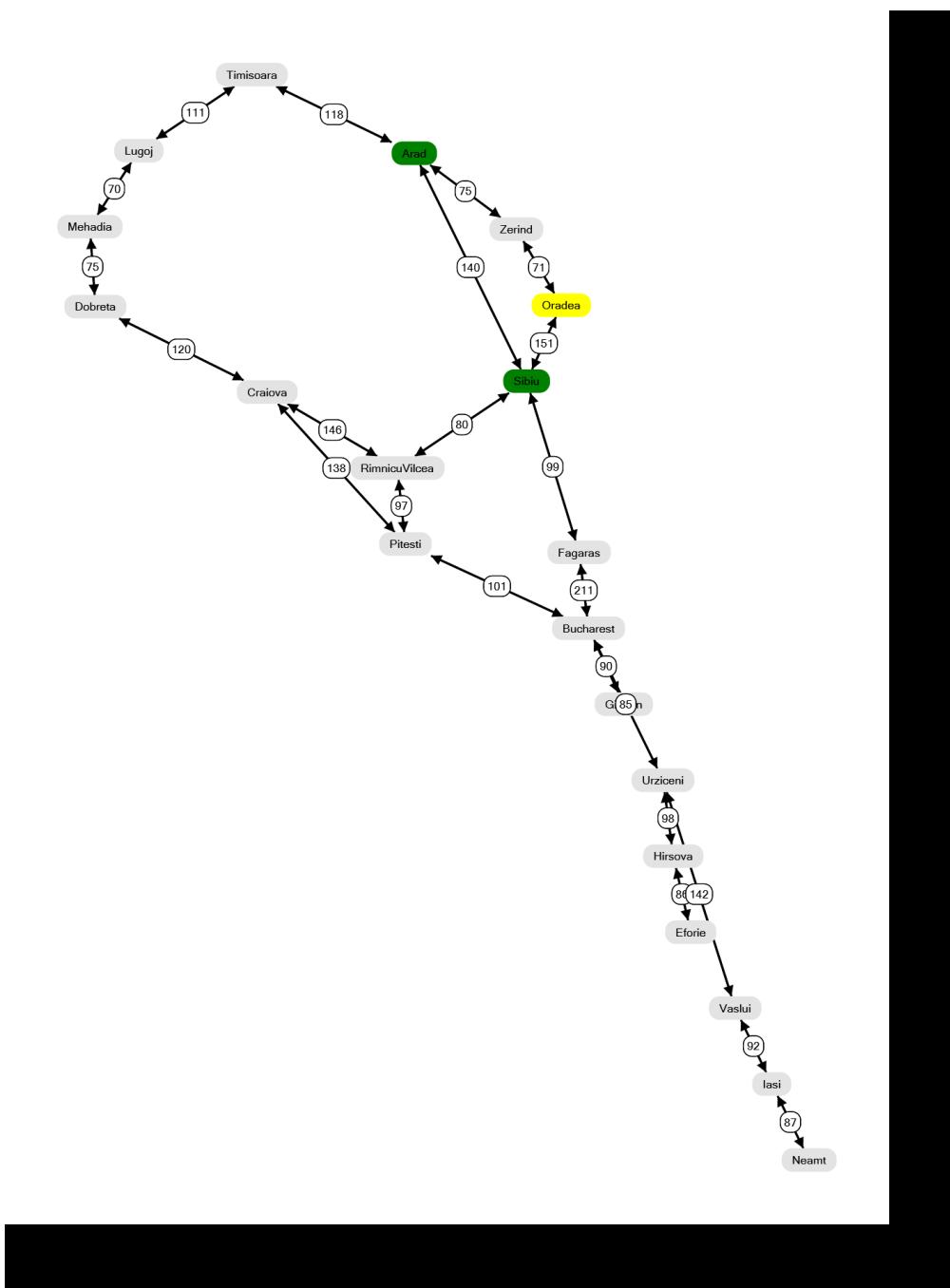
Visit

The algorithm has marked the node Sibiu as visited



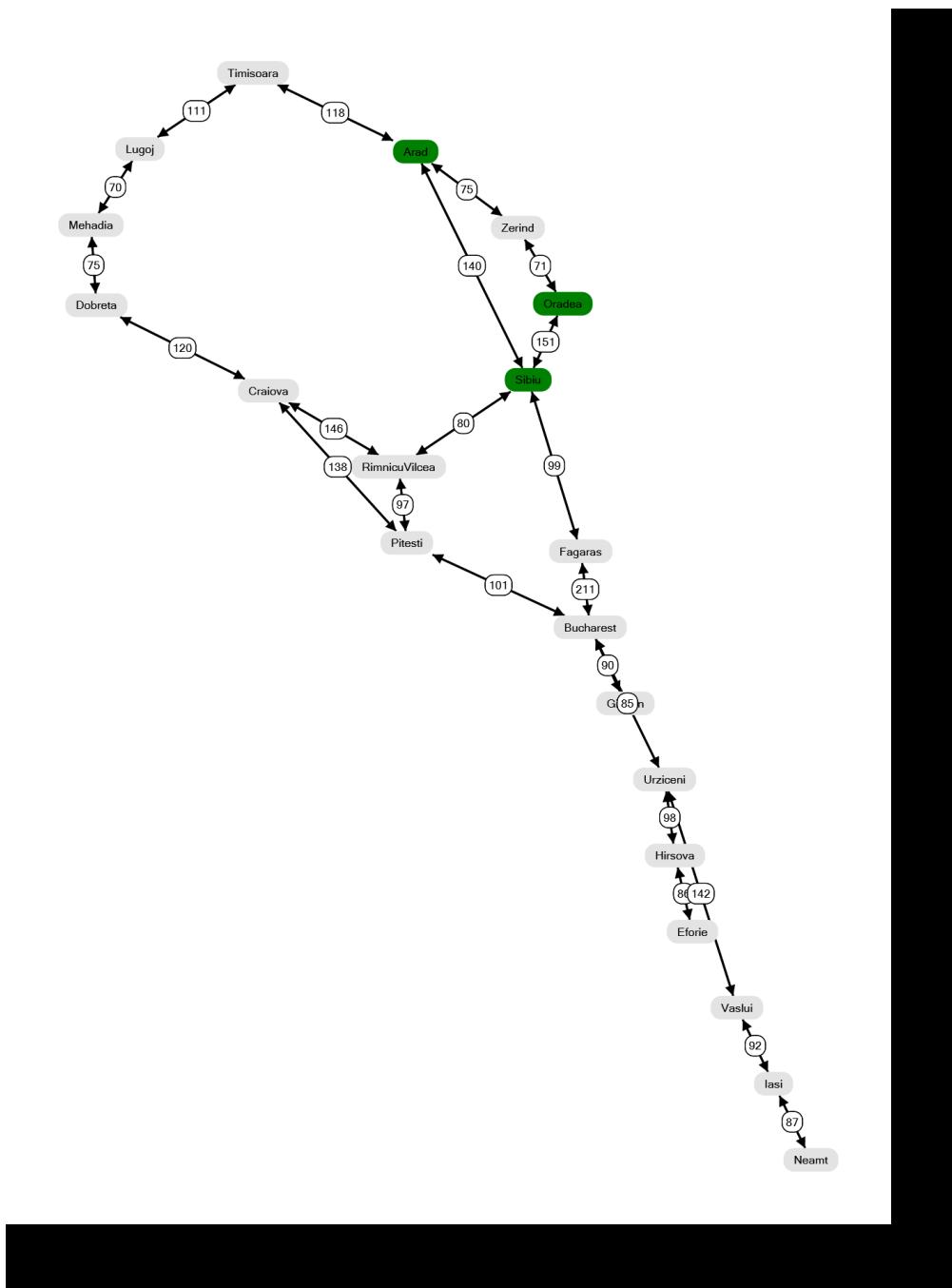
Pre-Visit

The algorithm has opened the node Oradea



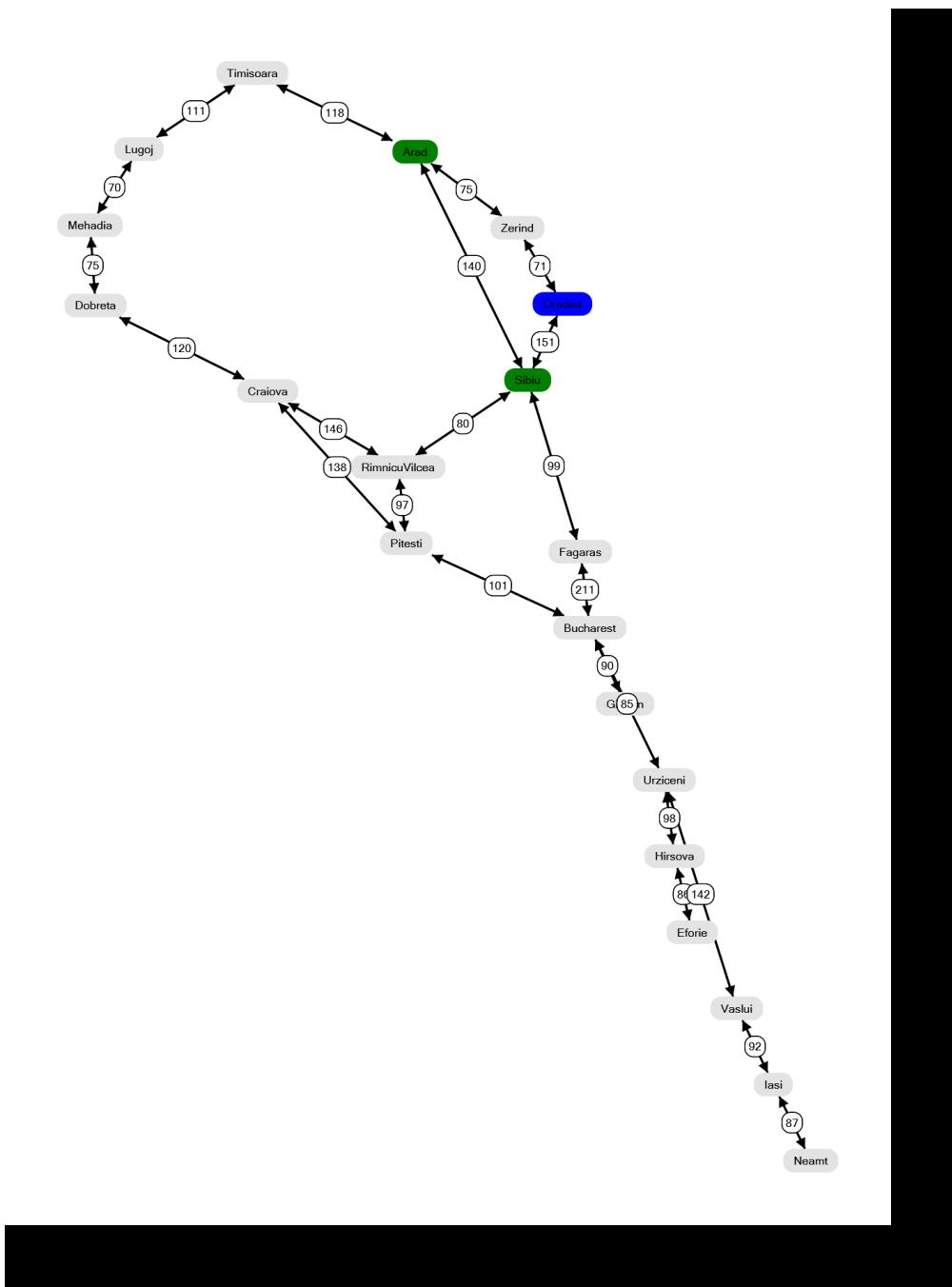
Visit

The algorithm has marked the node Oradea as visited



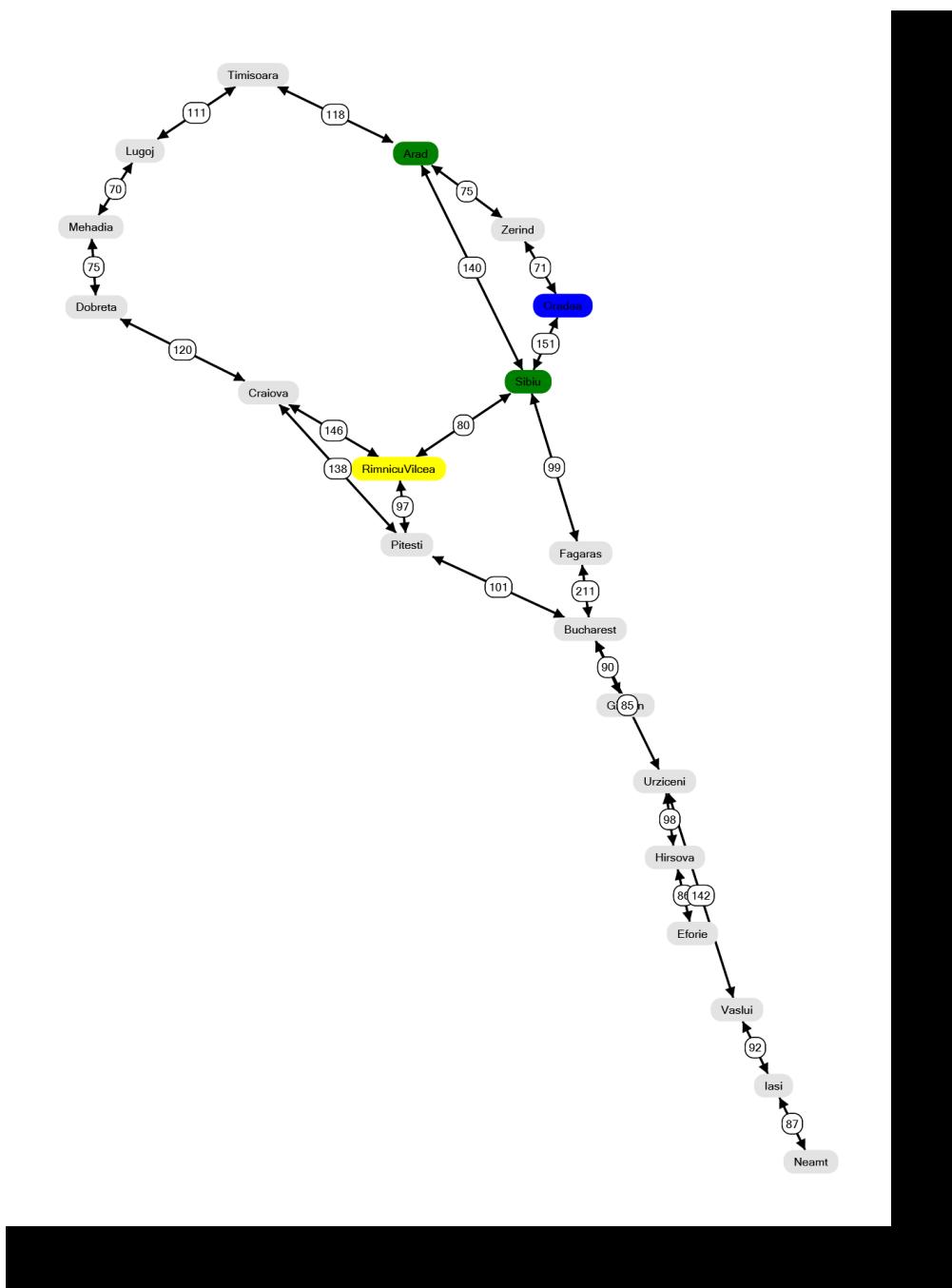
Post-Visit

The algorithm has closed the node Oradea



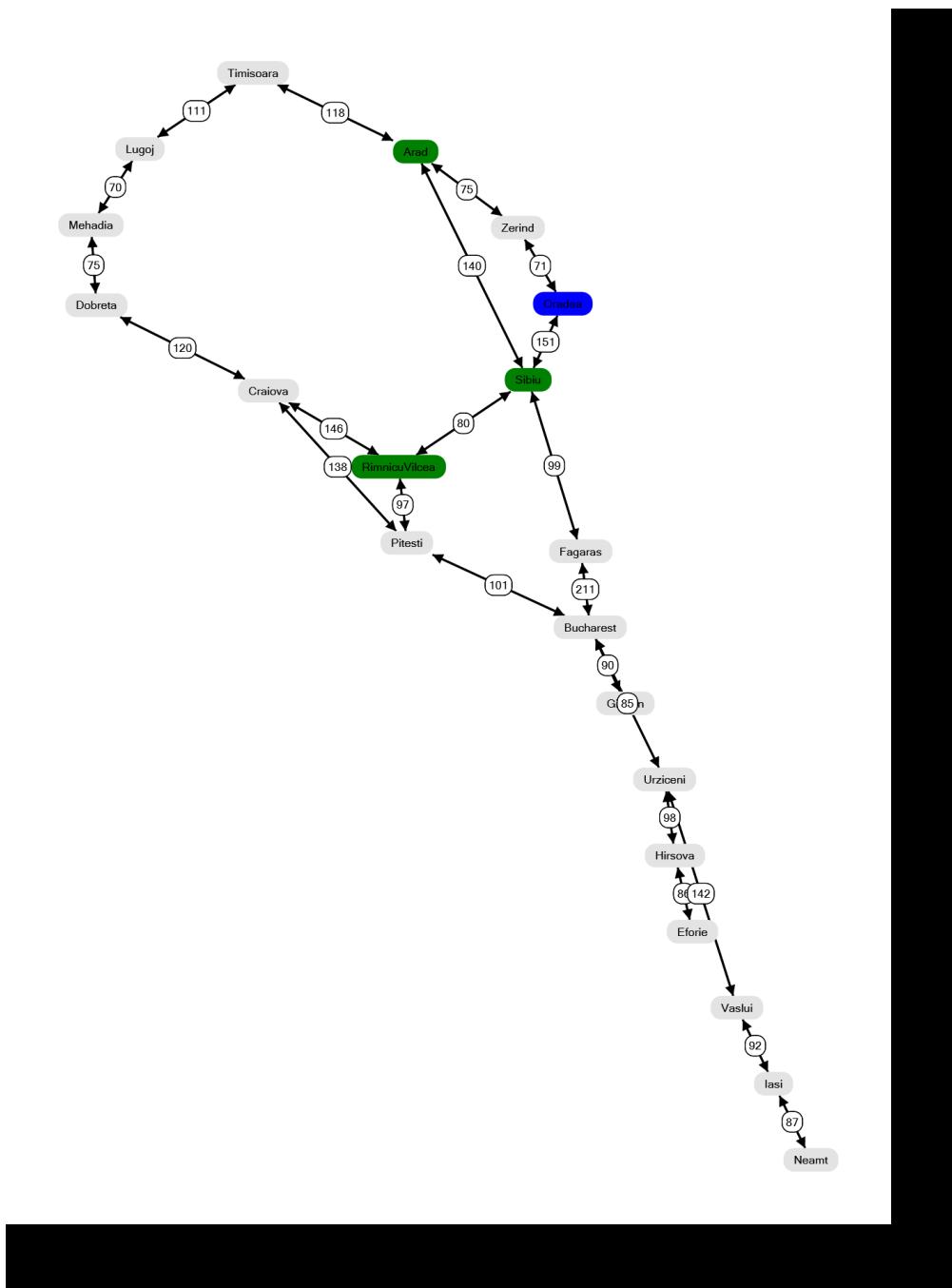
Pre-Visit

The algorithm has opened the node Rimnicu Vilcea



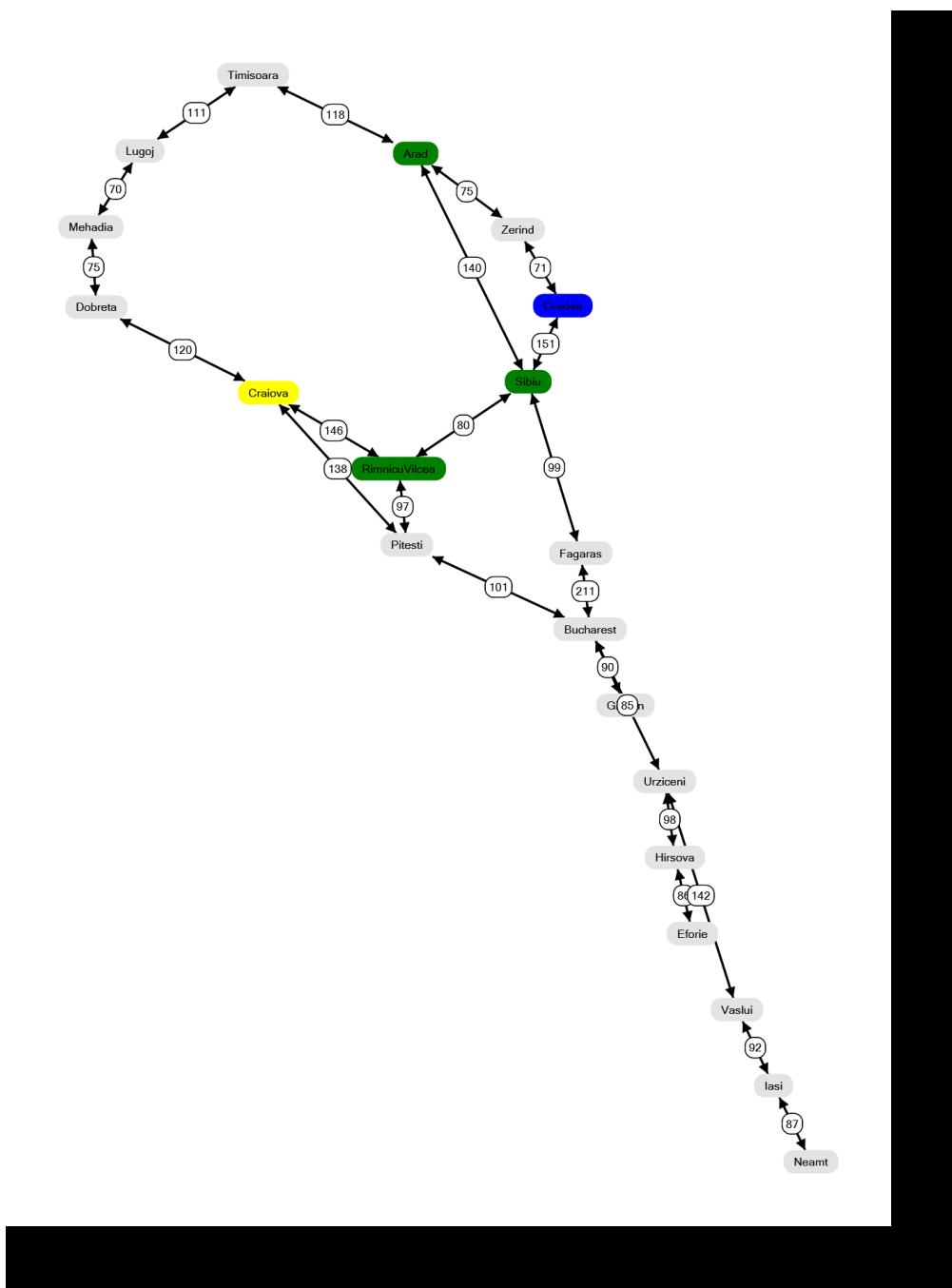
Visit

The algorithm has marked the node Rimnicu Vilcea as visited



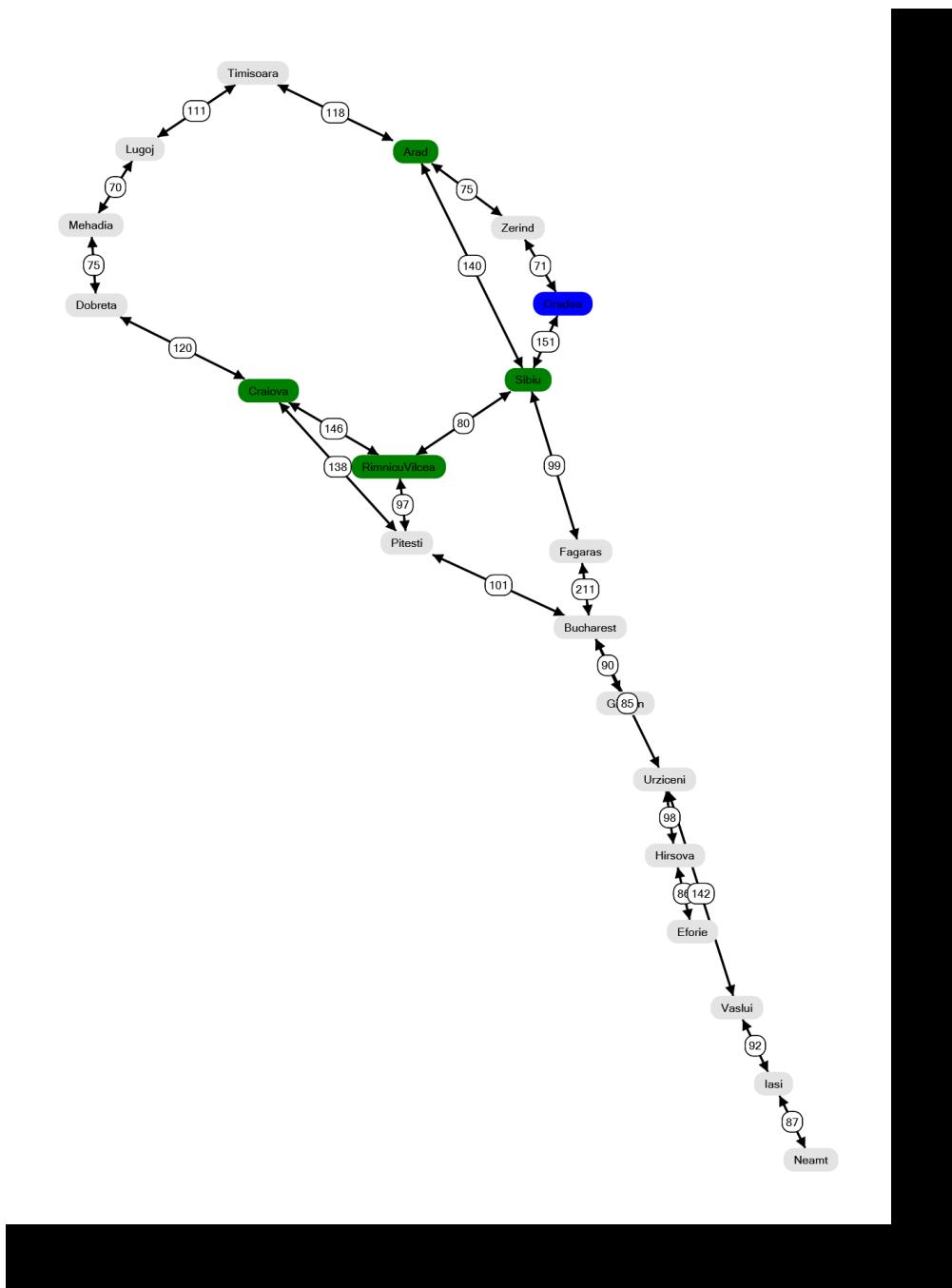
Pre-Visit

The algorithm has opened the node Craiova



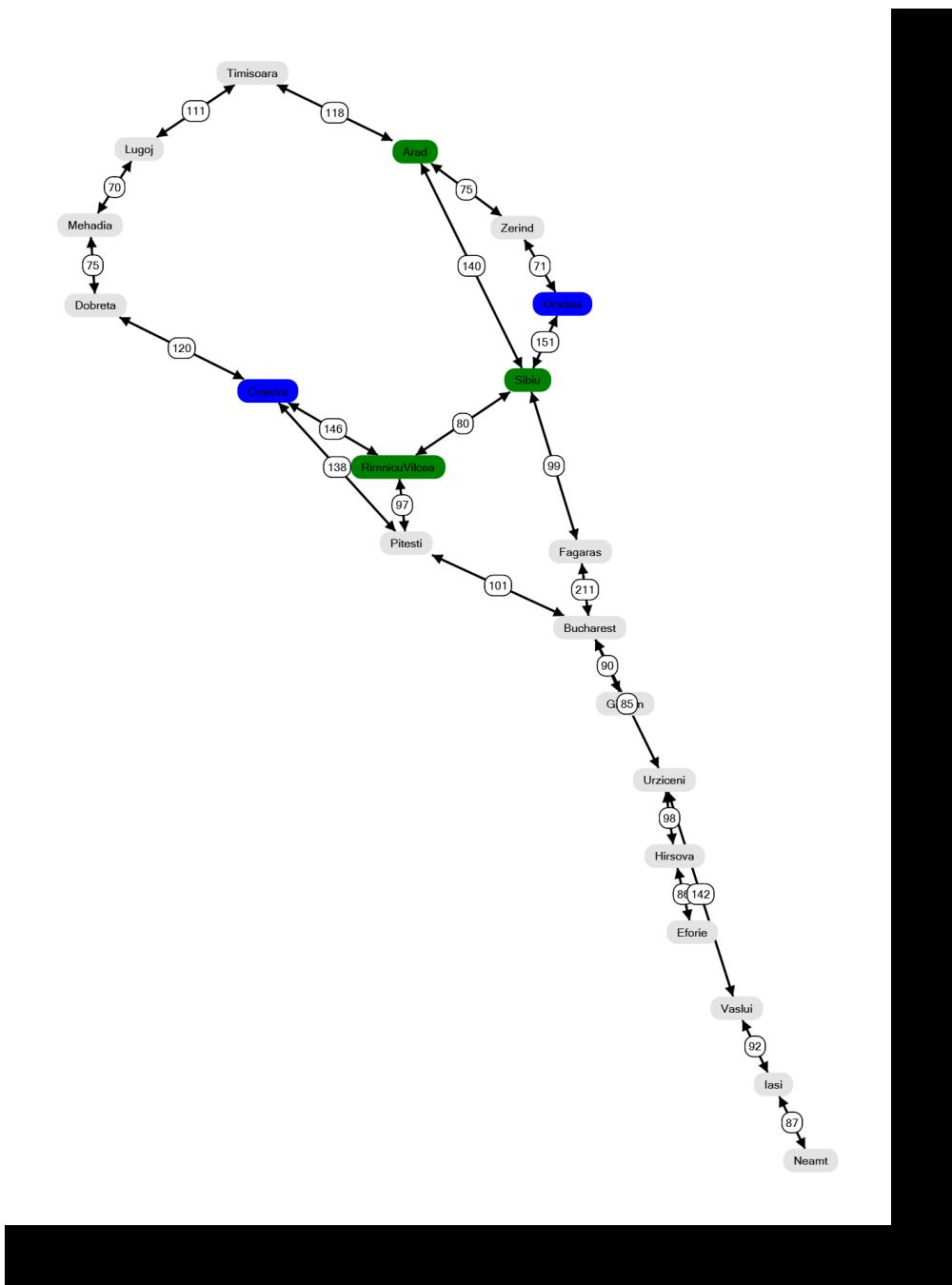
Visit

The algorithm has marked the node Craiova as visited



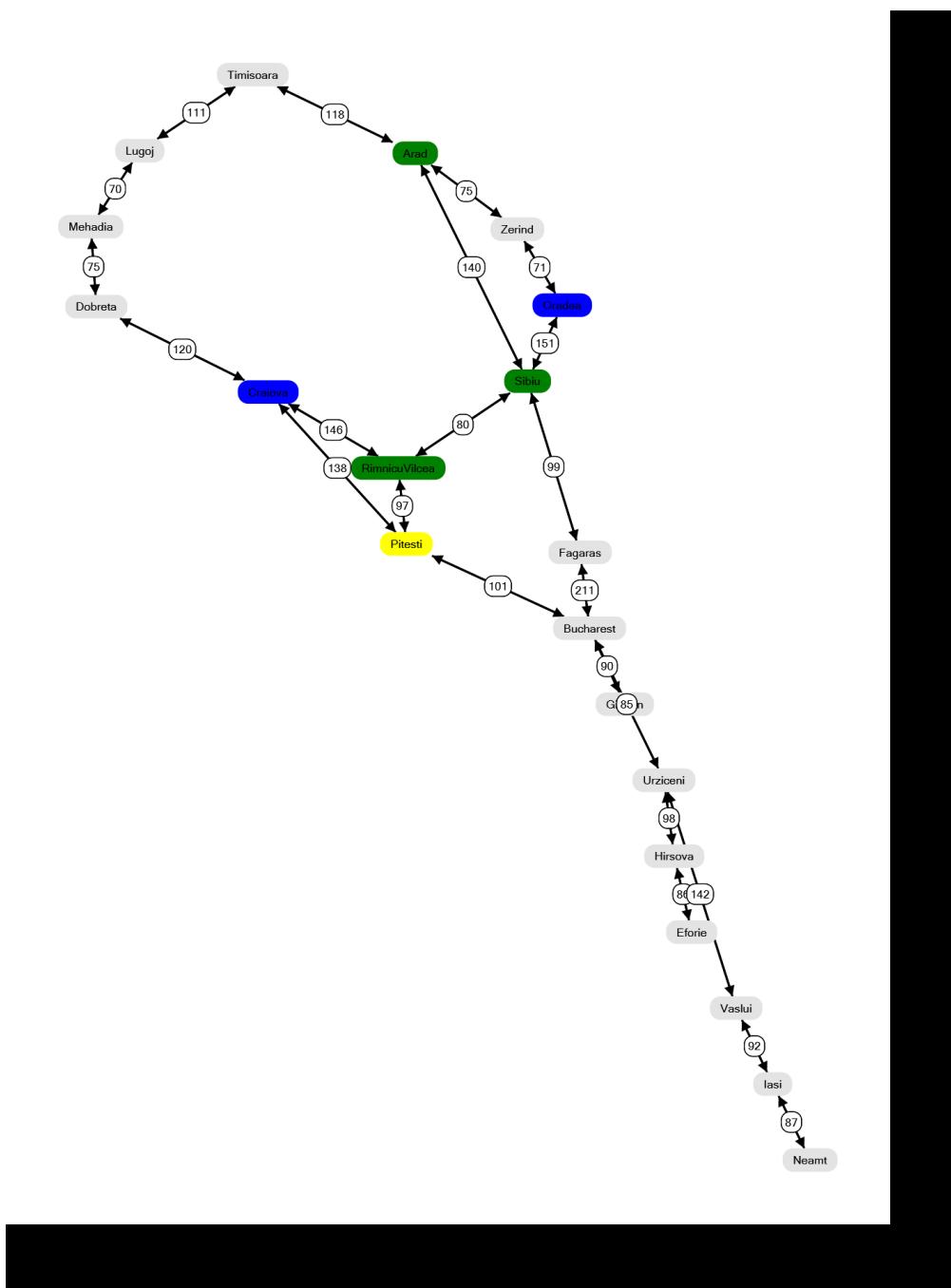
Post-Visit

The algorithm has closed the node Craiova



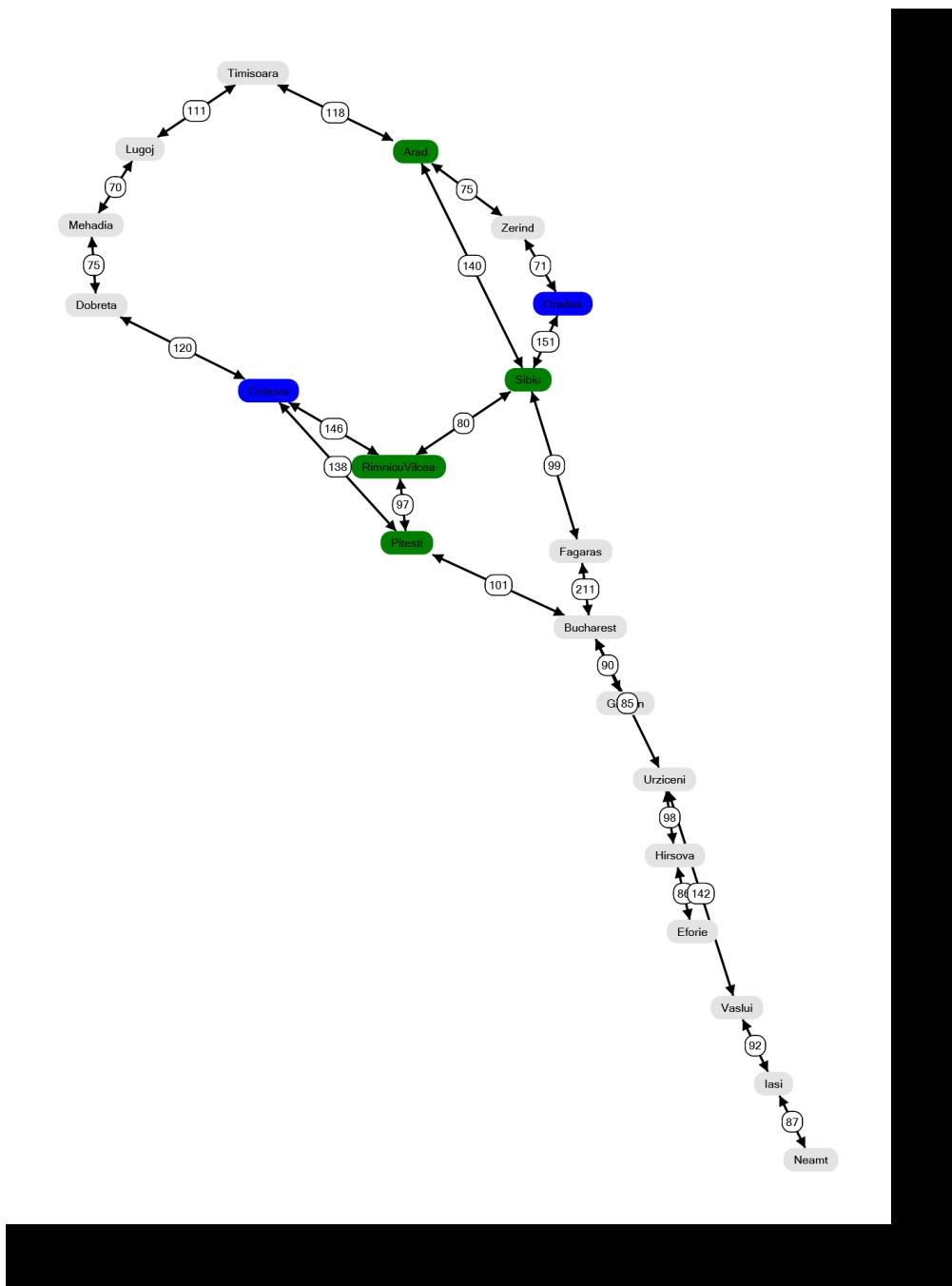
Pre-Visit

The algorithm has opened the node Pitesti



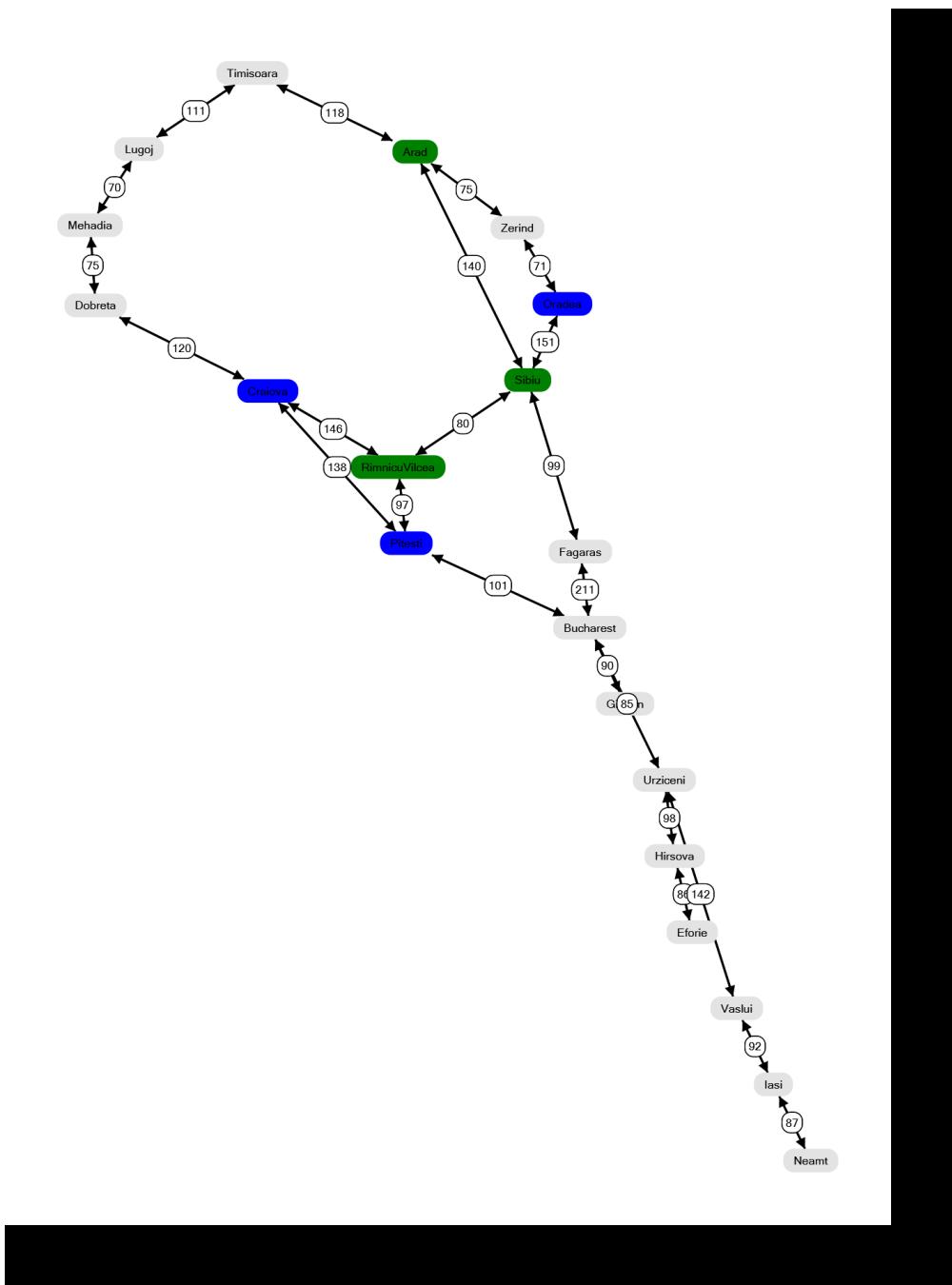
Visit

The algorithm has marked the node Pitesti as visited



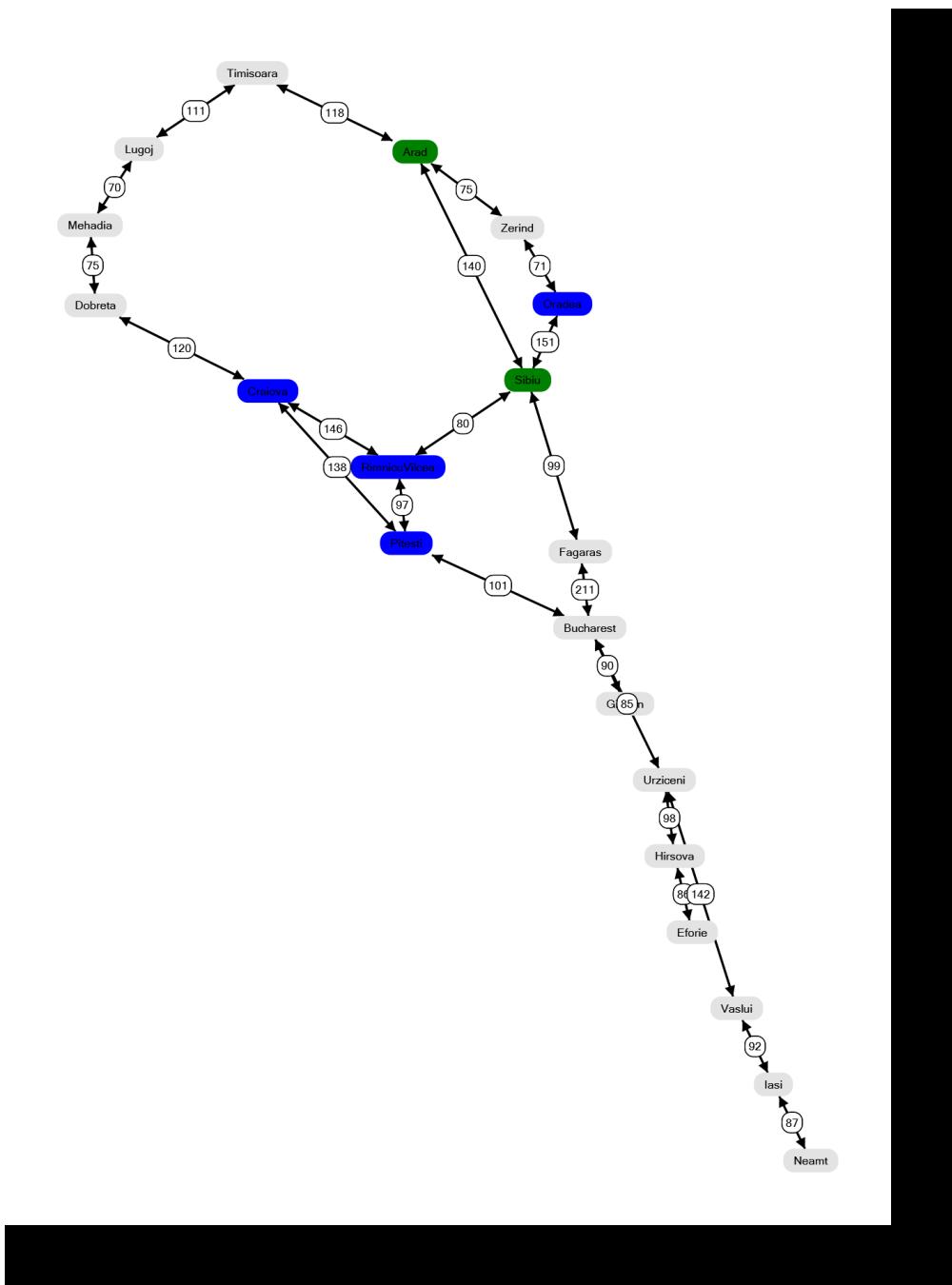
Post-Visit

The algorithm has closed the node Pitesti



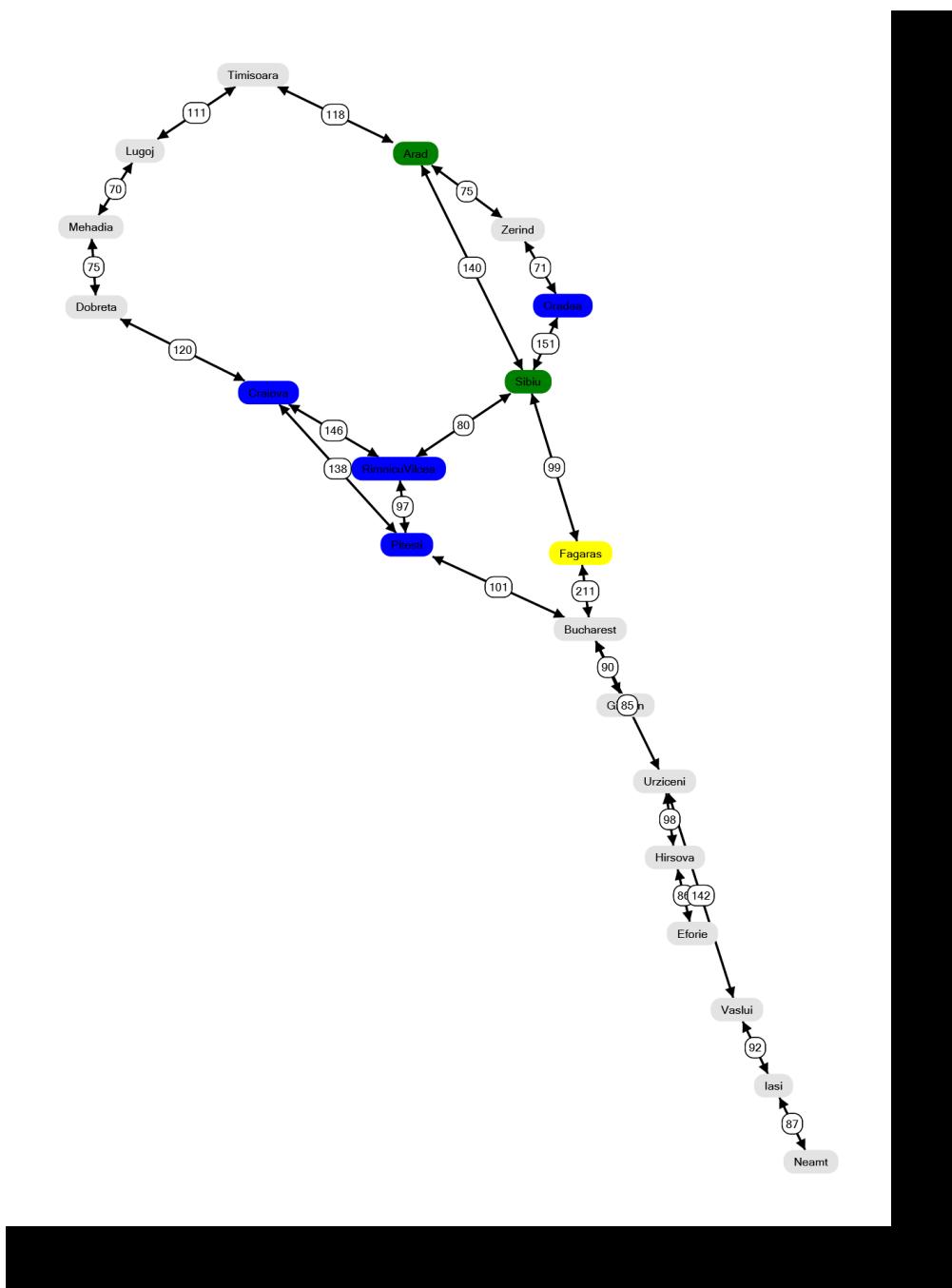
Post-Visit

The algorithm has closed the node Rimnicu Vilcea



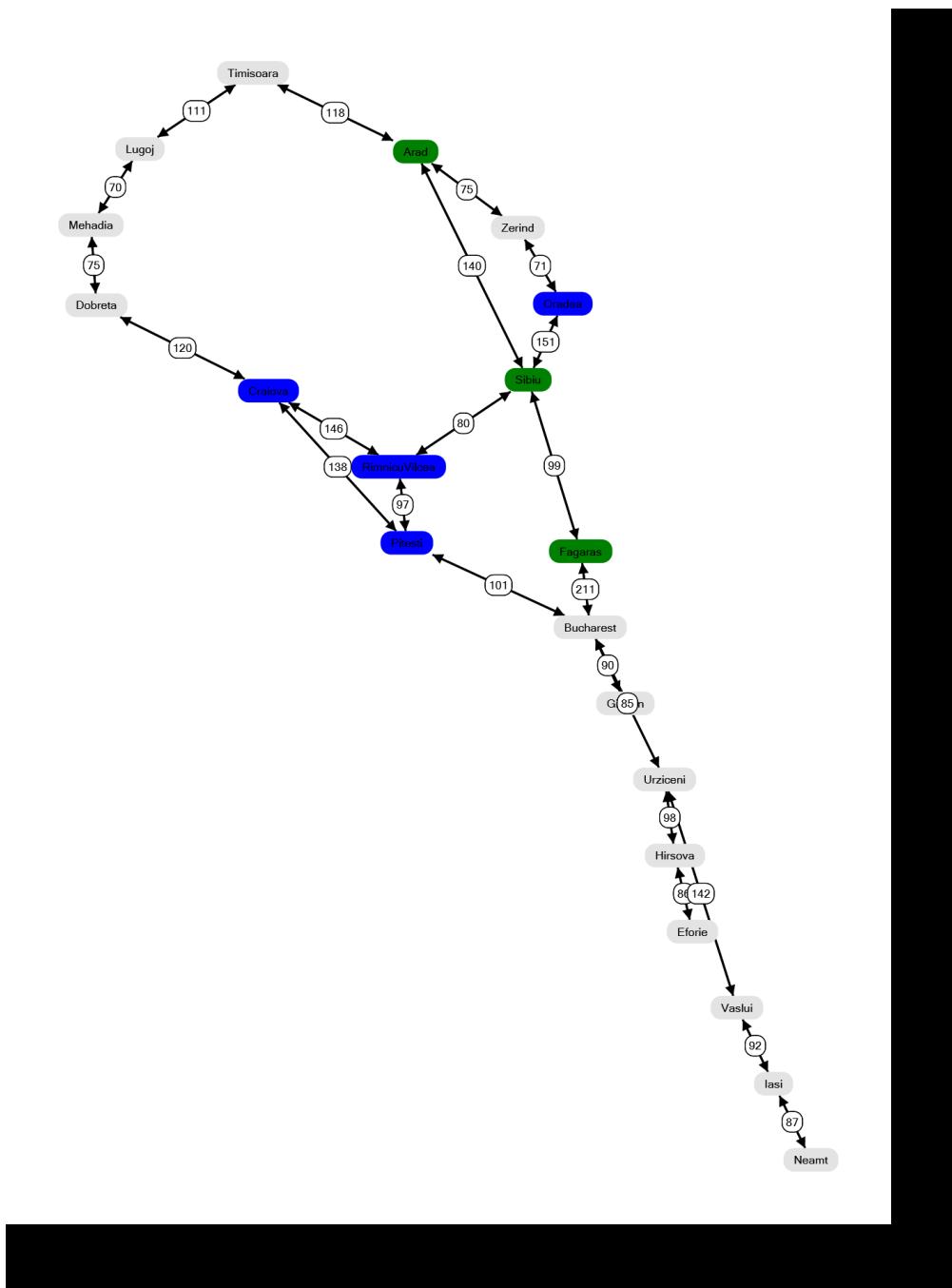
Pre-Visit

The algorithm has opened the node Fagaras



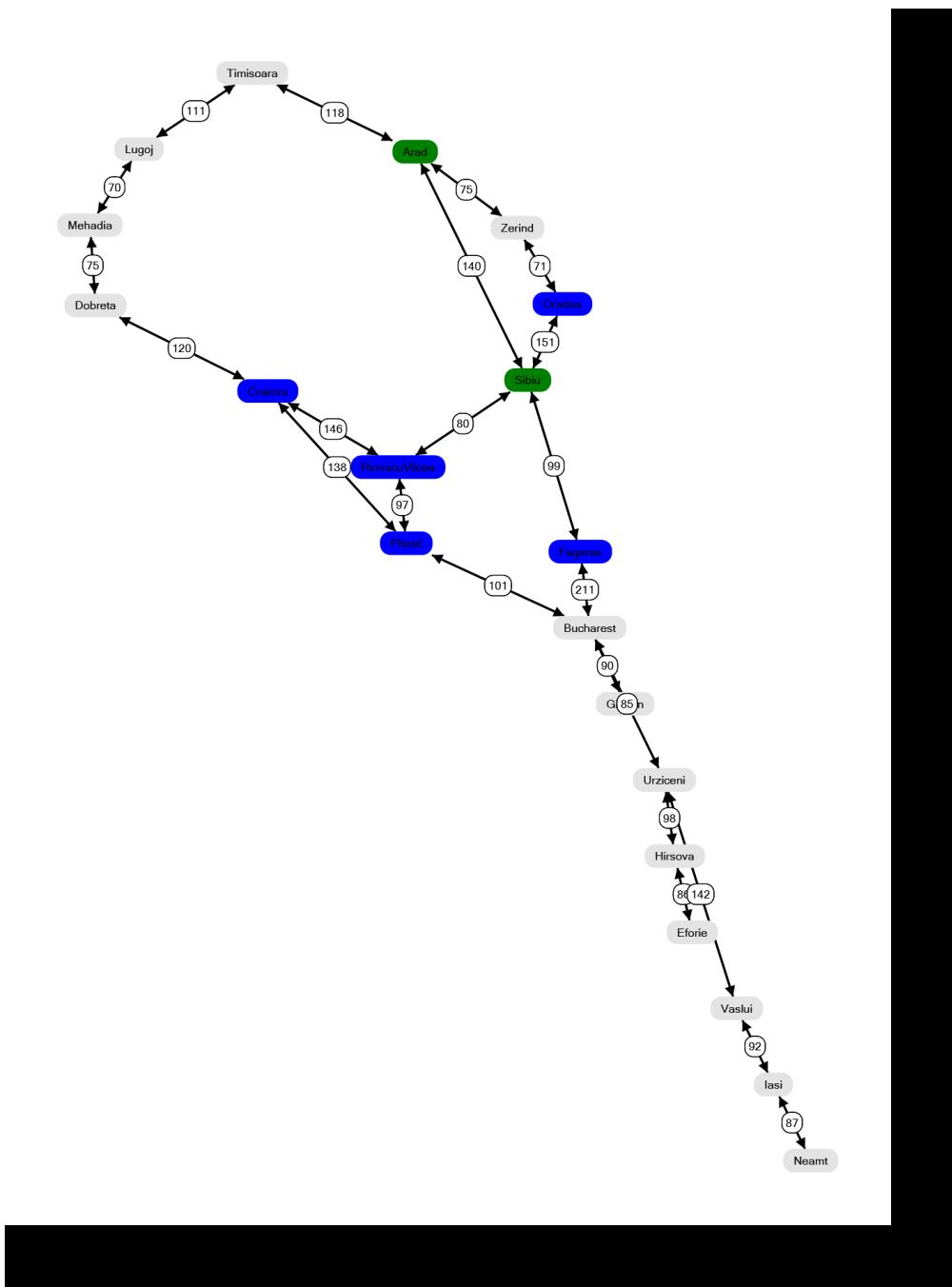
Visit

The algorithm has marked the node Fagaras as visited



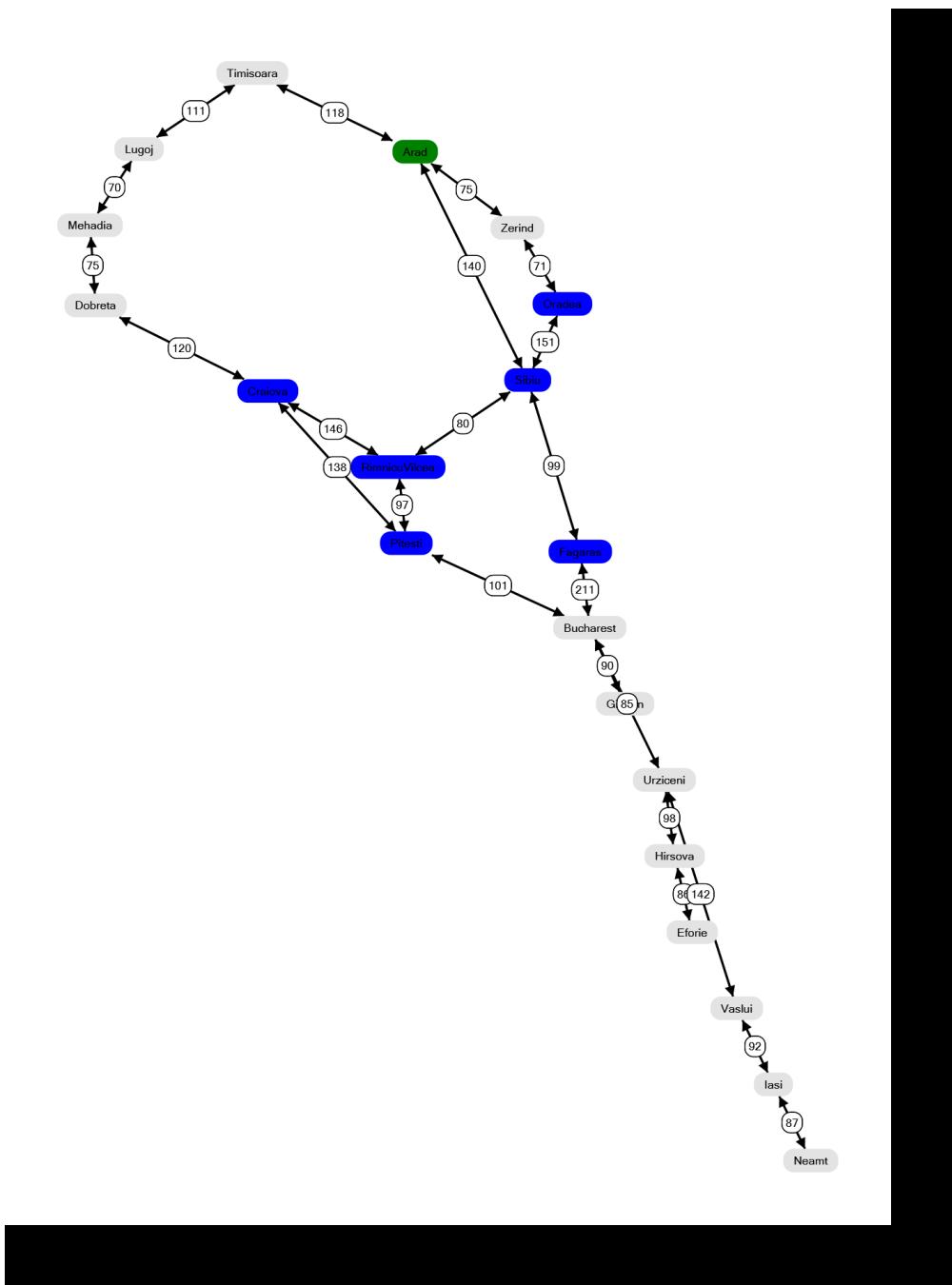
Post-Visit

The algorithm has closed the node Fagaras



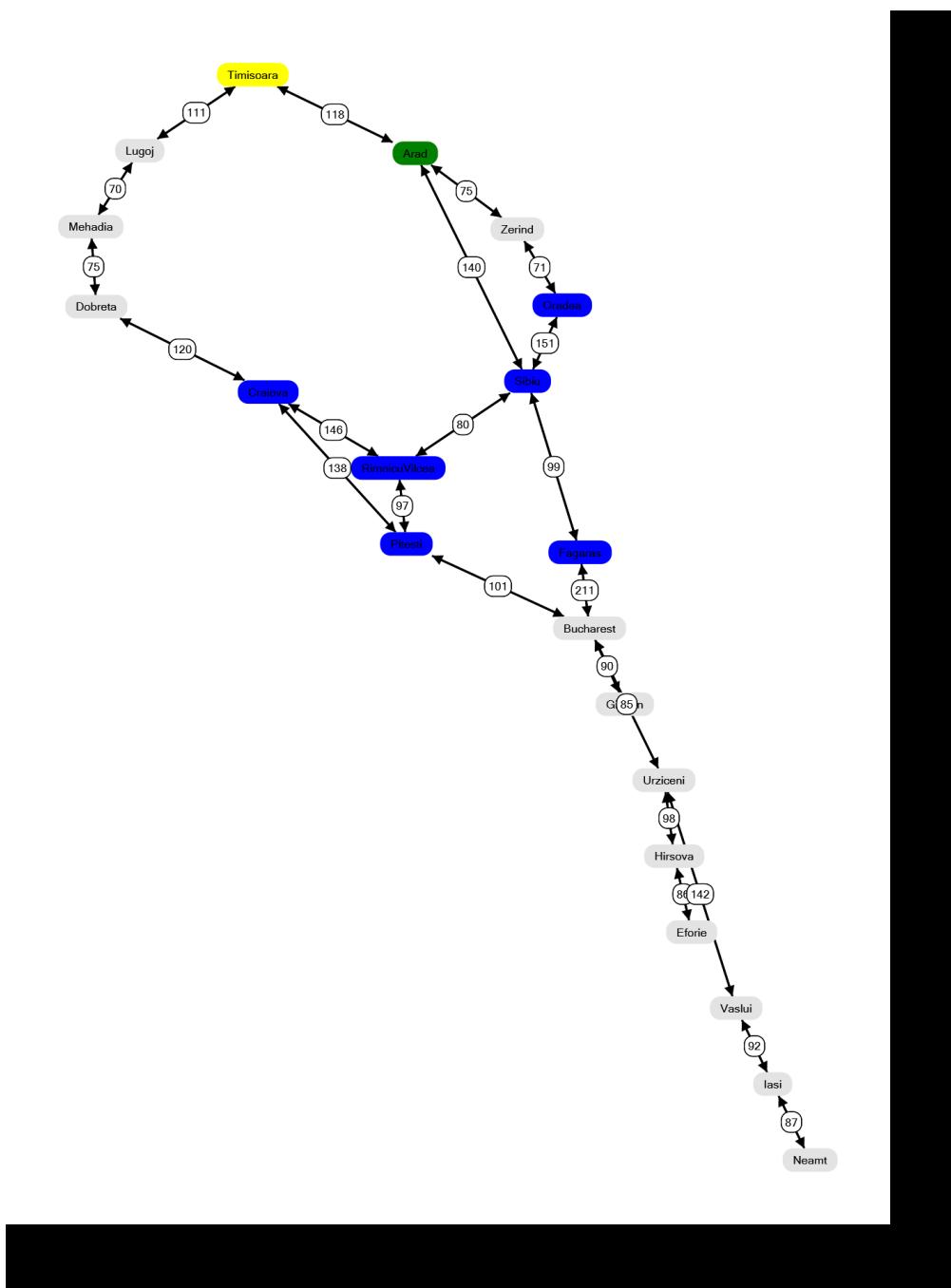
Post-Visit

The algorithm has closed the node Sibiu



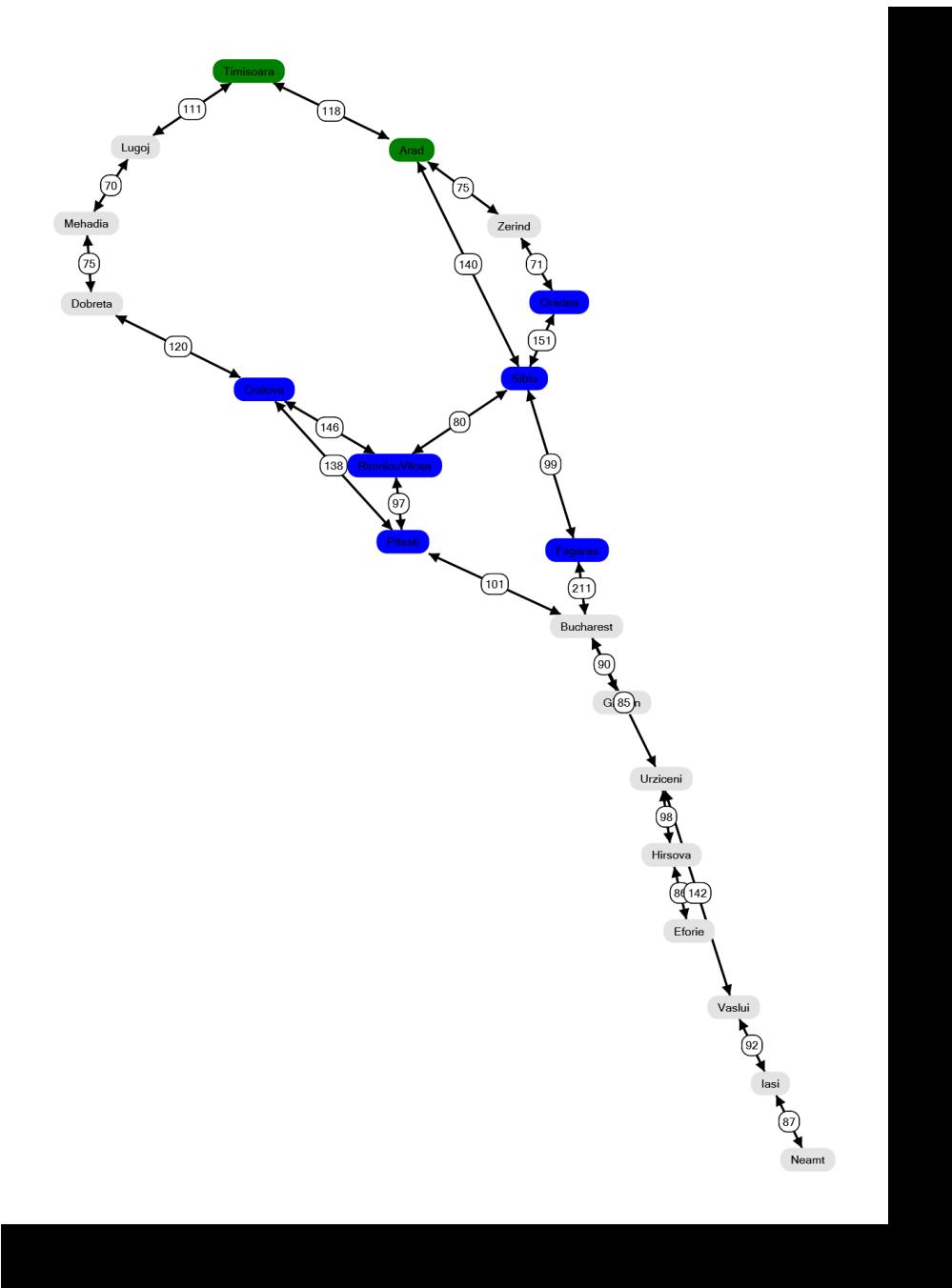
Pre-Visit

The algorithm has opened the node Timisoara



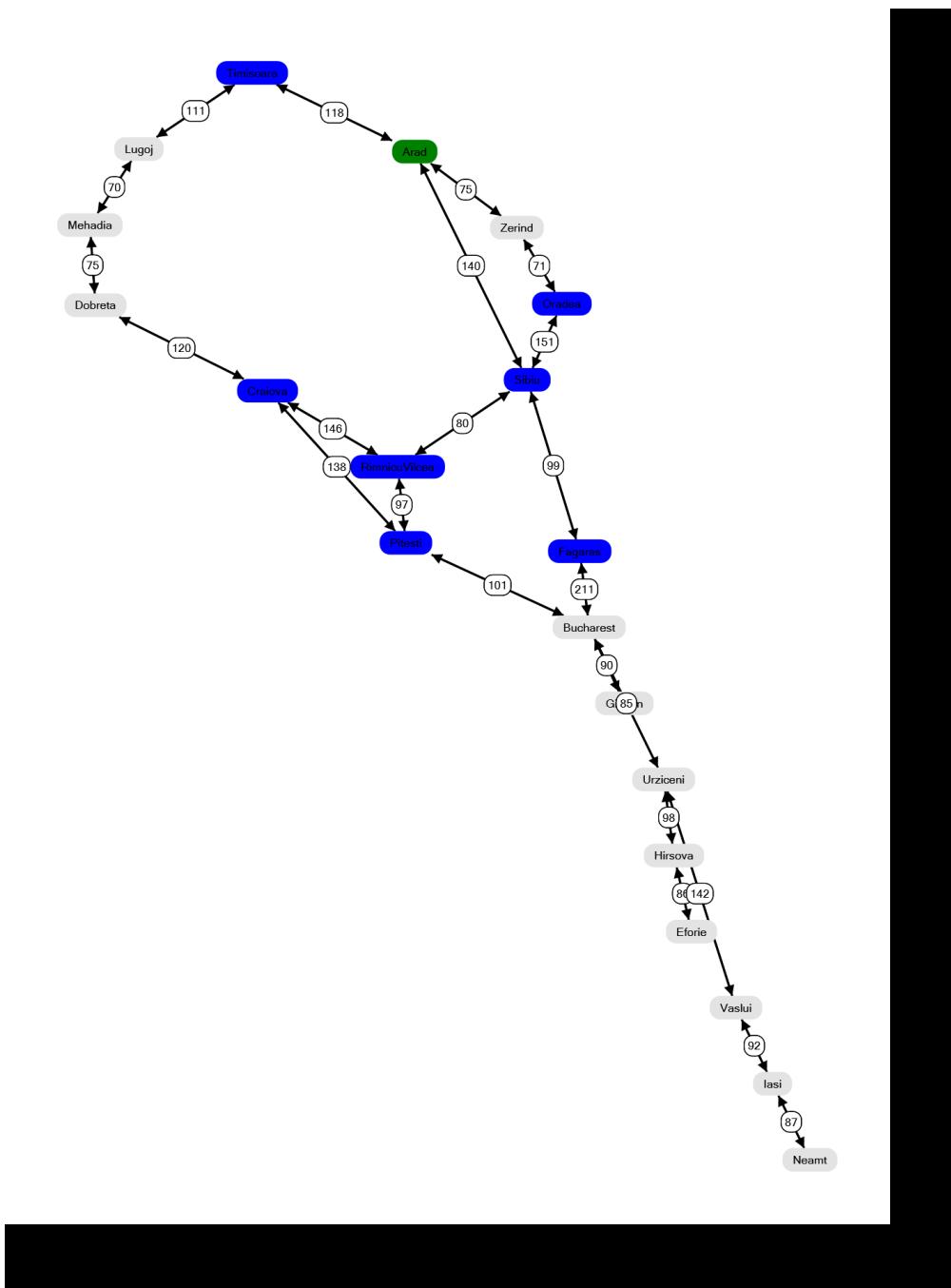
Visit

The algorithm has marked the node Timisoara as visited



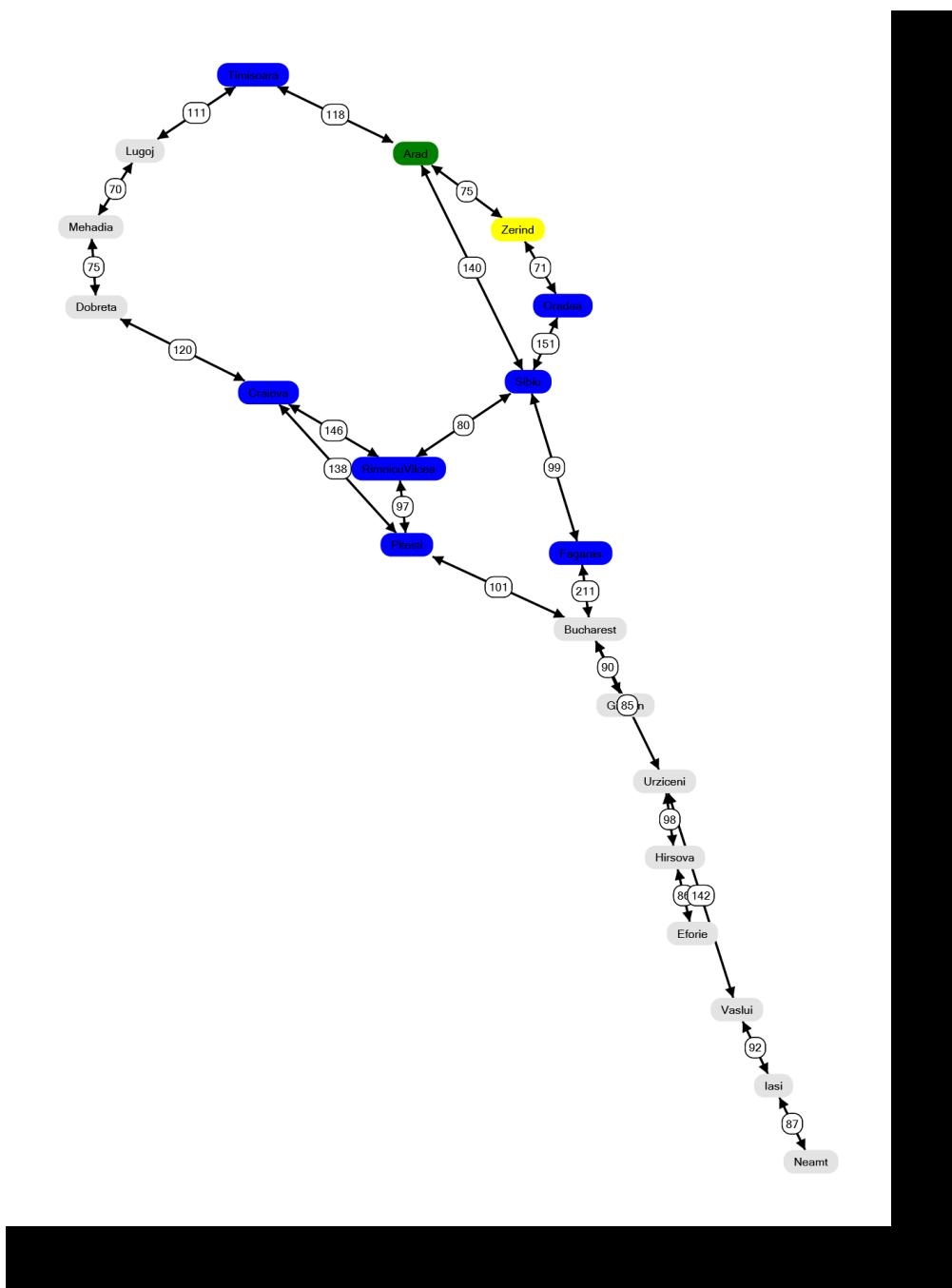
Post-Visit

The algorithm has closed the node Timisoara



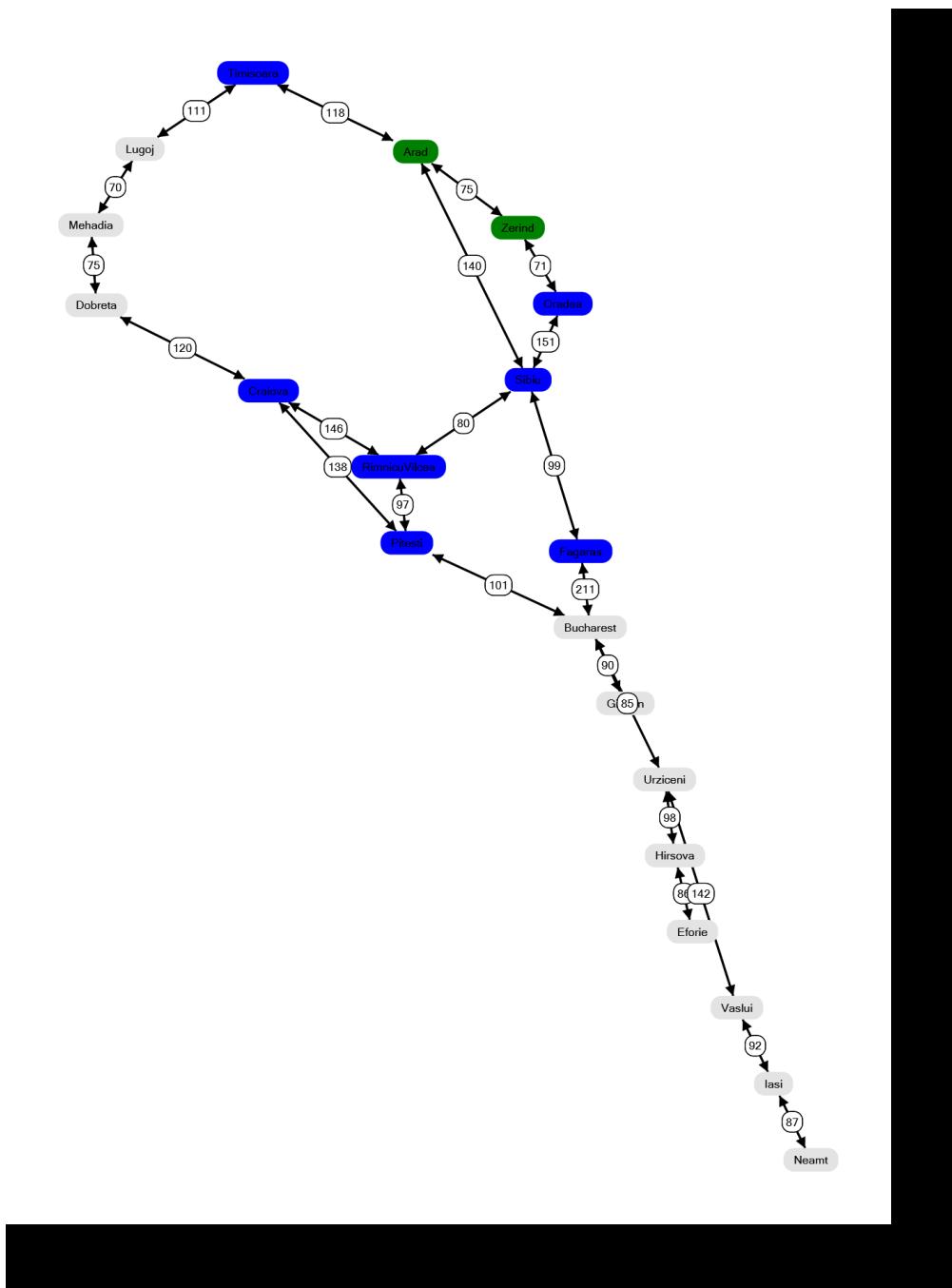
Pre-Visit

The algorithm has opened the node Zerind



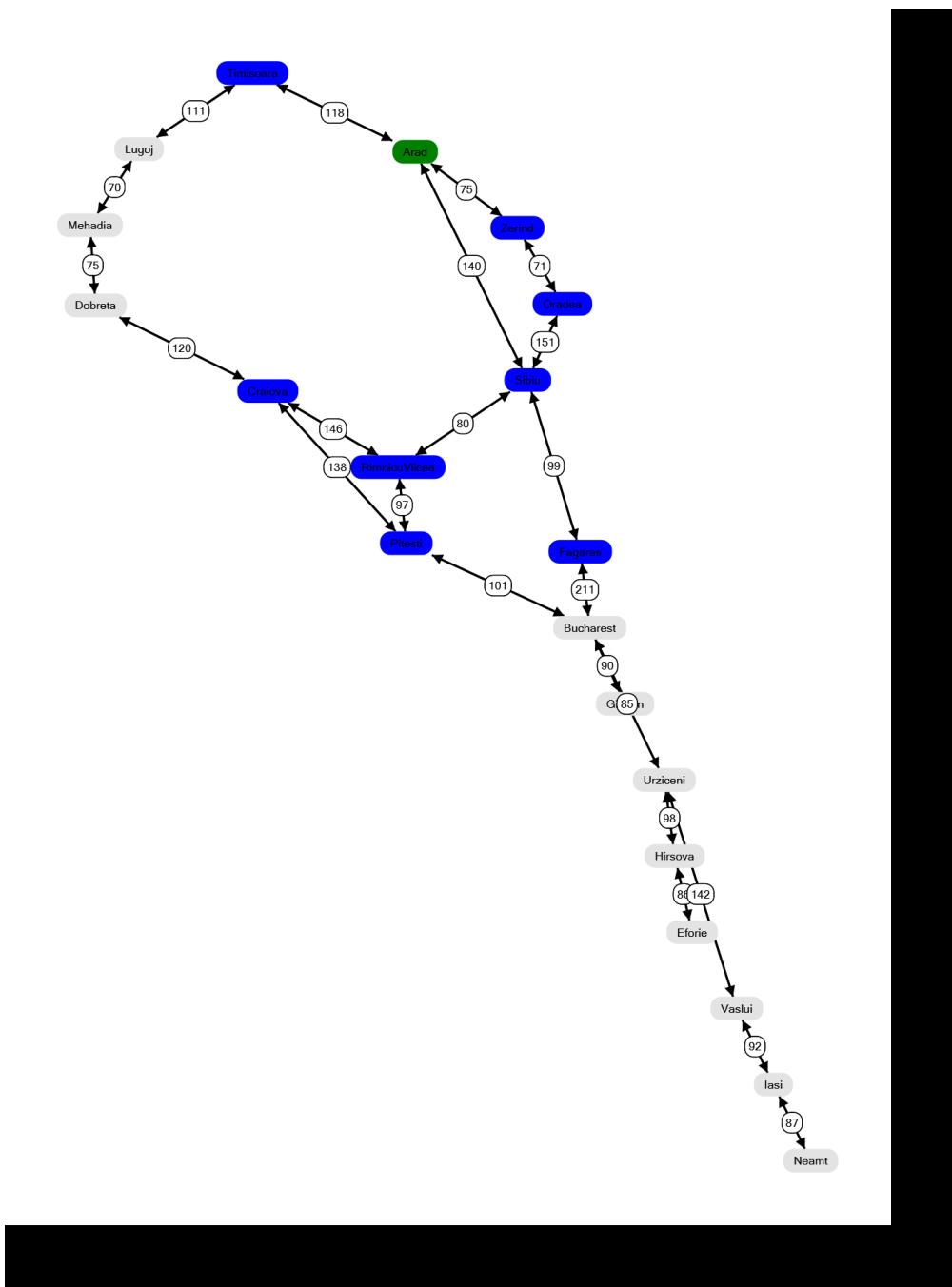
Visit

The algorithm has marked the node Zerind as visited



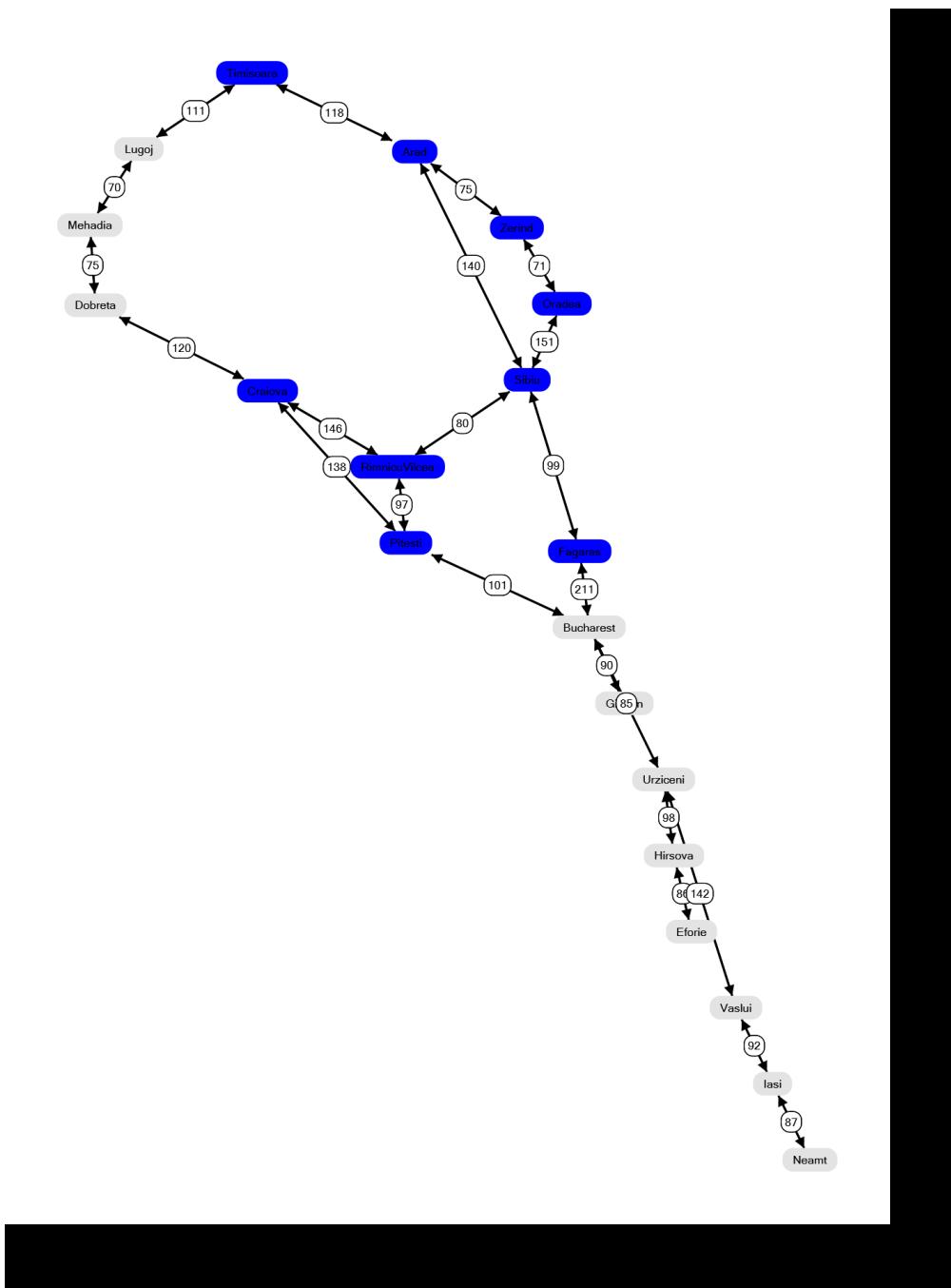
Post-Visit

The algorithm has closed the node Zerind



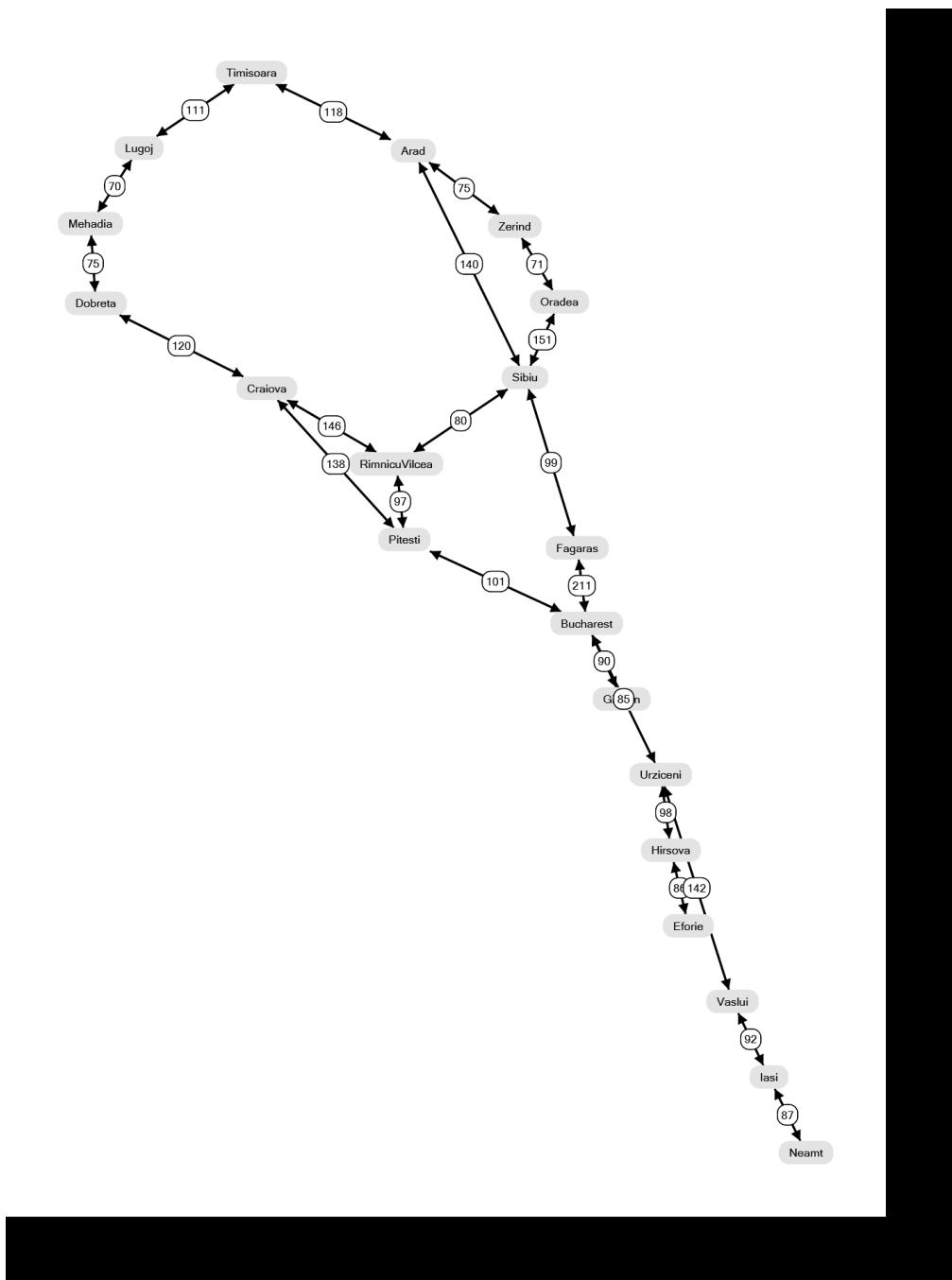
Post-Visit

The algorithm has closed the node Arad



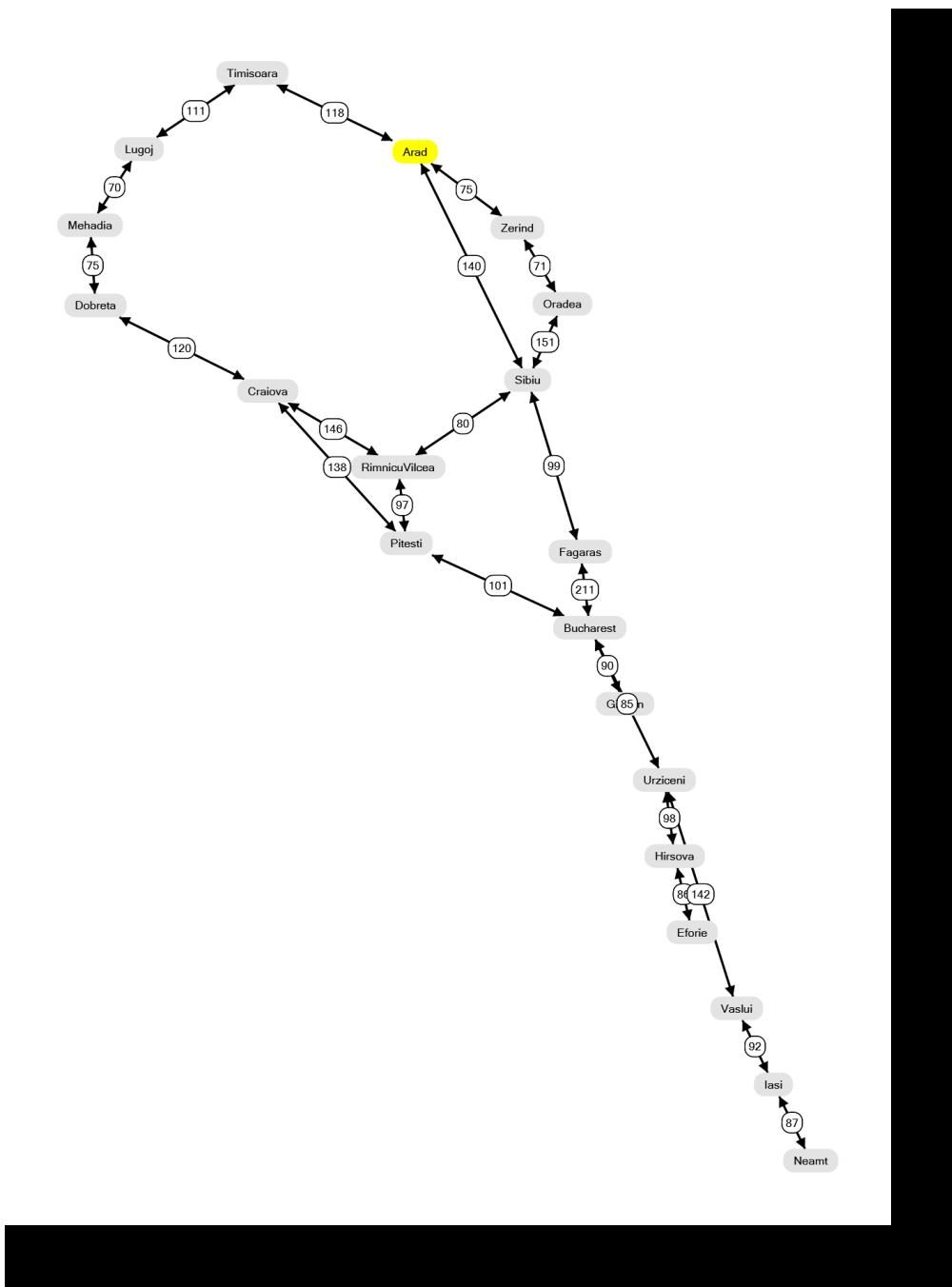
Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



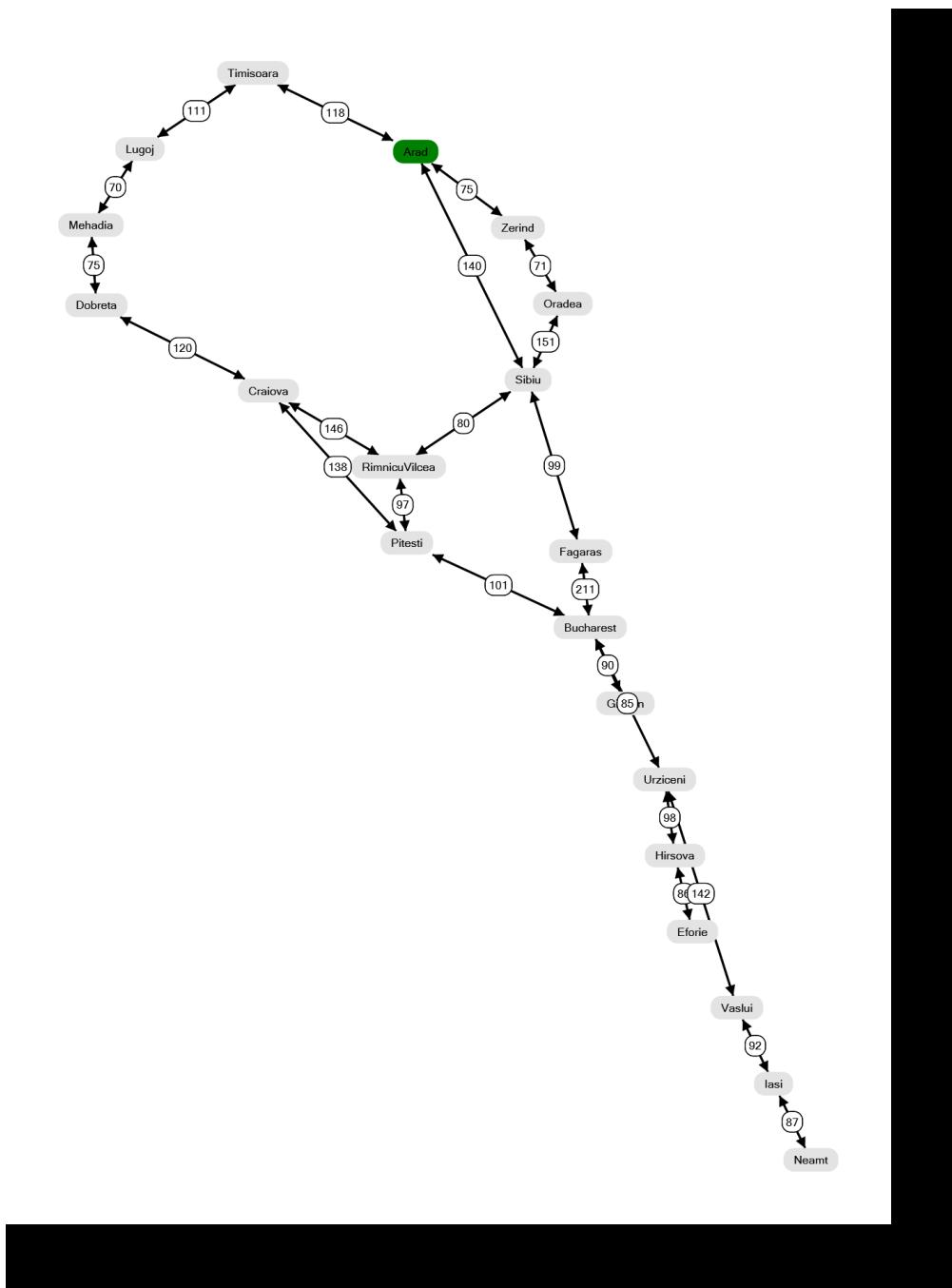
Pre-Visit

The algorithm has opened the node Arad



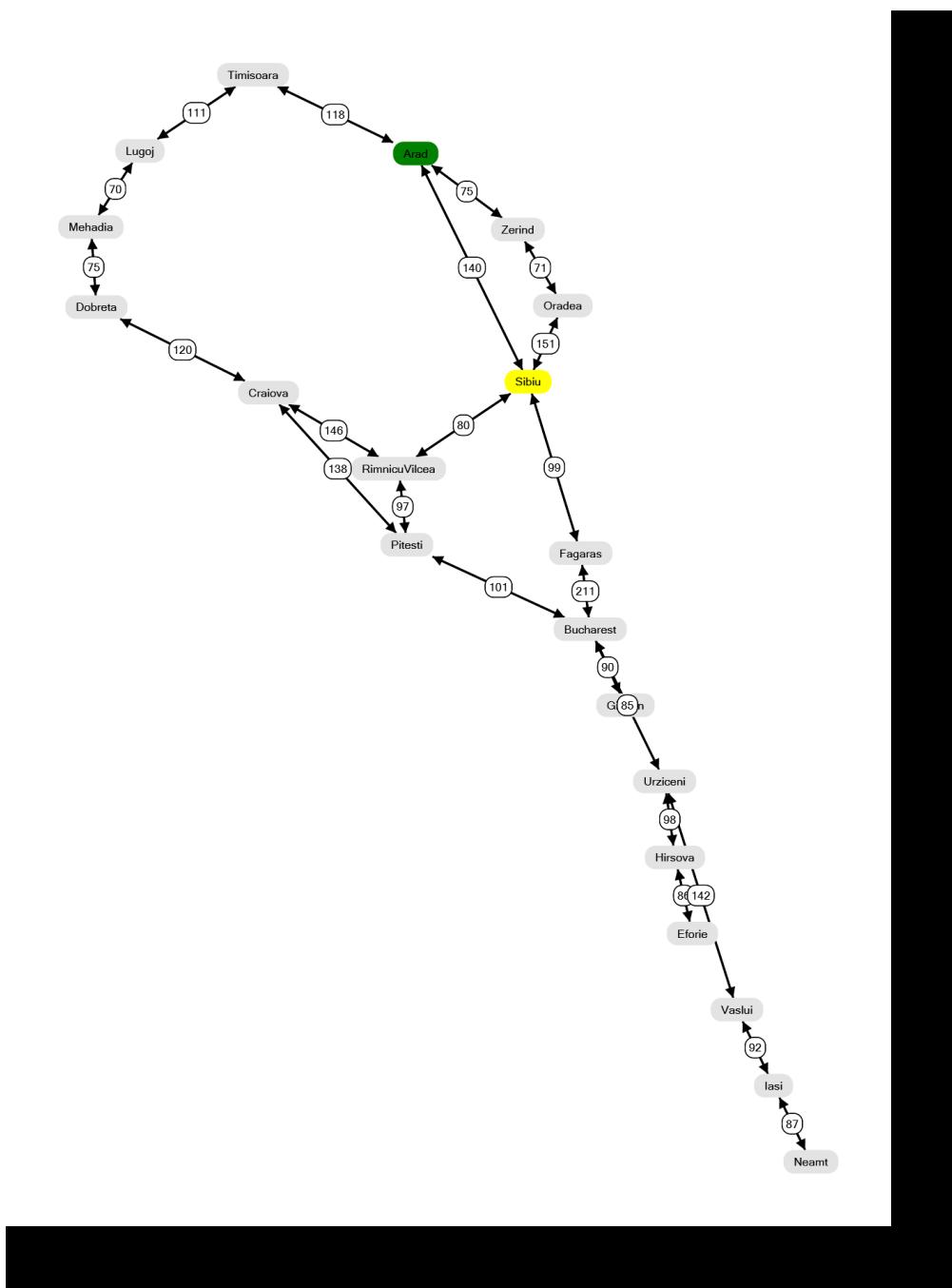
Visit

The algorithm has marked the node Arad as visited



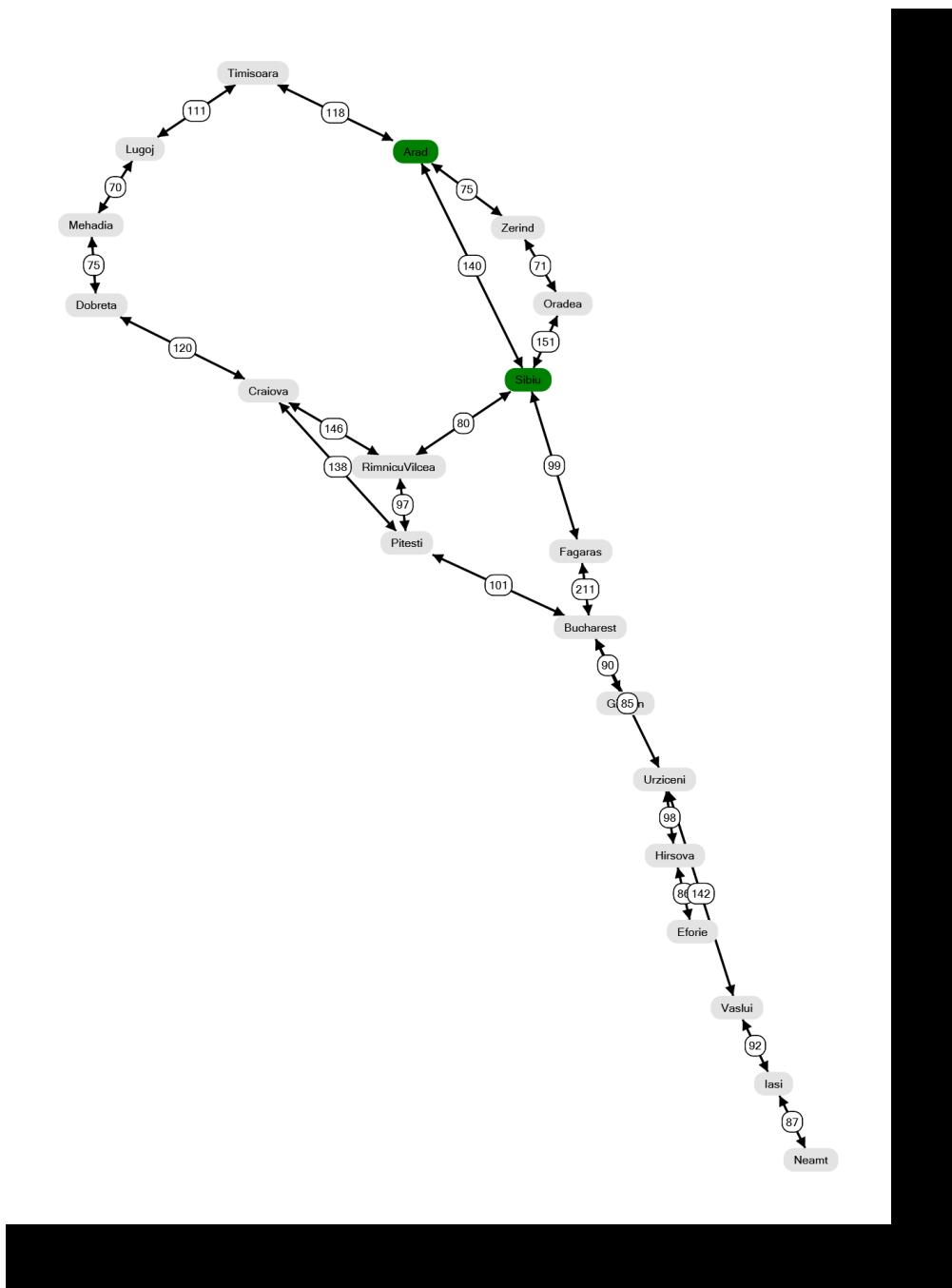
Pre-Visit

The algorithm has opened the node Sibiu



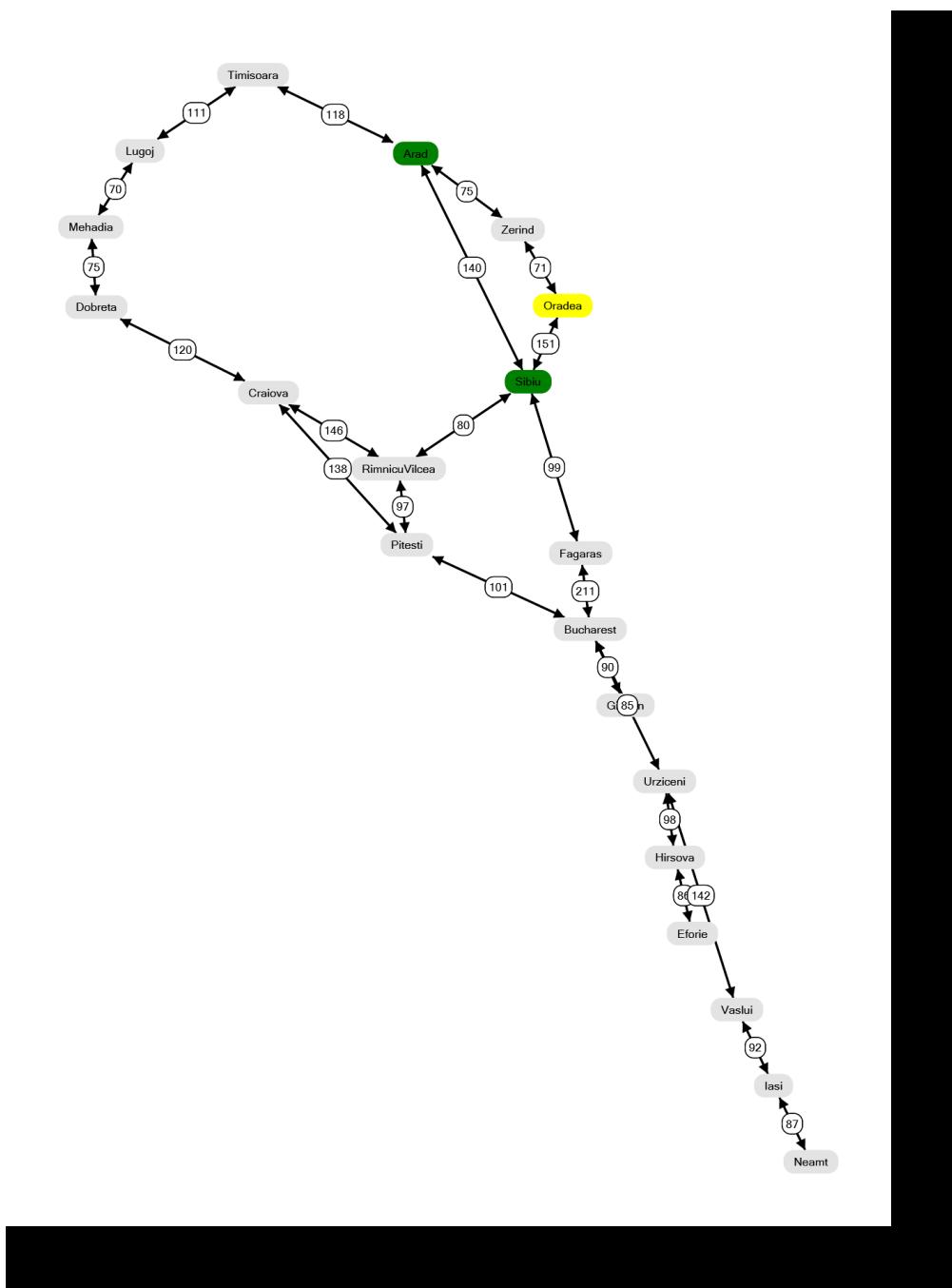
Visit

The algorithm has marked the node Sibiu as visited



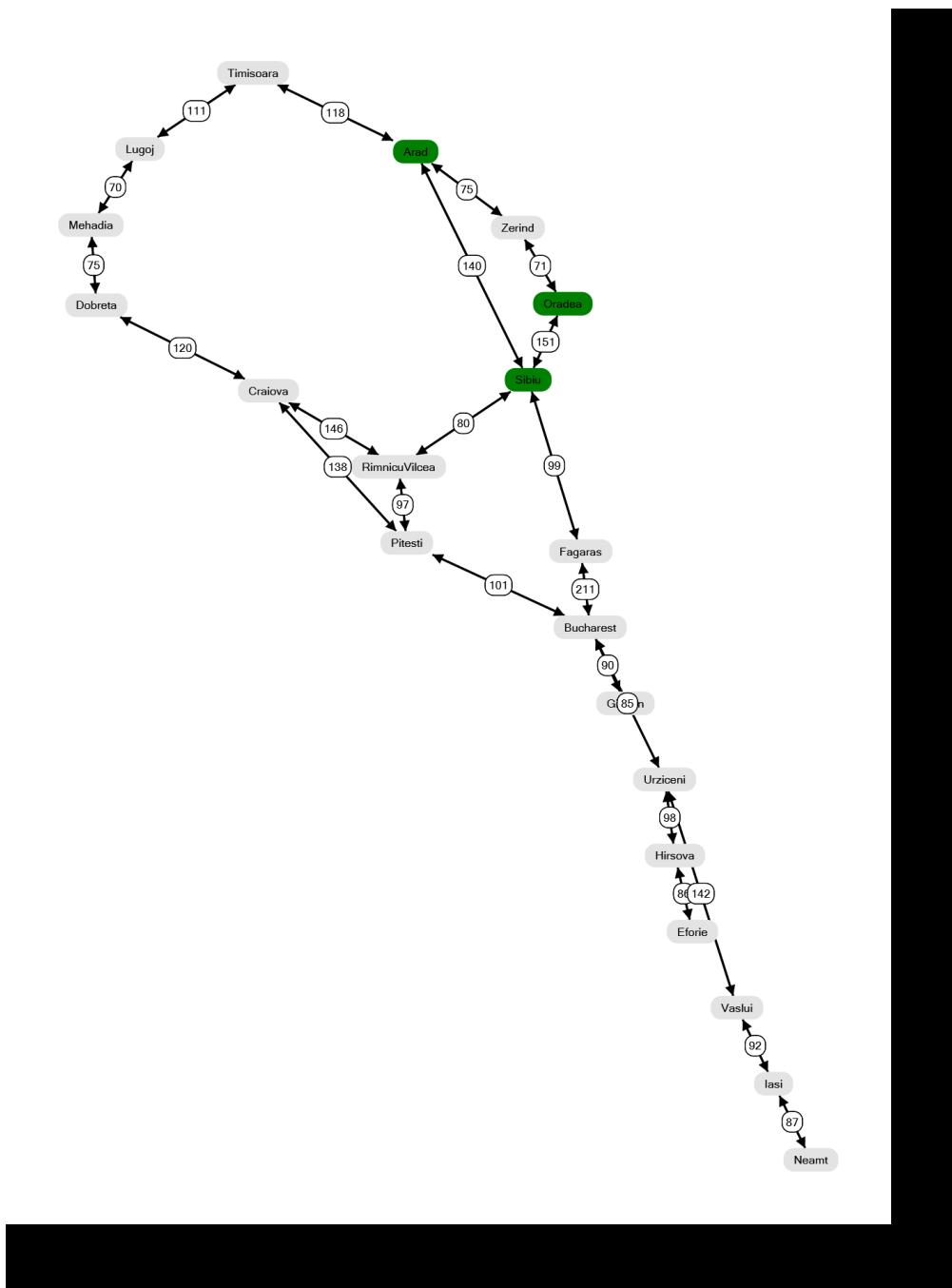
Pre-Visit

The algorithm has opened the node Oradea



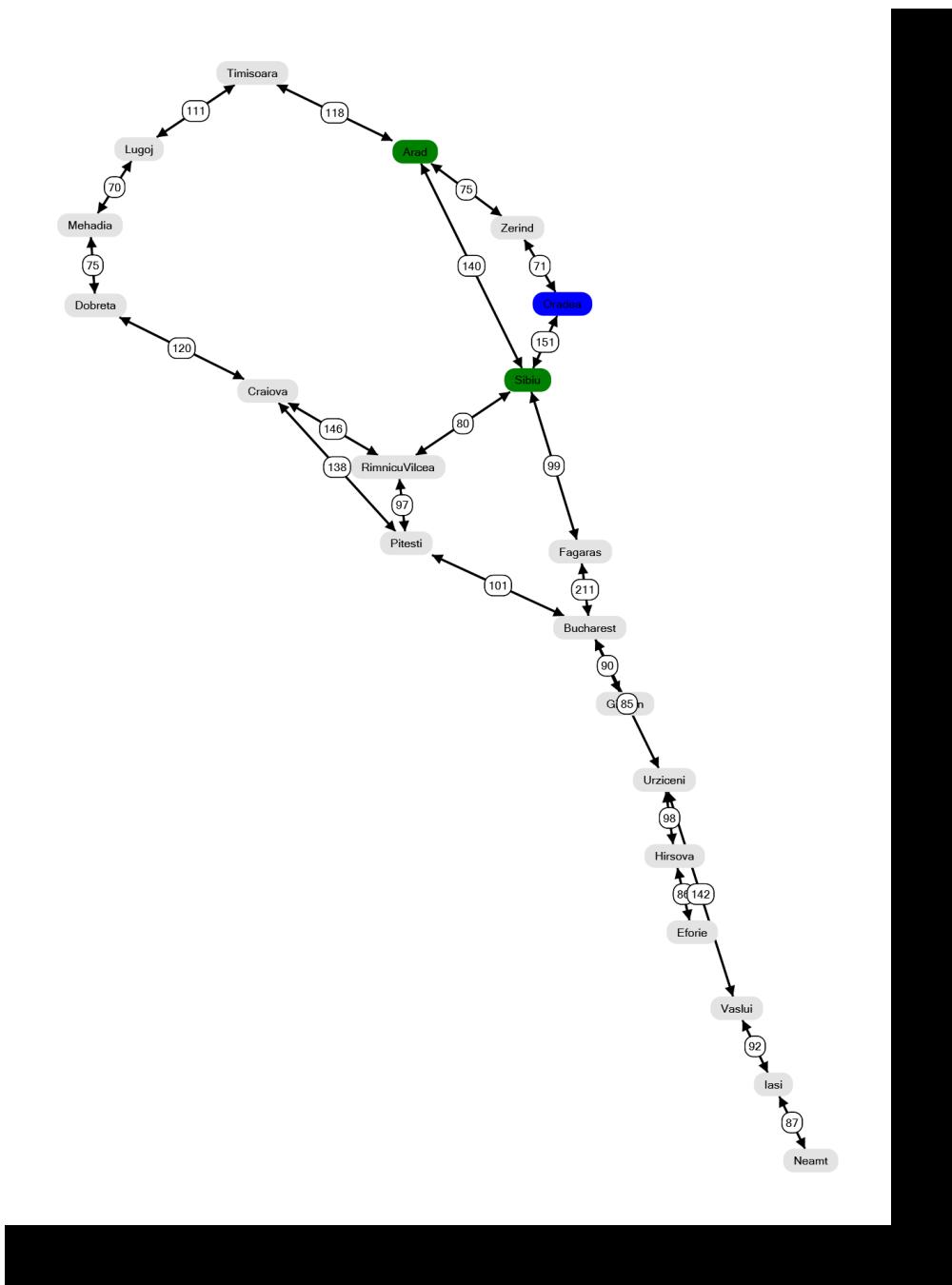
Visit

The algorithm has marked the node Oradea as visited



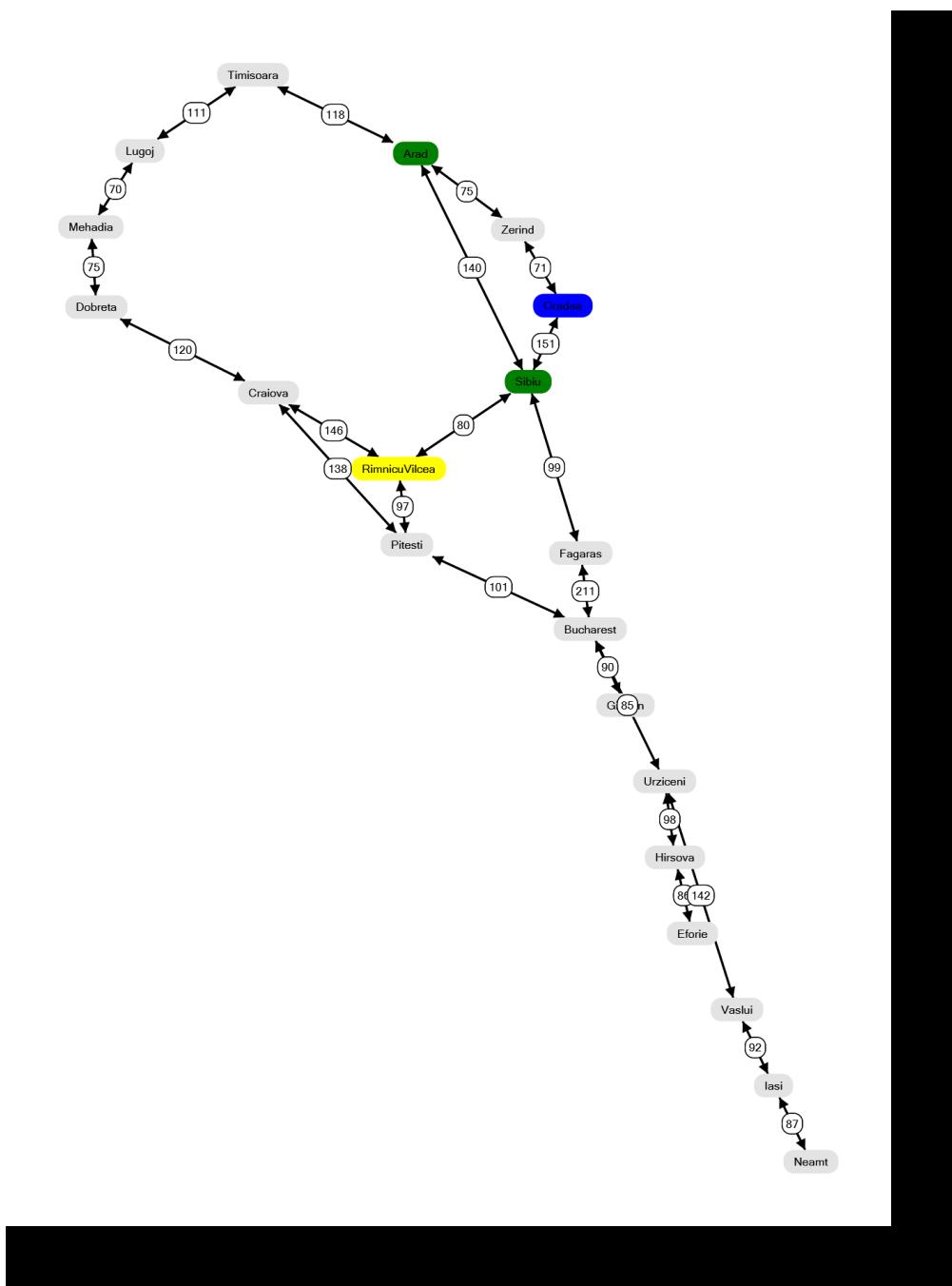
Post-Visit

The algorithm has closed the node Oradea



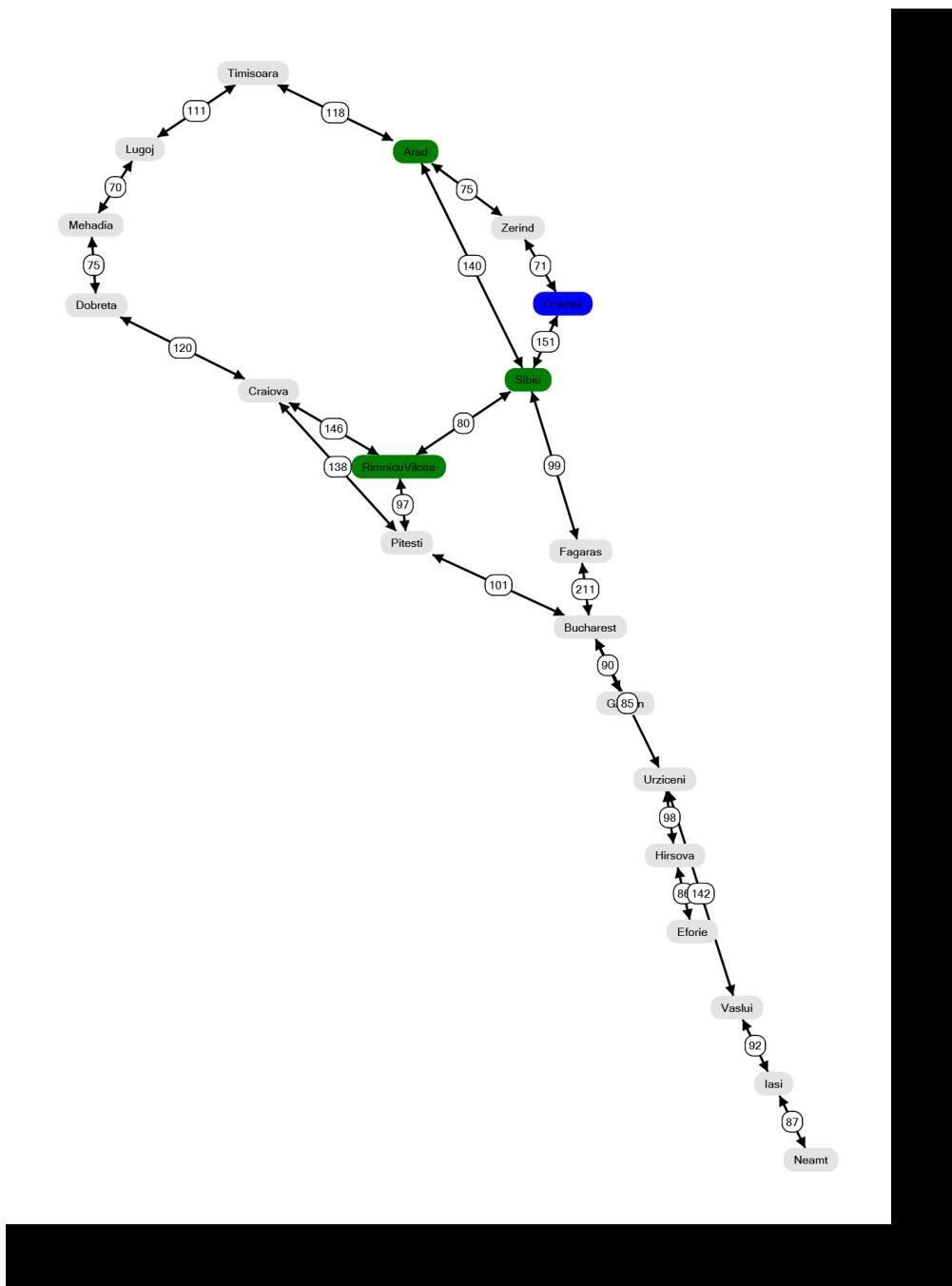
Pre-Visit

The algorithm has opened the node Rimnicu Vilcea



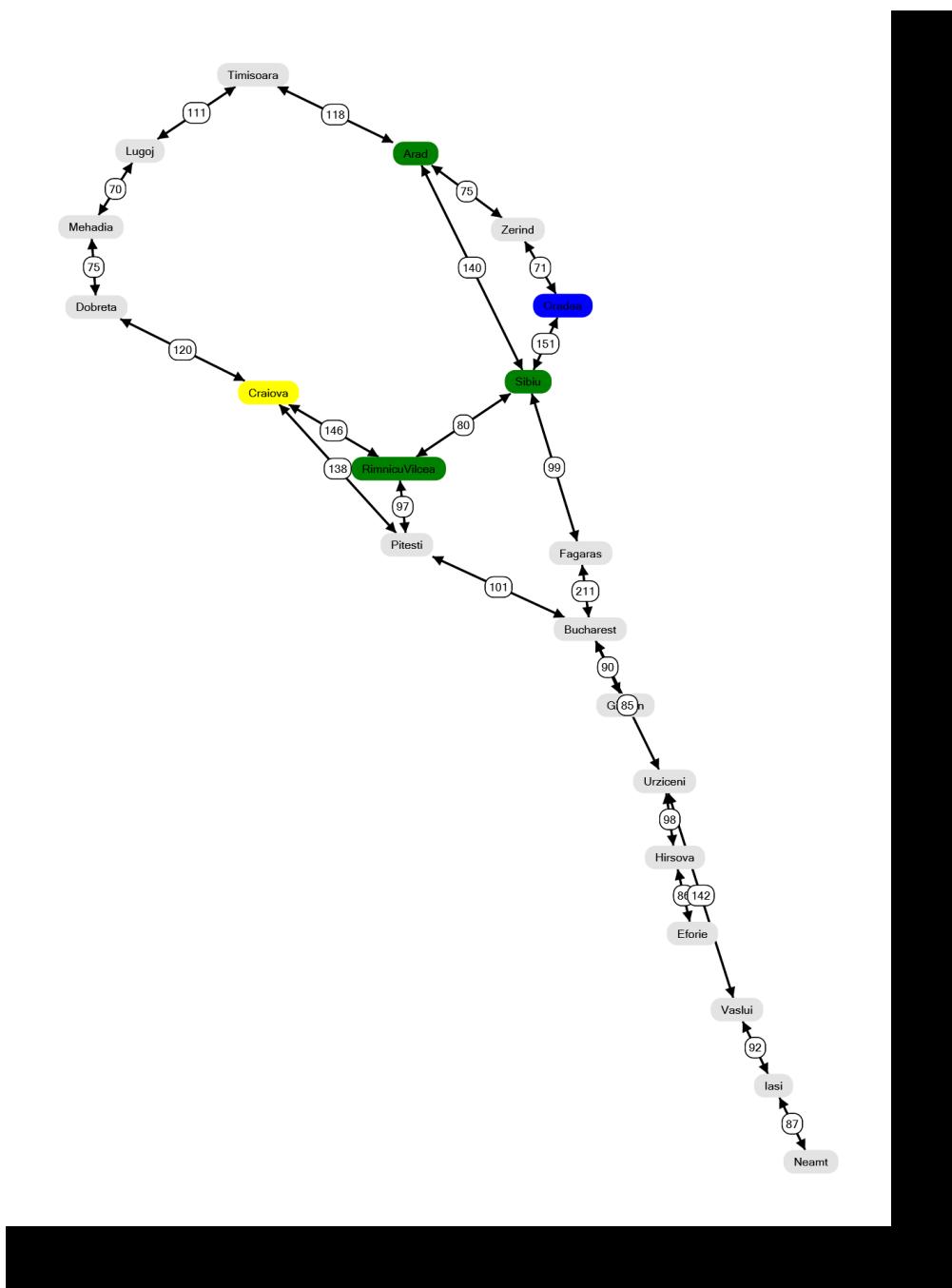
Visit

The algorithm has marked the node Rimnicu Vilcea as visited



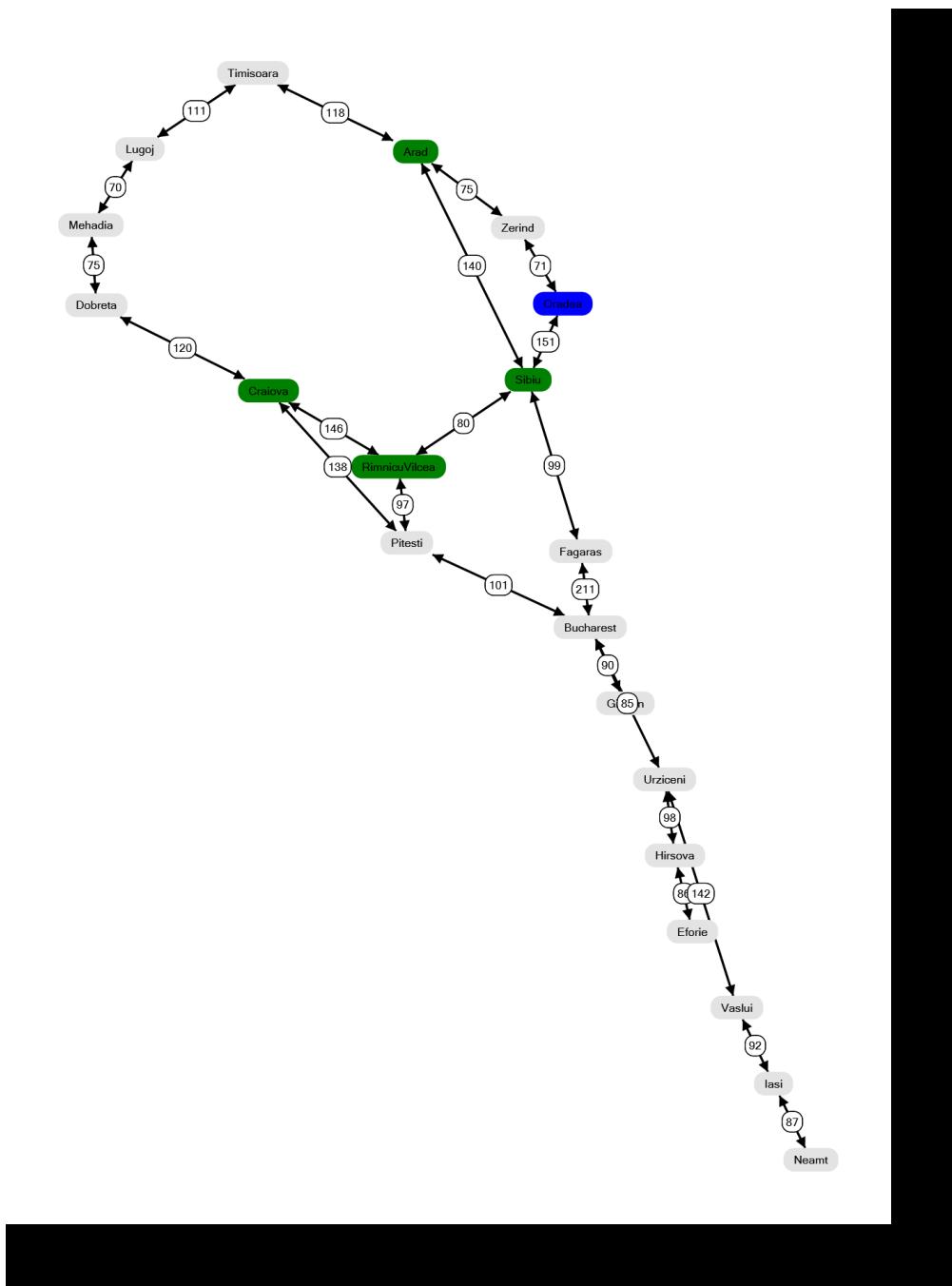
Pre-Visit

The algorithm has opened the node Craiova



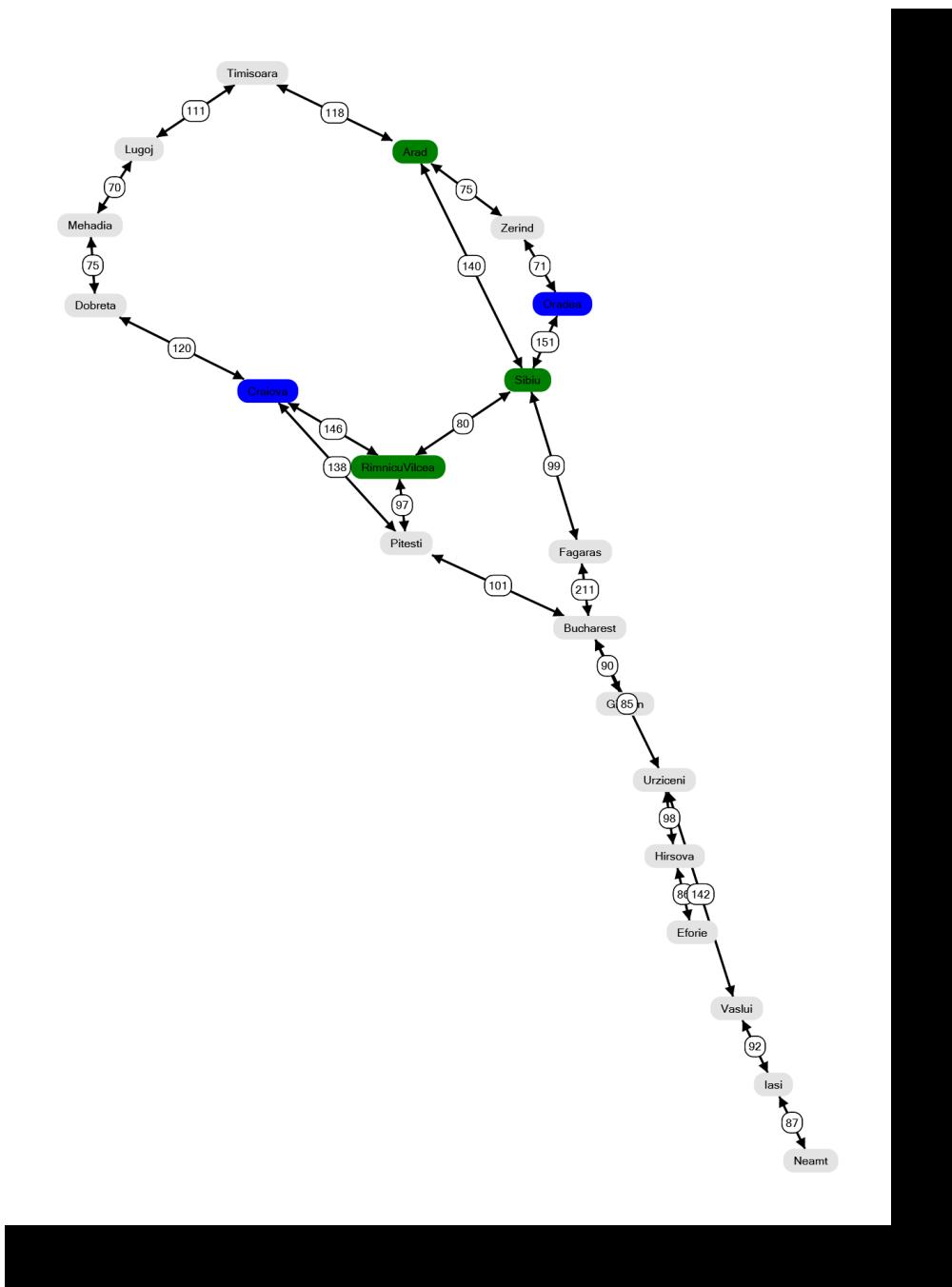
Visit

The algorithm has marked the node Craiova as visited



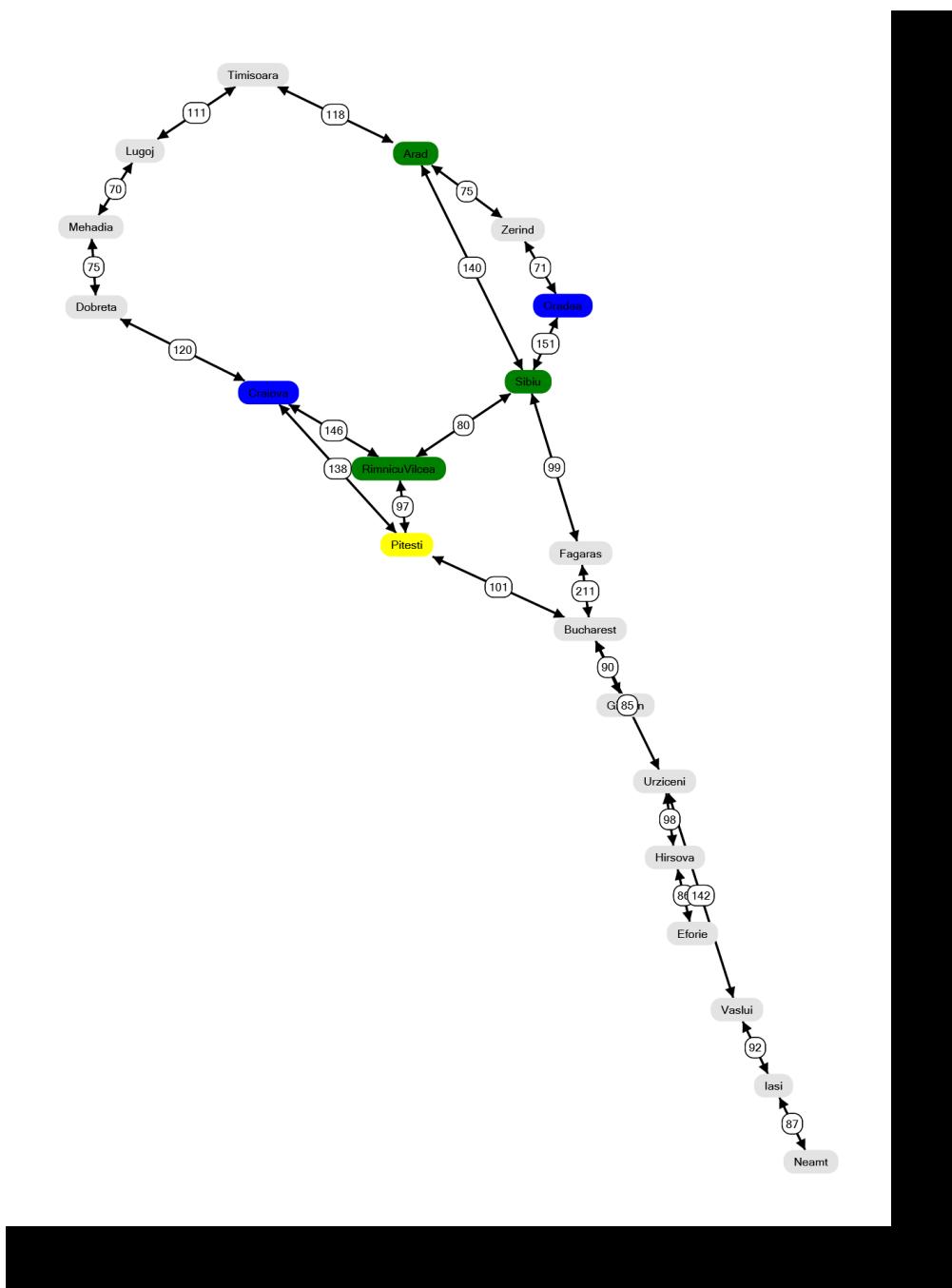
Post-Visit

The algorithm has closed the node Craiova



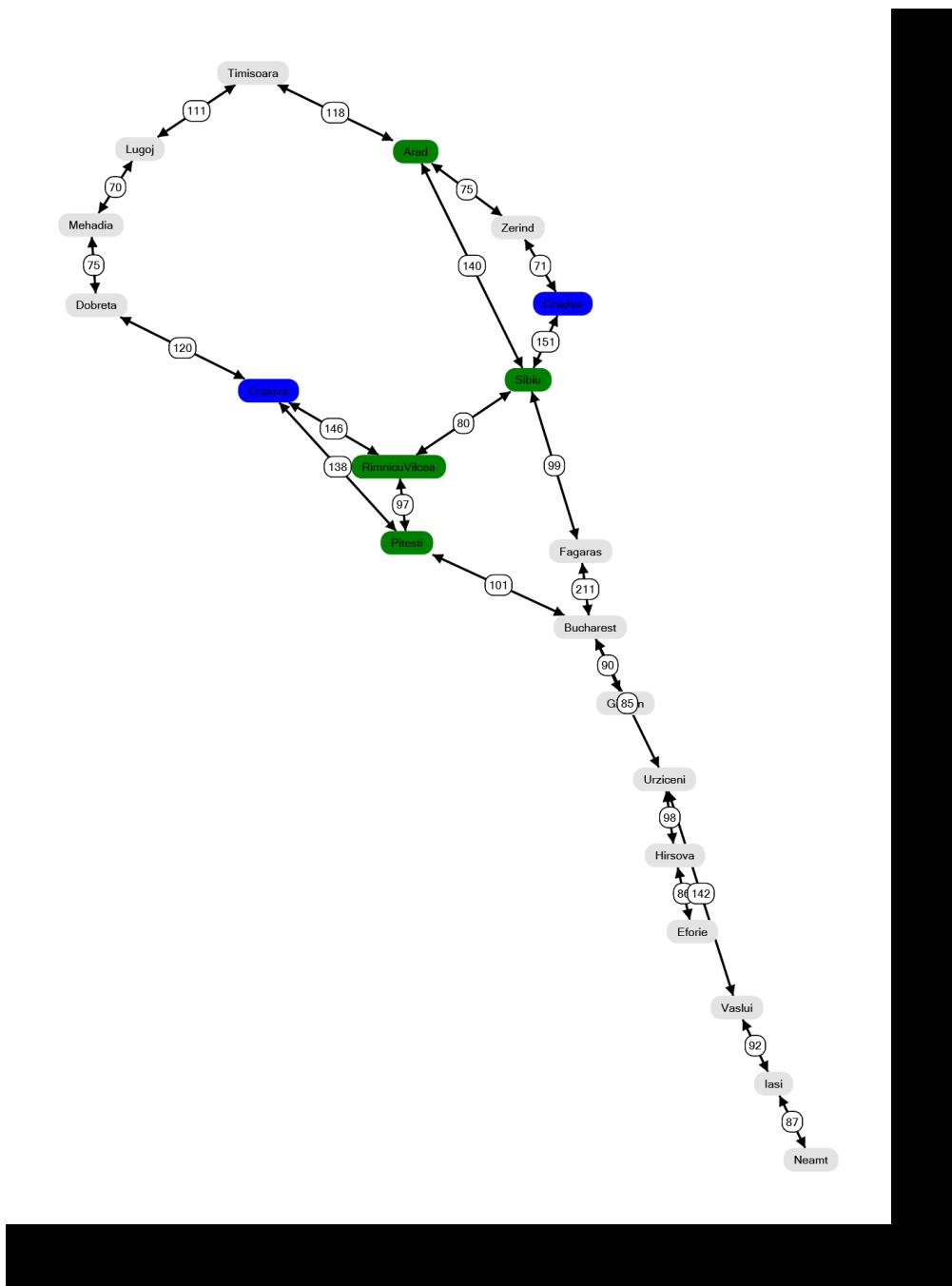
Pre-Visit

The algorithm has opened the node Pitesti



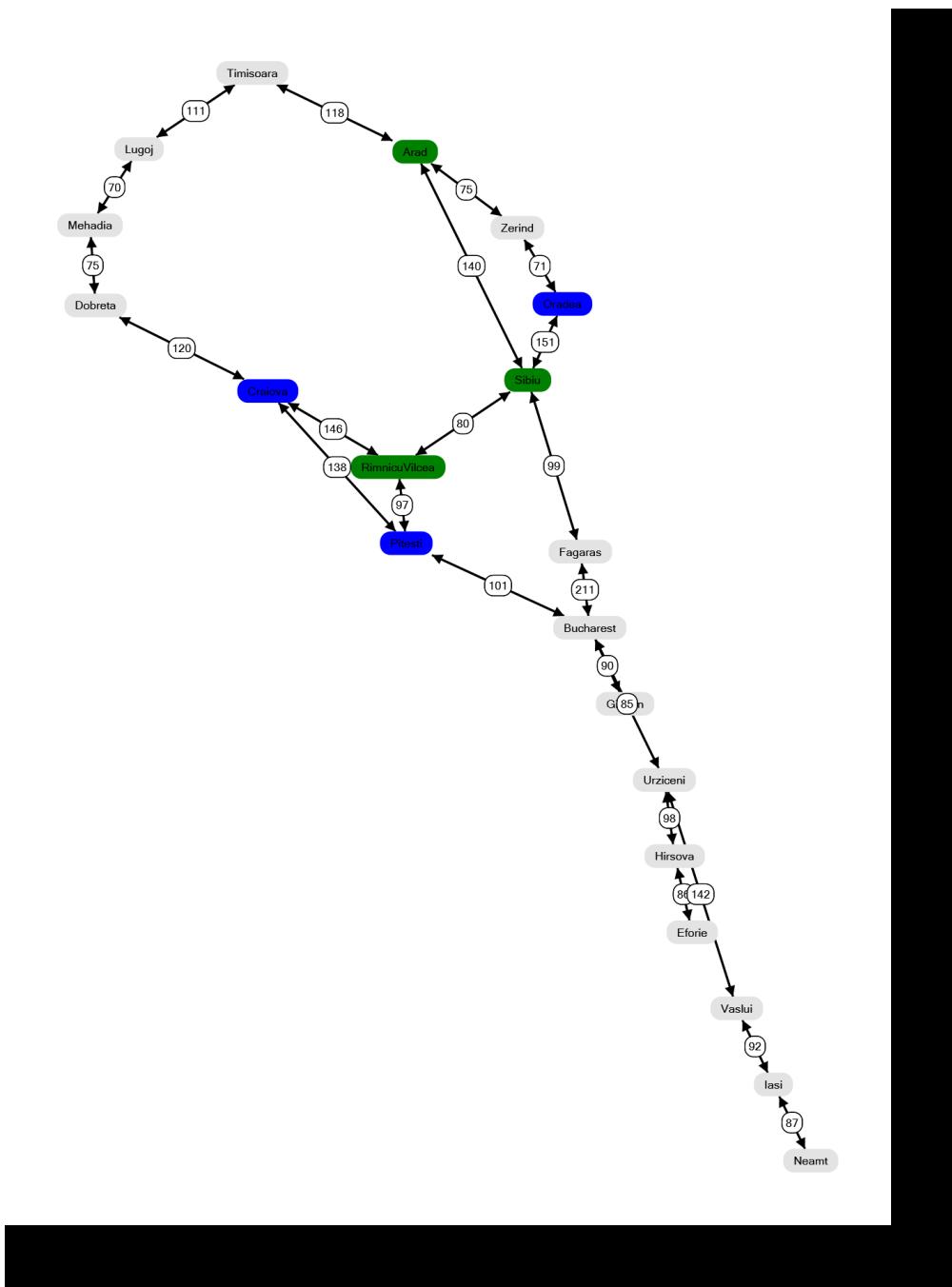
Visit

The algorithm has marked the node Pitesti as visited



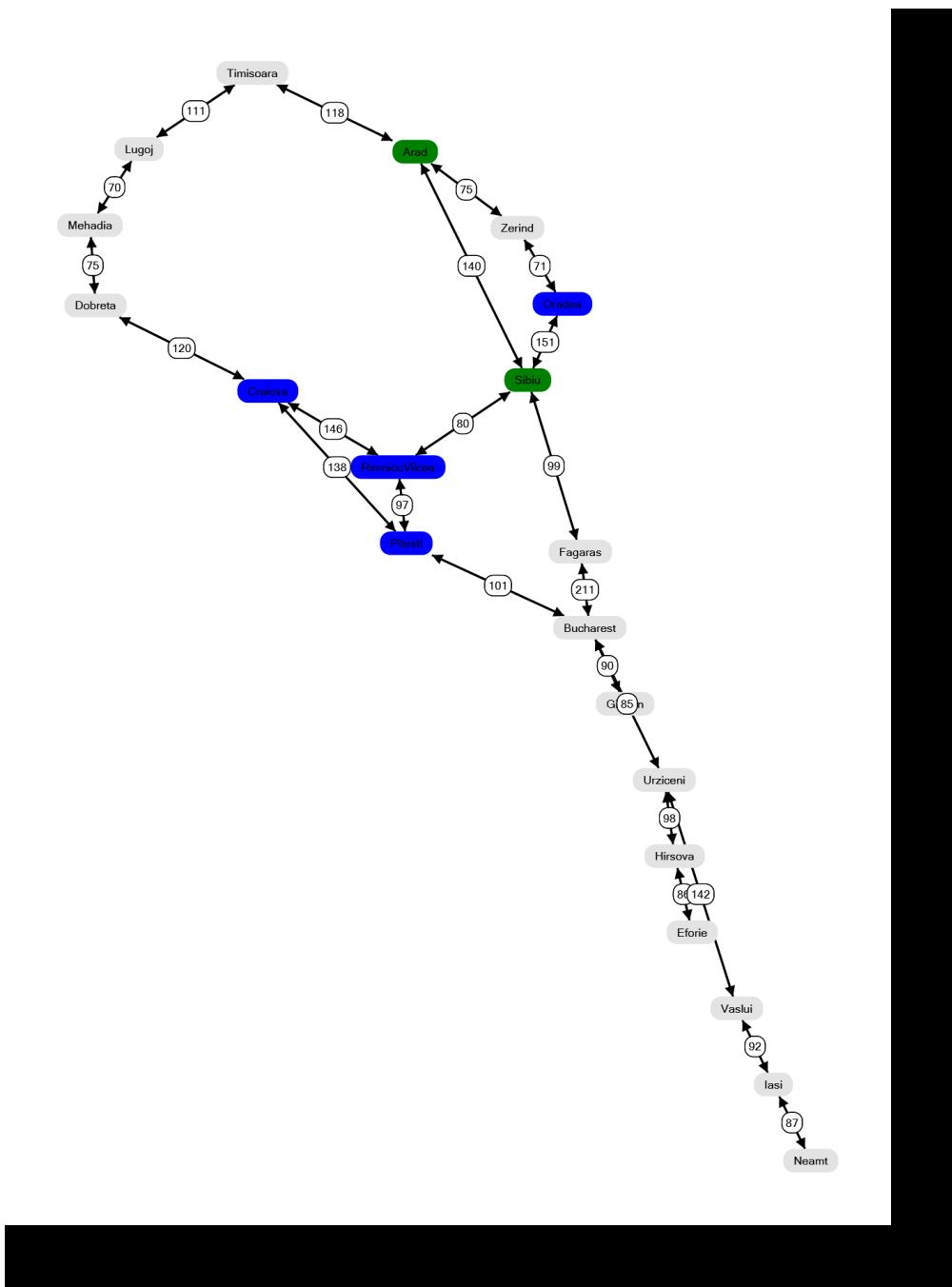
Post-Visit

The algorithm has closed the node Pitesti



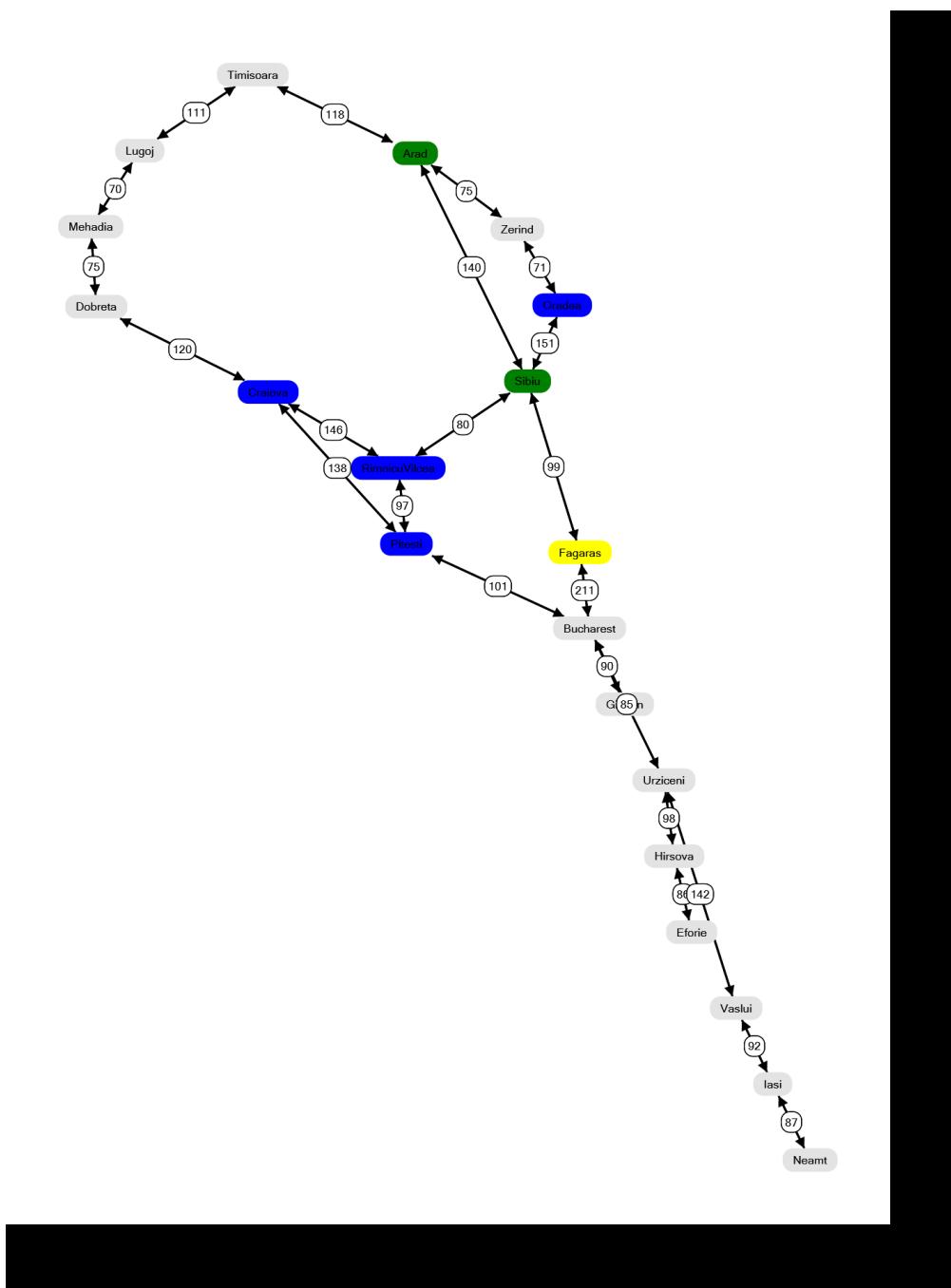
Post-Visit

The algorithm has closed the node Rimnicu Vilcea



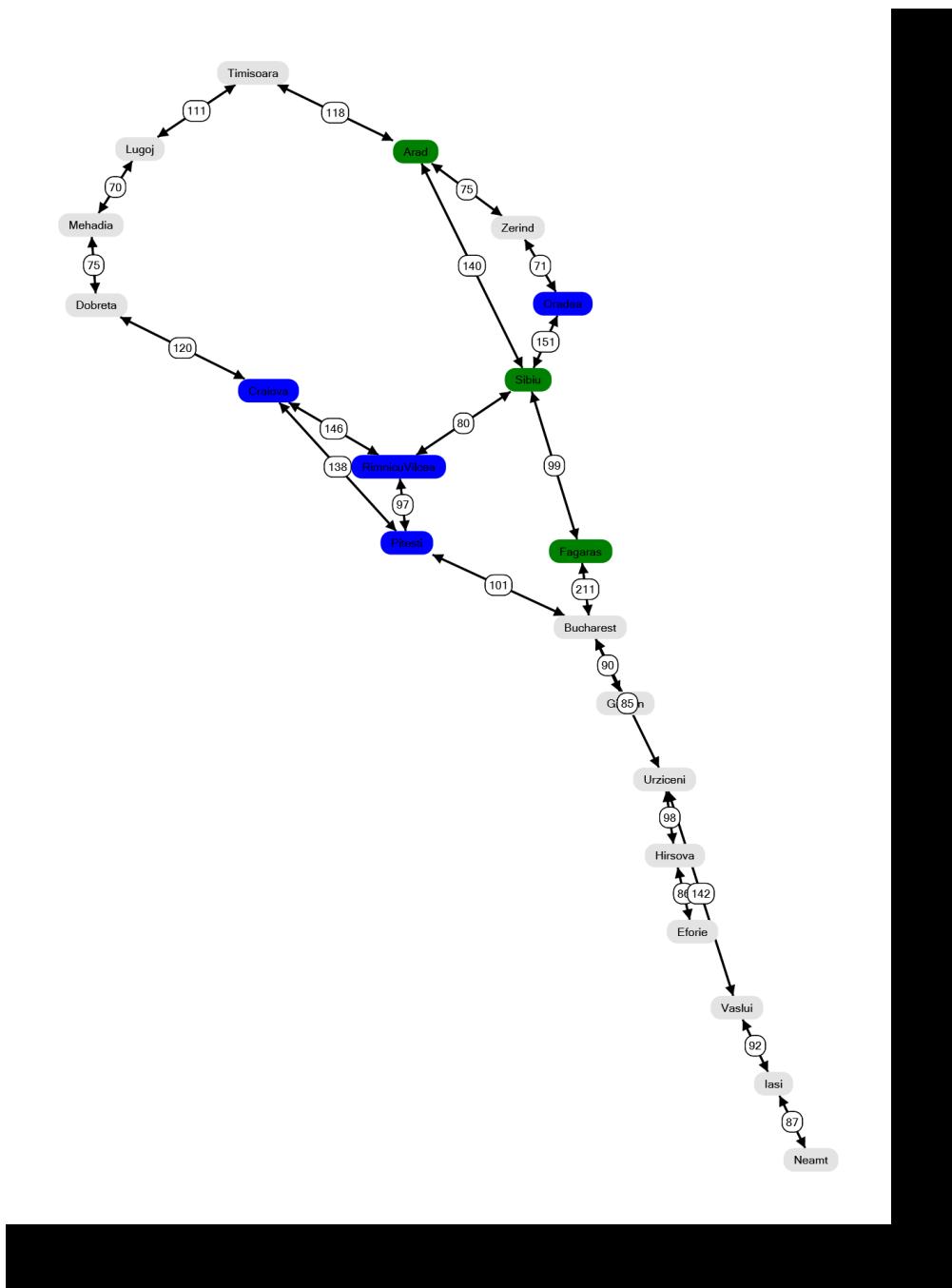
Pre-Visit

The algorithm has opened the node Fagaras



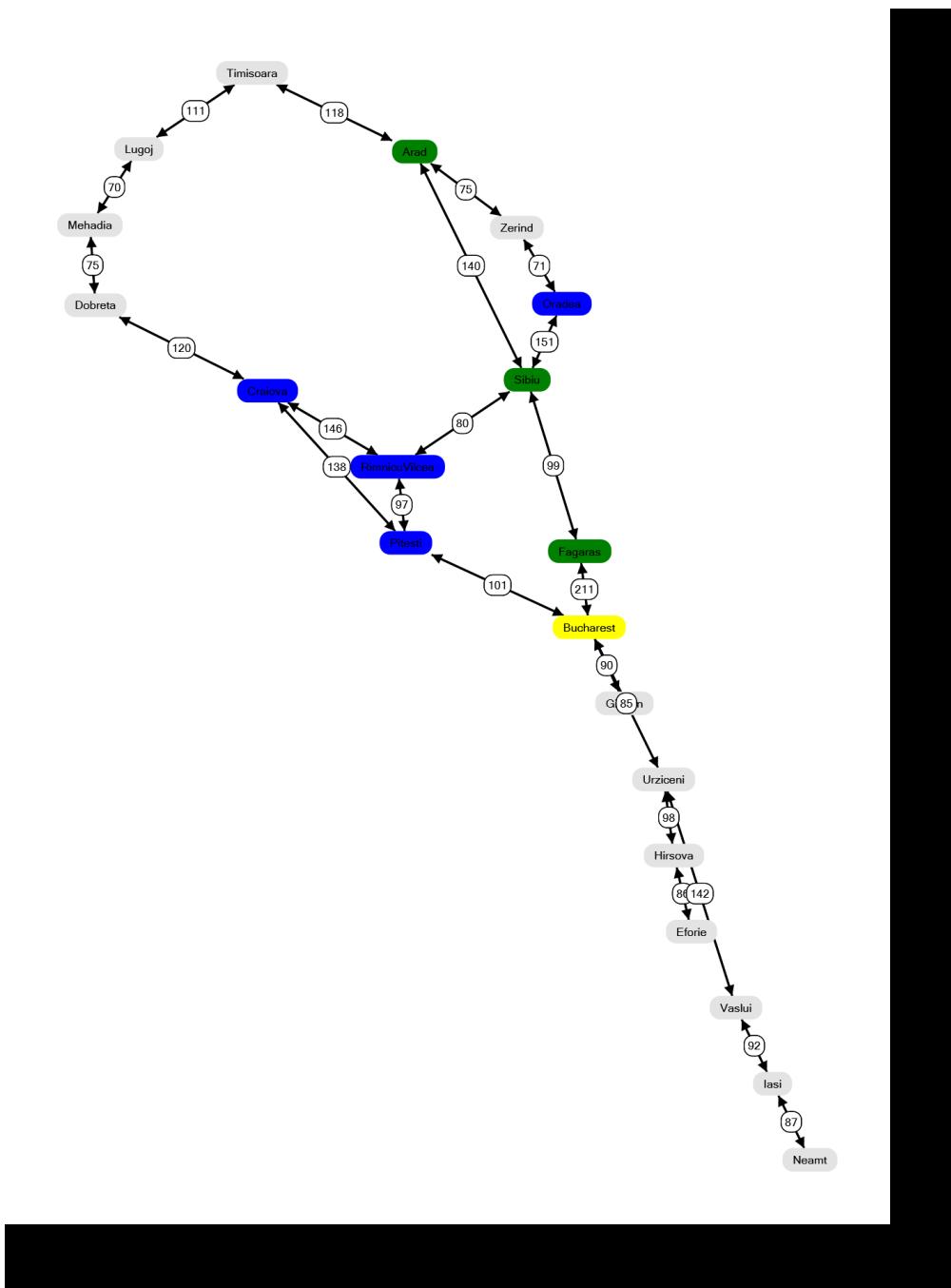
Visit

The algorithm has marked the node Fagaras as visited



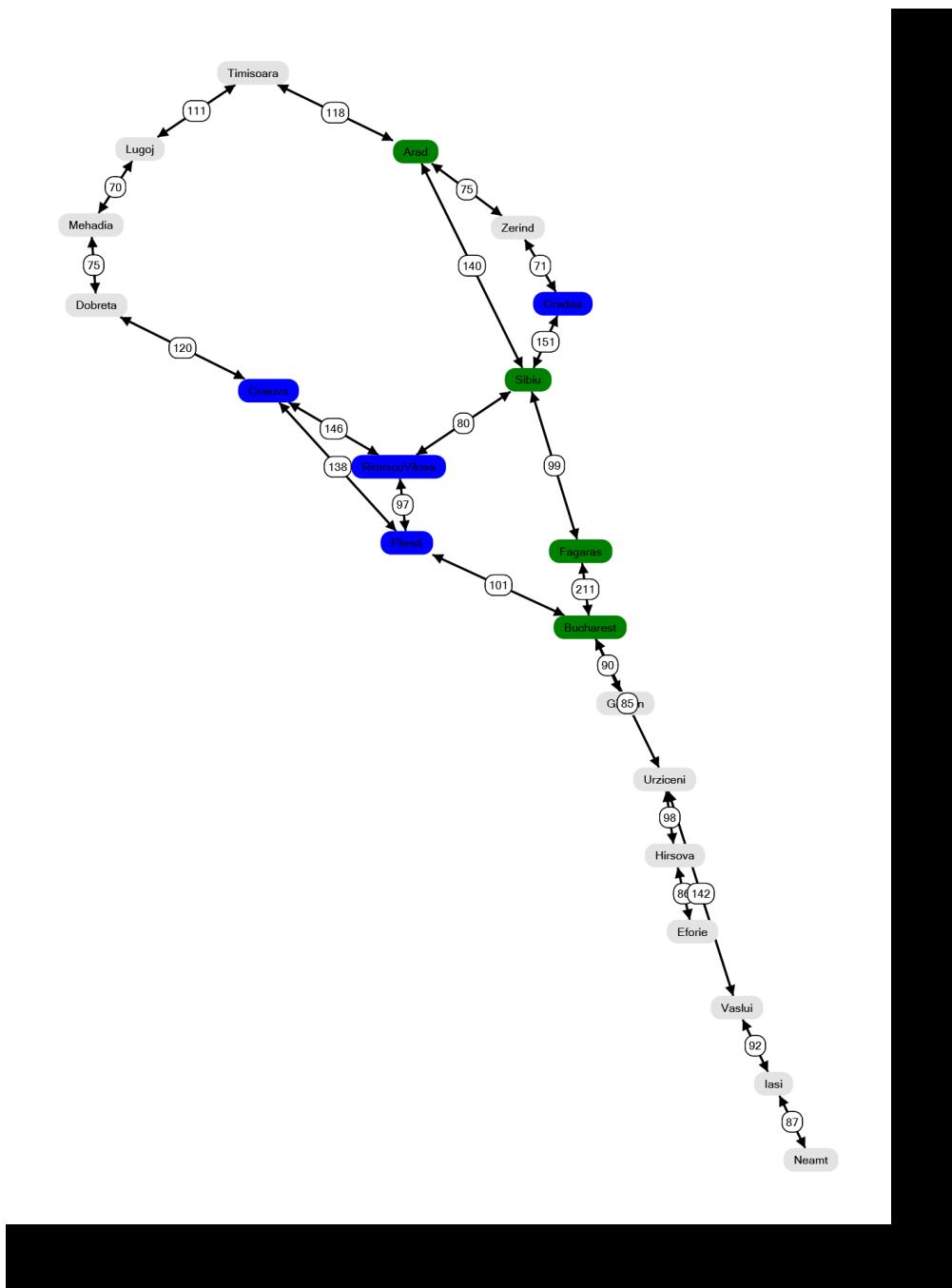
Pre-Visit

The algorithm has opened the node Bucharest



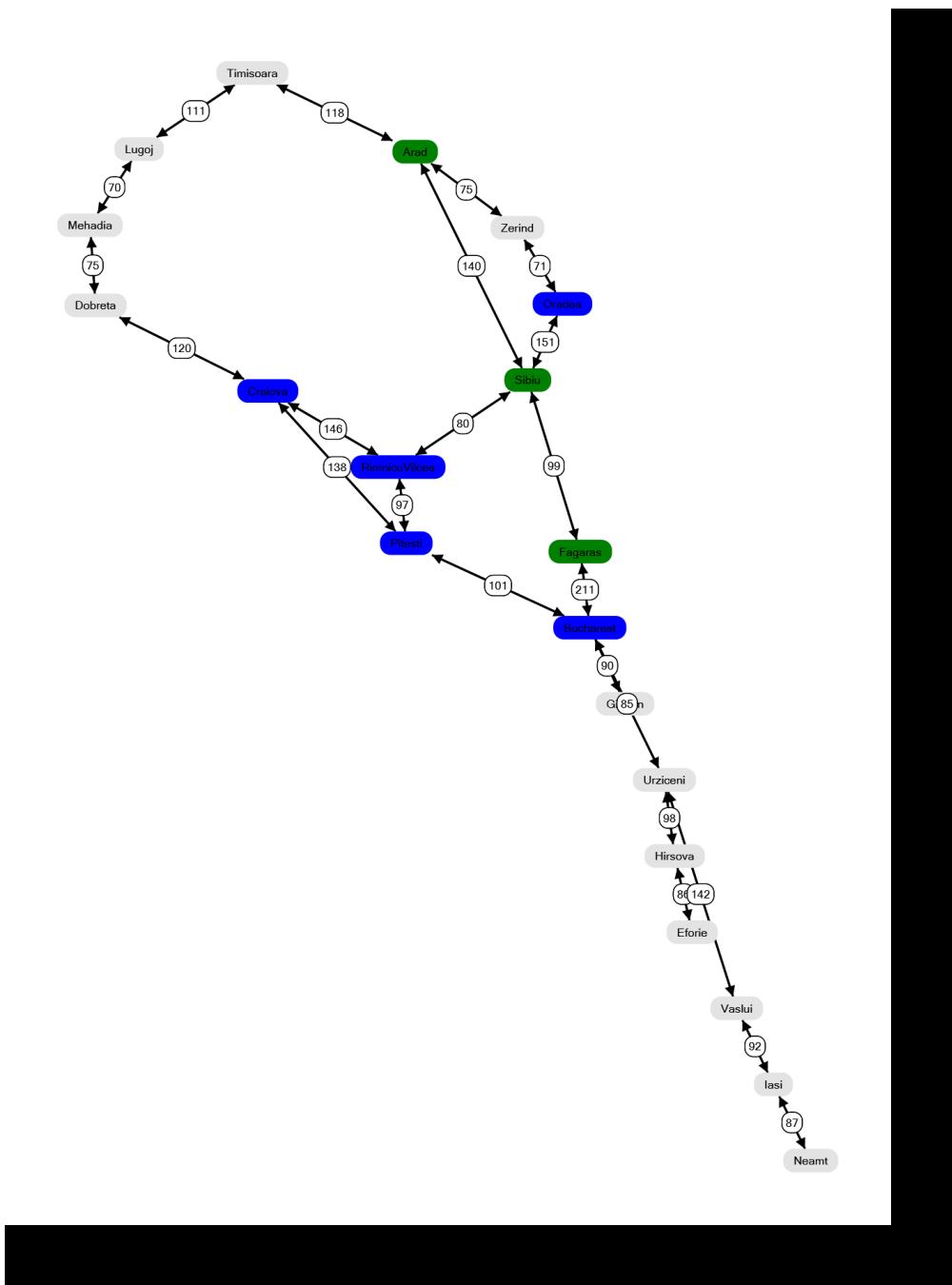
Visit

The algorithm has marked the node Bucharest as visited



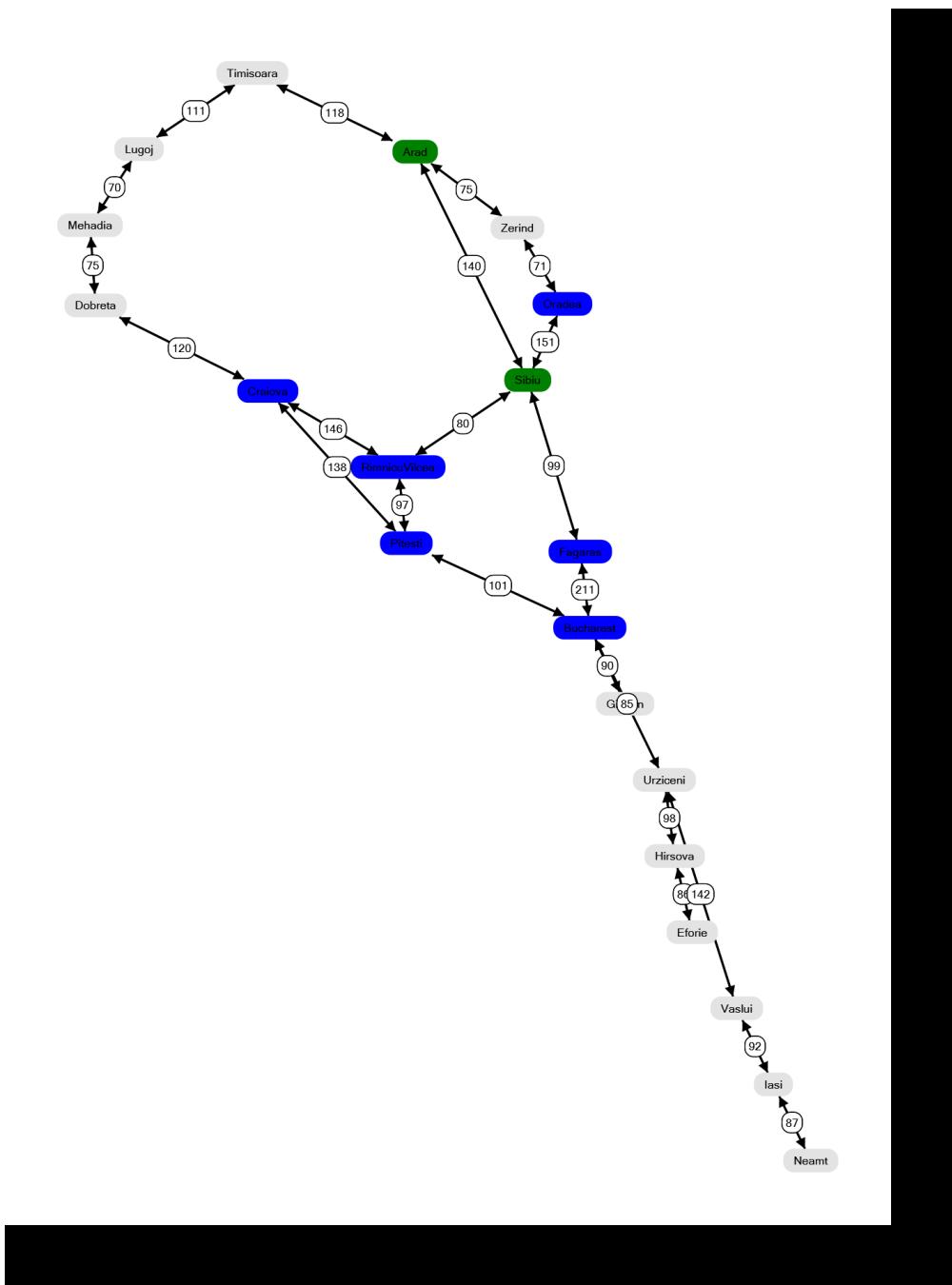
Post-Visit

The algorithm has closed the node Bucharest



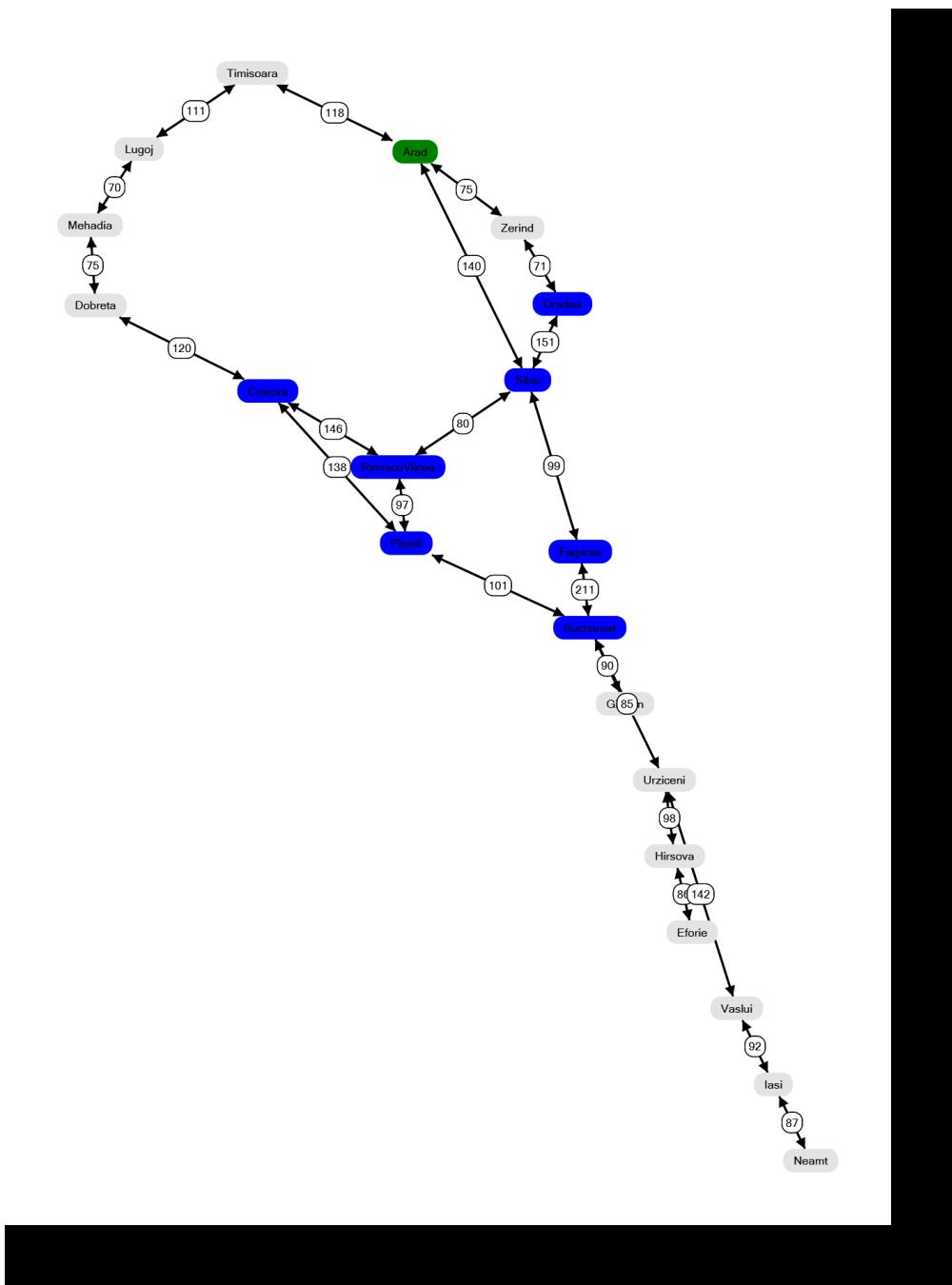
Post-Visit

The algorithm has closed the node Fagaras



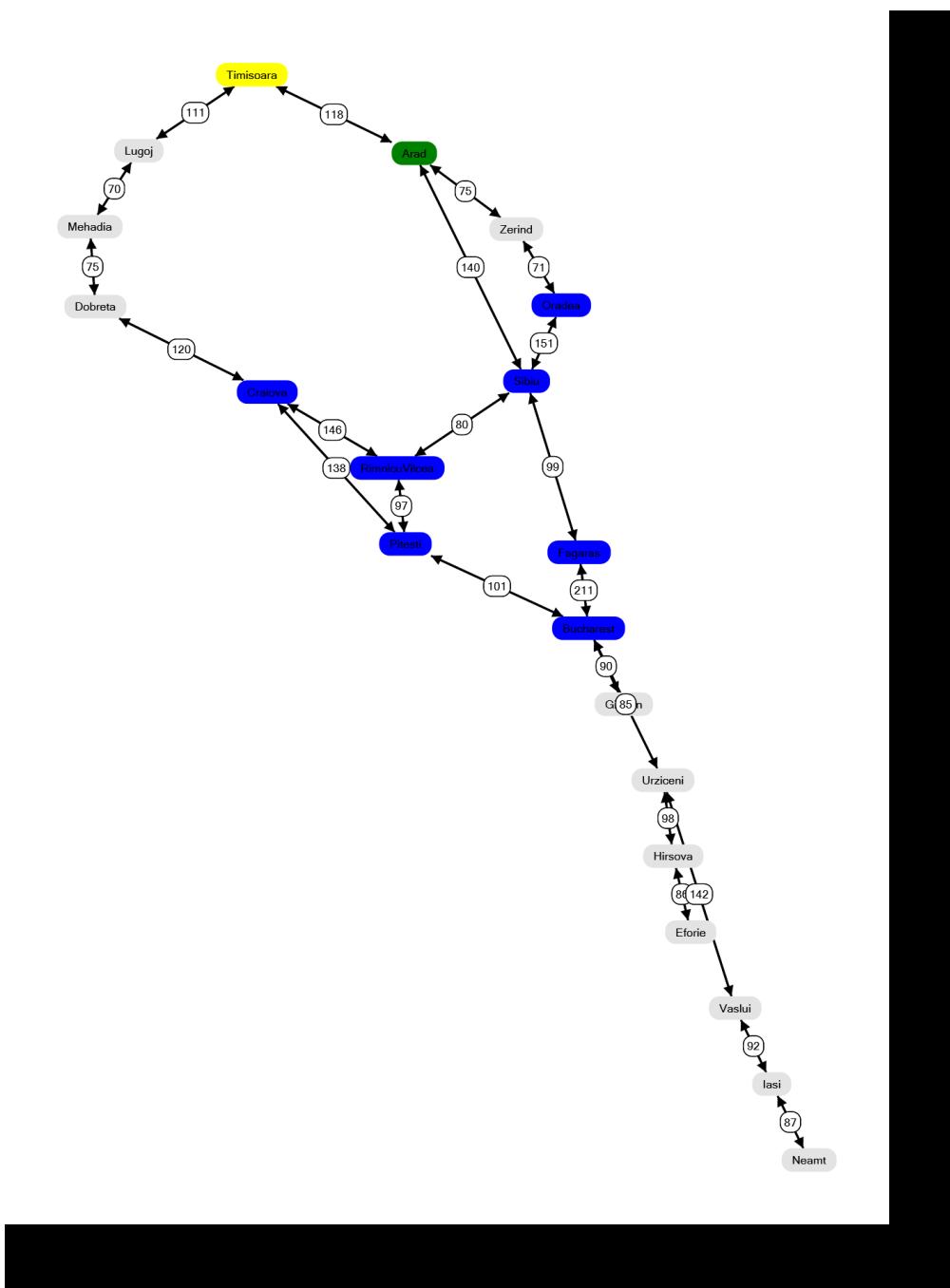
Post-Visit

The algorithm has closed the node Sibiu



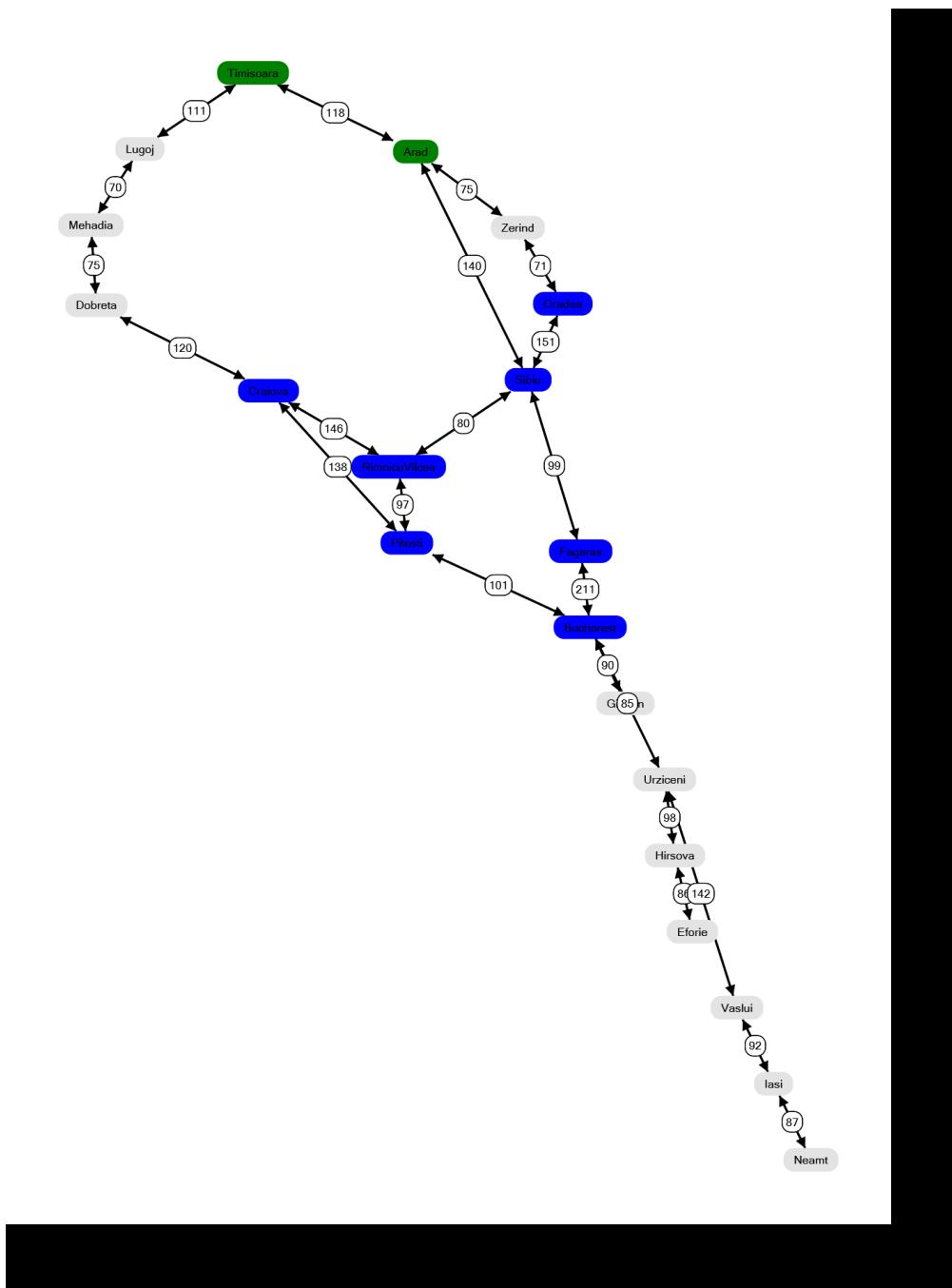
Pre-Visit

The algorithm has opened the node Timisoara



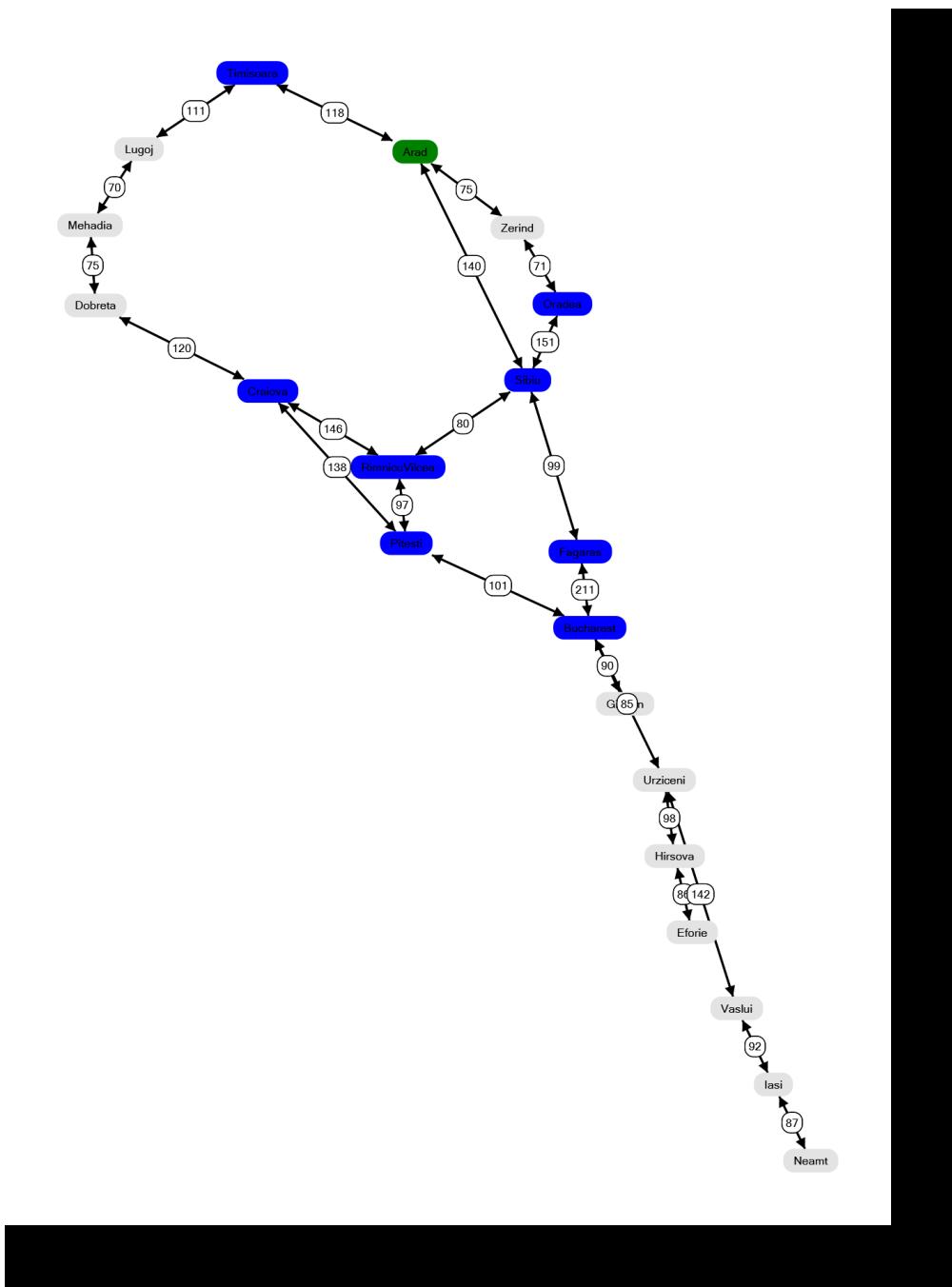
Visit

The algorithm has marked the node Timisoara as visited



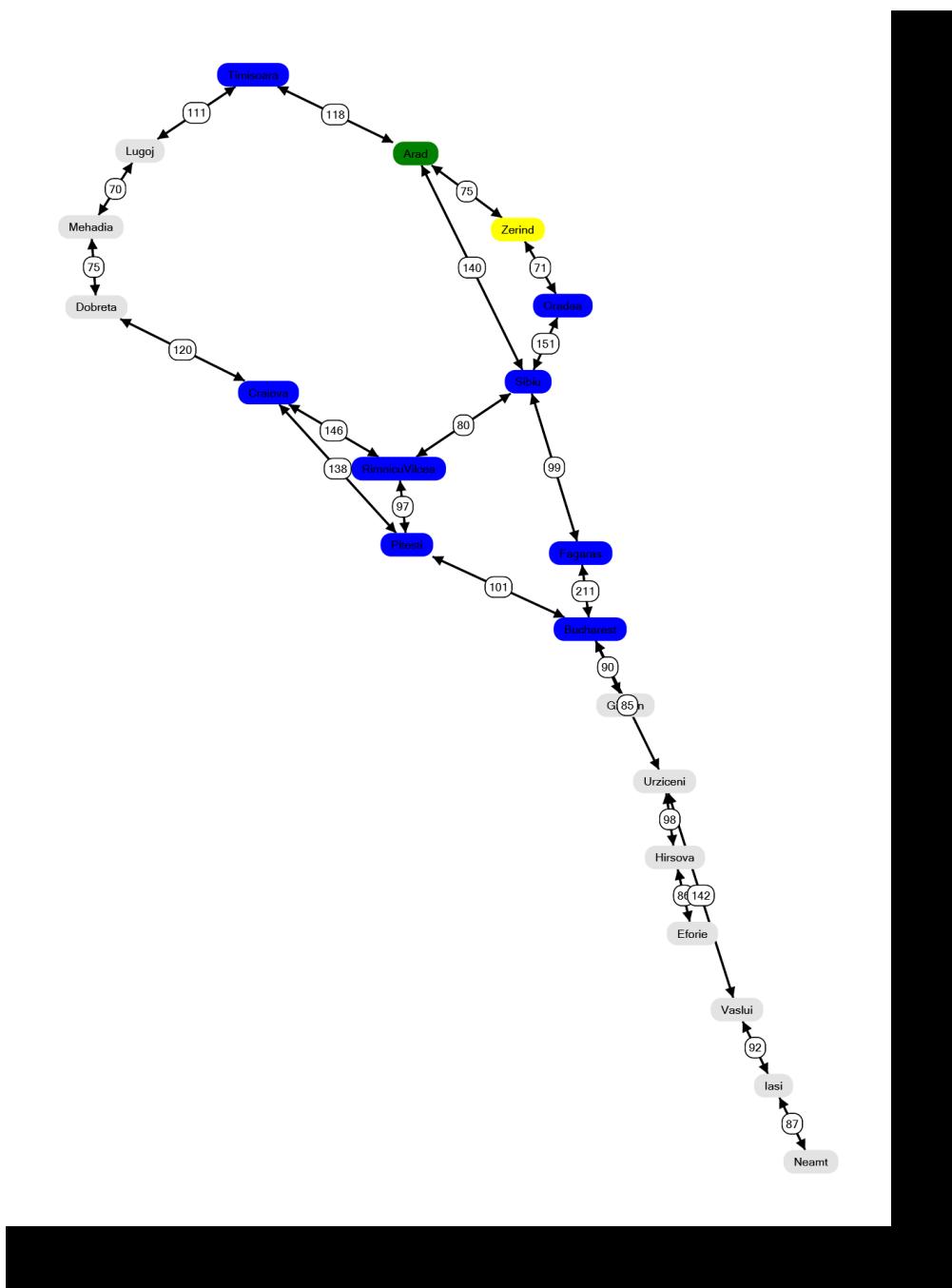
Post-Visit

The algorithm has closed the node Timisoara



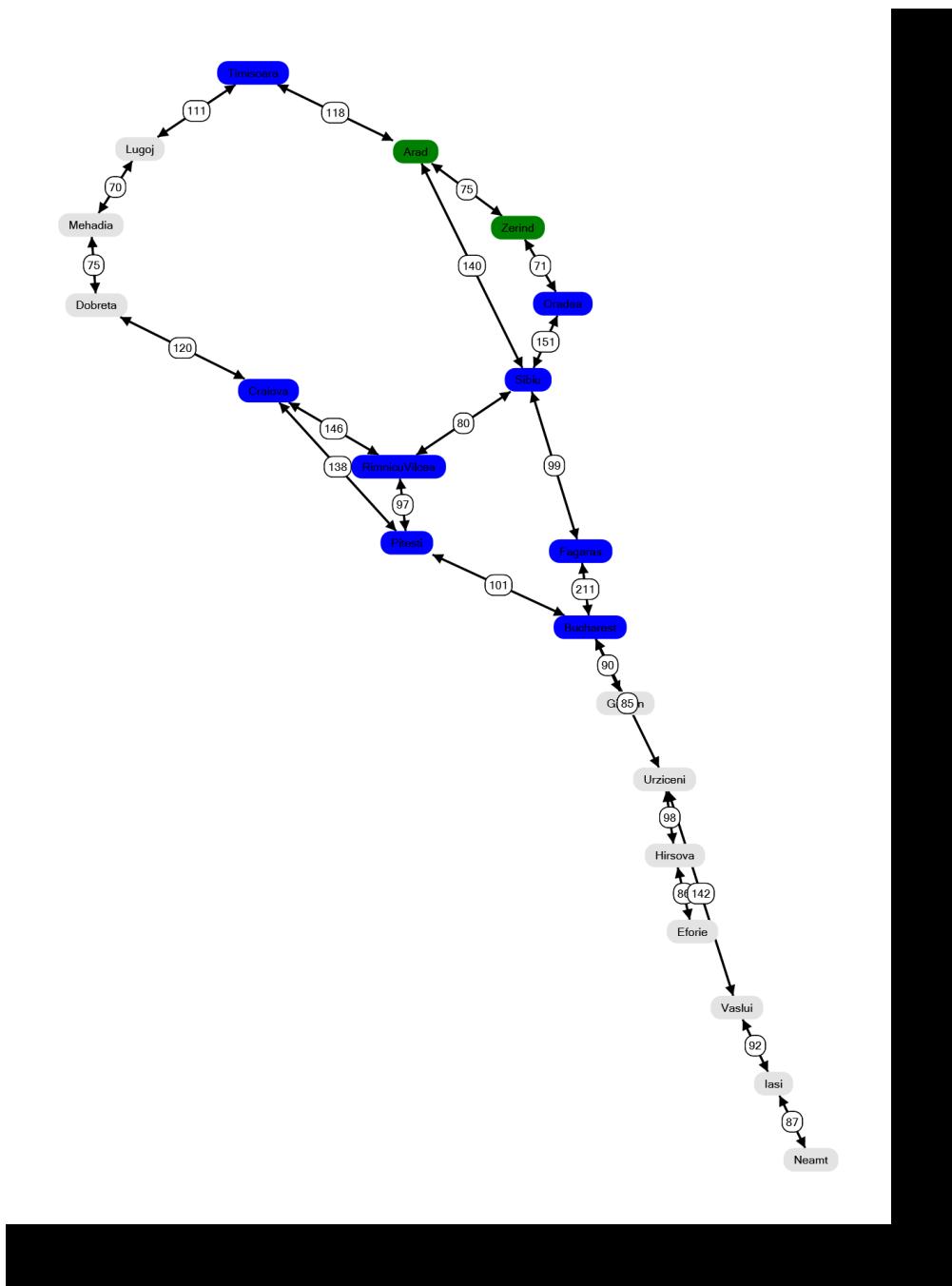
Pre-Visit

The algorithm has opened the node Zerind



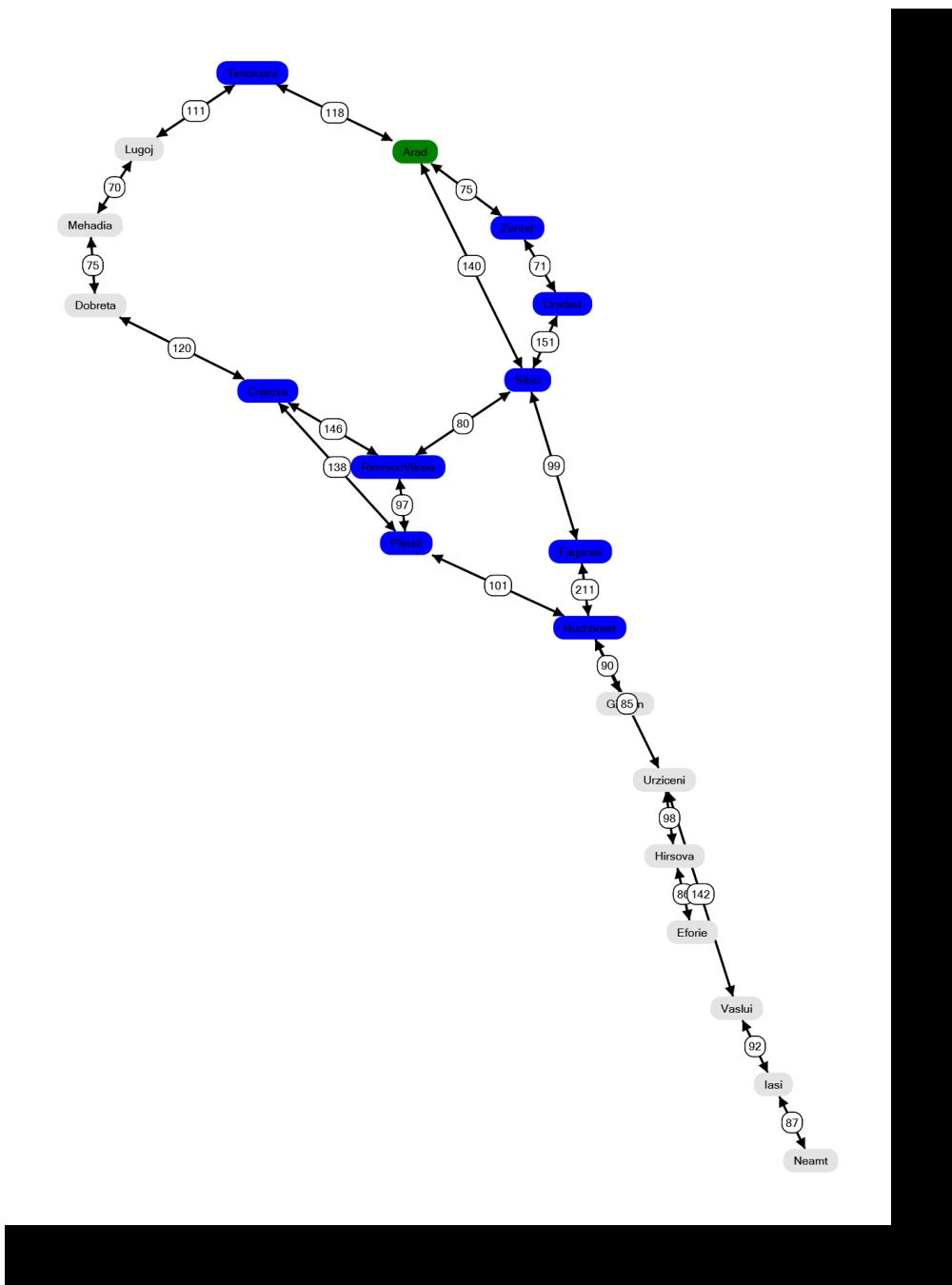
Visit

The algorithm has marked the node Zerind as visited



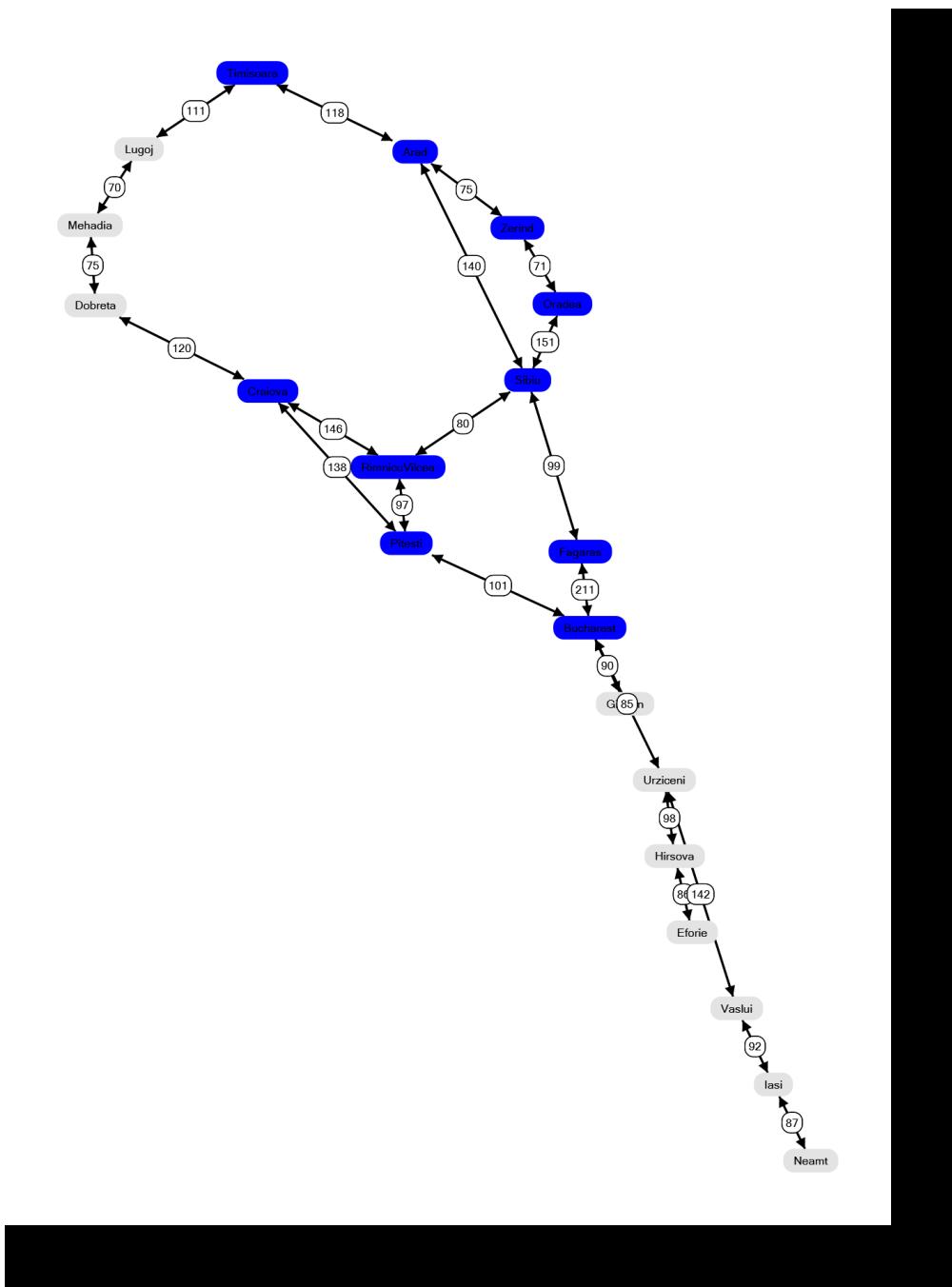
Post-Visit

The algorithm has closed the node Zerind



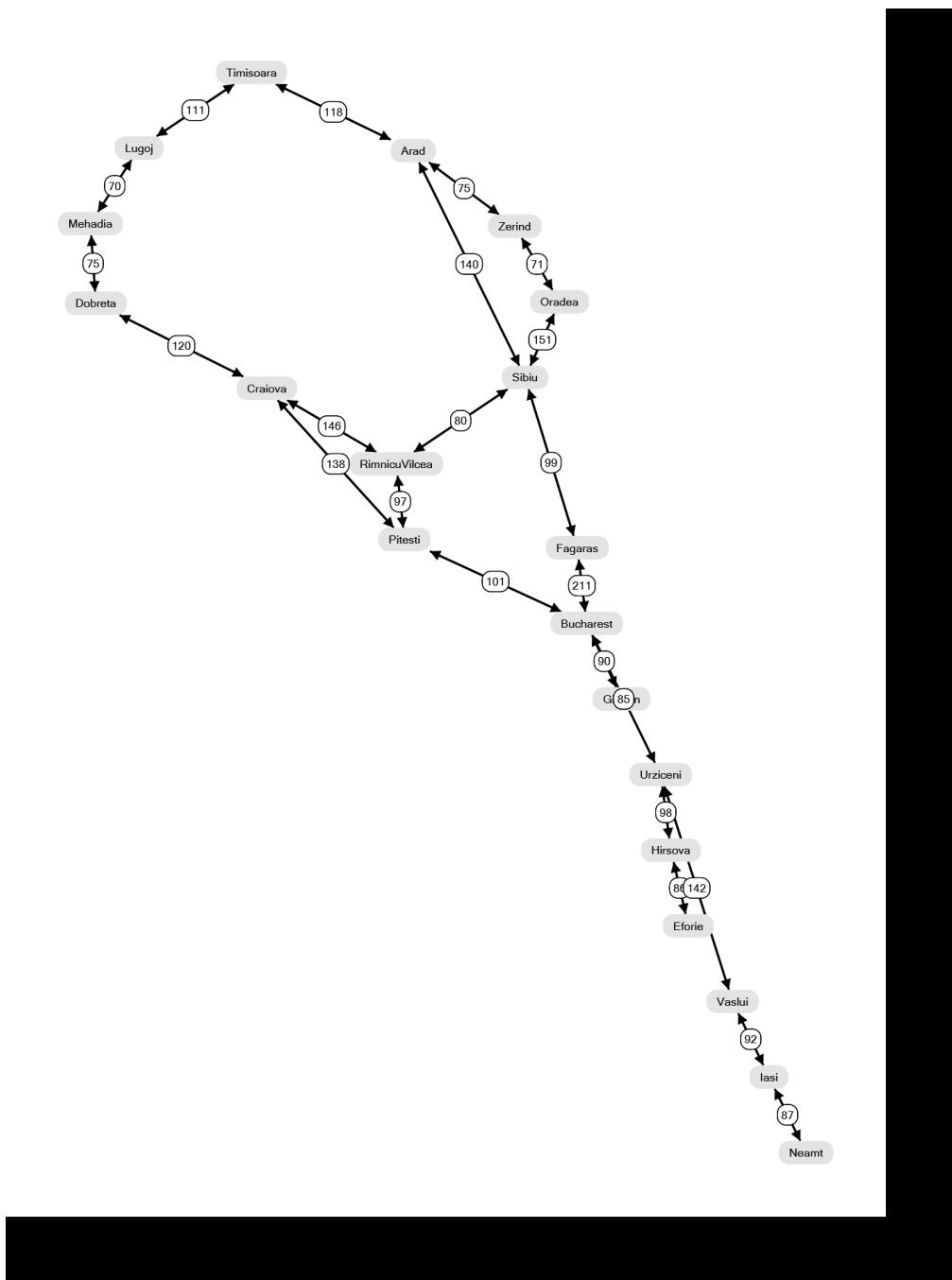
Post-Visit

The algorithm has closed the node Arad



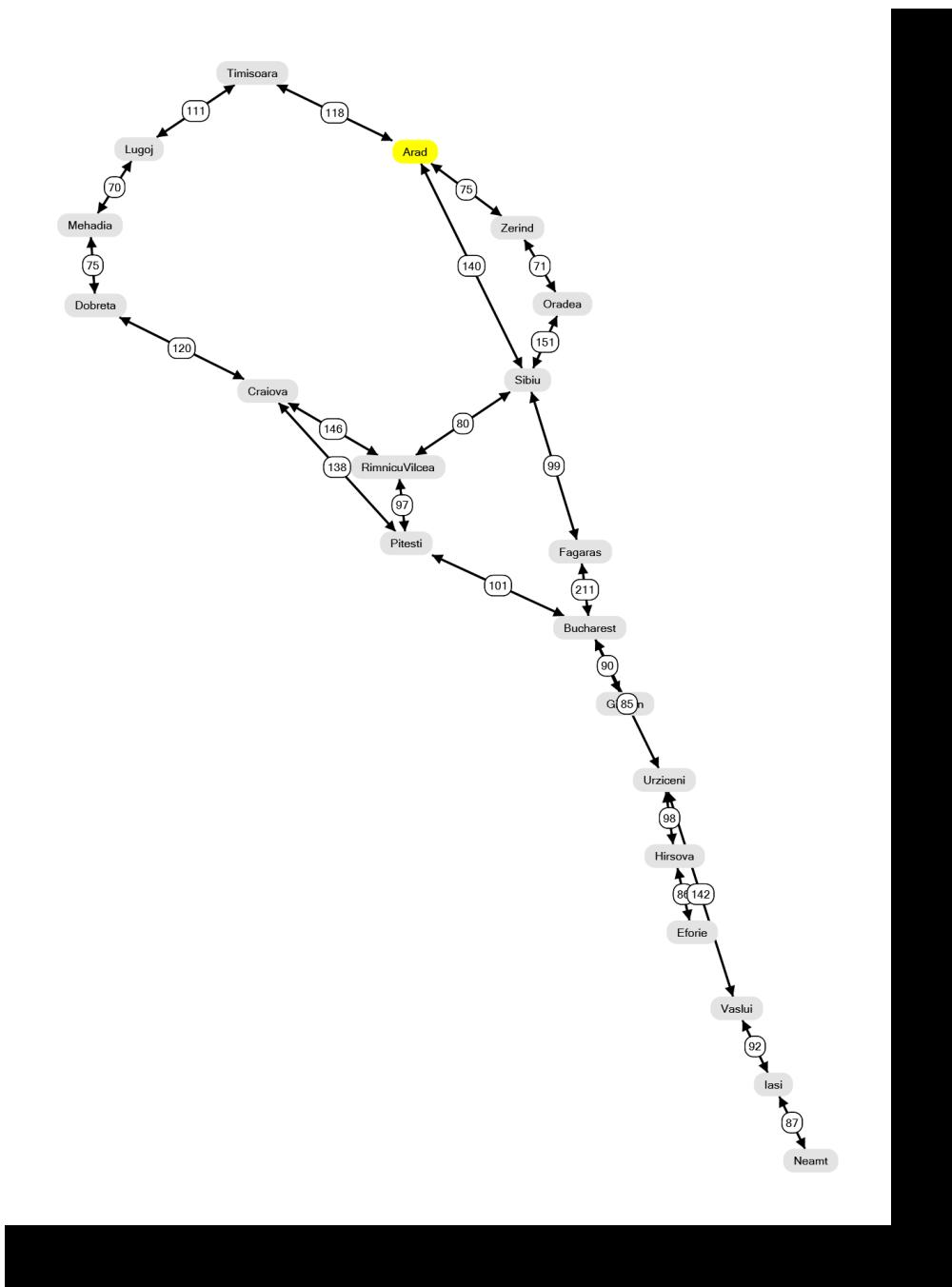
Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



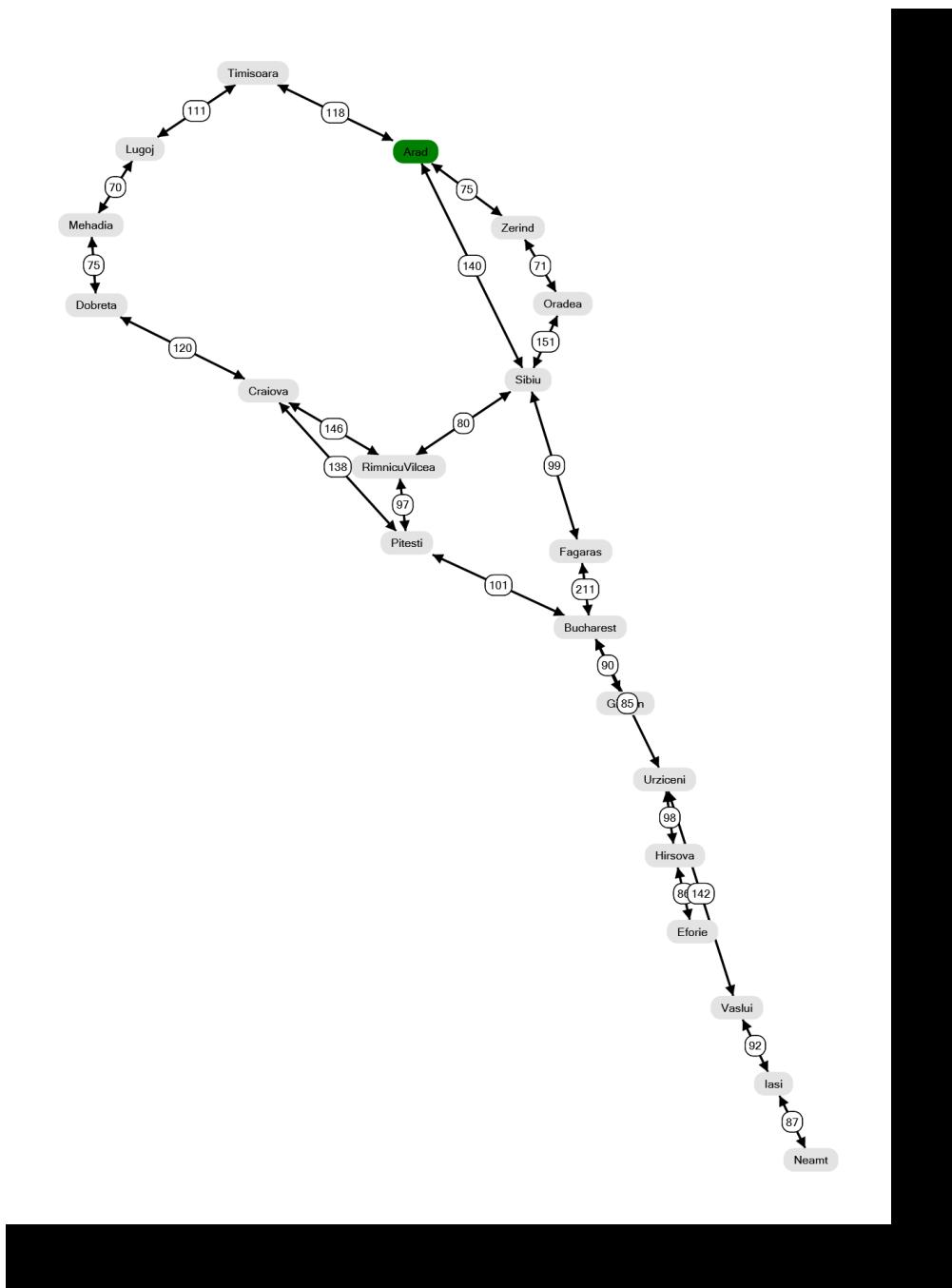
Pre-Visit

The algorithm has opened the node Arad



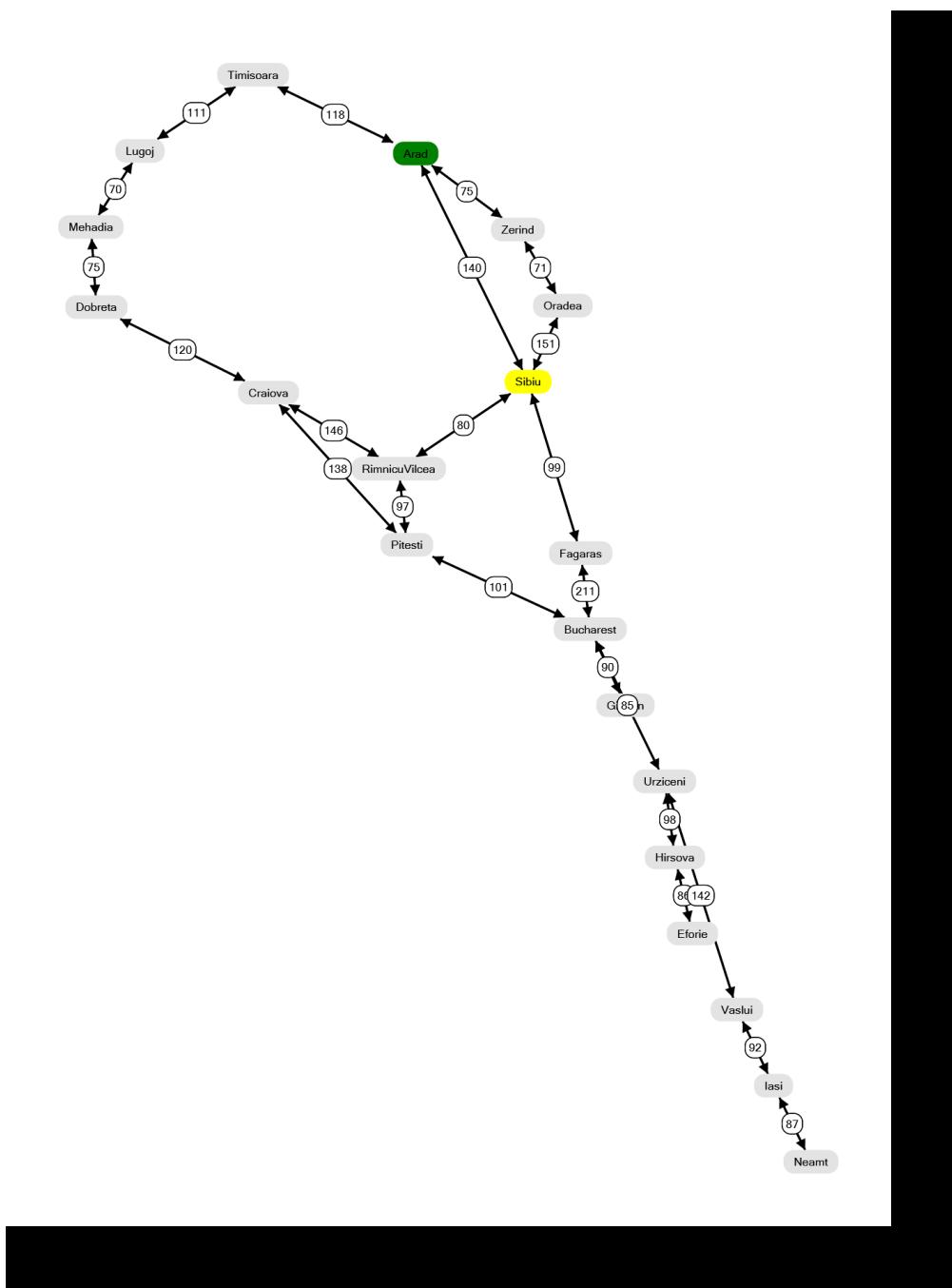
Visit

The algorithm has marked the node Arad as visited



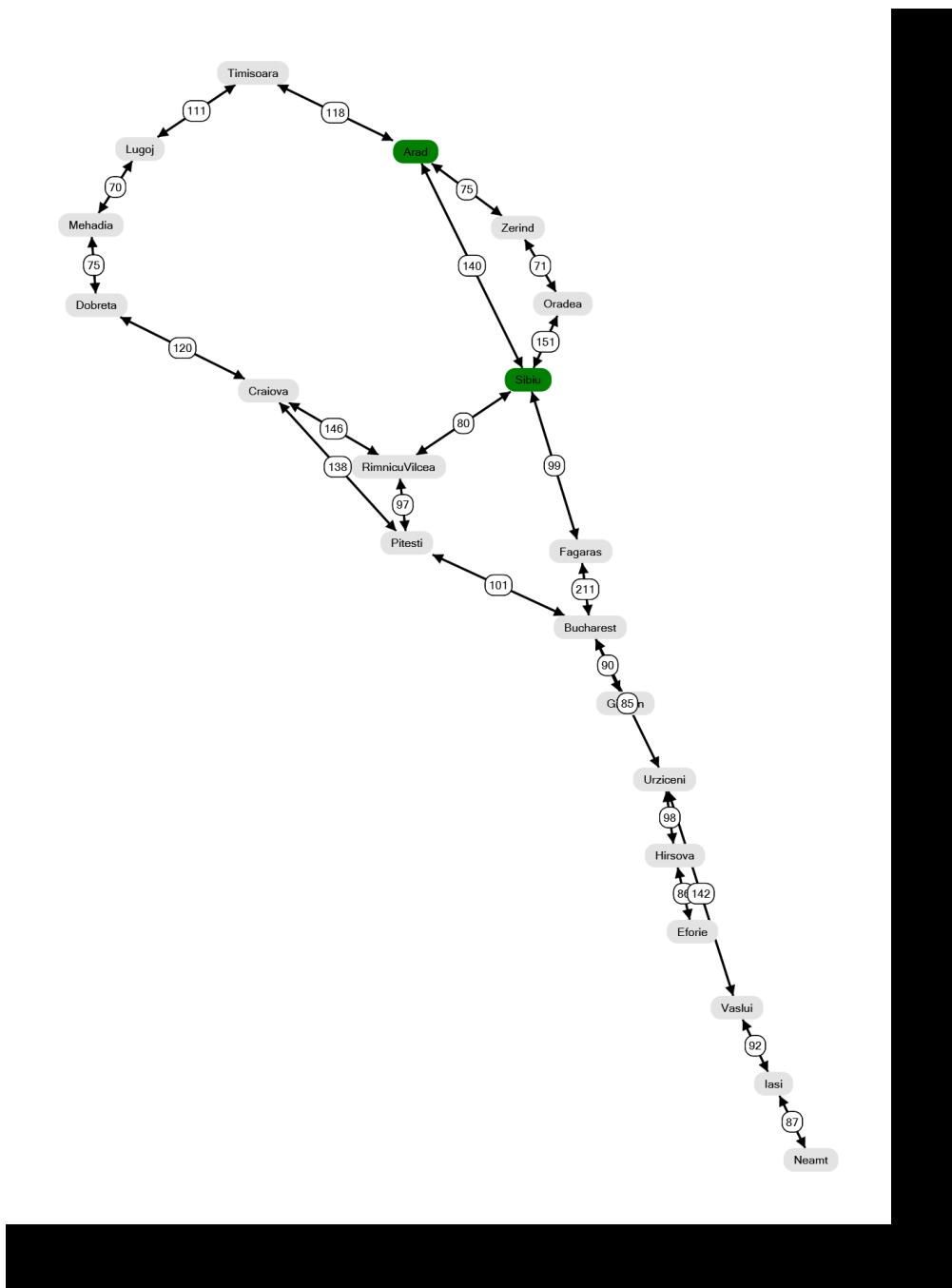
Pre-Visit

The algorithm has opened the node Sibiu



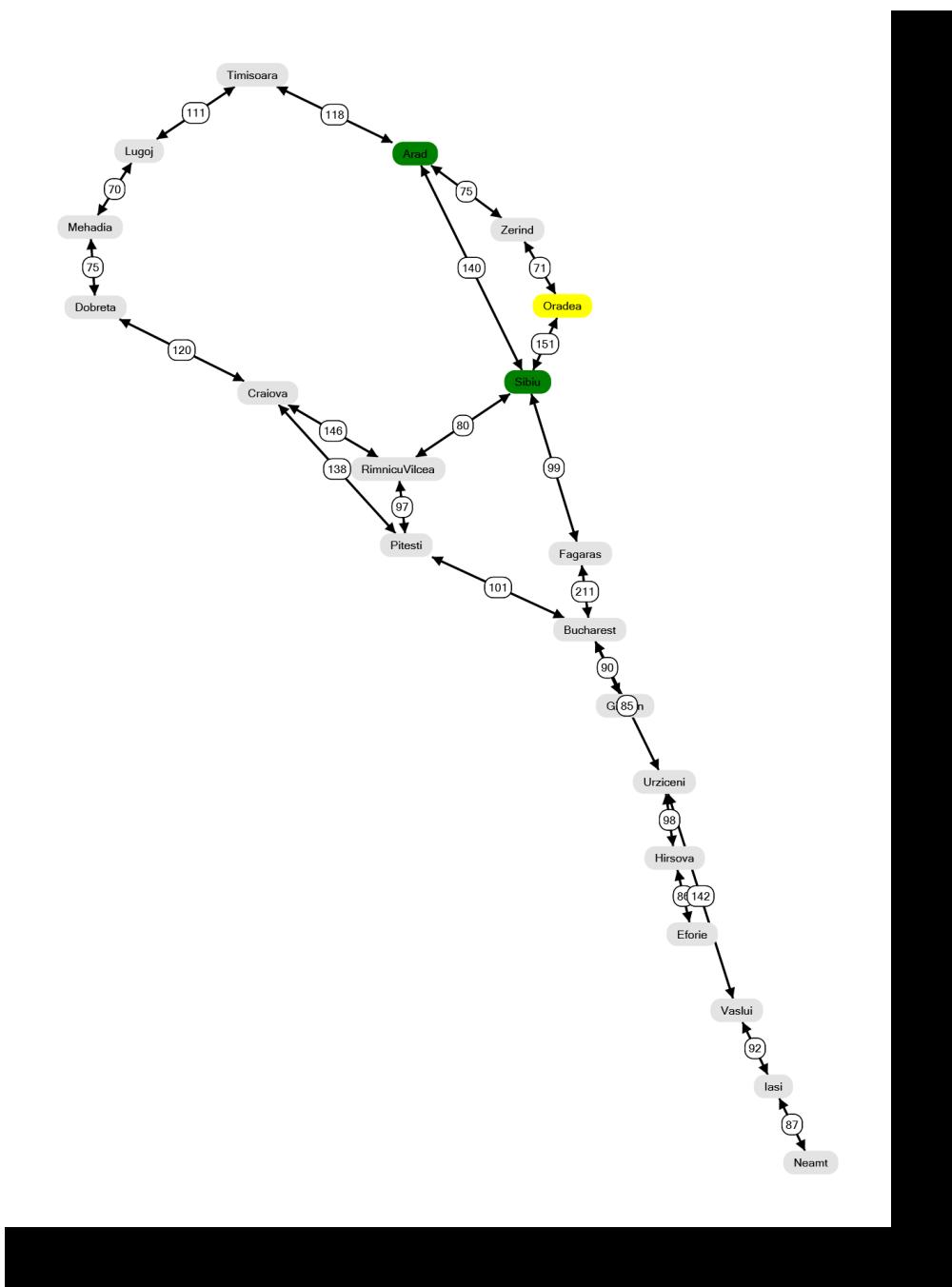
Visit

The algorithm has marked the node Sibiu as visited



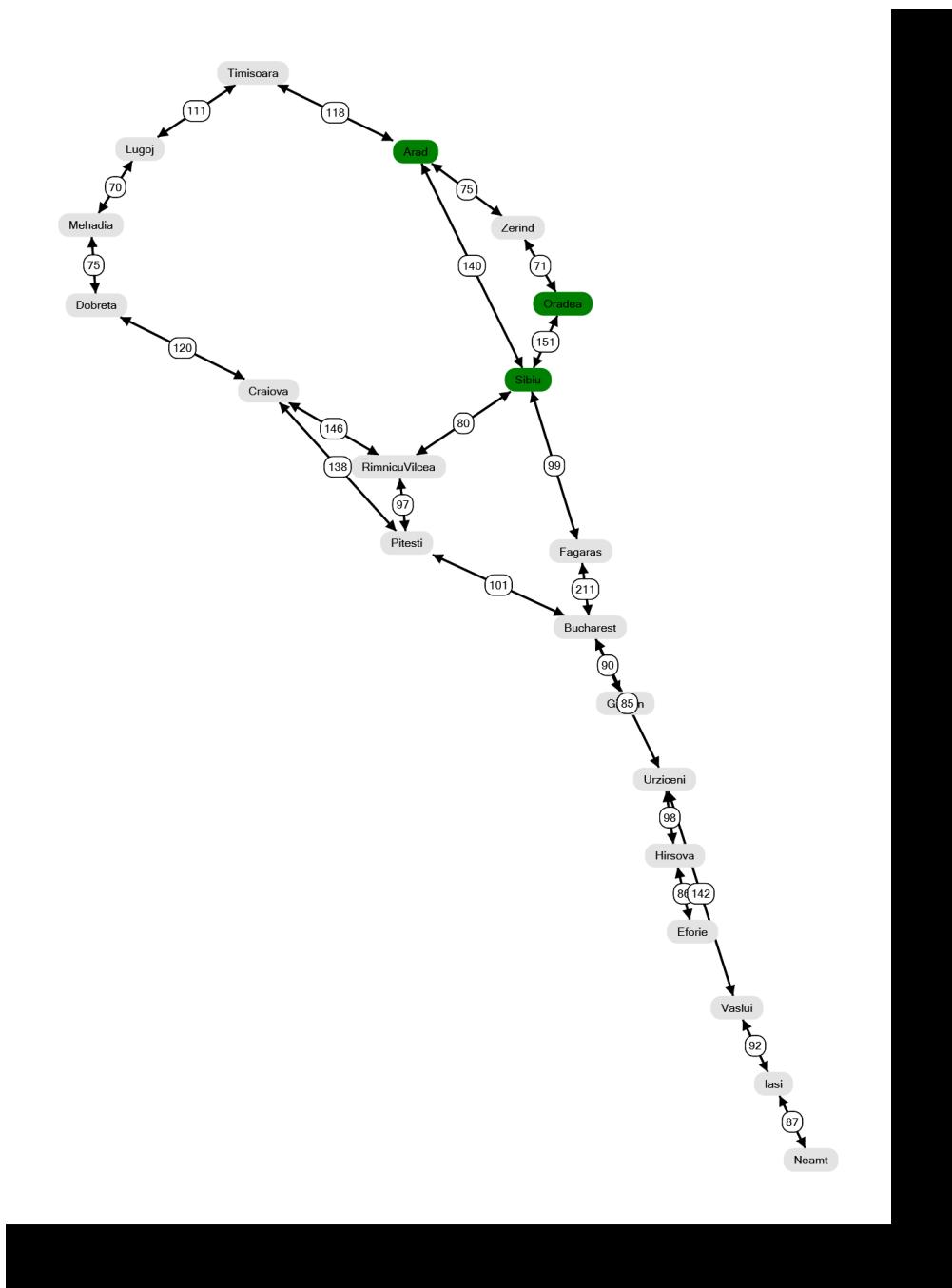
Pre-Visit

The algorithm has opened the node Oradea



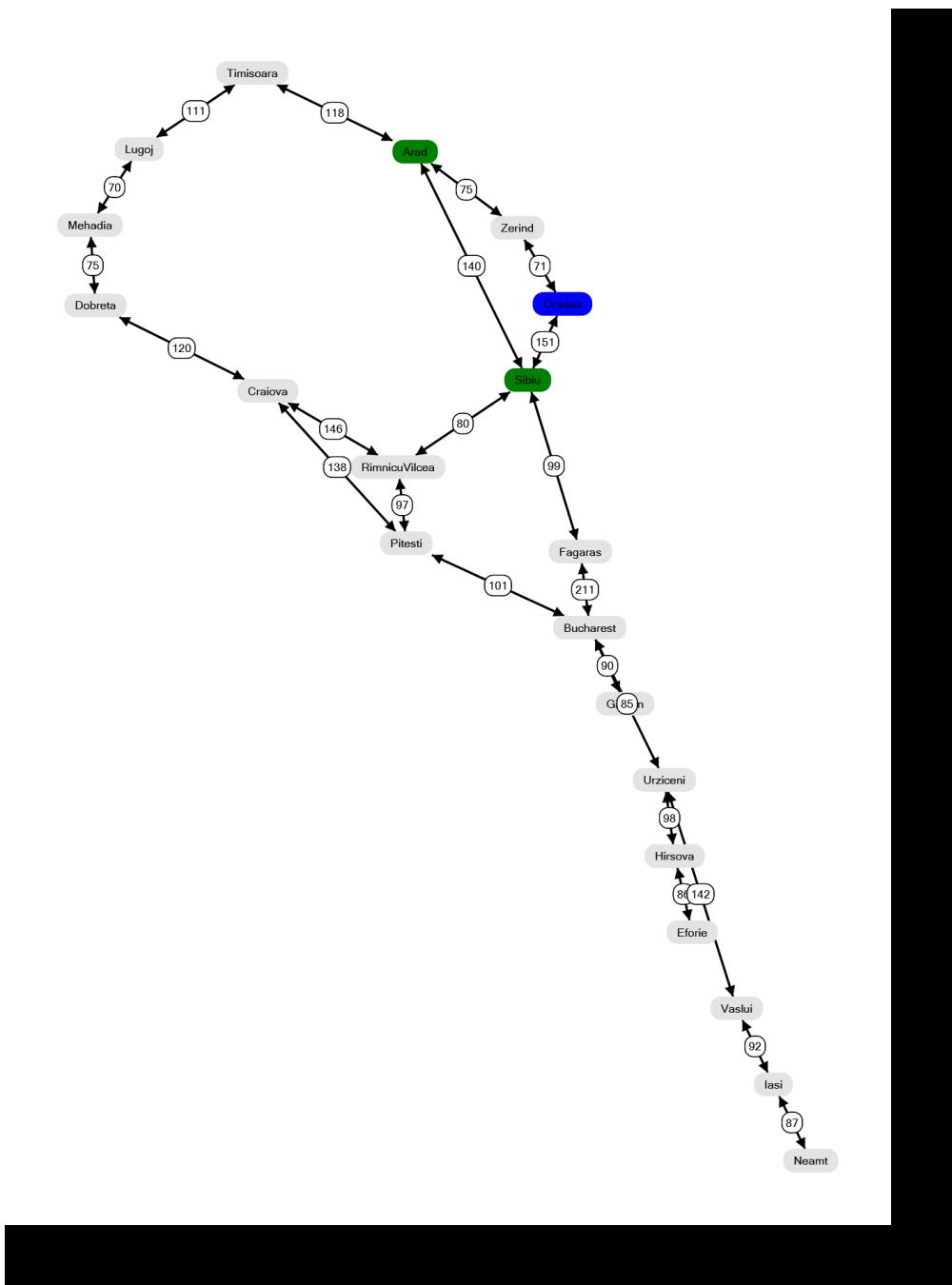
Visit

The algorithm has marked the node Oradea as visited



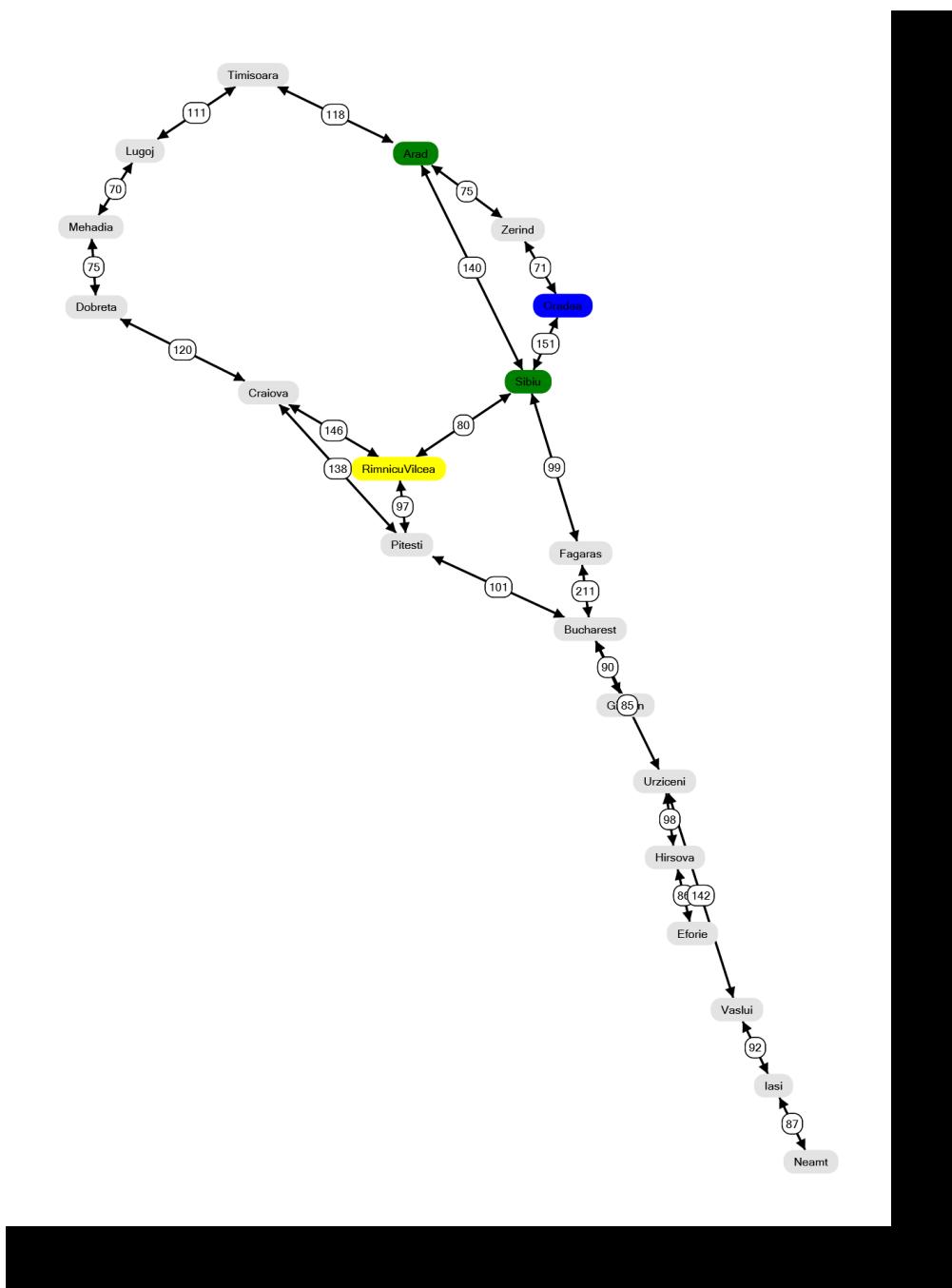
Post-Visit

The algorithm has closed the node Oradea



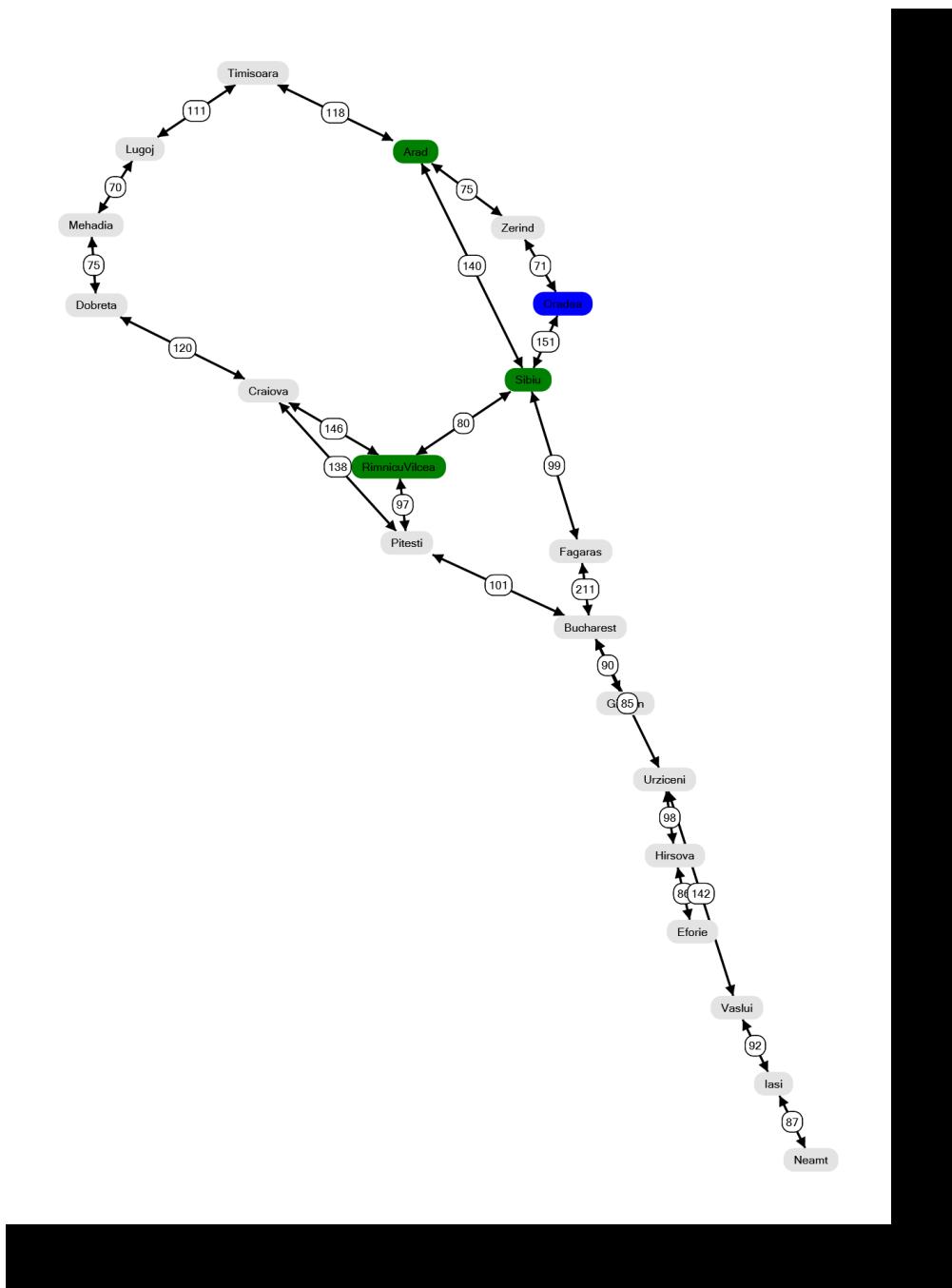
Pre-Visit

The algorithm has opened the node Rimnicu Vilcea



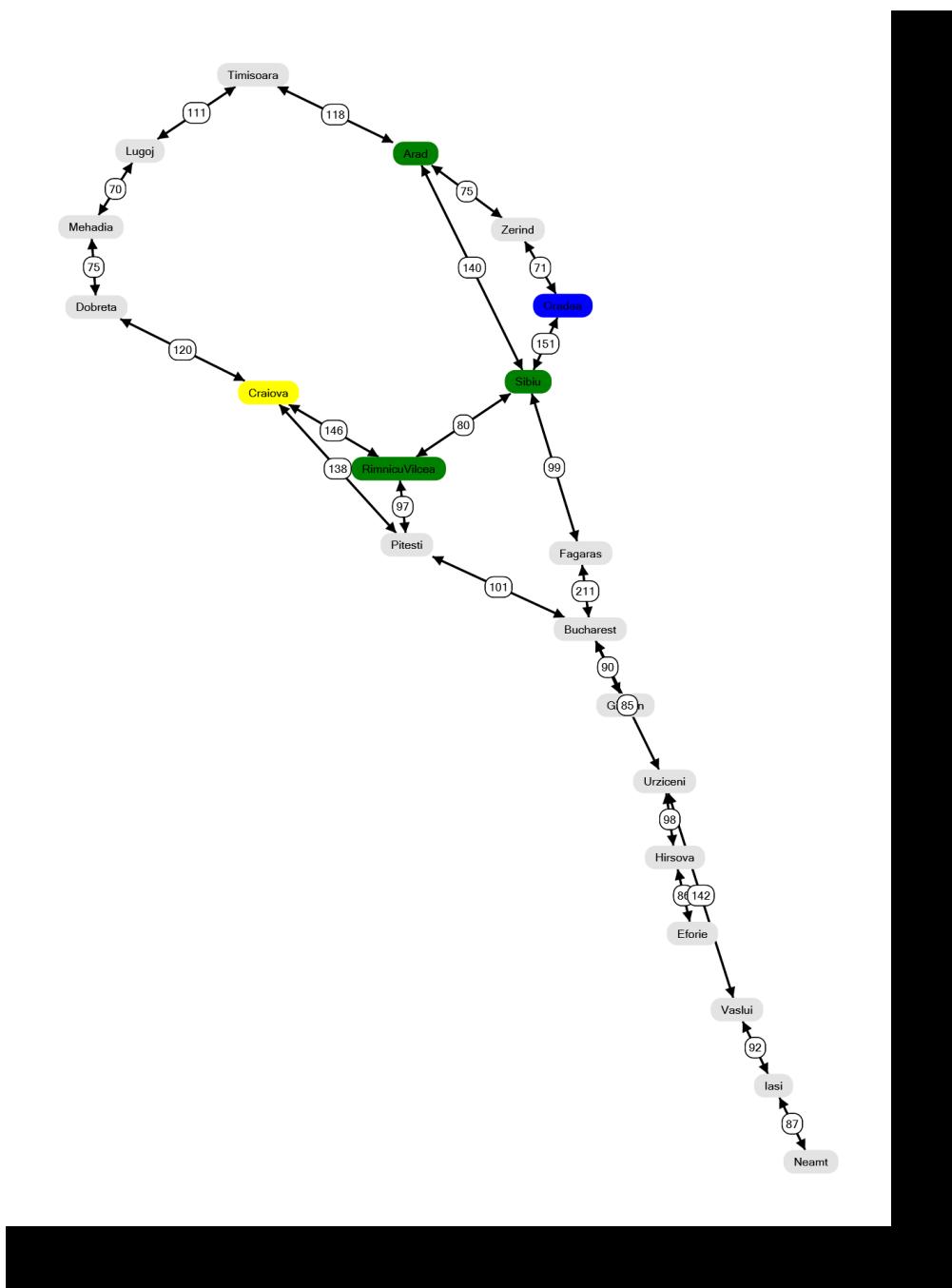
Visit

The algorithm has marked the node Rimnicu Vilcea as visited



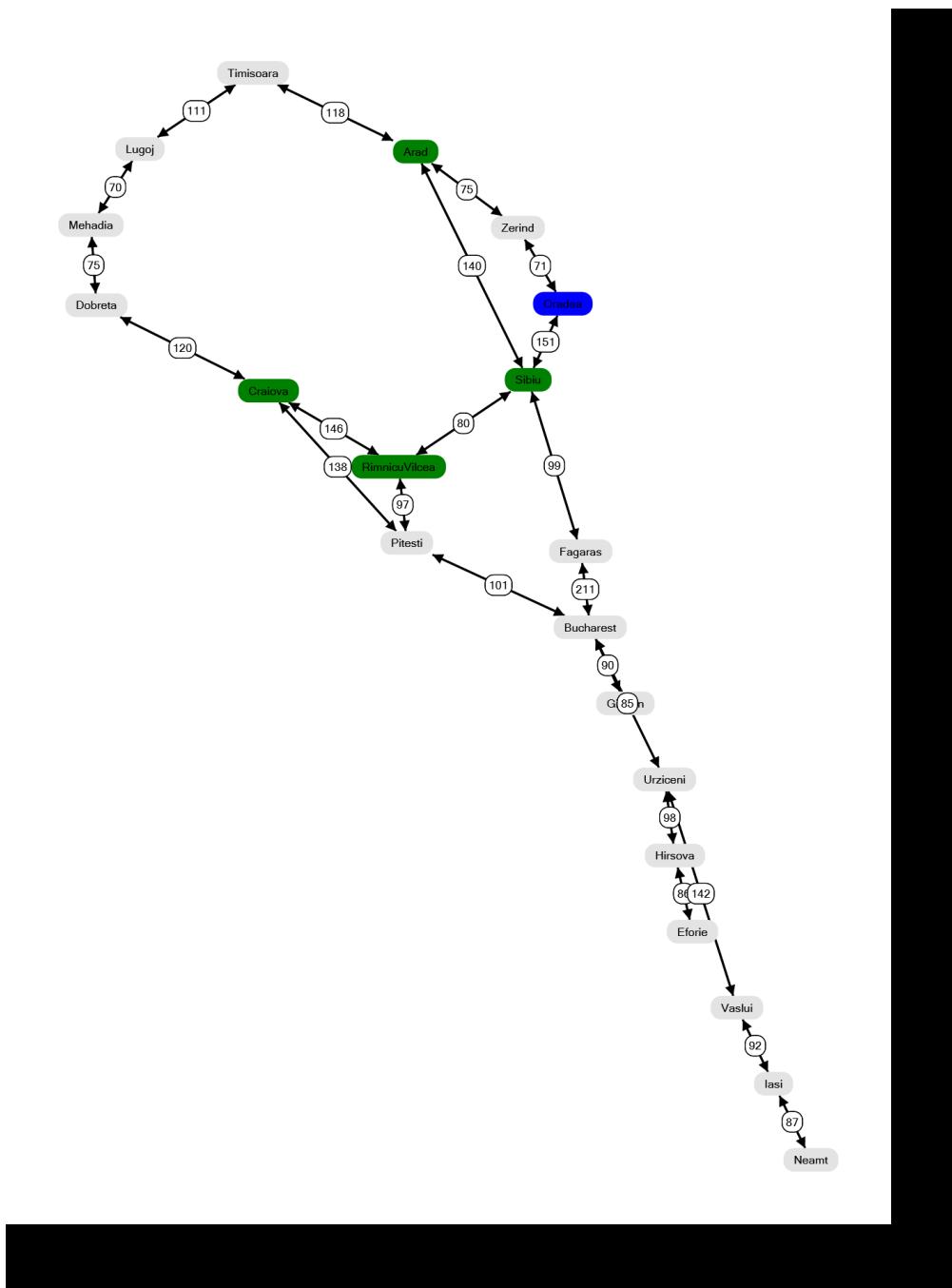
Pre-Visit

The algorithm has opened the node Craiova



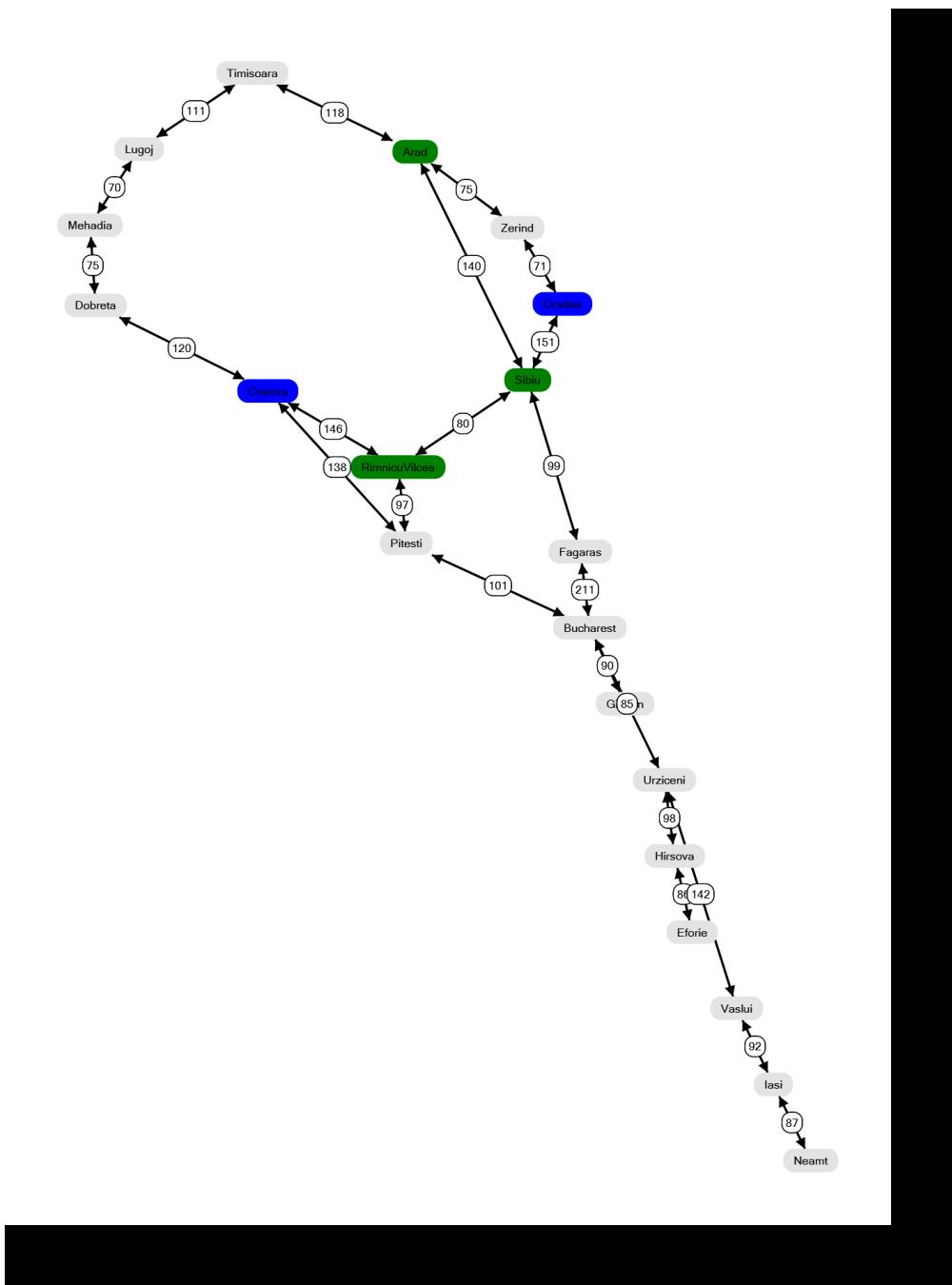
Visit

The algorithm has marked the node Craiova as visited



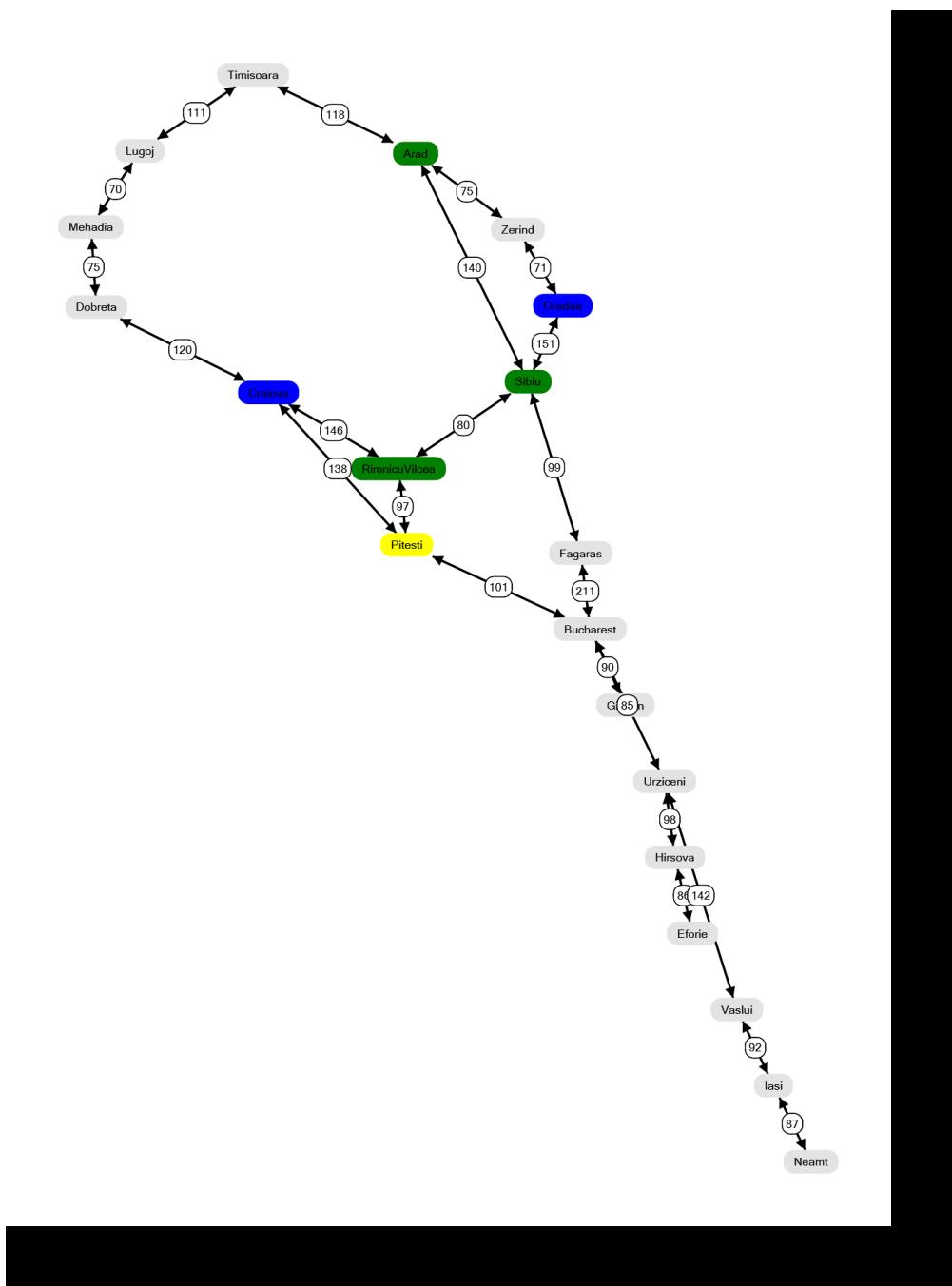
Post-Visit

The algorithm has closed the node Craiova



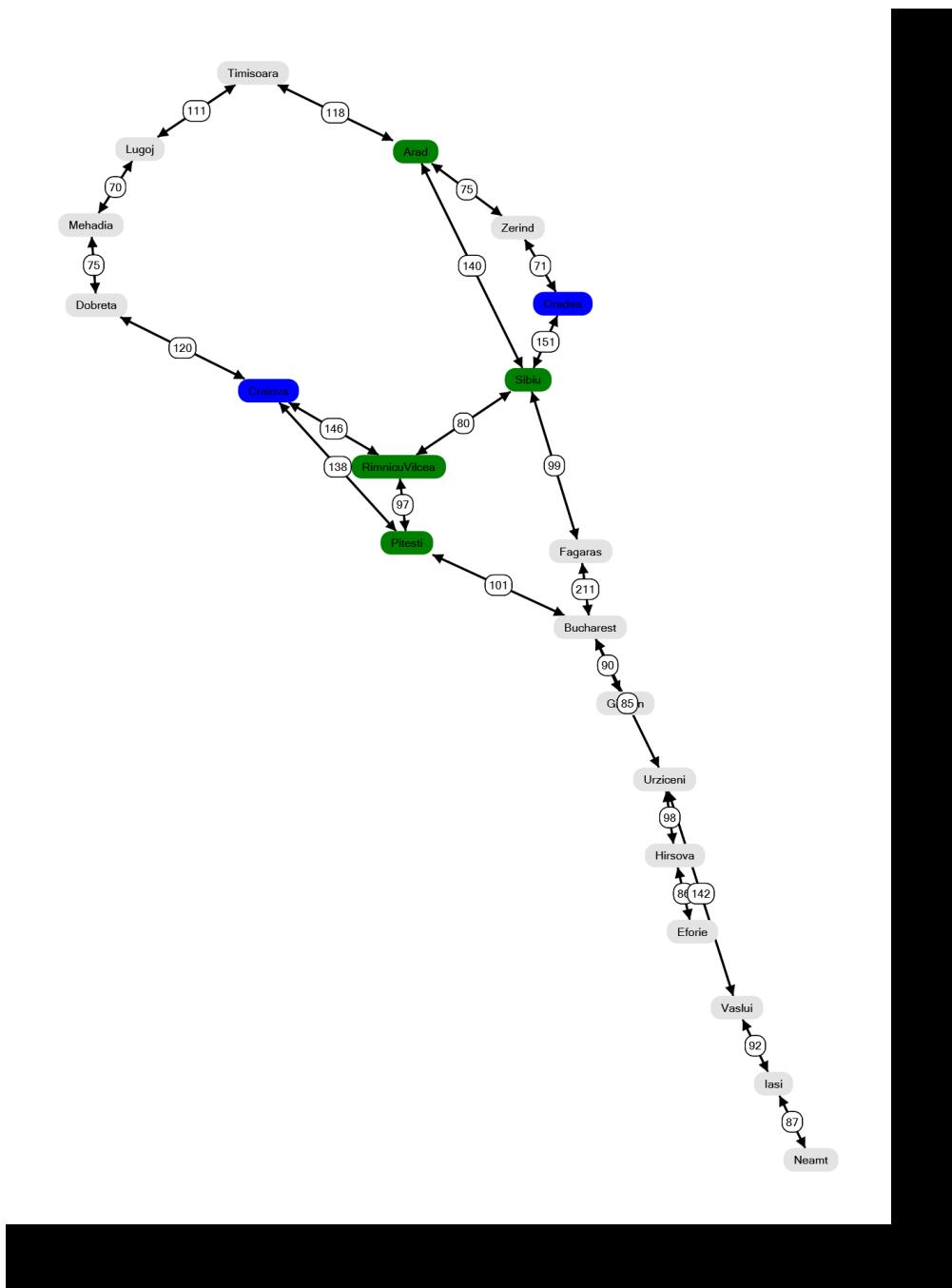
Pre-Visit

The algorithm has opened the node Pitesti



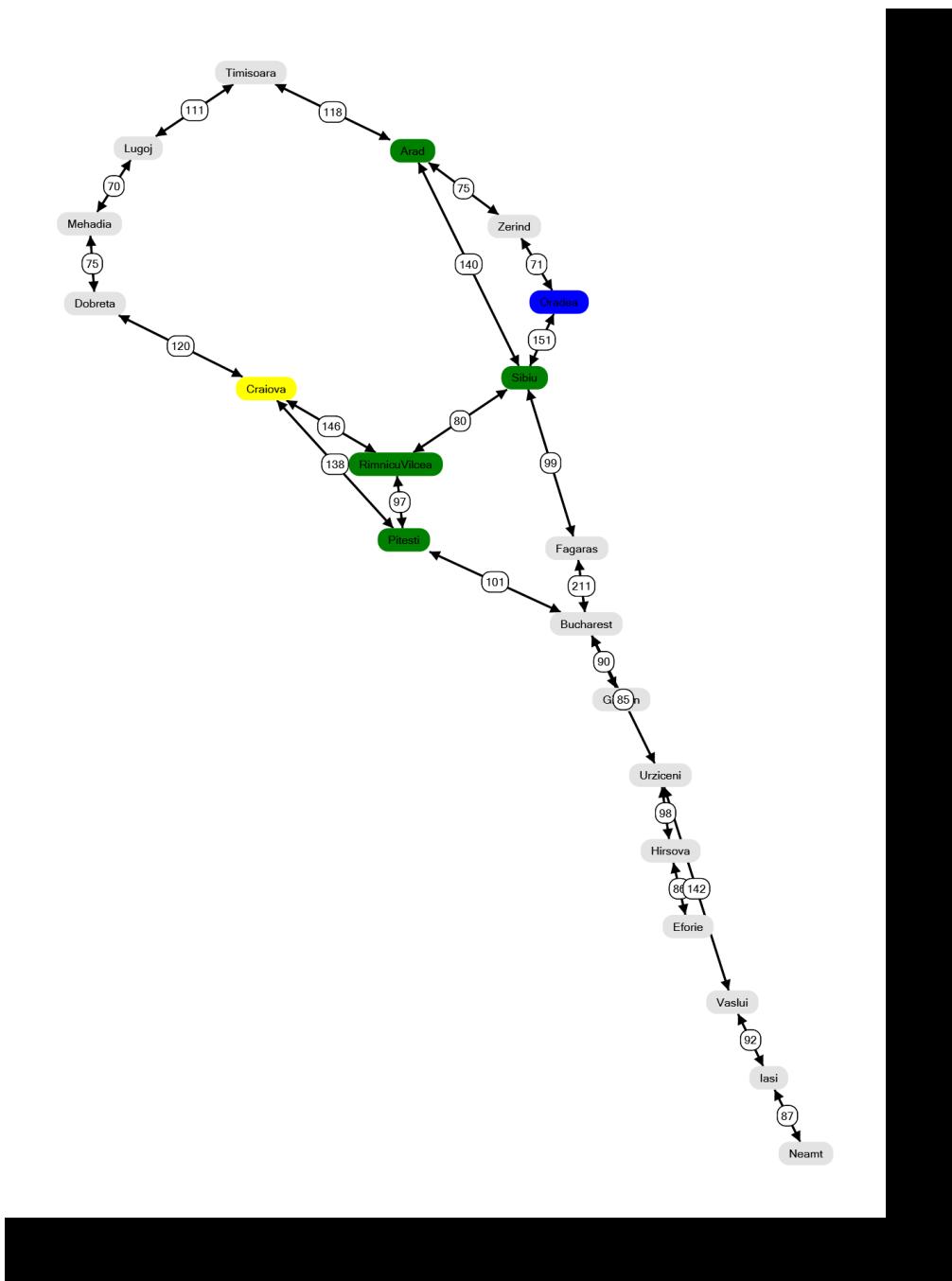
Visit

The algorithm has marked the node Pitesti as visited



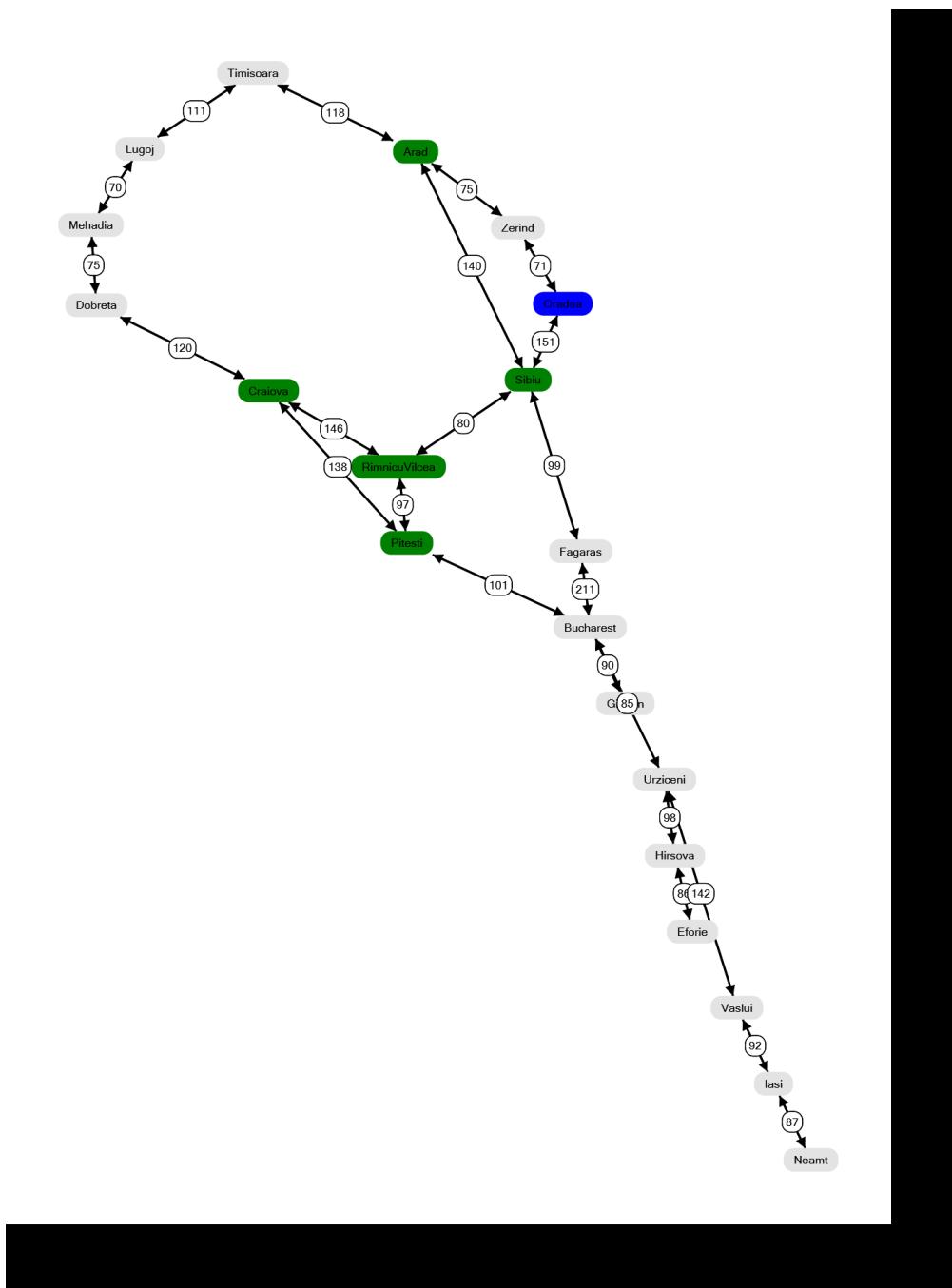
Pre-Visit

The algorithm has opened the node Craiova



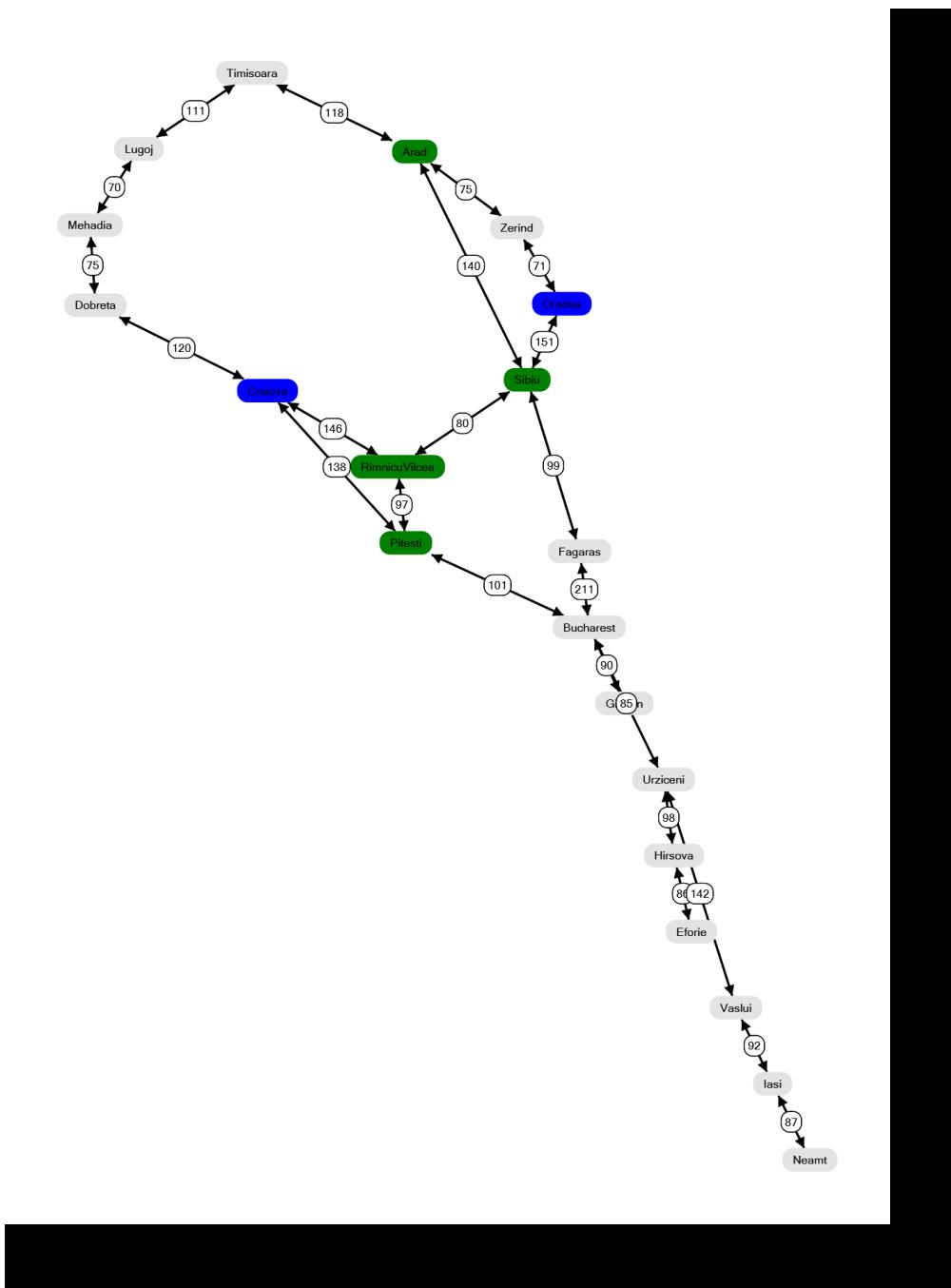
Visit

The algorithm has marked the node Craiova as visited



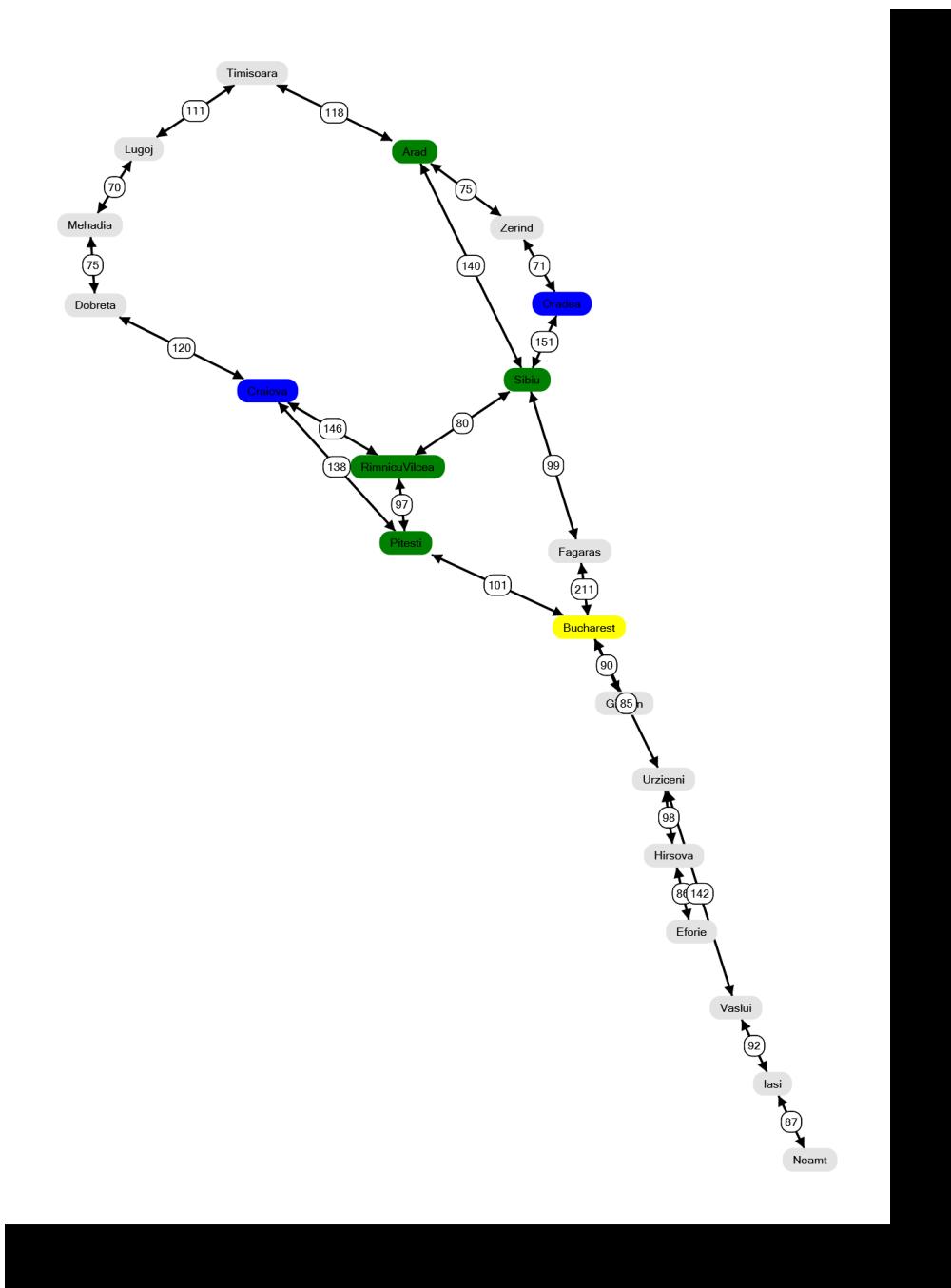
Post-Visit

The algorithm has closed the node Craiova



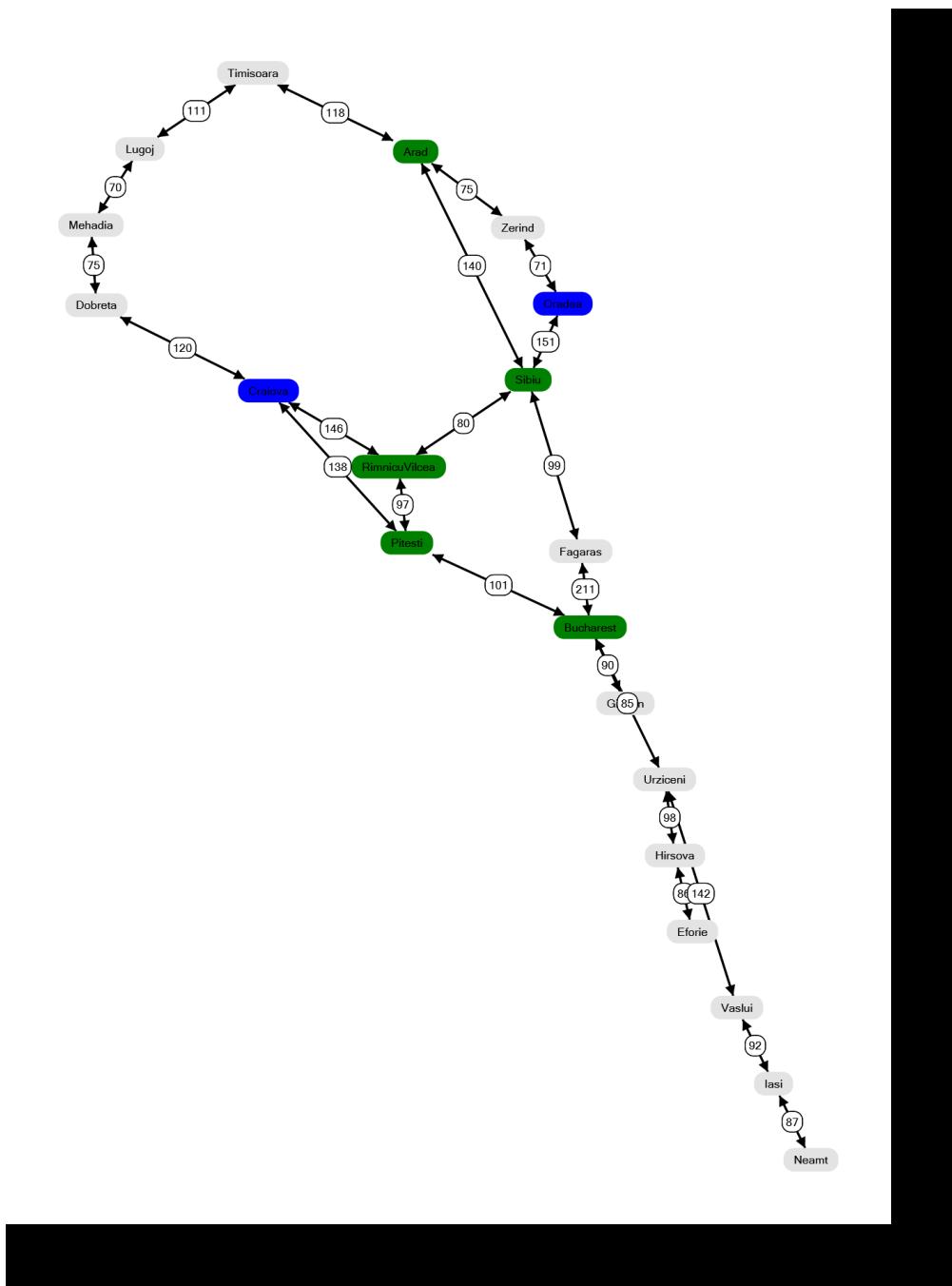
Pre-Visit

The algorithm has opened the node Bucharest



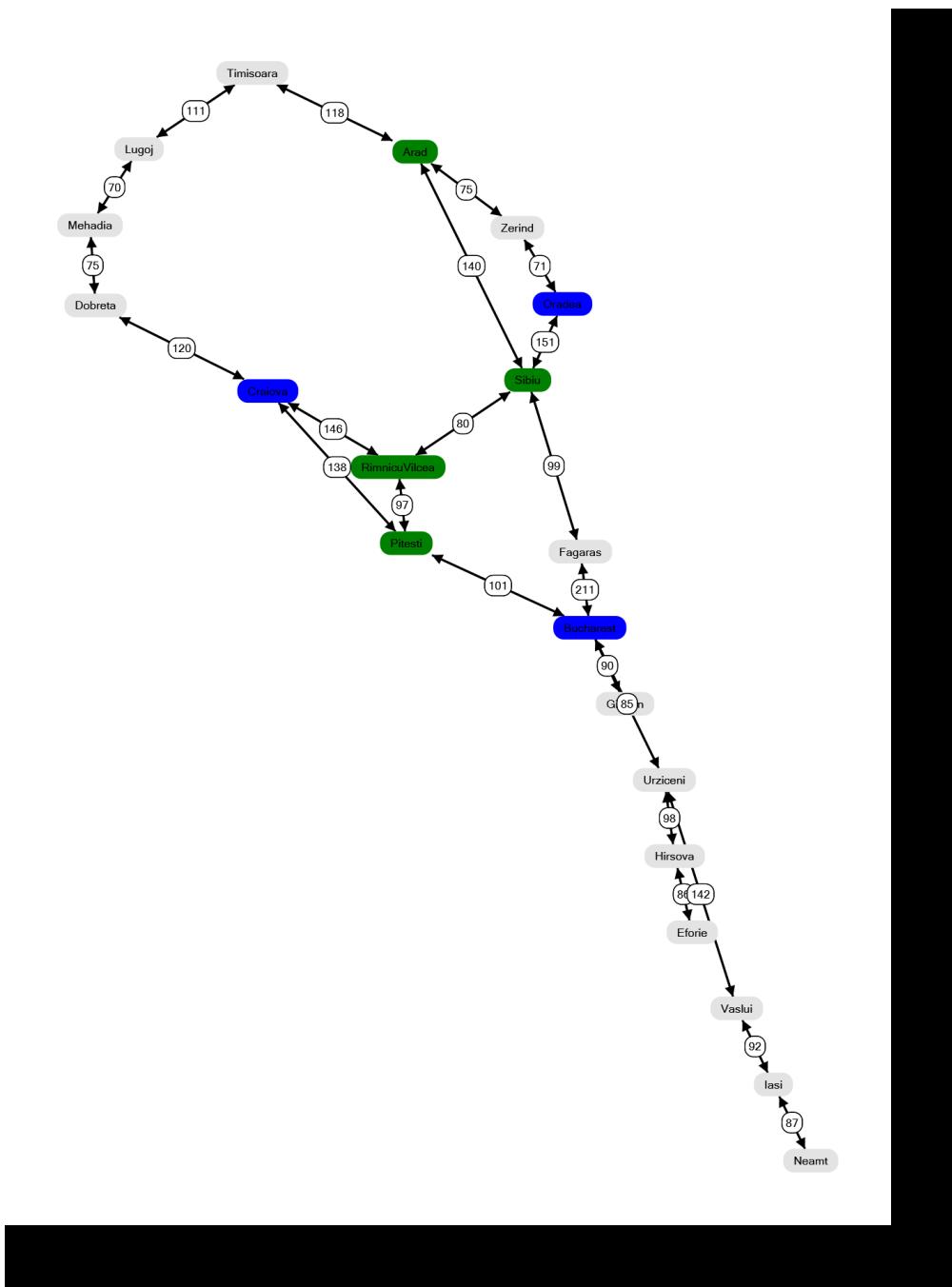
Visit

The algorithm has marked the node Bucharest as visited



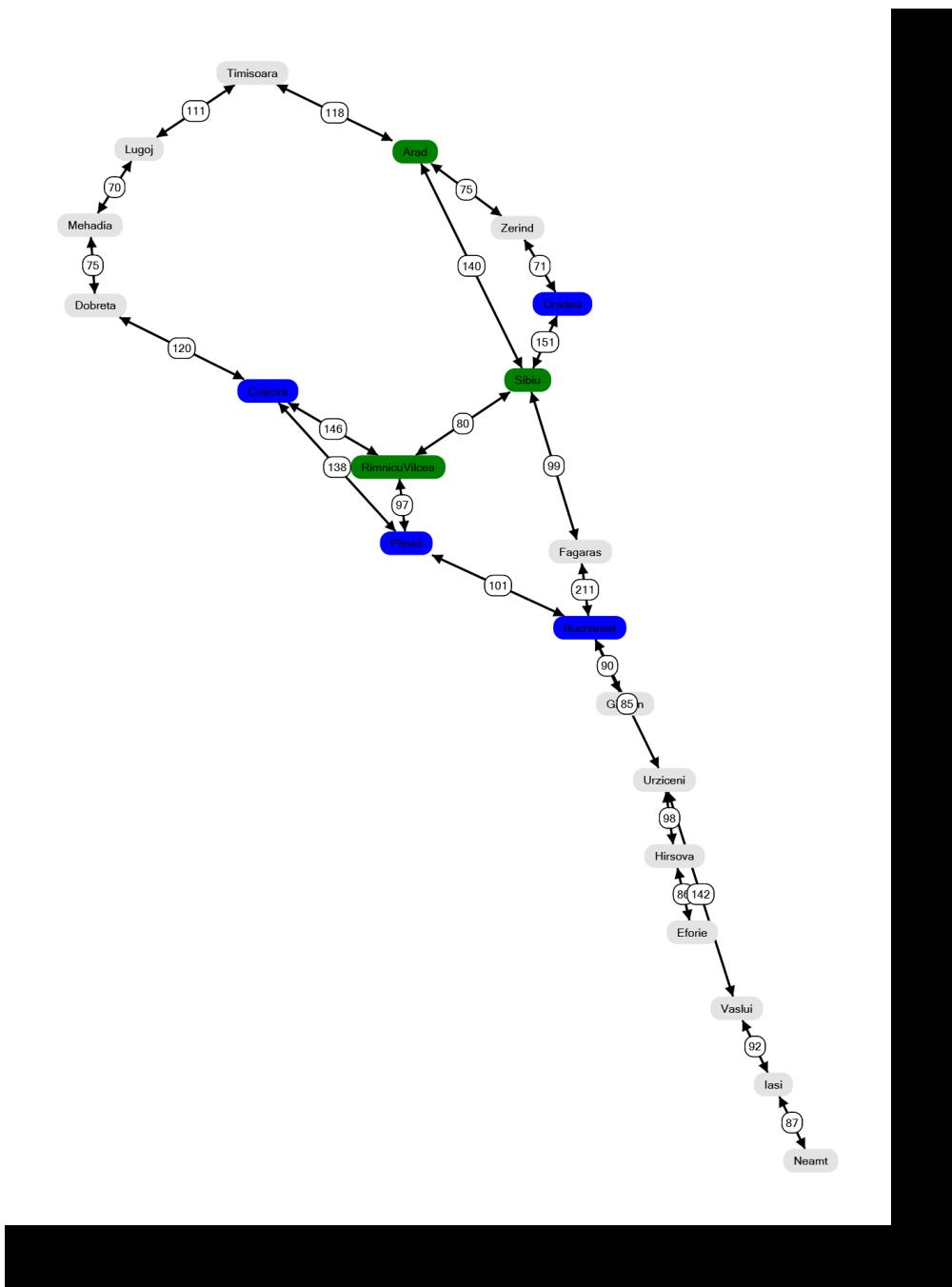
Post-Visit

The algorithm has closed the node Bucharest



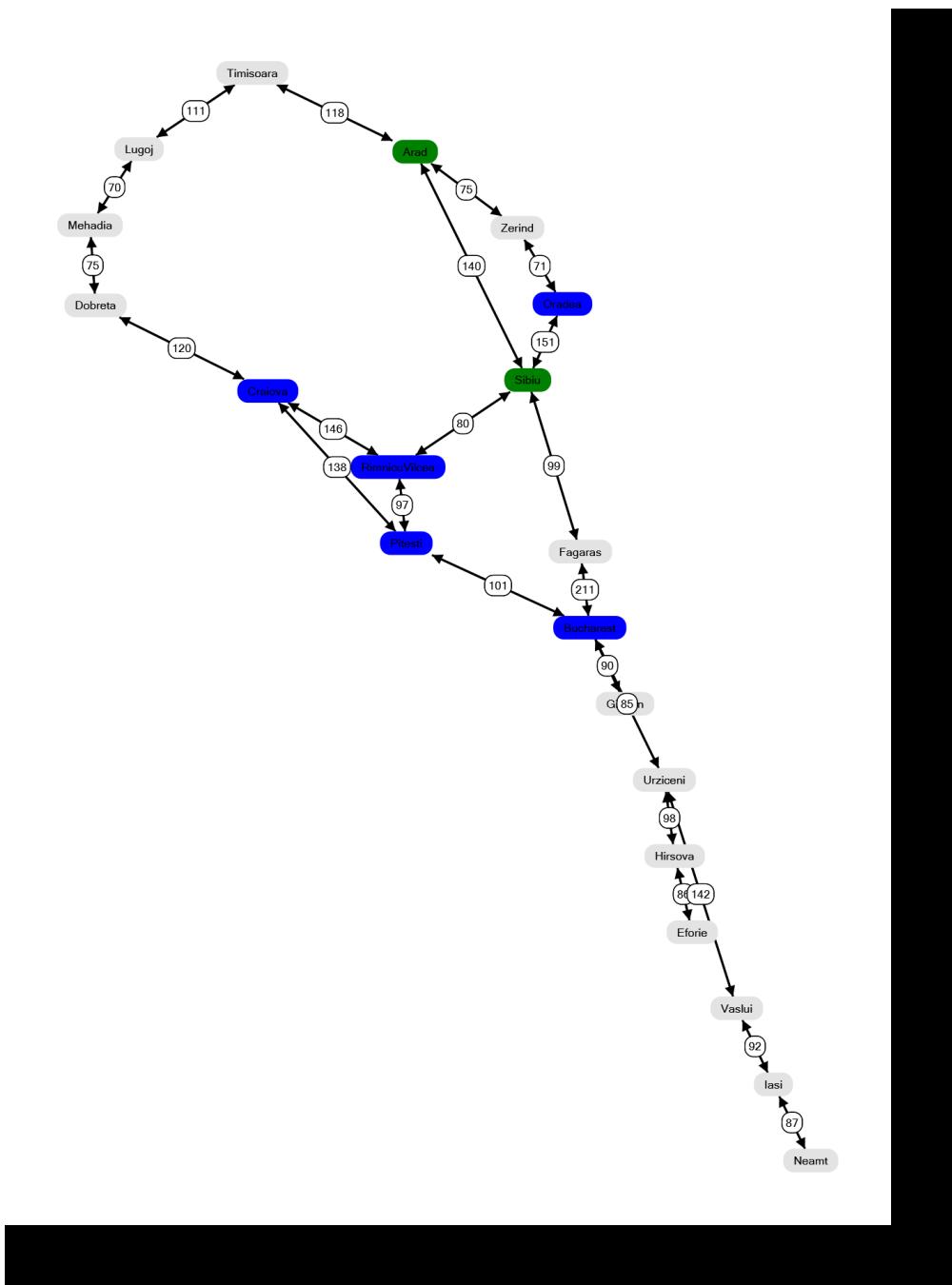
Post-Visit

The algorithm has closed the node Pitesti



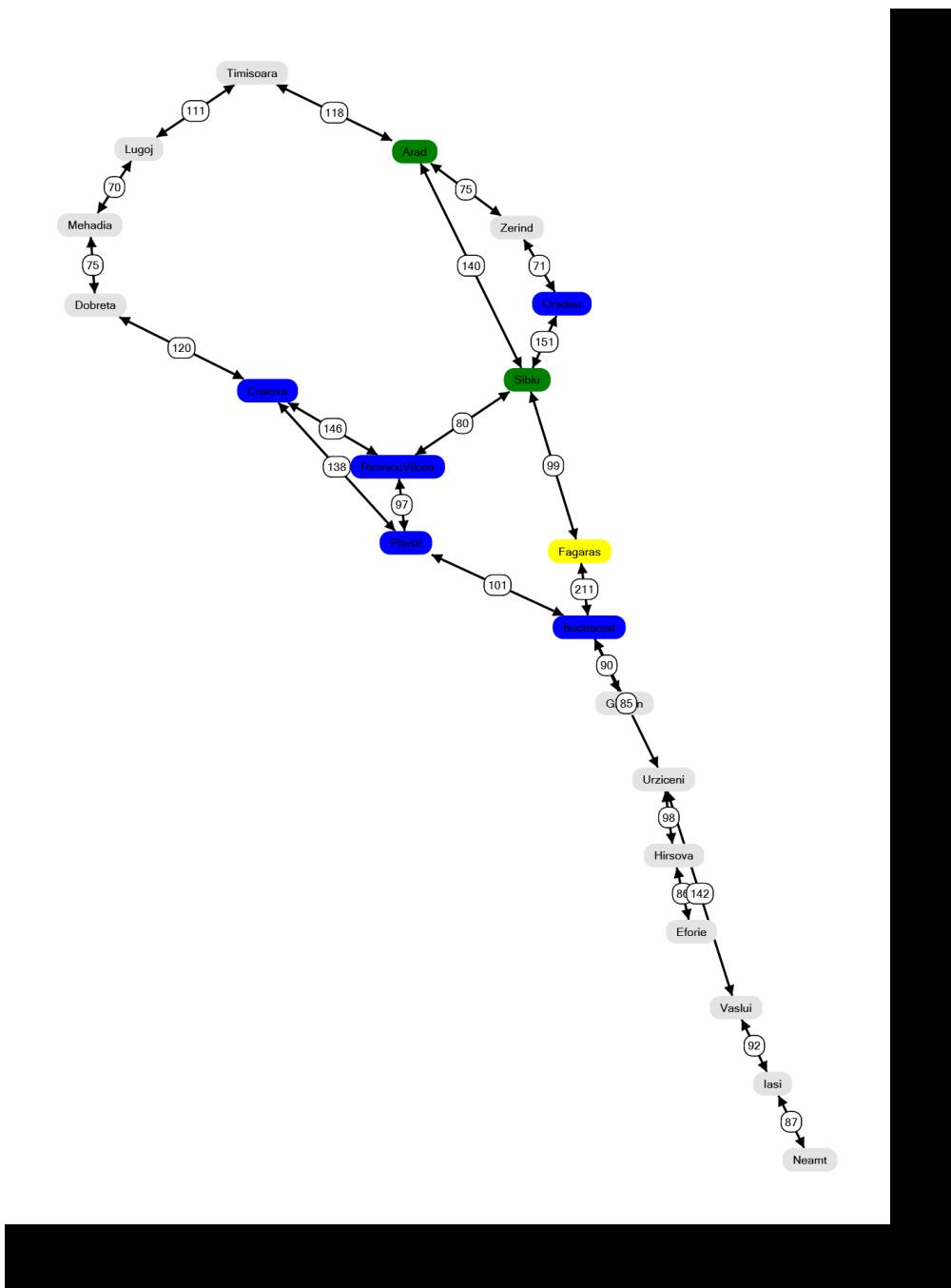
Post-Visit

The algorithm has closed the node Rimnicu Vilcea



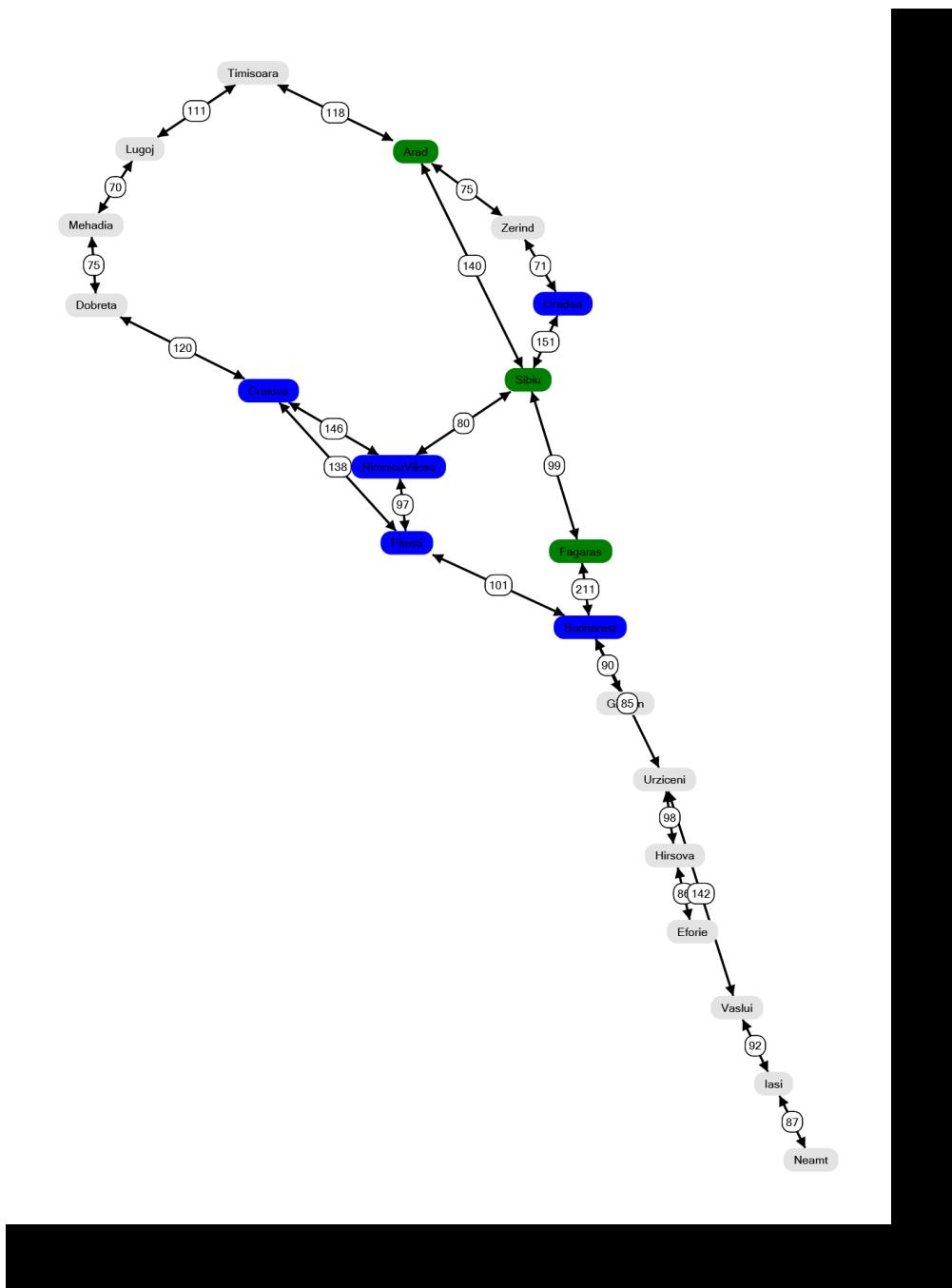
Pre-Visit

The algorithm has opened the node Fagaras



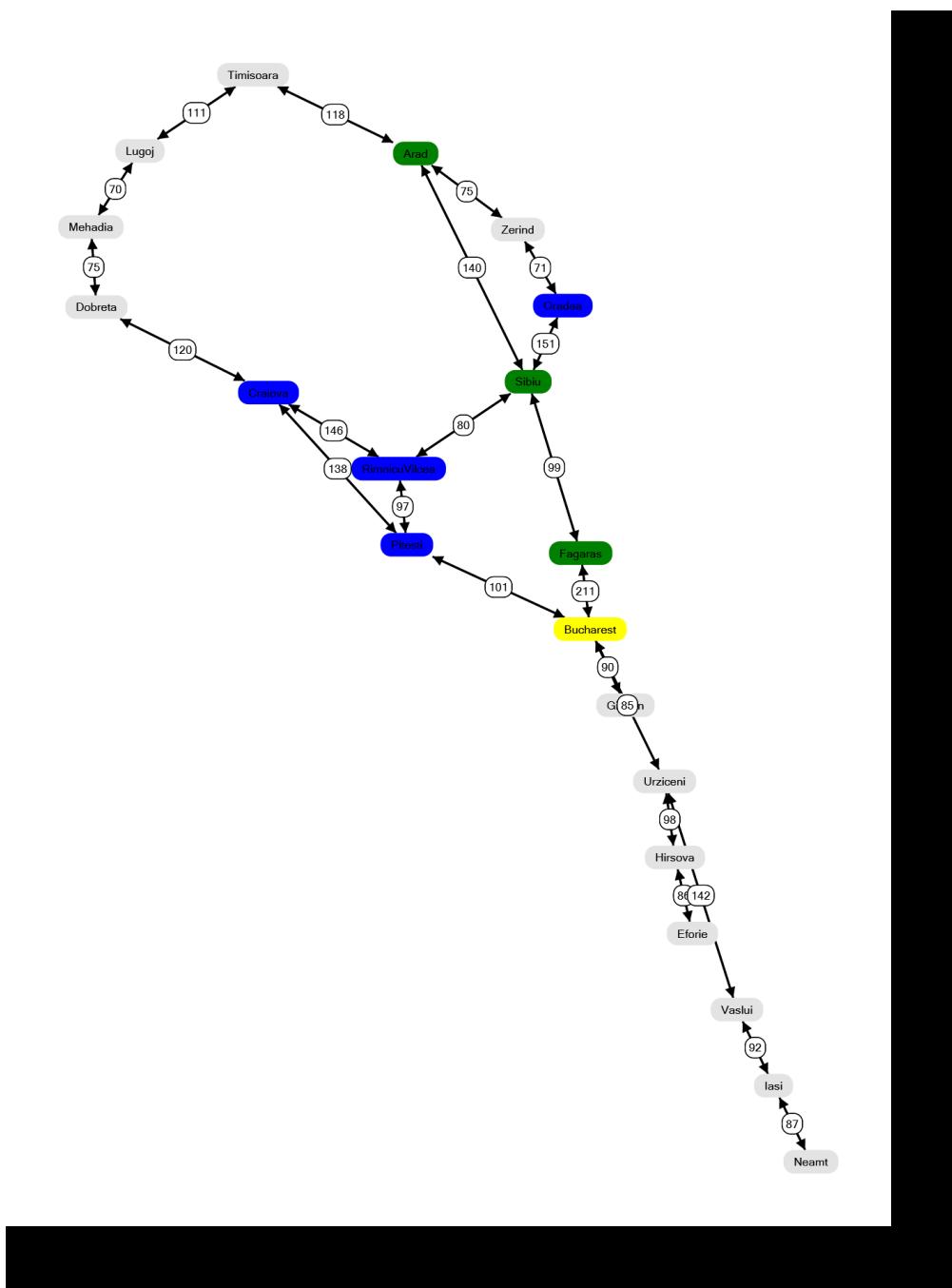
Visit

The algorithm has marked the node Fagaras as visited



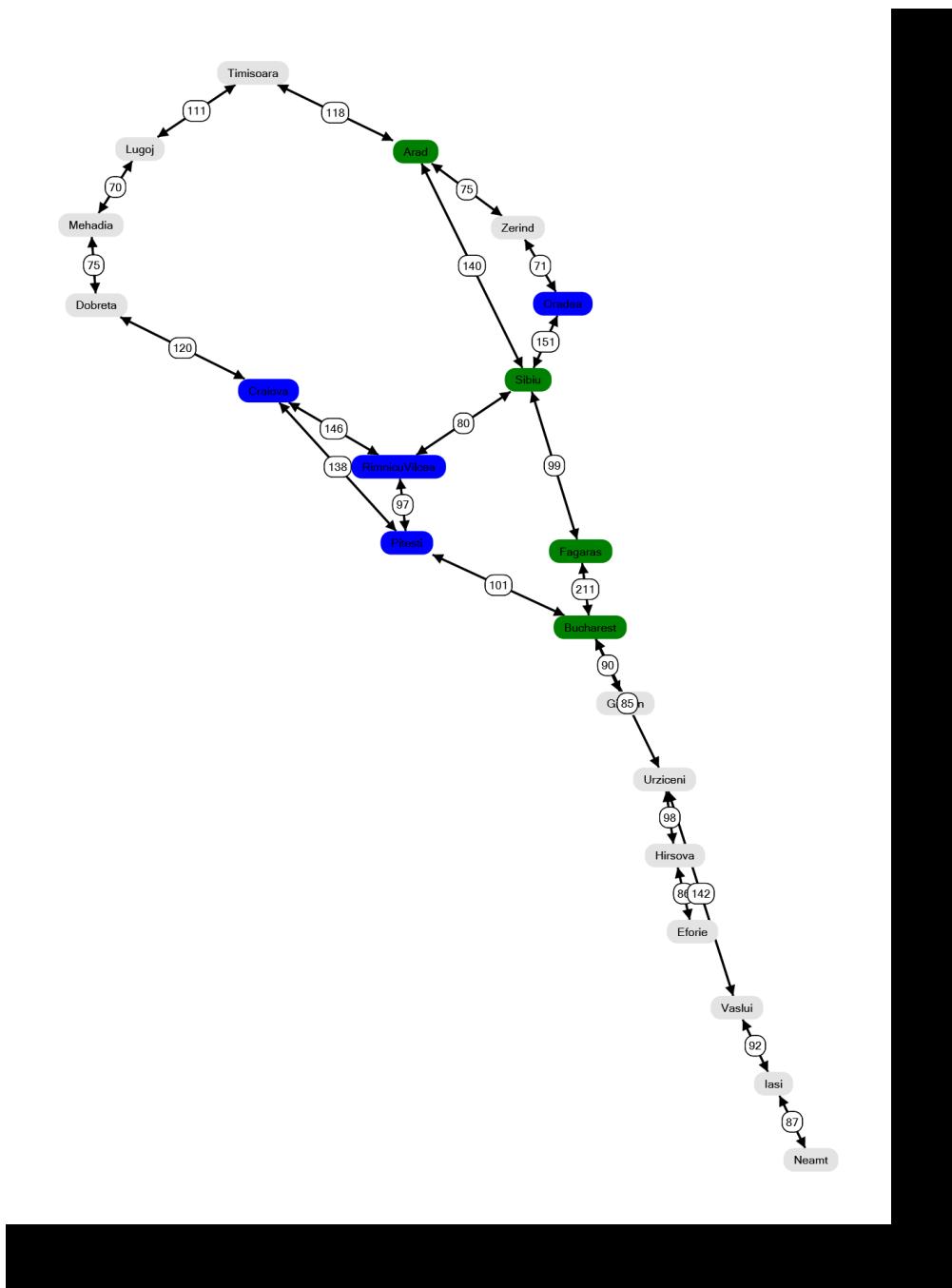
Pre-Visit

The algorithm has opened the node Bucharest



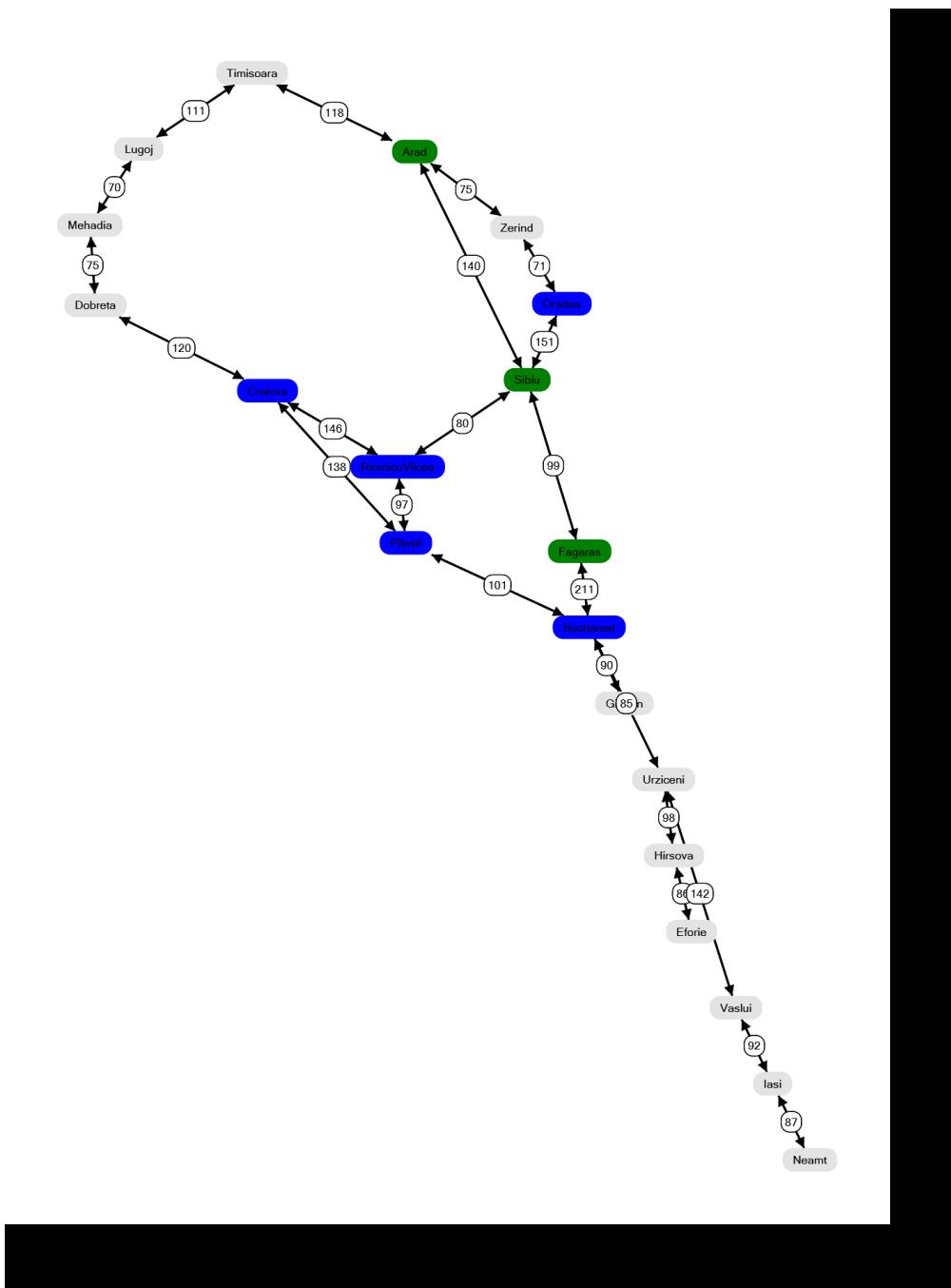
Visit

The algorithm has marked the node Bucharest as visited



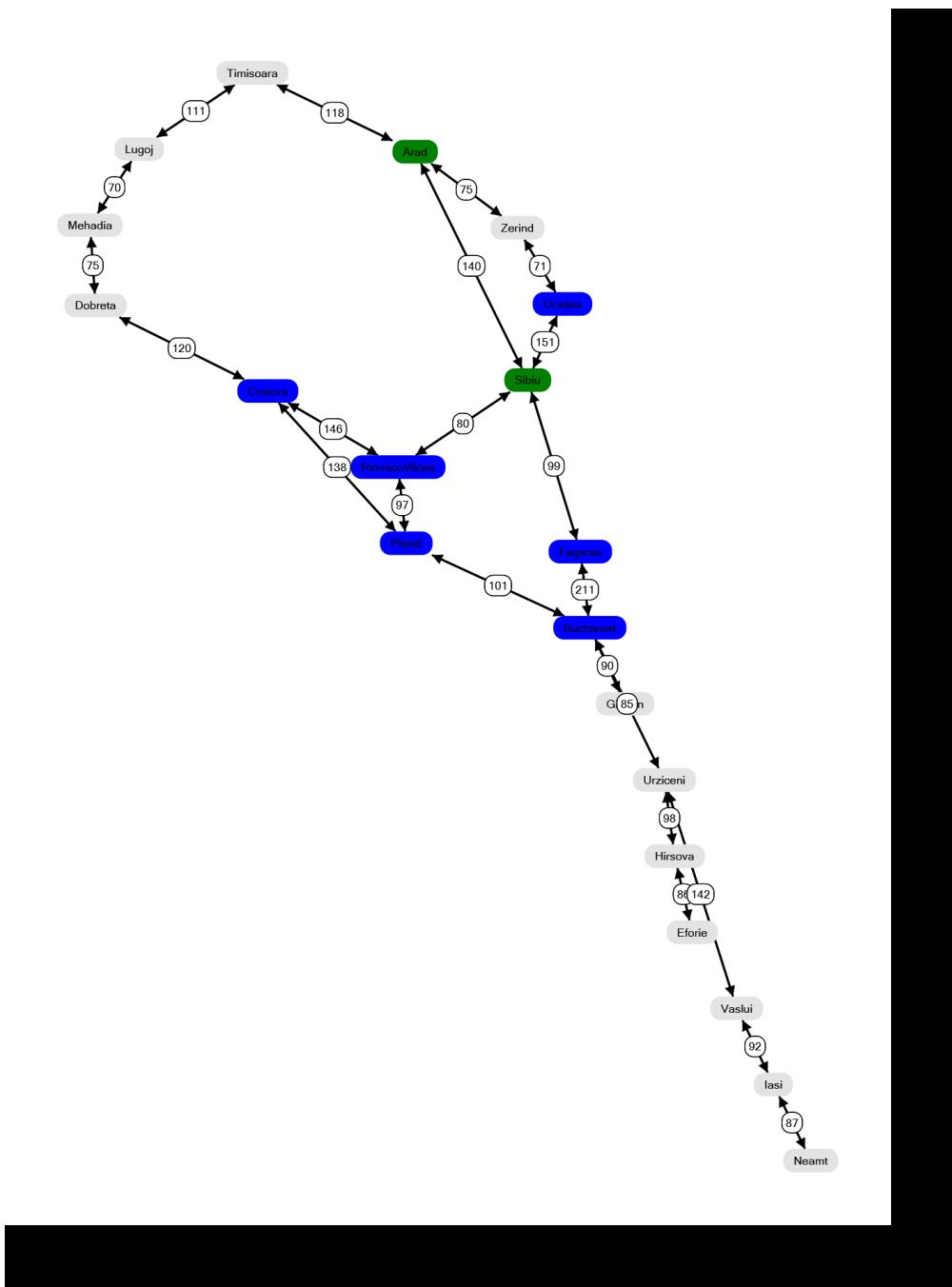
Post-Visit

The algorithm has closed the node Bucharest



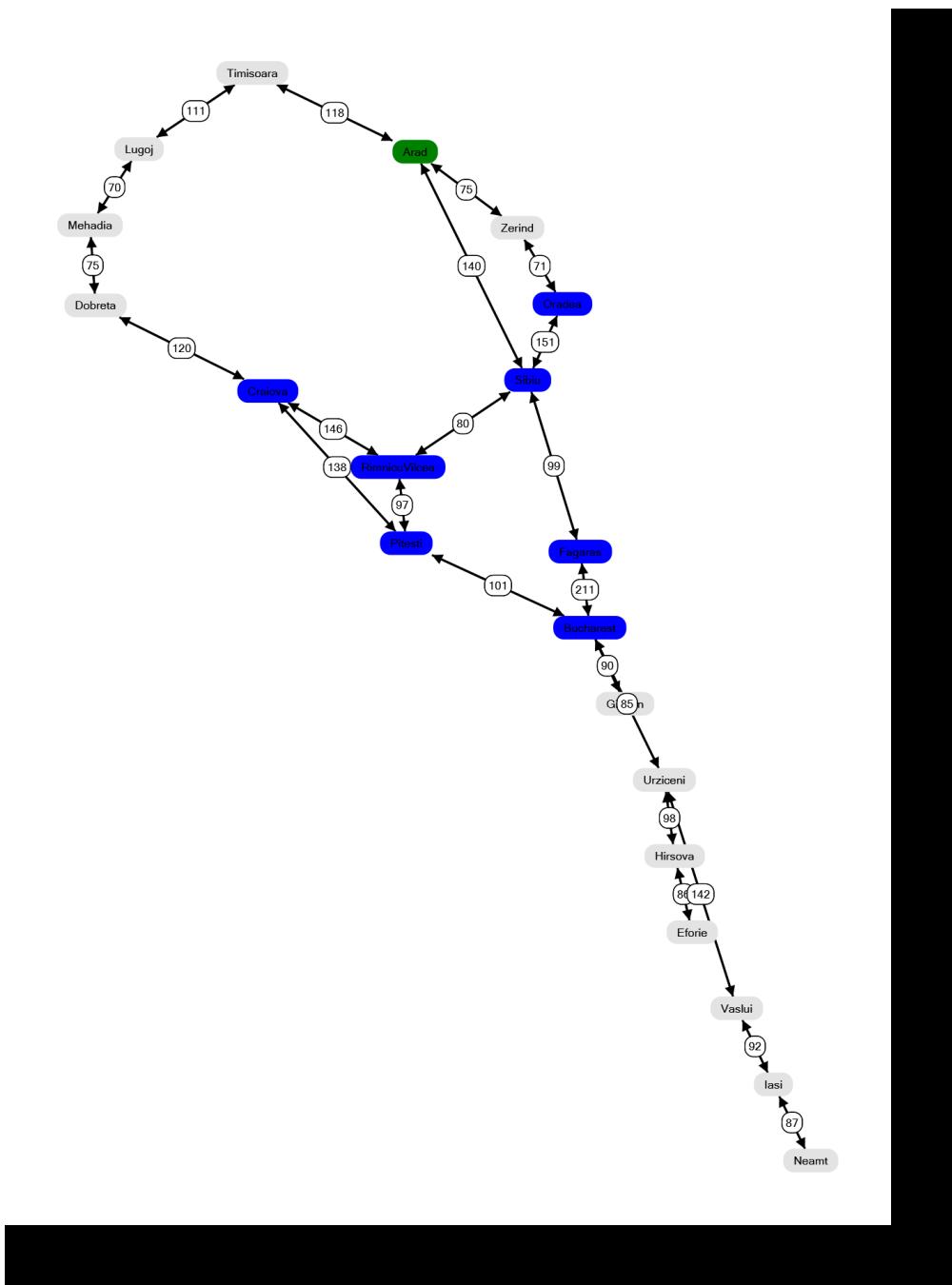
Post-Visit

The algorithm has closed the node Fagaras



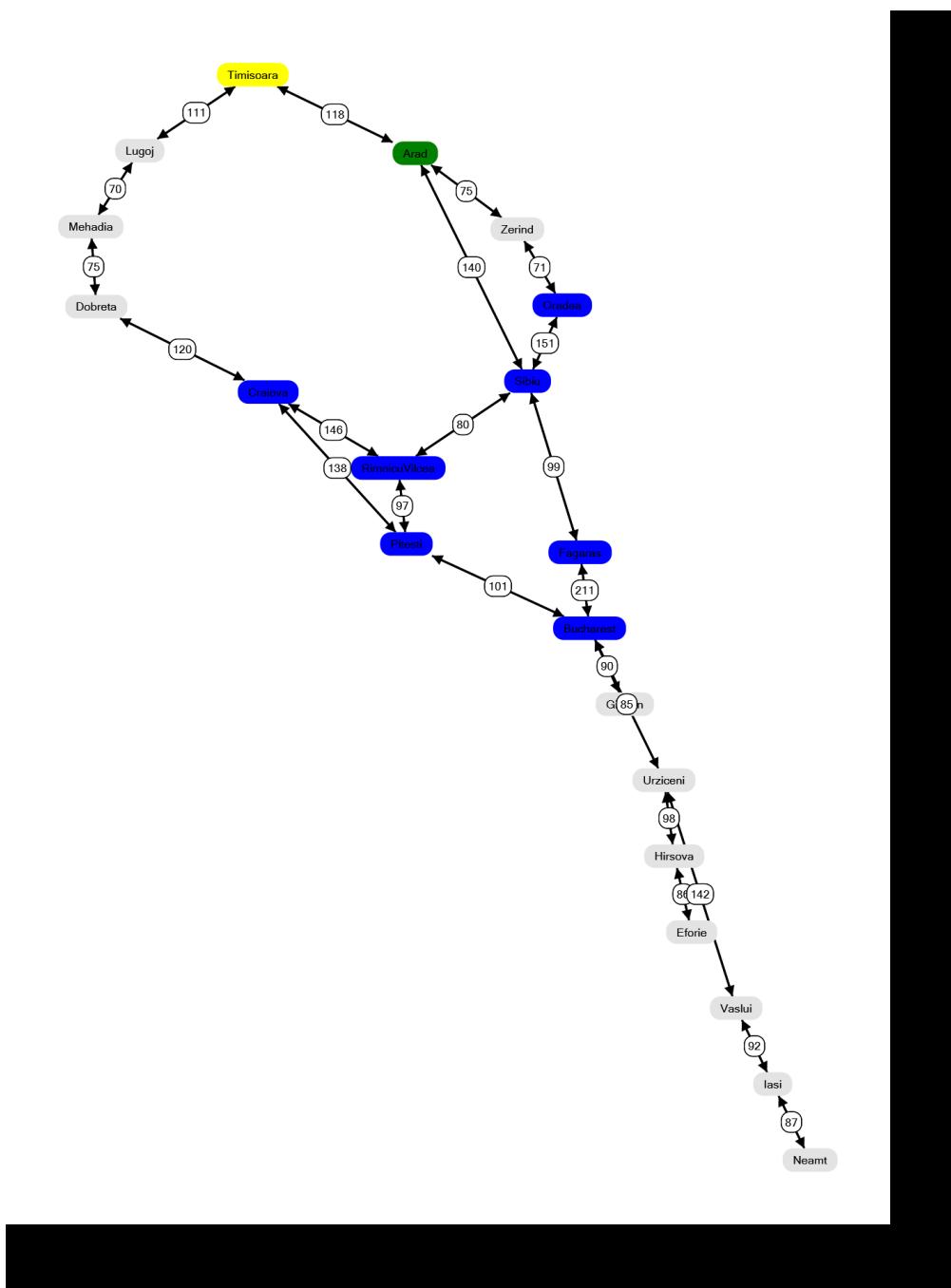
Post-Visit

The algorithm has closed the node Sibiu



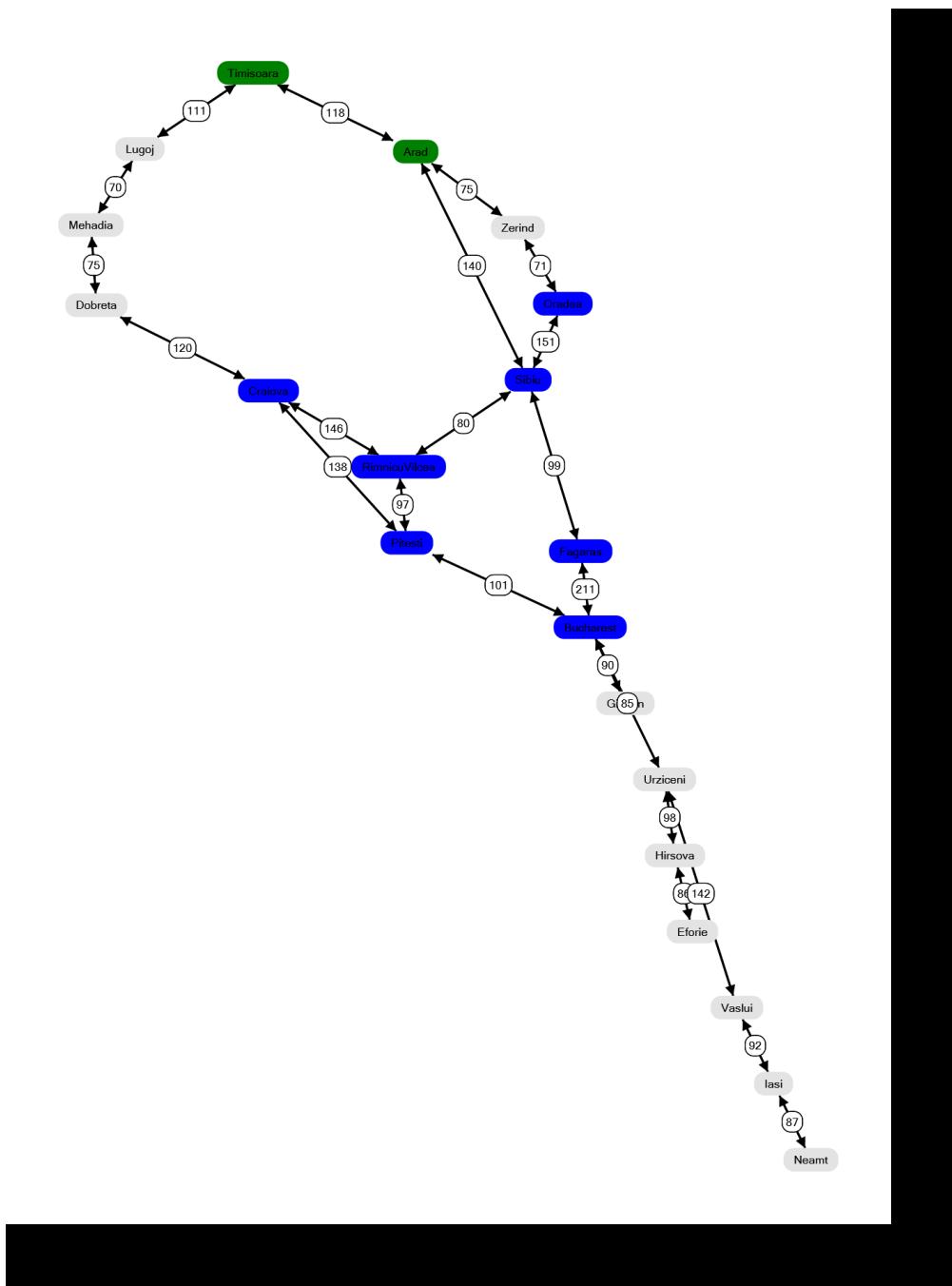
Pre-Visit

The algorithm has opened the node Timisoara



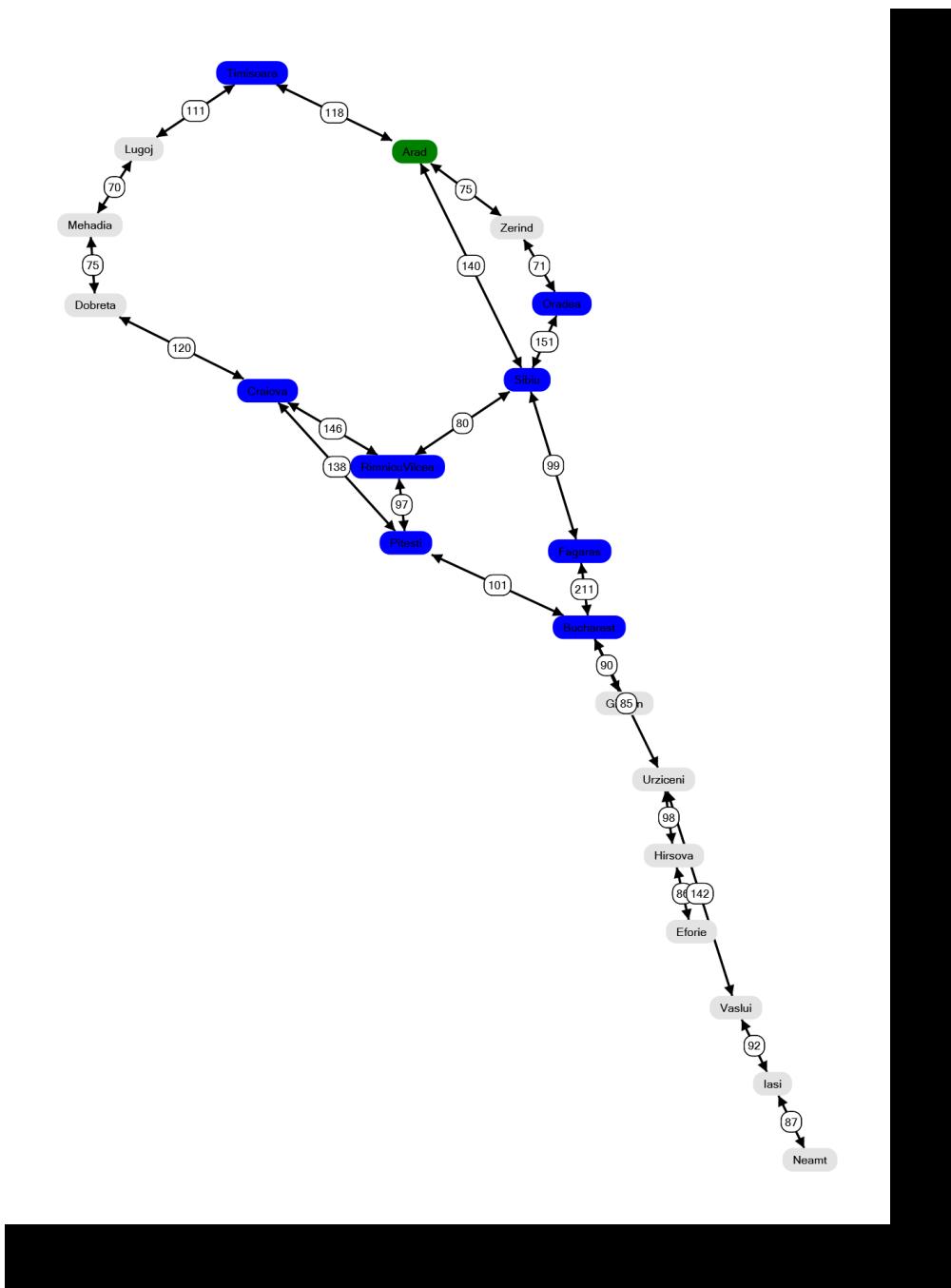
Visit

The algorithm has marked the node Timisoara as visited



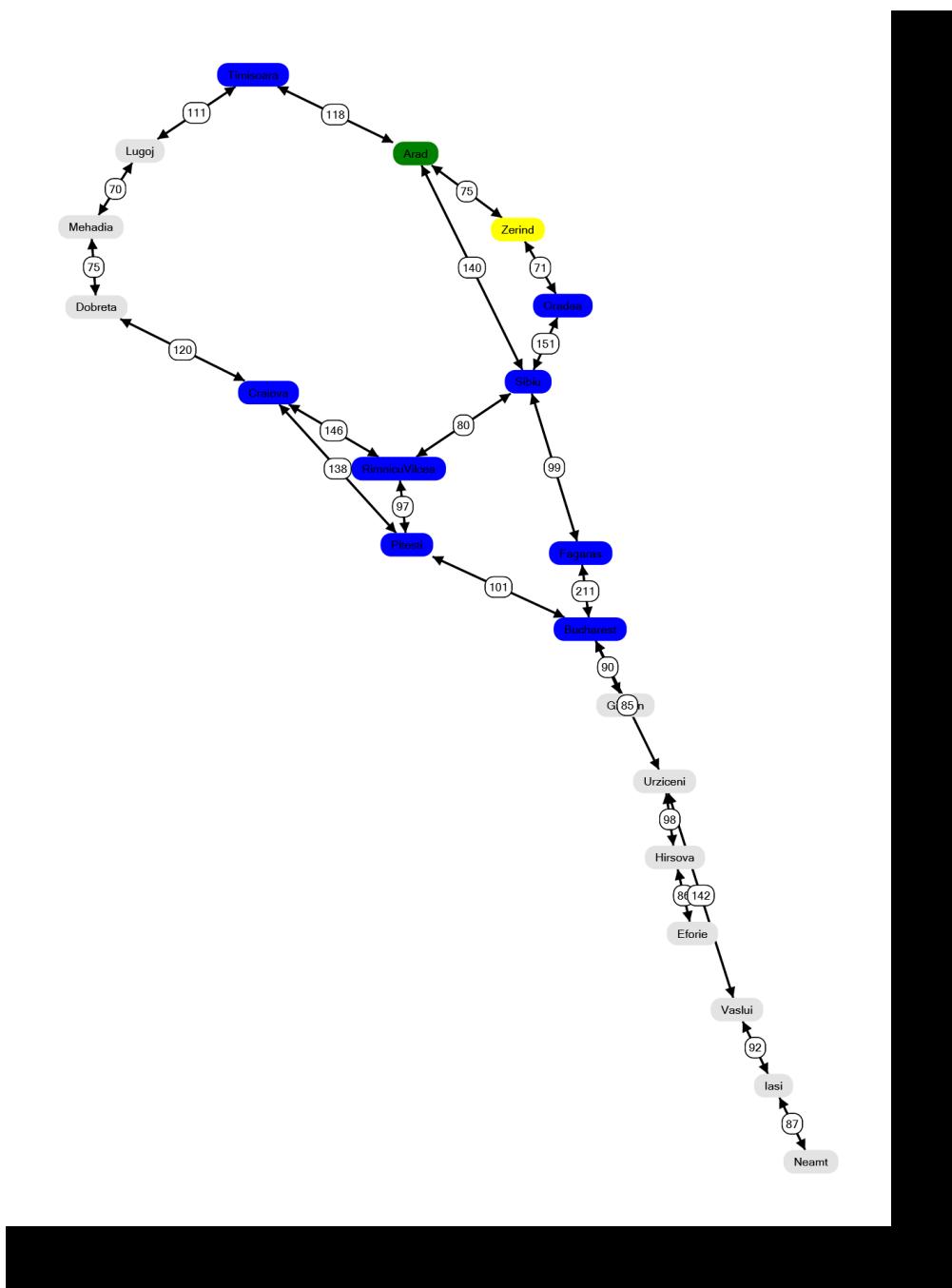
Post-Visit

The algorithm has closed the node Timisoara



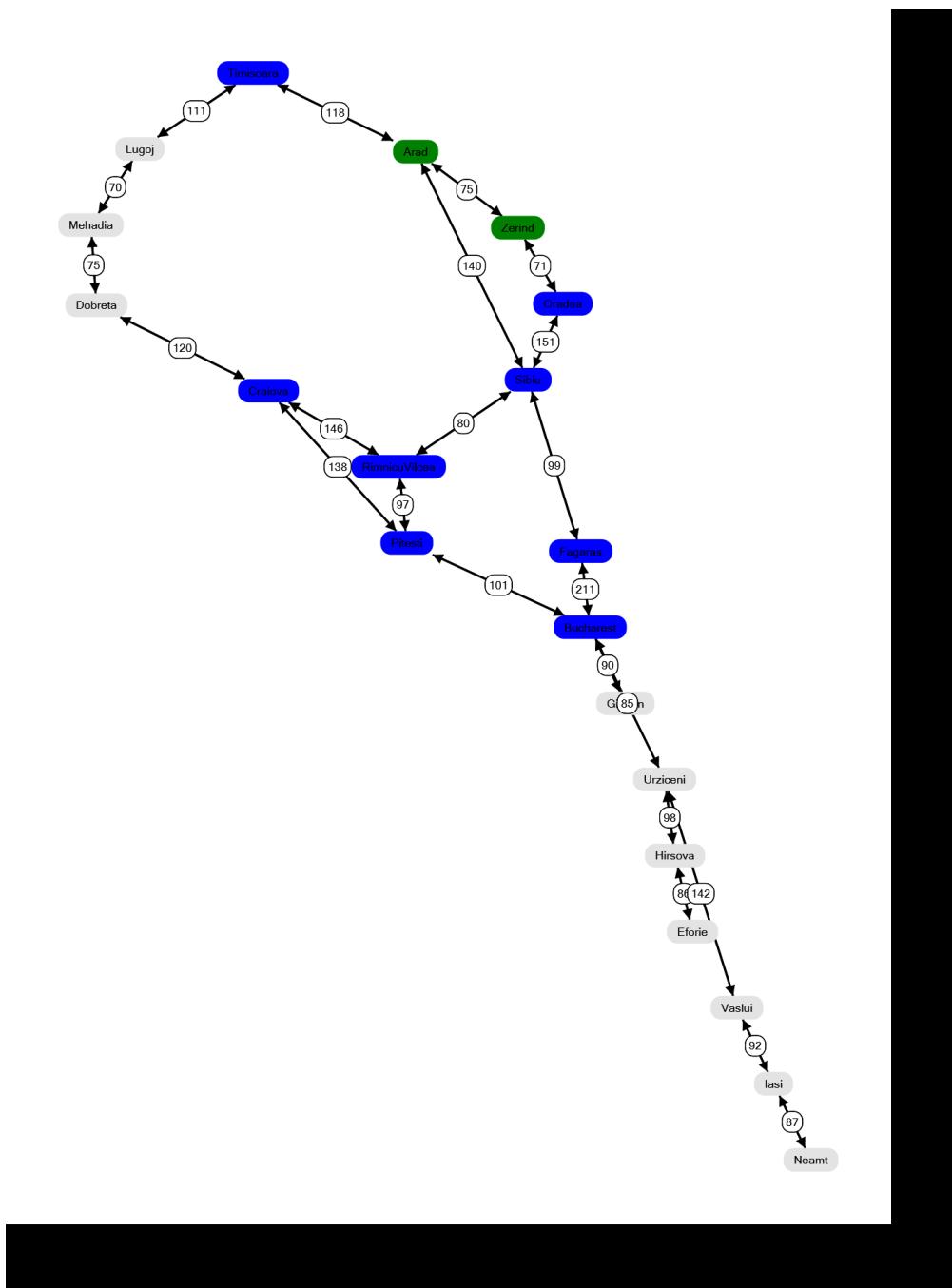
Pre-Visit

The algorithm has opened the node Zerind



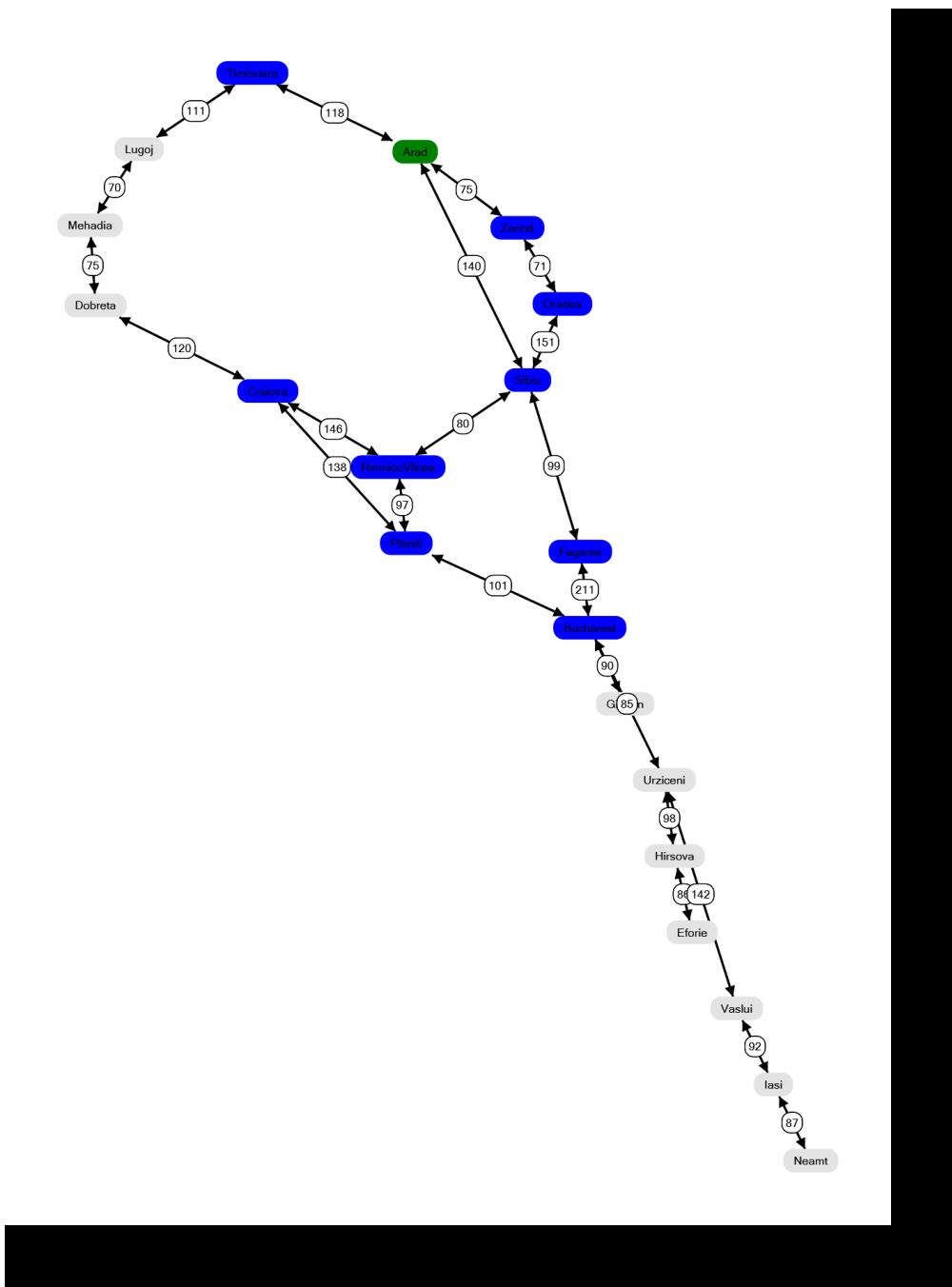
Visit

The algorithm has marked the node Zerind as visited



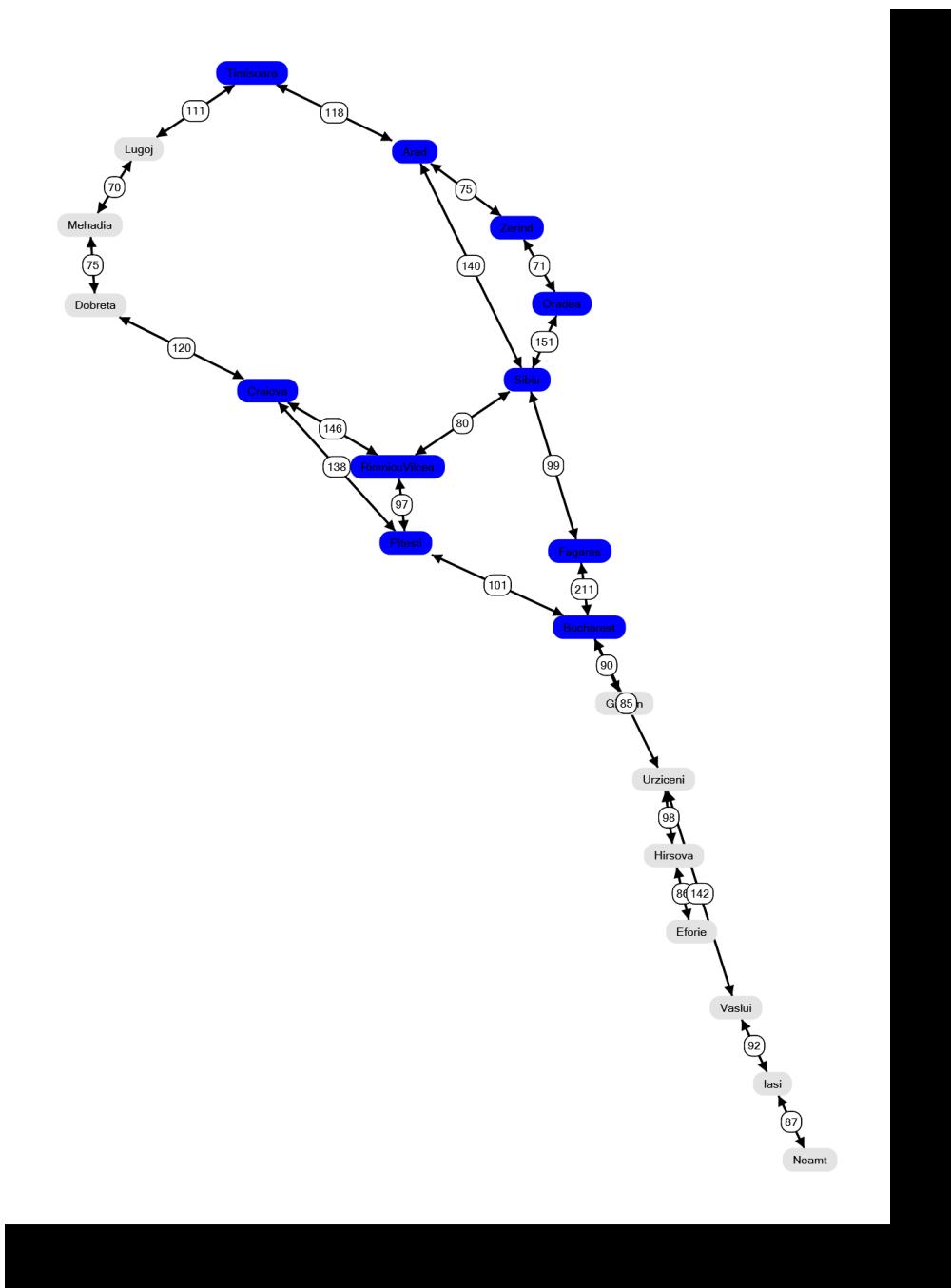
Post-Visit

The algorithm has closed the node Zerind



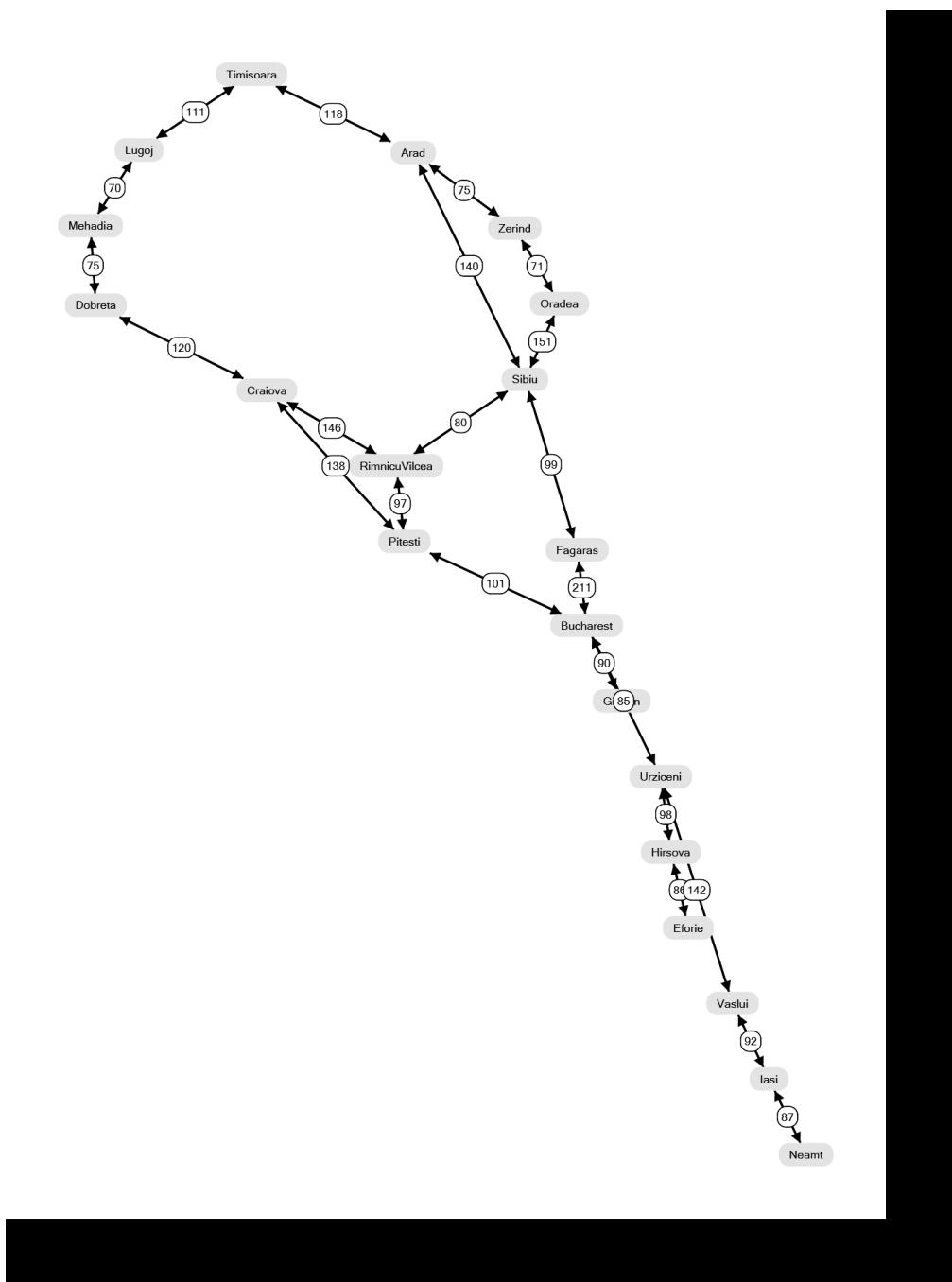
Post-Visit

The algorithm has closed the node Arad



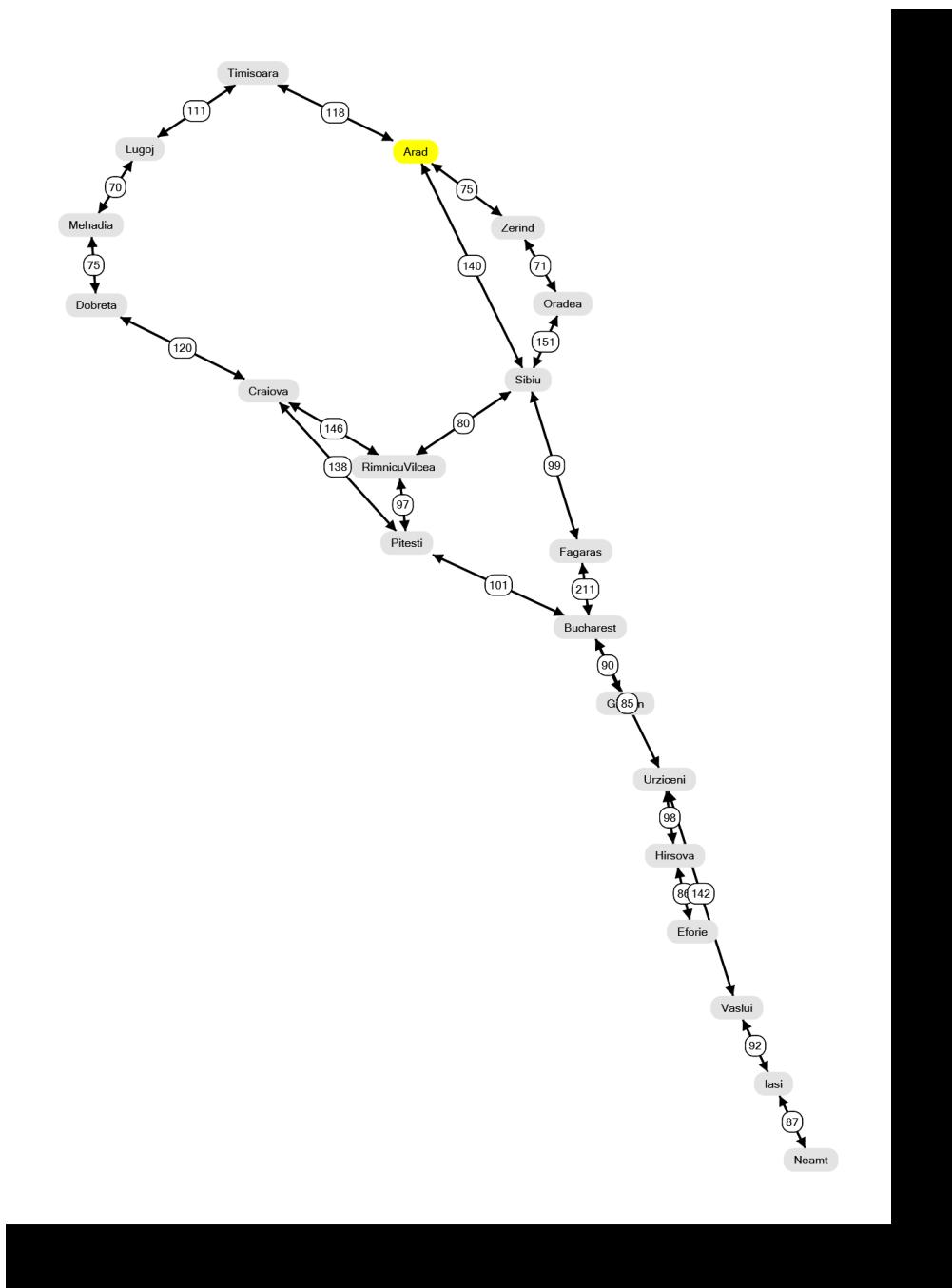
Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))



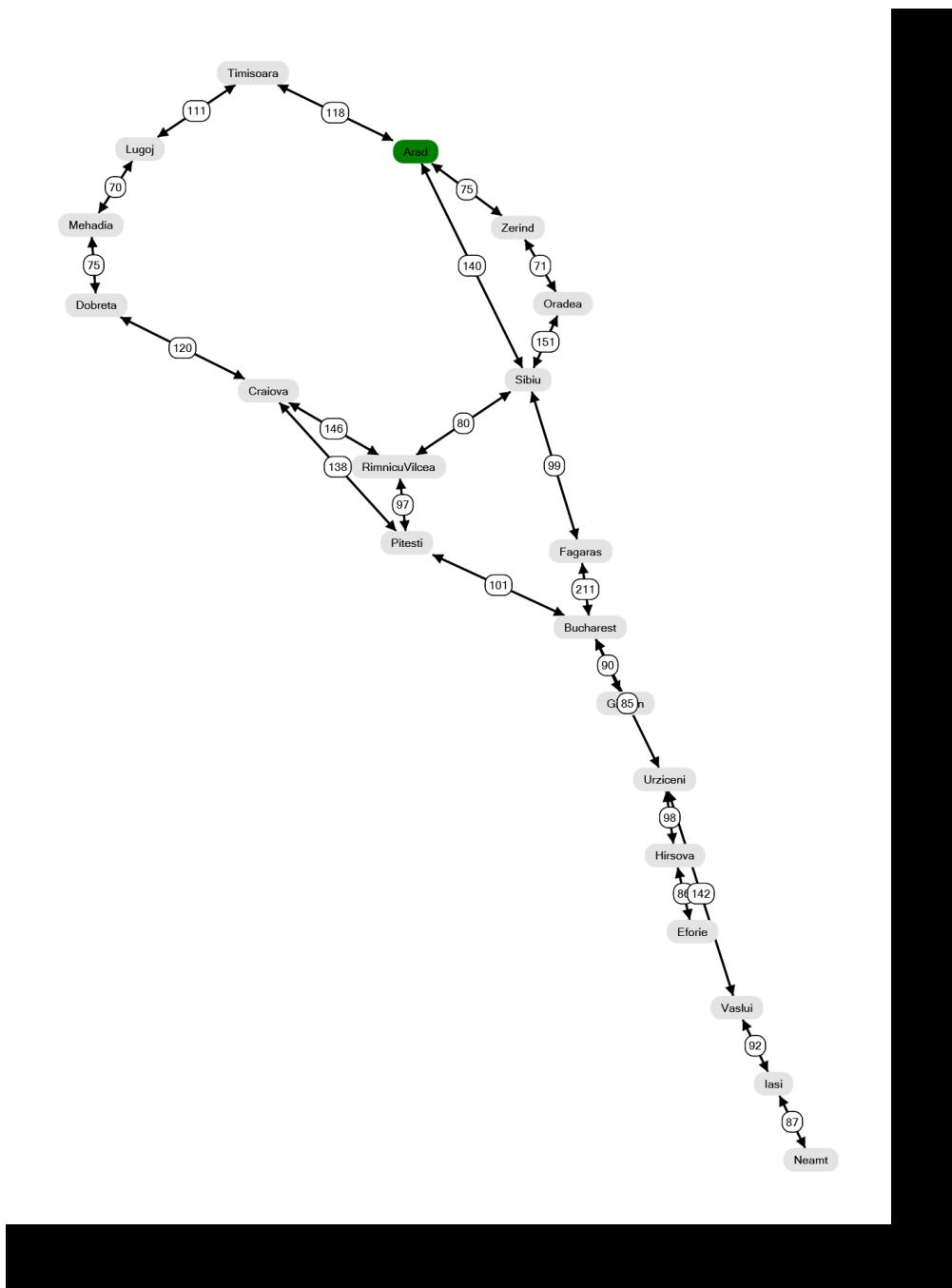
Pre-Visit

The algorithm has opened the node Arad



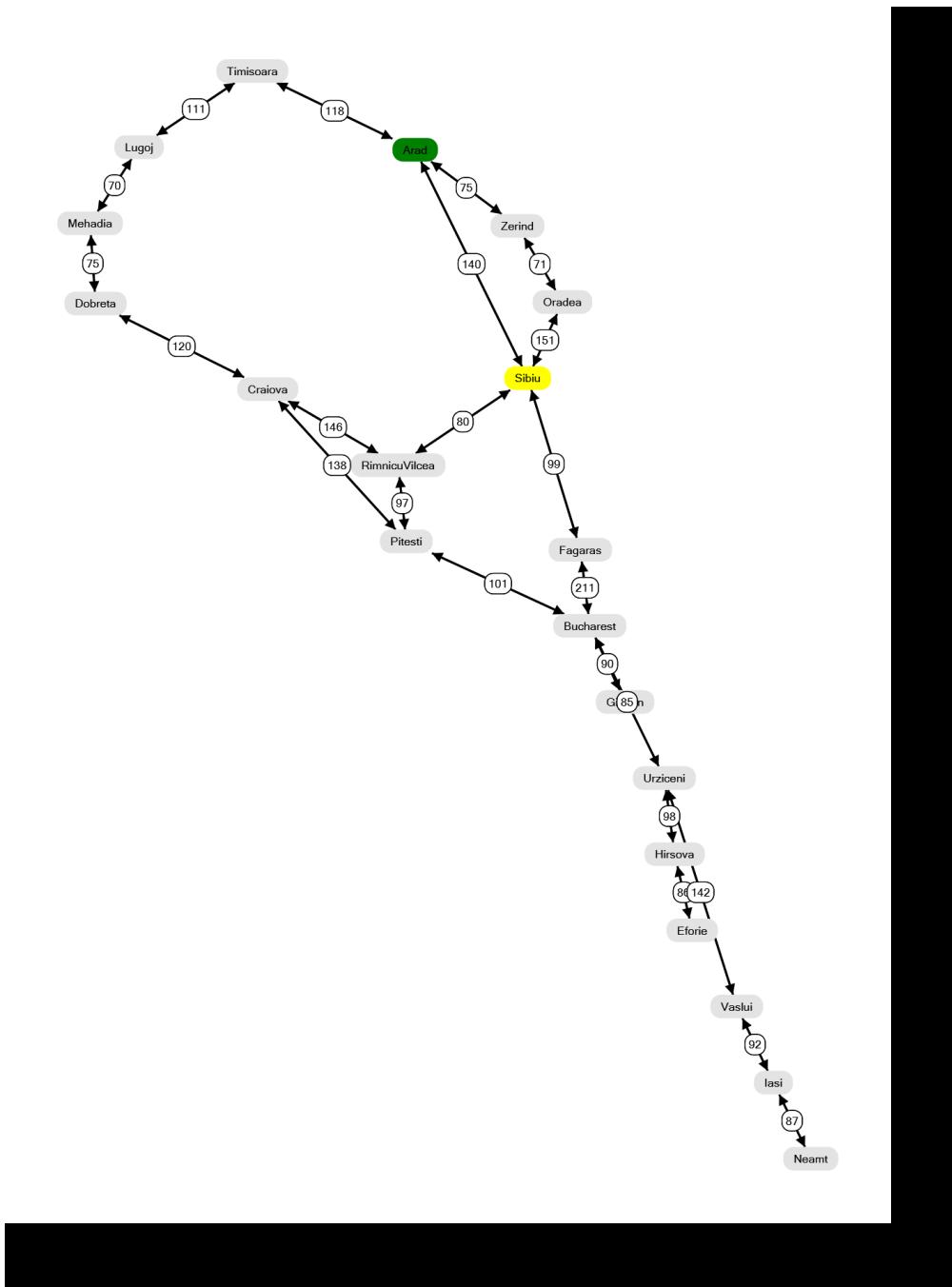
Visit

The algorithm has marked the node Arad as visited



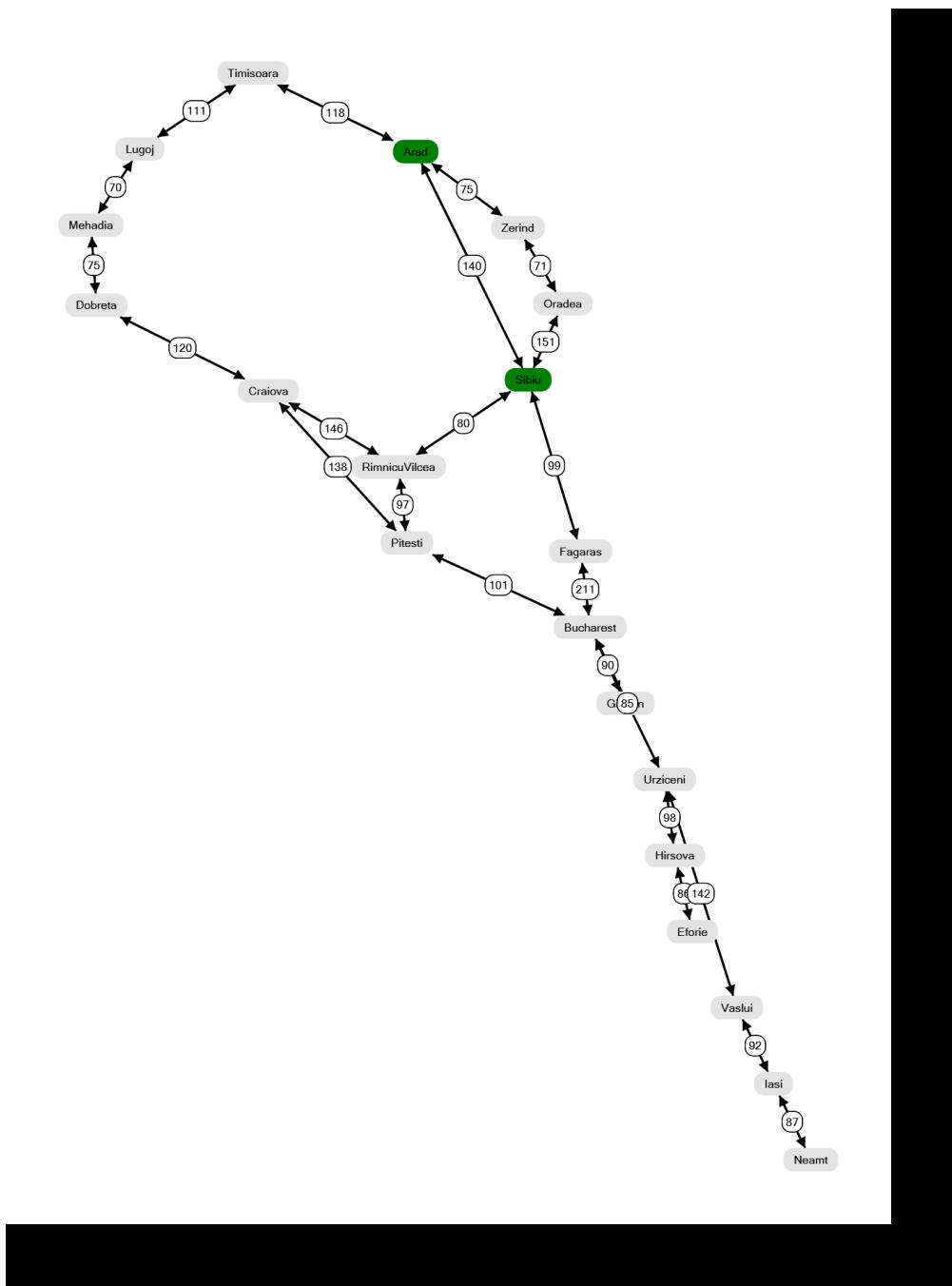
Pre-Visit

The algorithm has opened the node Sibiu



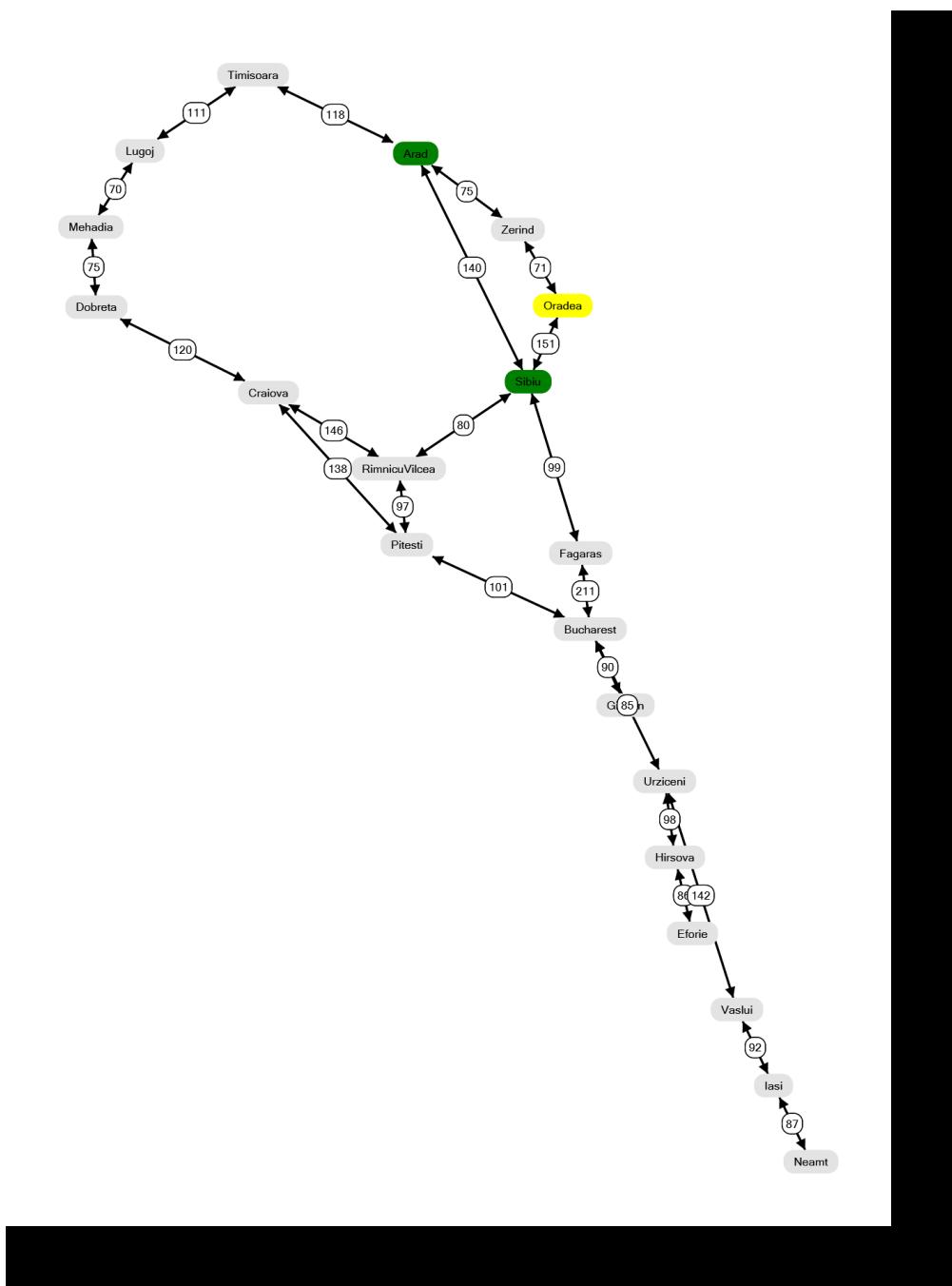
Visit

The algorithm has marked the node Sibiu as visited



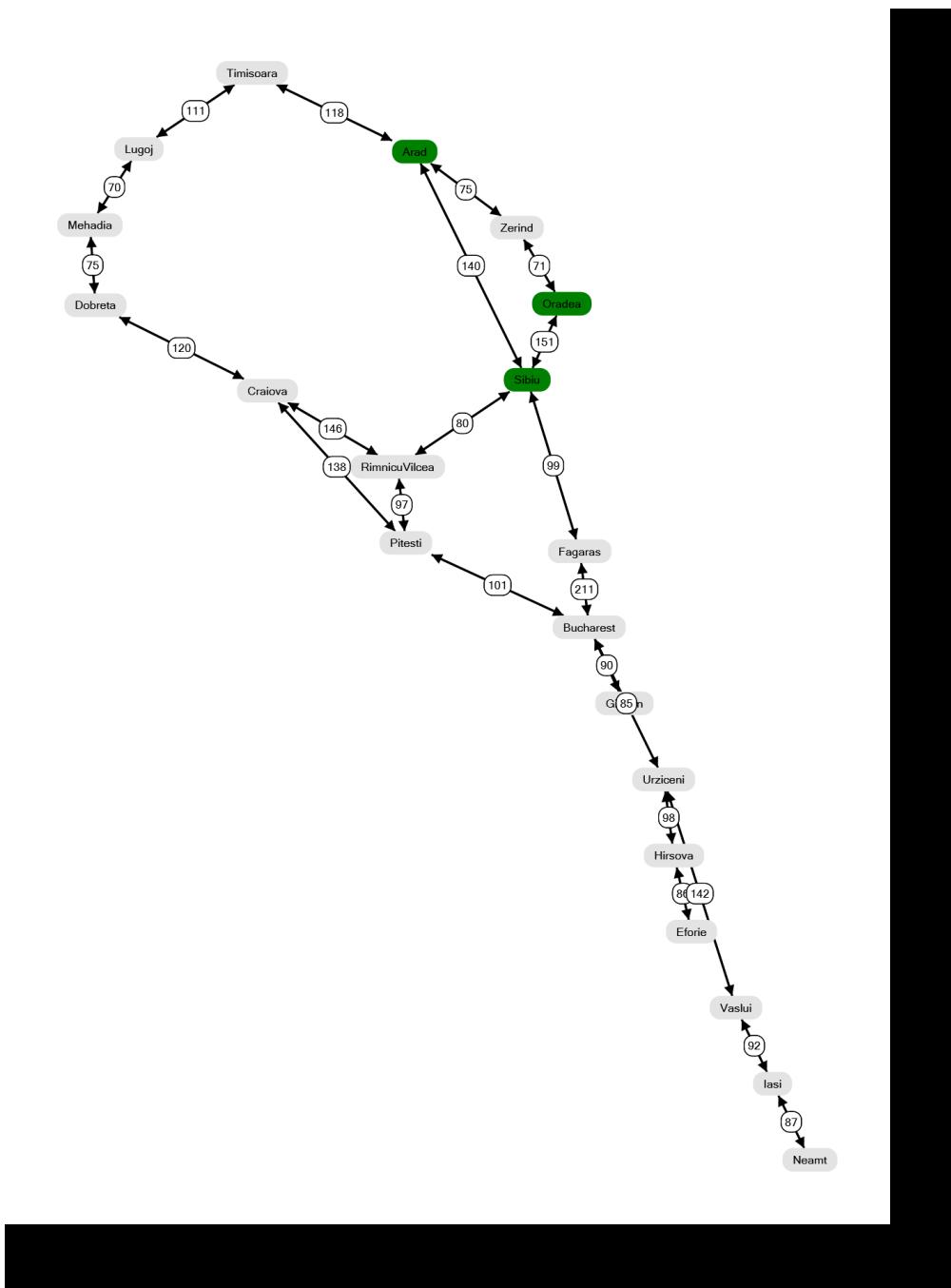
Pre-Visit

The algorithm has opened the node Oradea



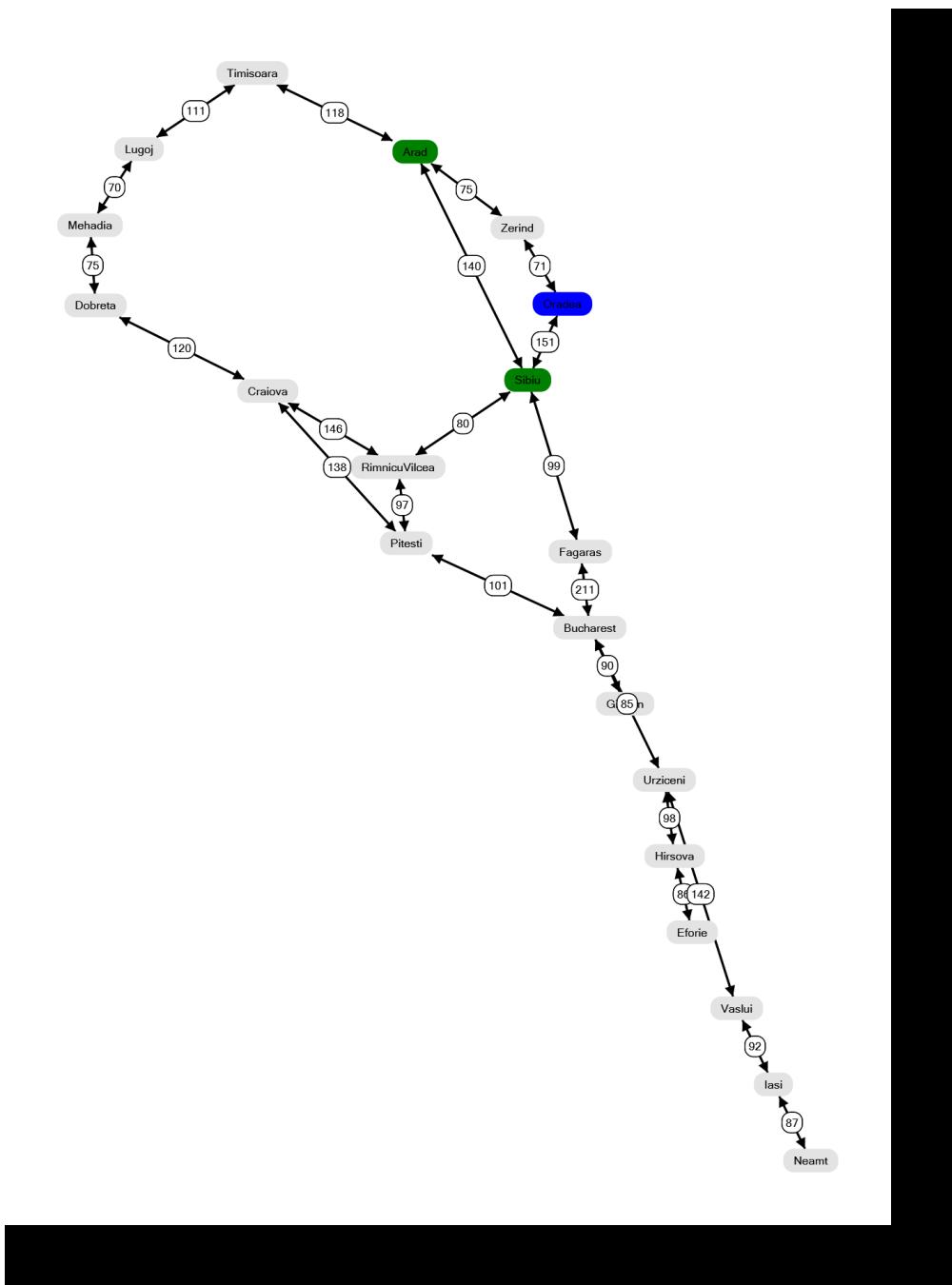
Visit

The algorithm has marked the node Oradea as visited



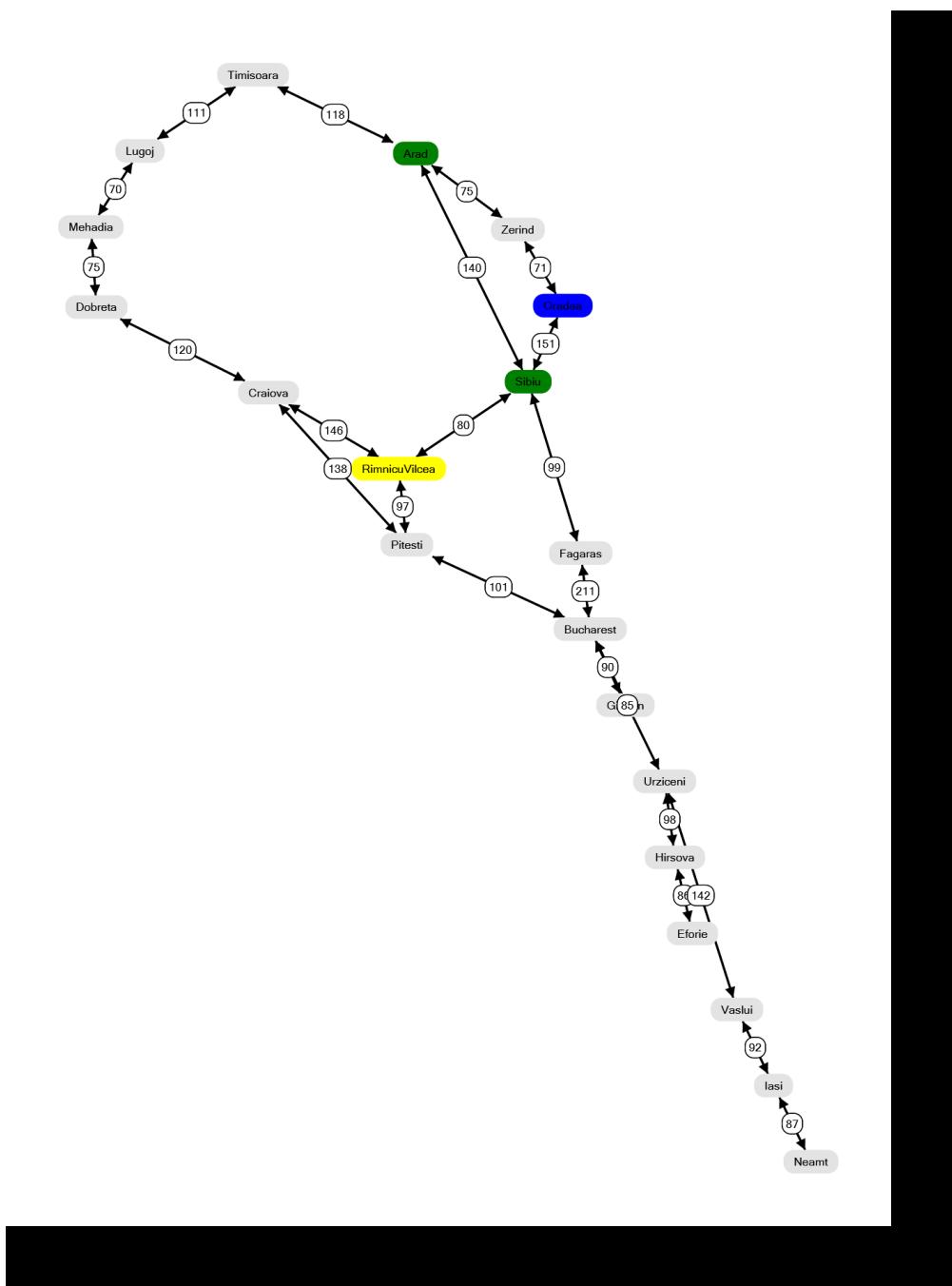
Post-Visit

The algorithm has closed the node Oradea



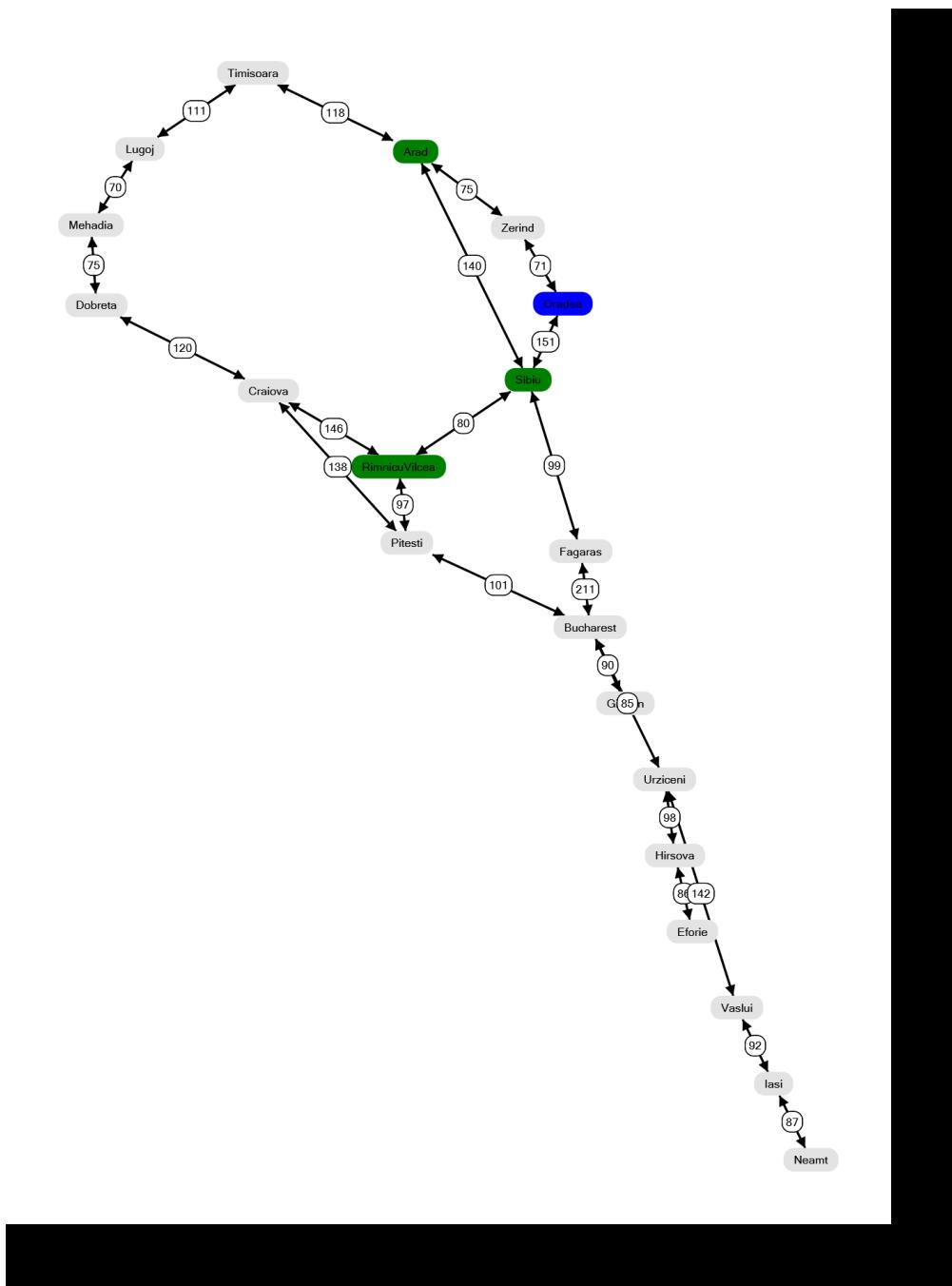
Pre-Visit

The algorithm has opened the node Rimnicu Vilcea



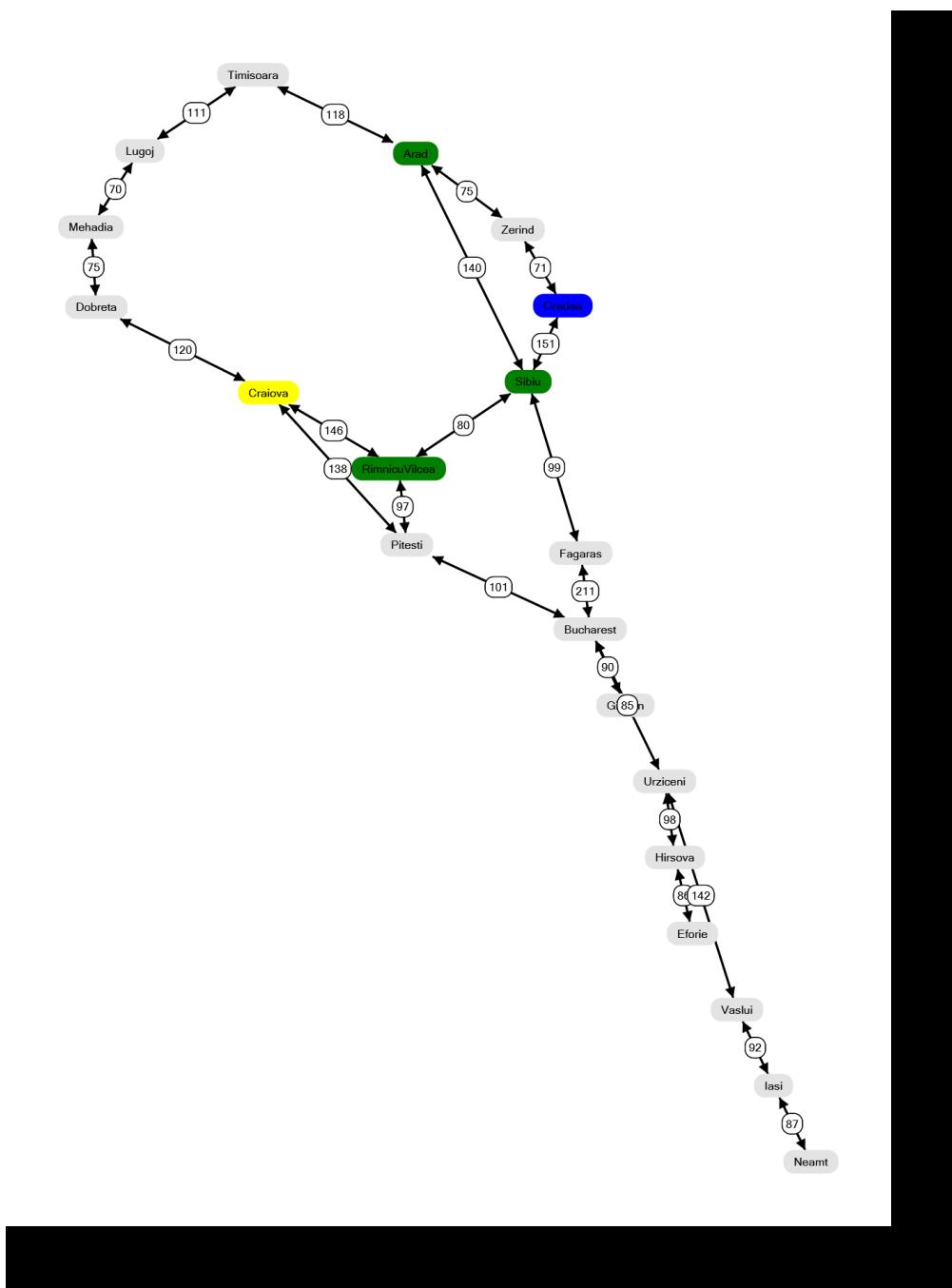
Visit

The algorithm has marked the node Rimnicu Vilcea as visited



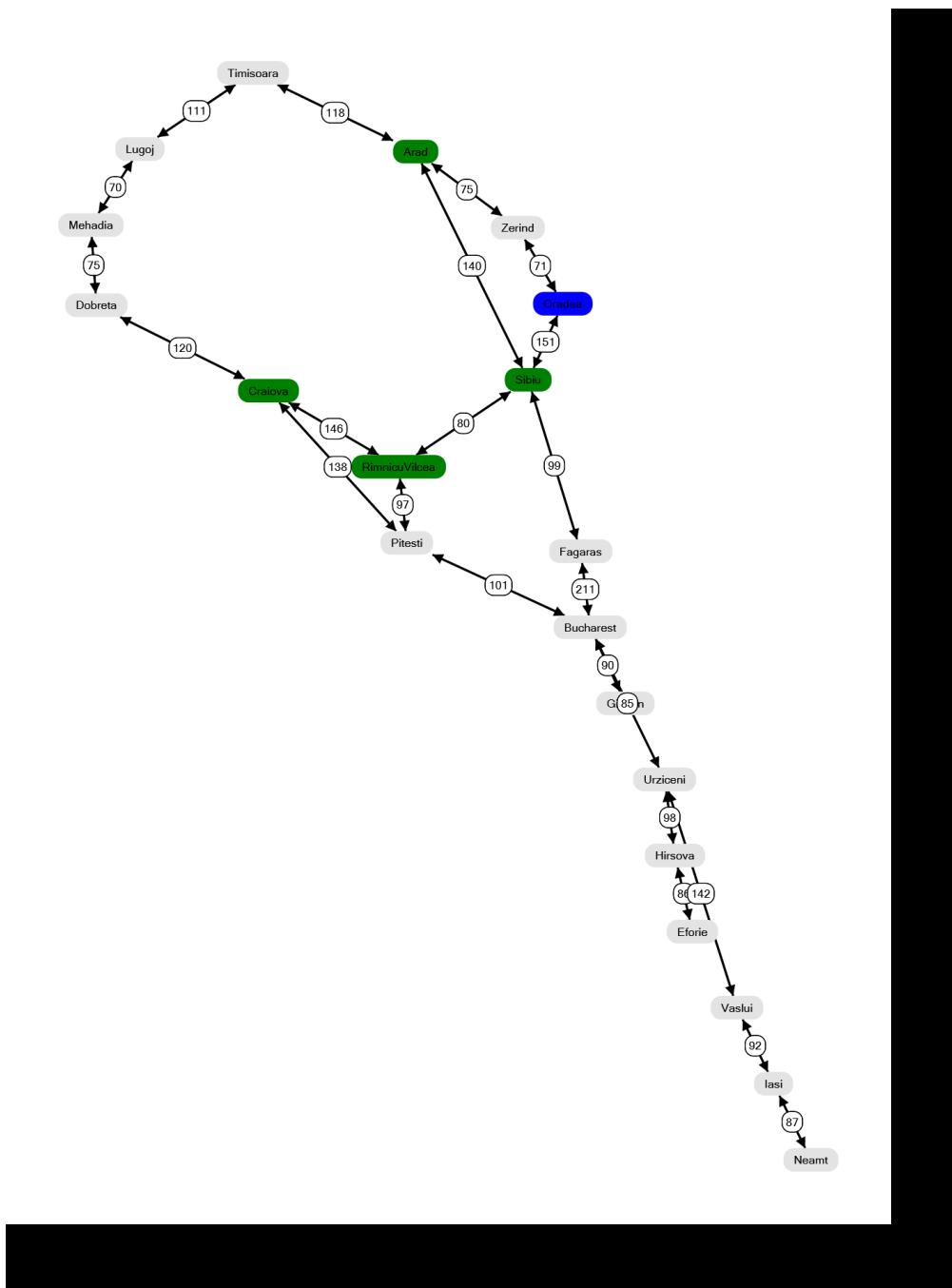
Pre-Visit

The algorithm has opened the node Craiova



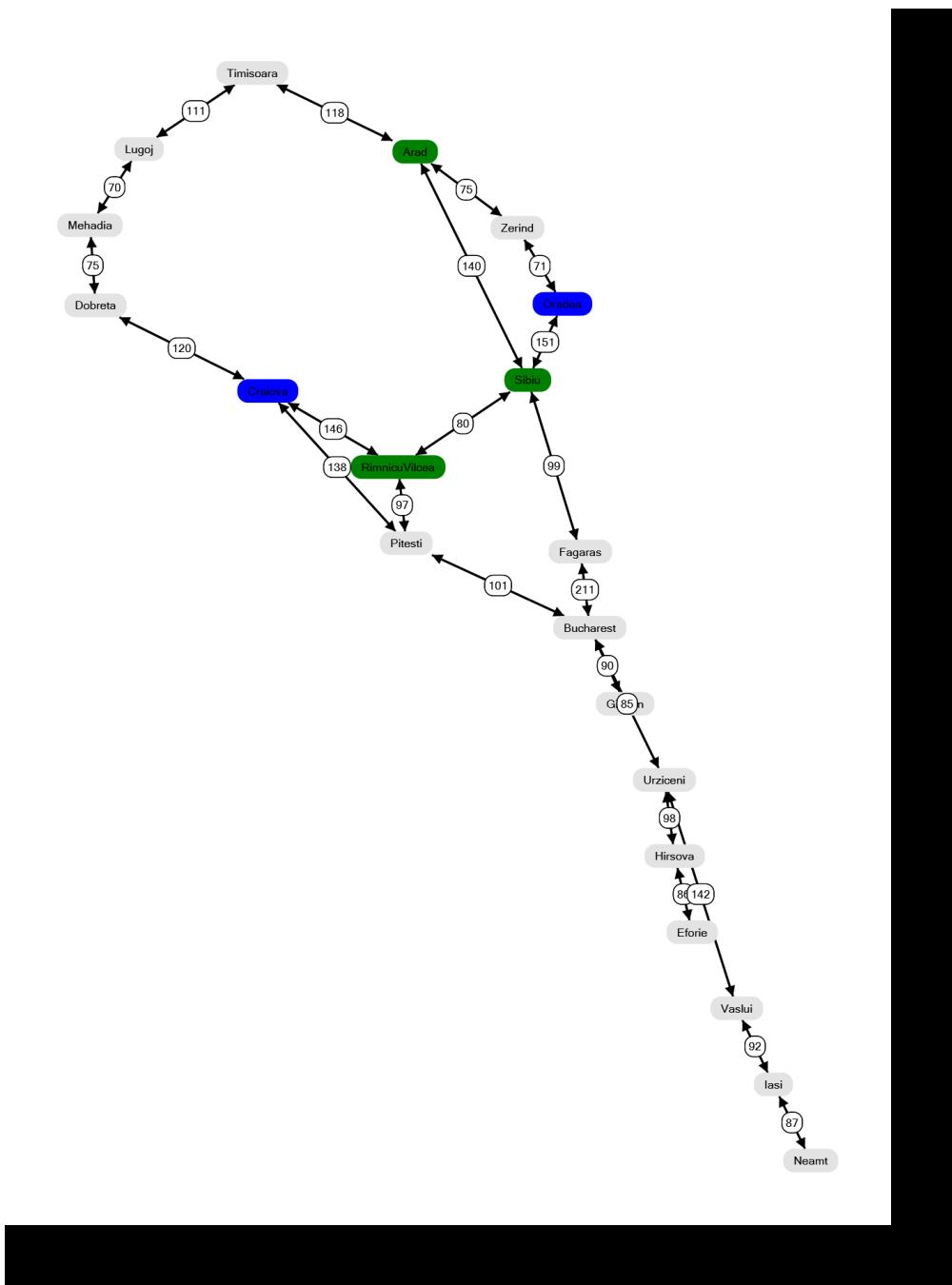
Visit

The algorithm has marked the node Craiova as visited



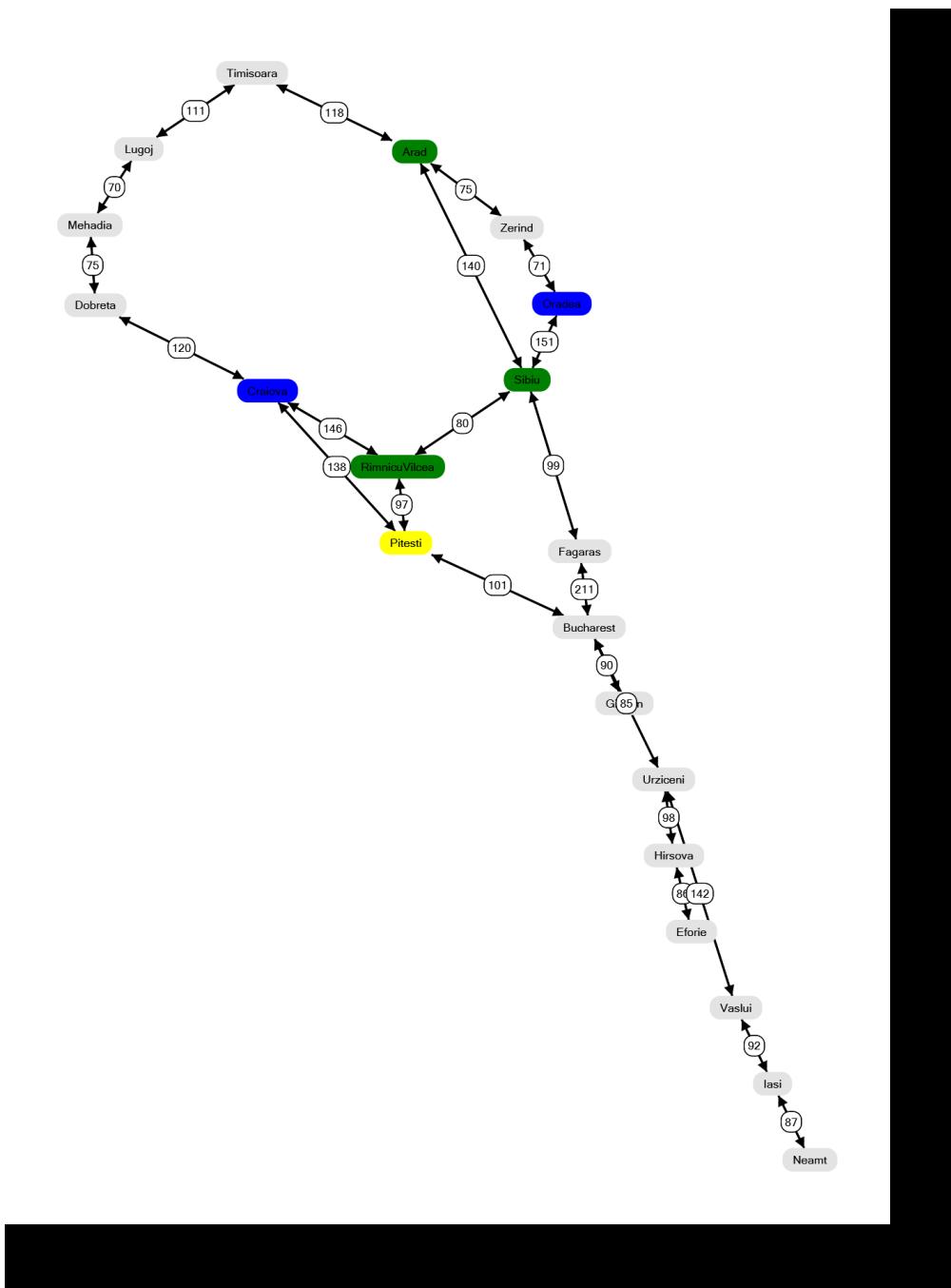
Post-Visit

The algorithm has closed the node Craiova



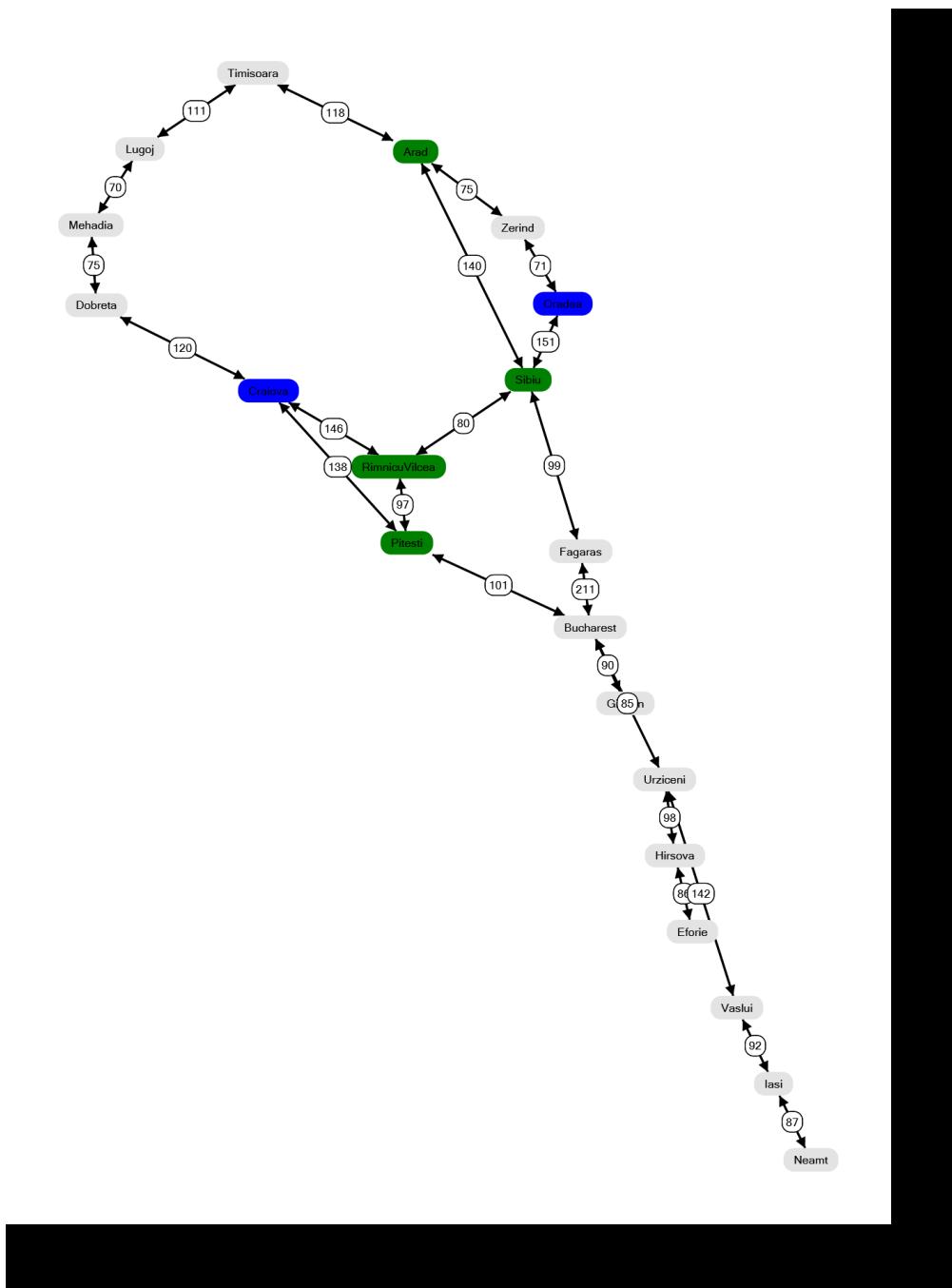
Pre-Visit

The algorithm has opened the node Pitesti



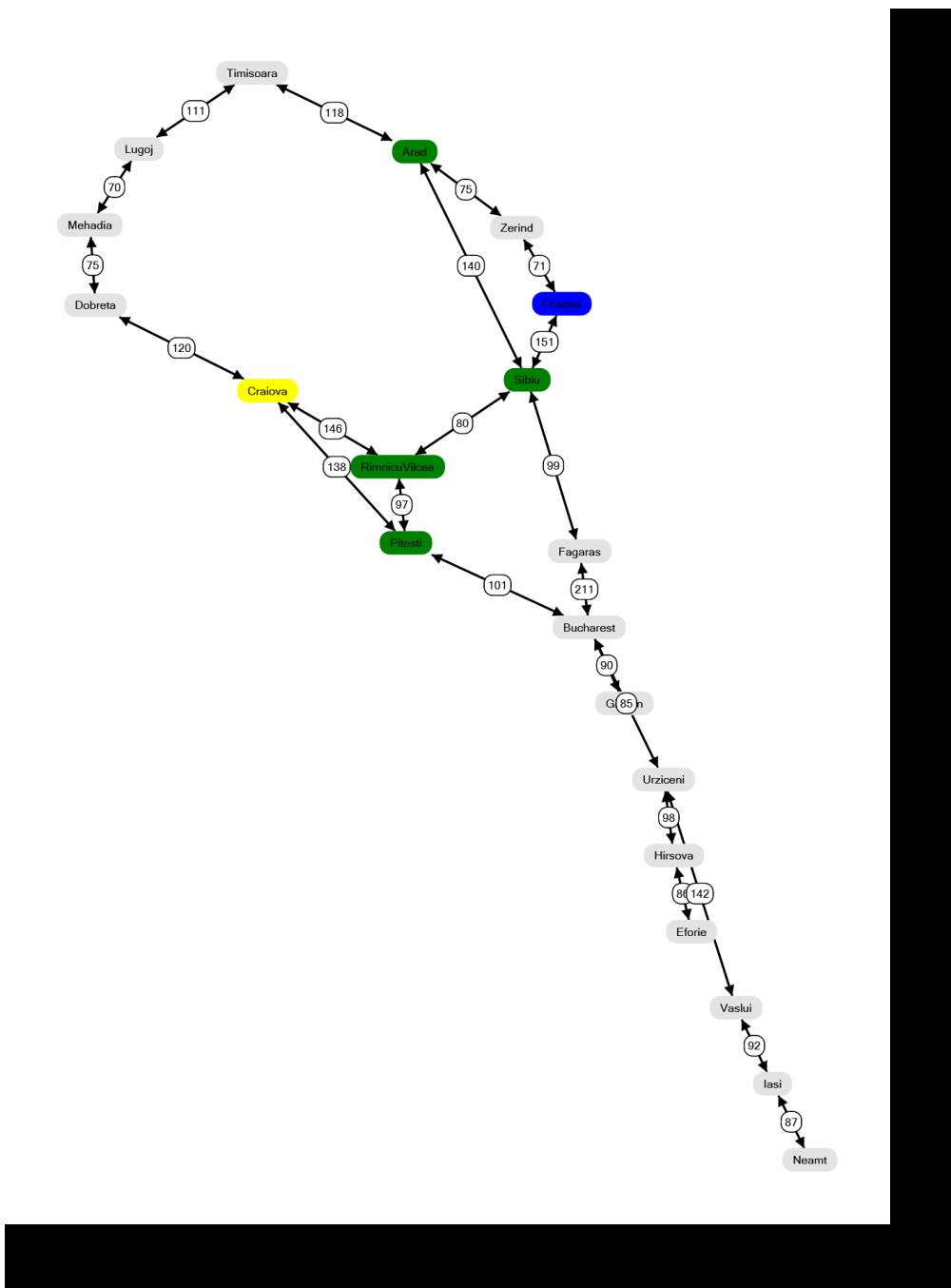
Visit

The algorithm has marked the node Pitesti as visited



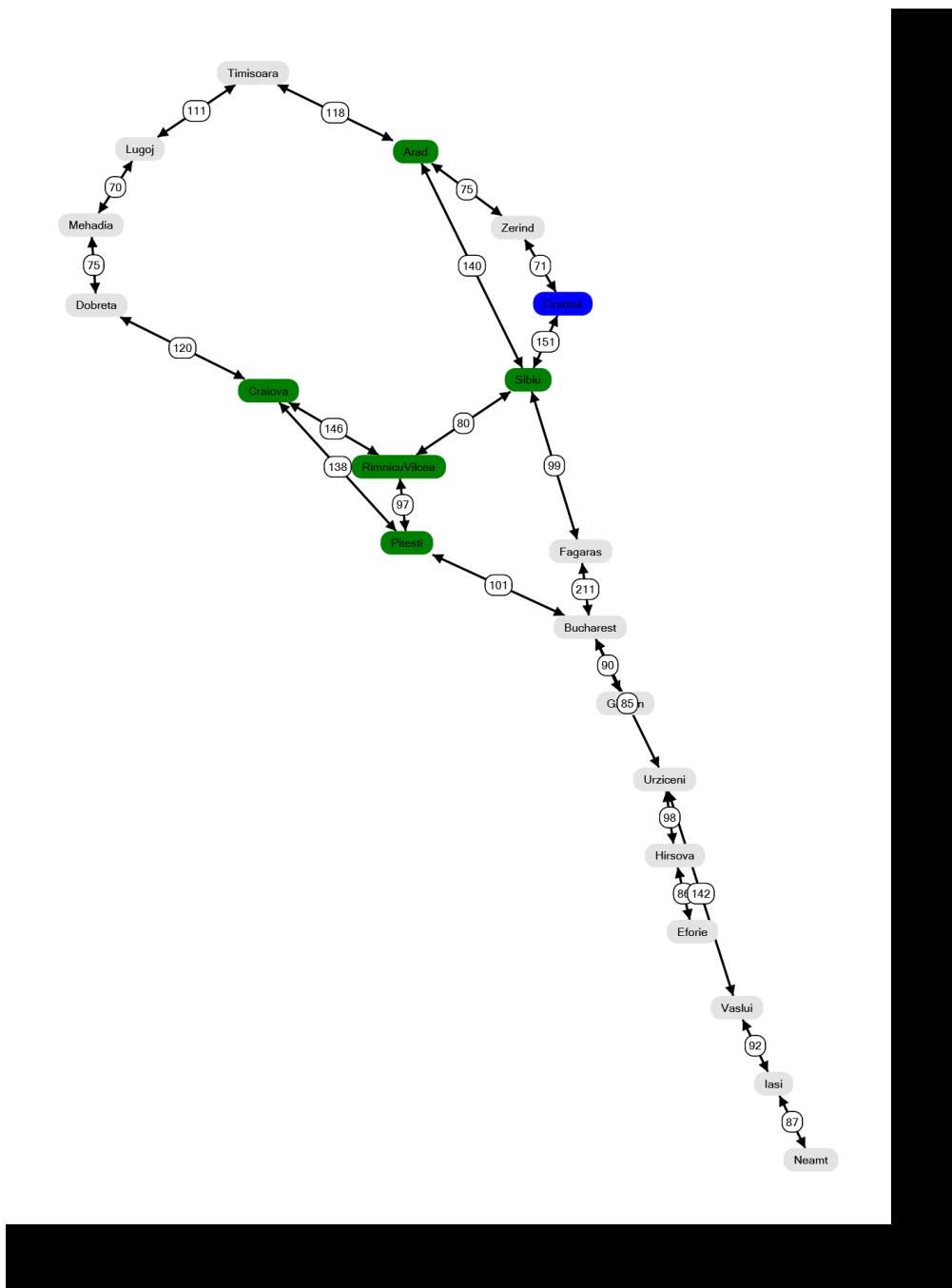
Pre-Visit

The algorithm has opened the node Craiova



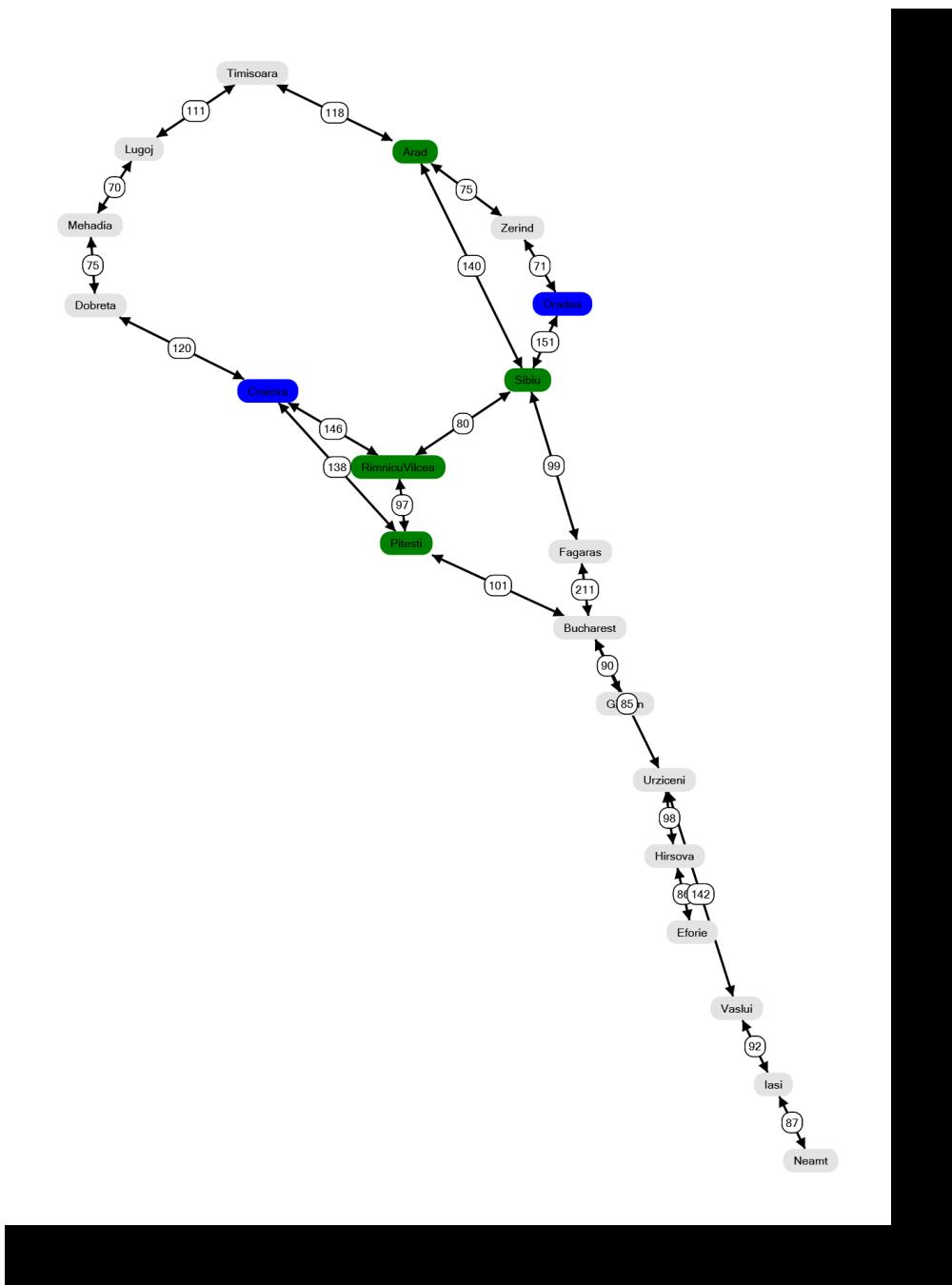
Visit

The algorithm has marked the node Craiova as visited



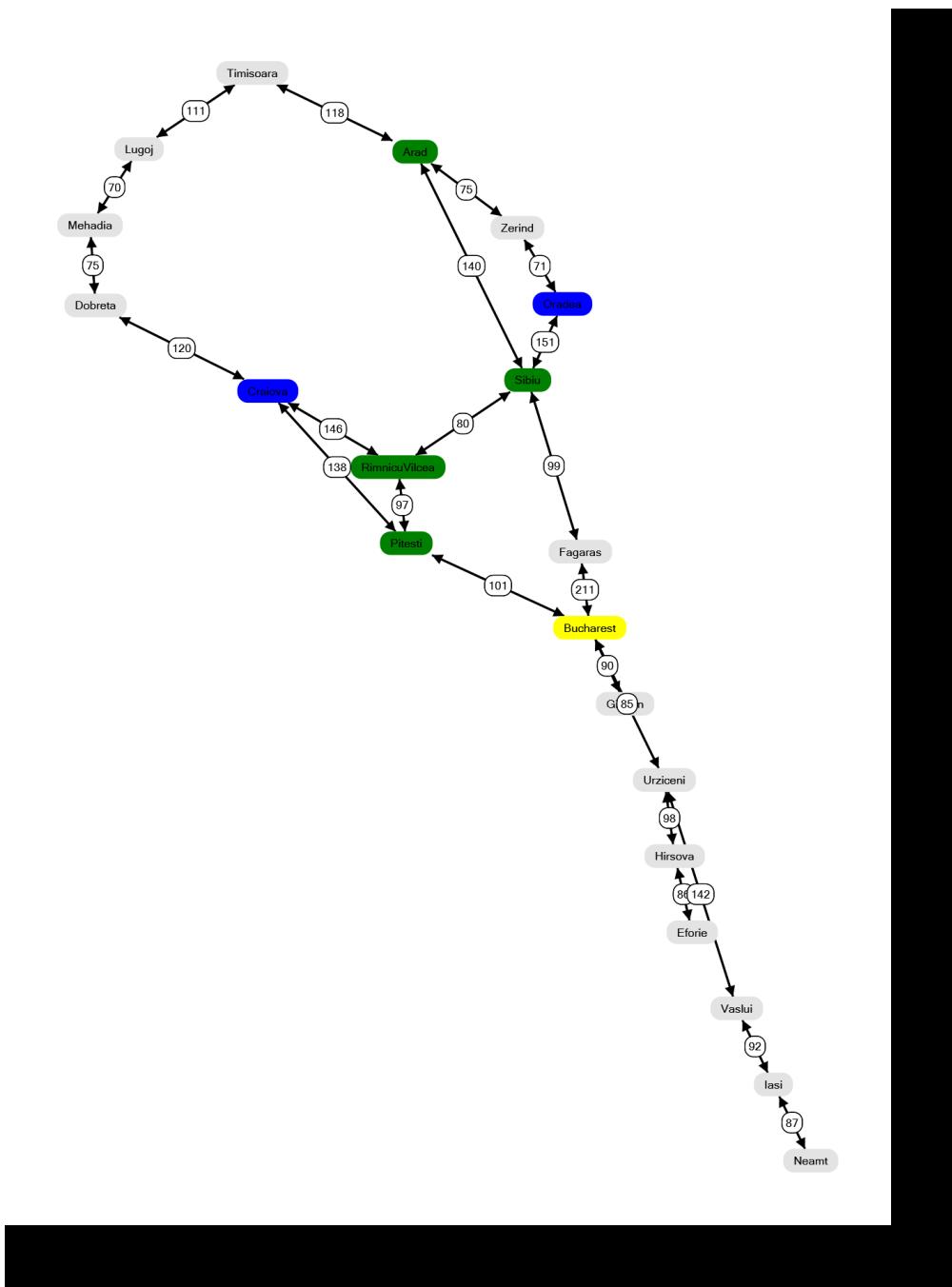
Post-Visit

The algorithm has closed the node Craiova



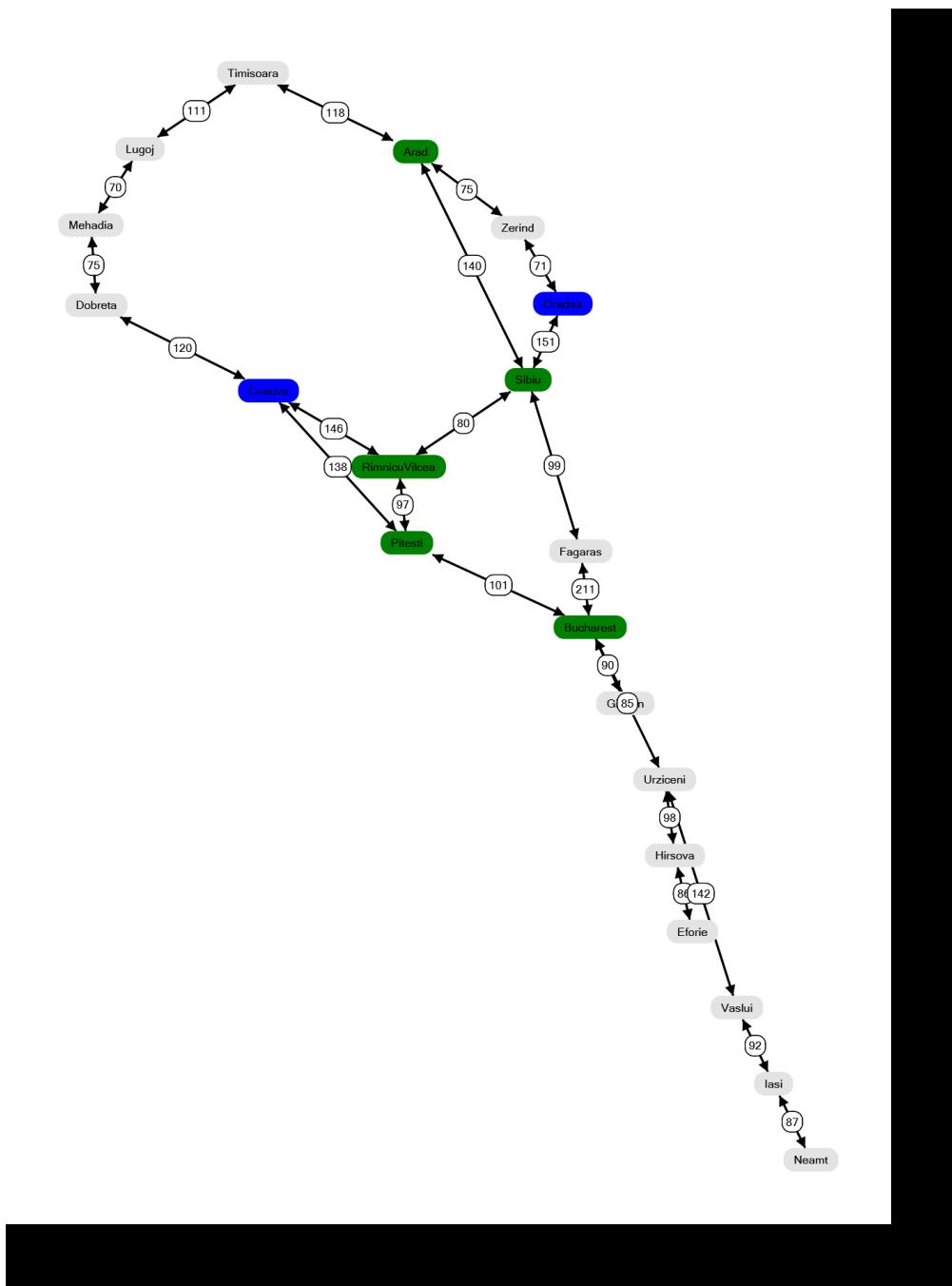
Pre-Visit

The algorithm has opened the node Bucharest



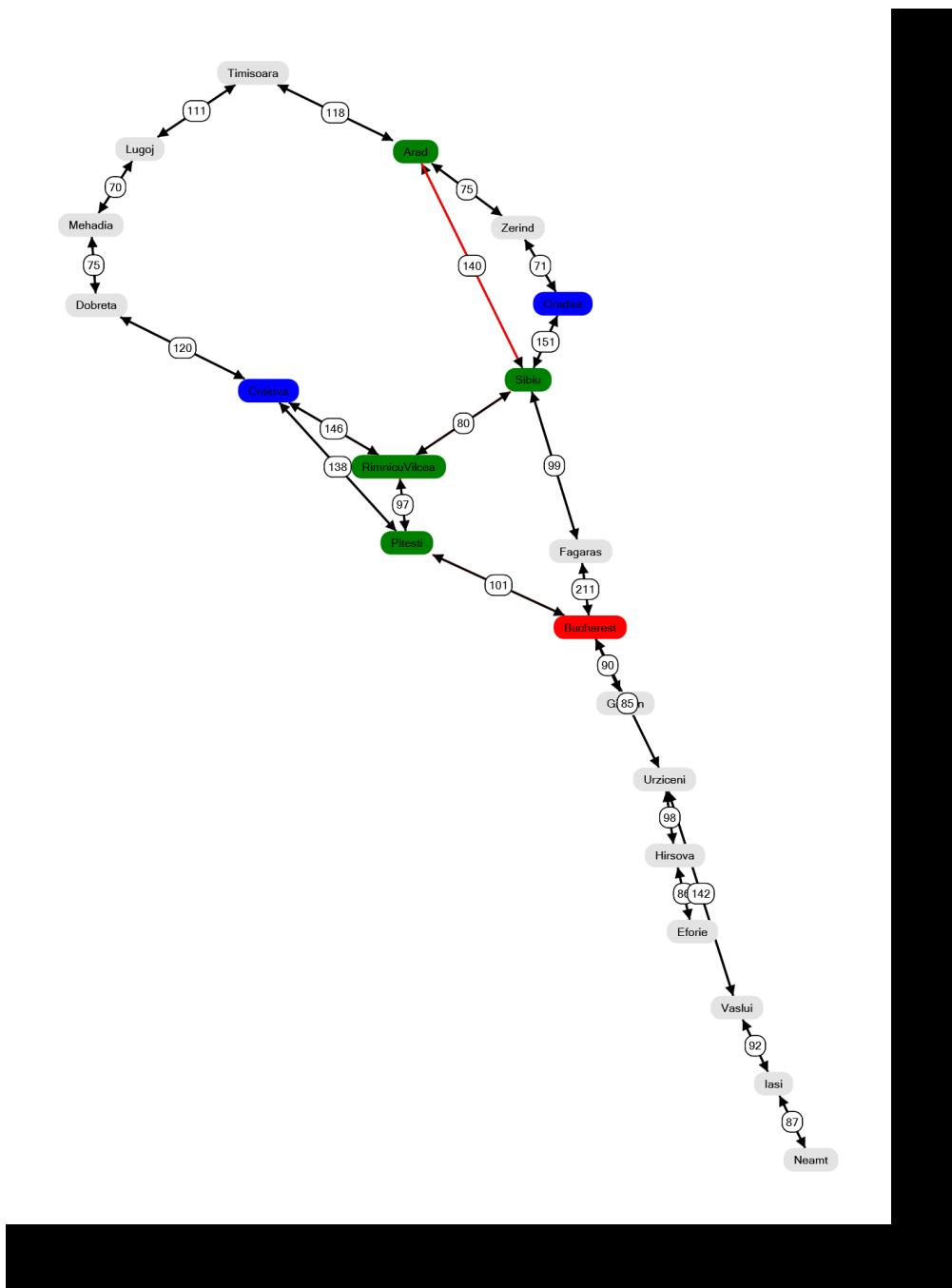
Visit

The algorithm has marked the node Bucharest as visited



Result Found

The algorithm has found the target node Bucharest



Reset

The algorithm is starting a new iteration with a different parameter (depth (IDLS) or Threshold (IDA*))

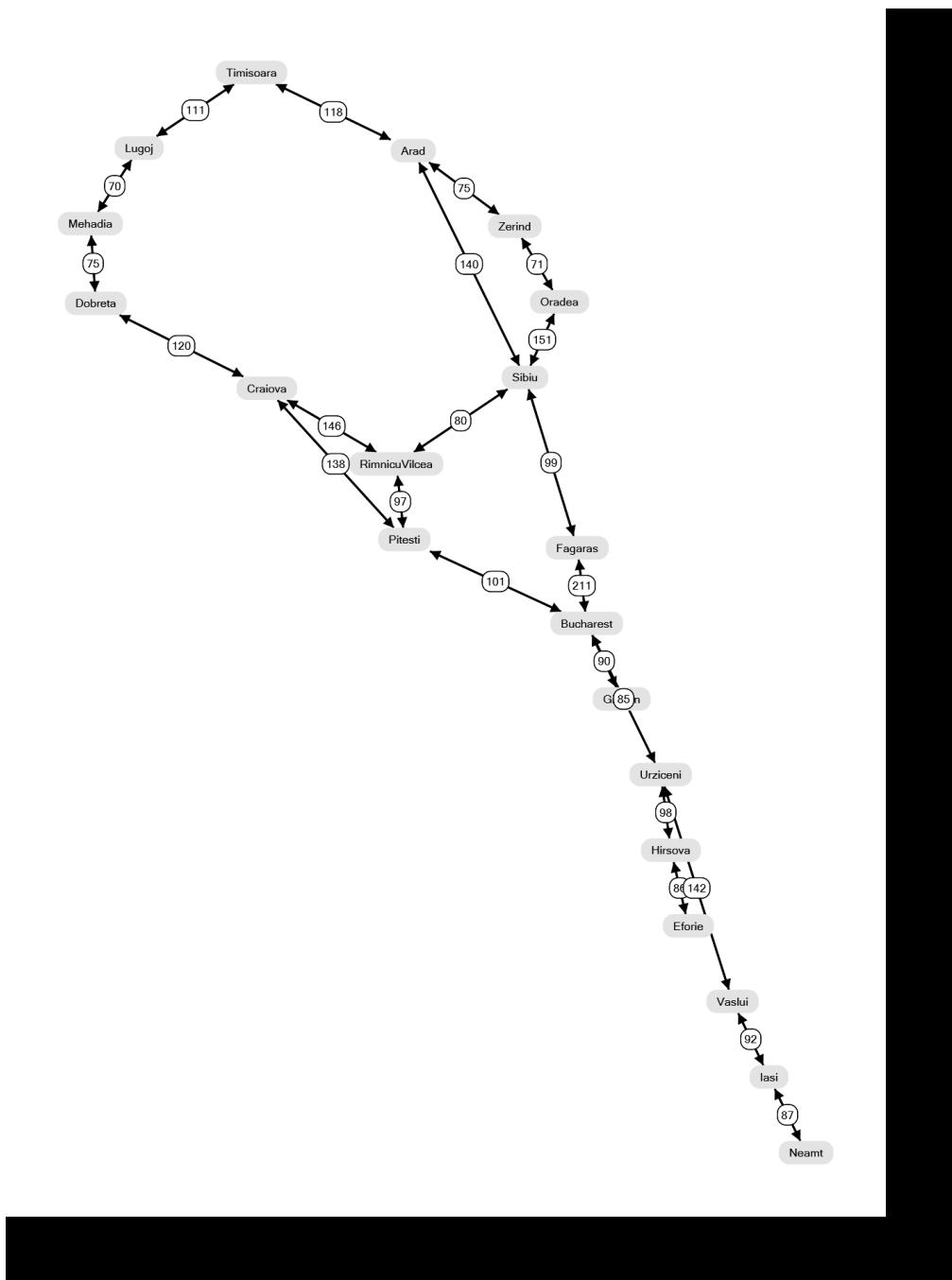


Table of Contents

Breadth-first search.....	4
Depth-first search.....	39
Uniform Cost Search.....	65
Iterative Depth limited search.....	106
A*.....	180
Greedy Best-first search.....	200
Iterative deepening A*.....	214