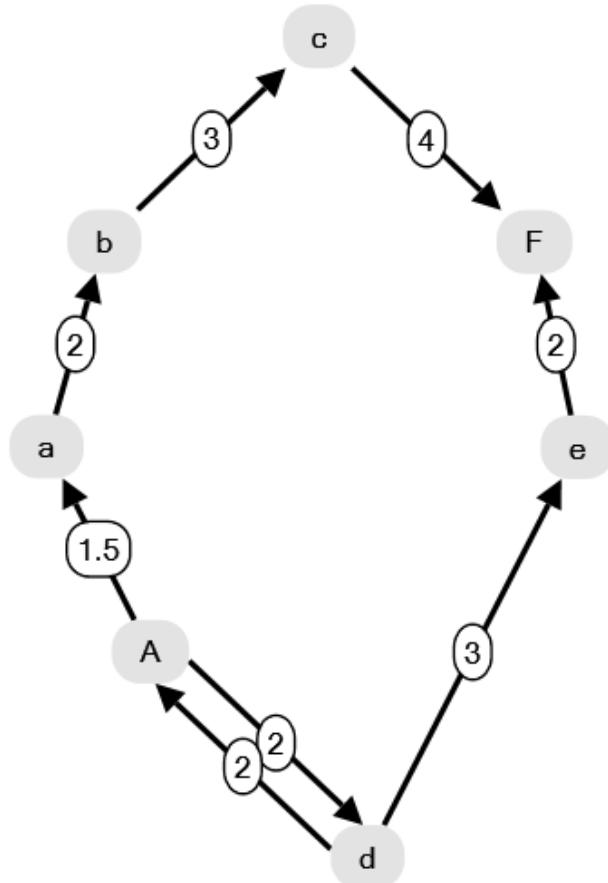


GraphSEA

Graph Search Algorithms

Benchmark Report

The Graph



We will perform a search for F starting from A 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	00:00:00.0007510
Depth-first search	00:00:00.0005072
Uniform Cost Search	00:00:00.0008861
Iterative Depth limited search	00:00:00.0007905
A*	00:00:00.0012005
Greedy Best-first search	00:00:00.0008568
Iterative deepening A*	00:00:00.0007336

Breadth-first search

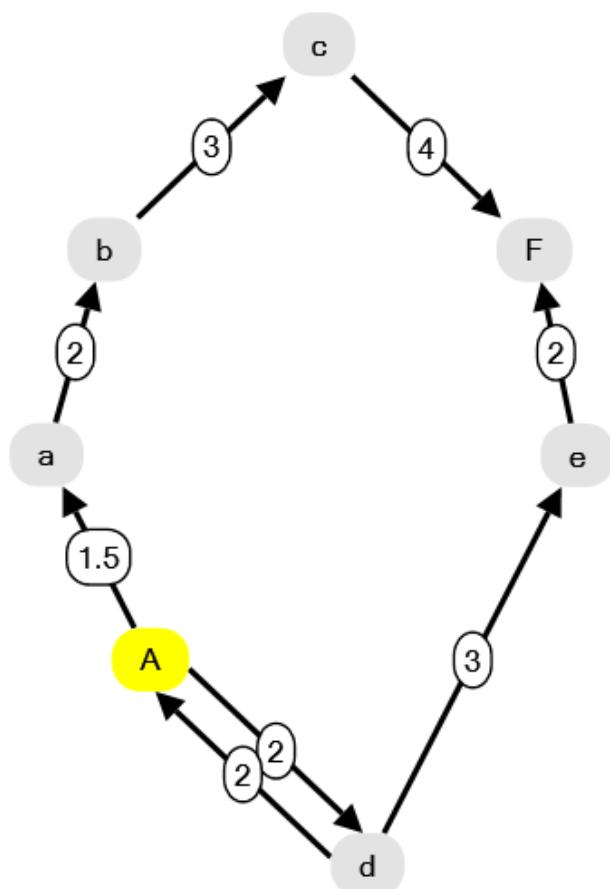
Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a 'search key') and explores the neighbor nodes first, before moving to the next level neighbours.

The algorithm took 00:00:00.0007510 to find the goal.

The result path costs 7

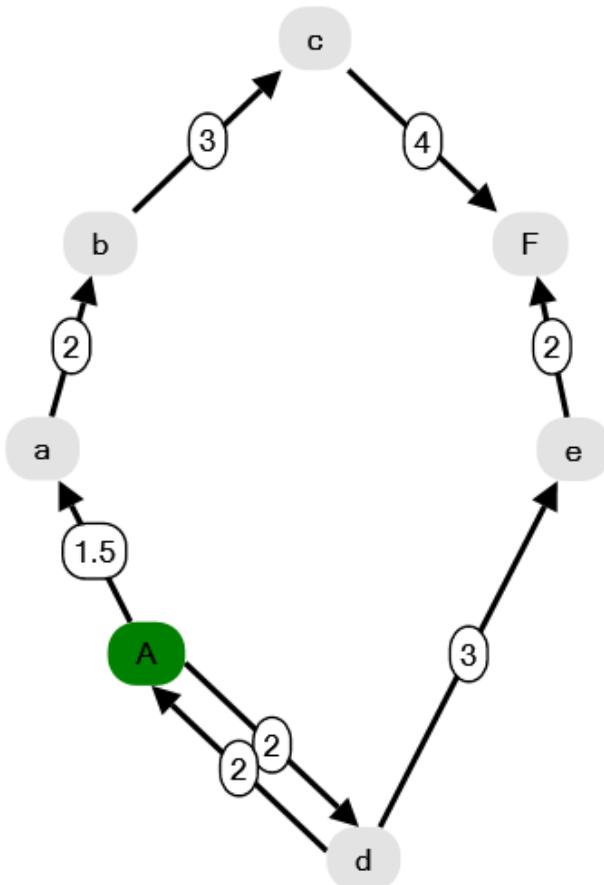
Pre-Visit

The algorithm has opened the node A



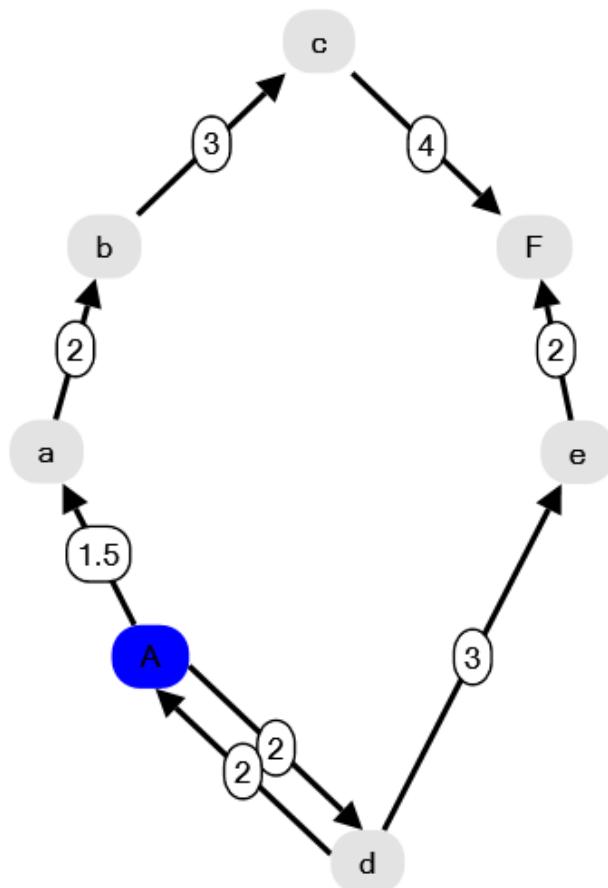
Visit

The algorithm has marked the node A as visited



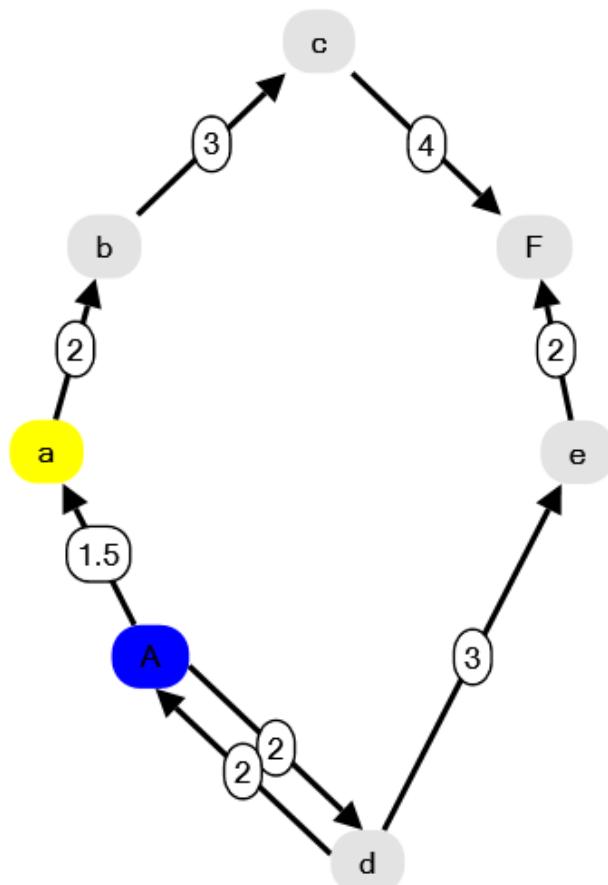
Post-Visit

The algorithm has closed the node A



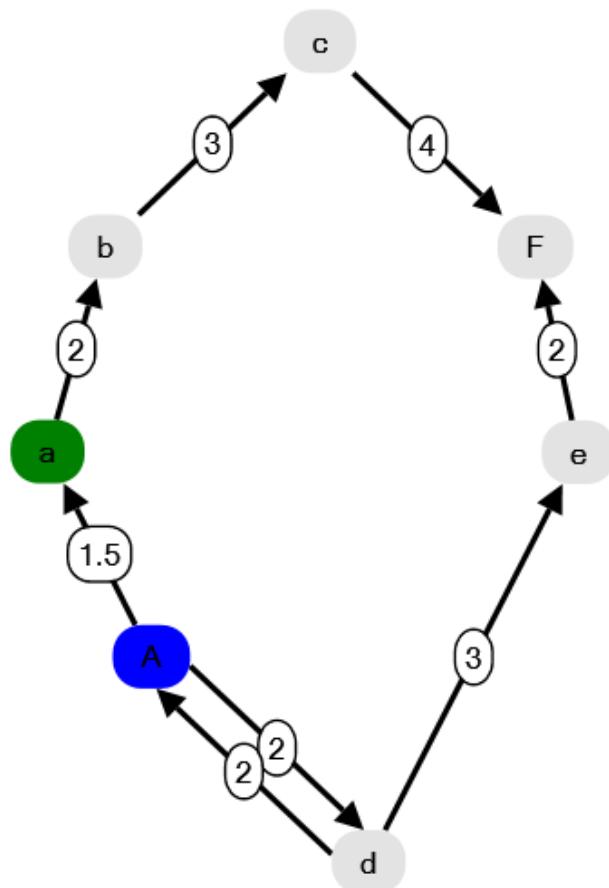
Pre-Visit

The algorithm has opened the node *a*



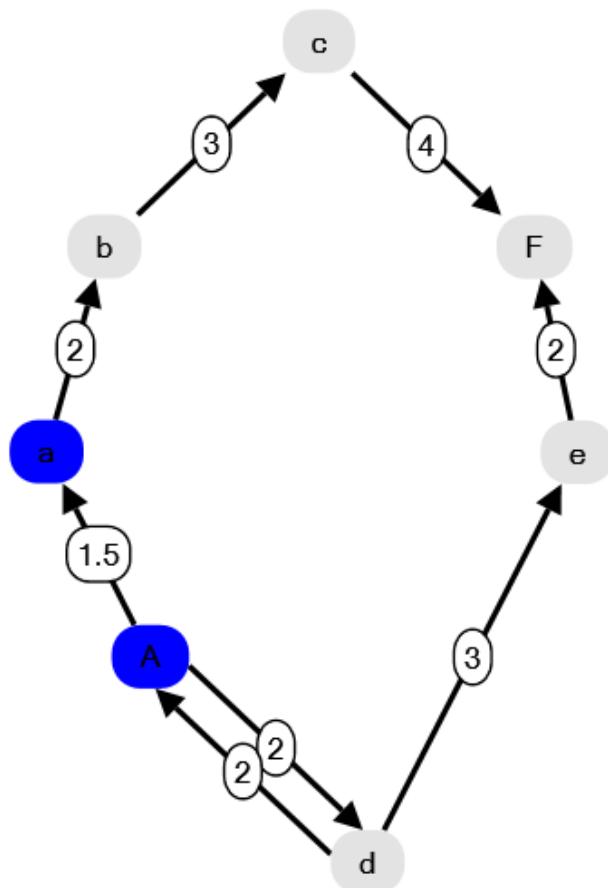
Visit

The algorithm has marked the node *a* as visited



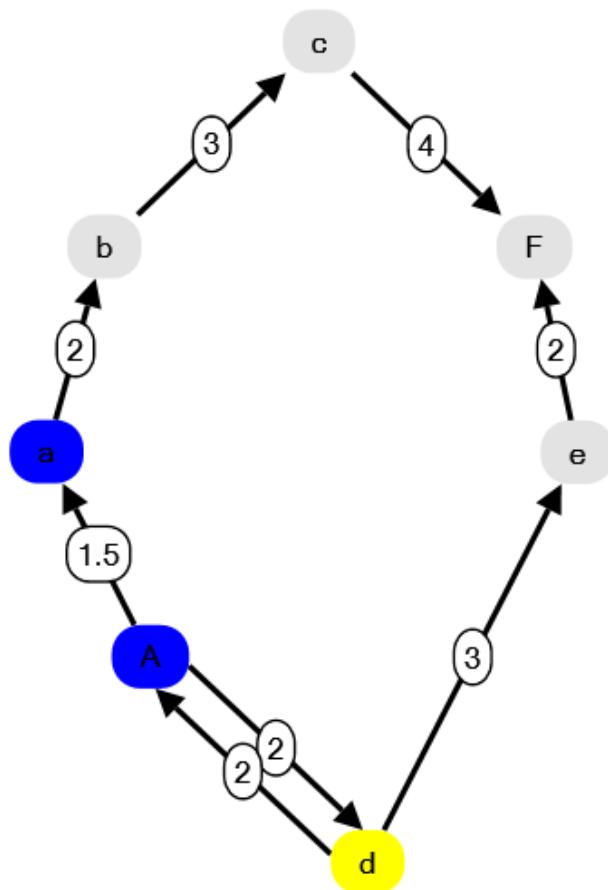
Post-Visit

The algorithm has closed the node *a*



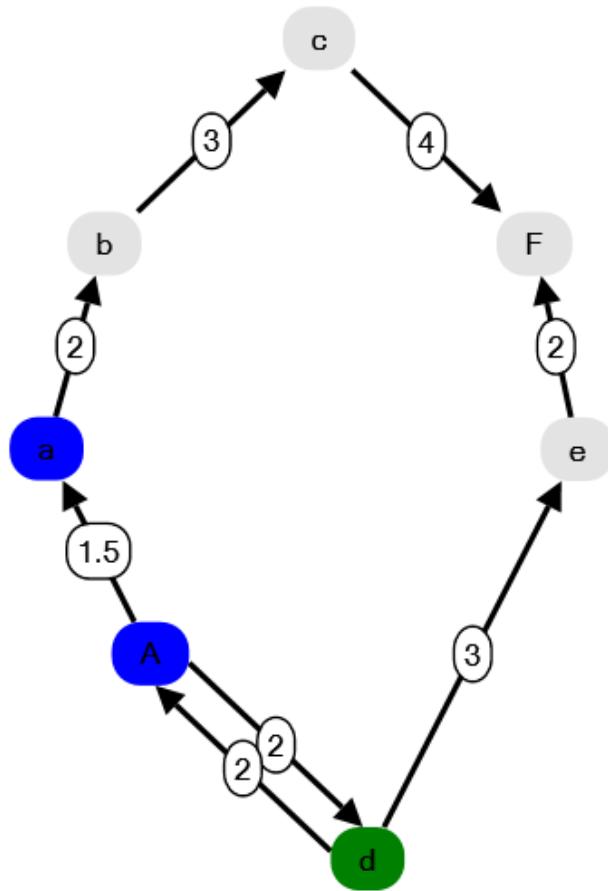
Pre-Visit

The algorithm has opened the node *d*



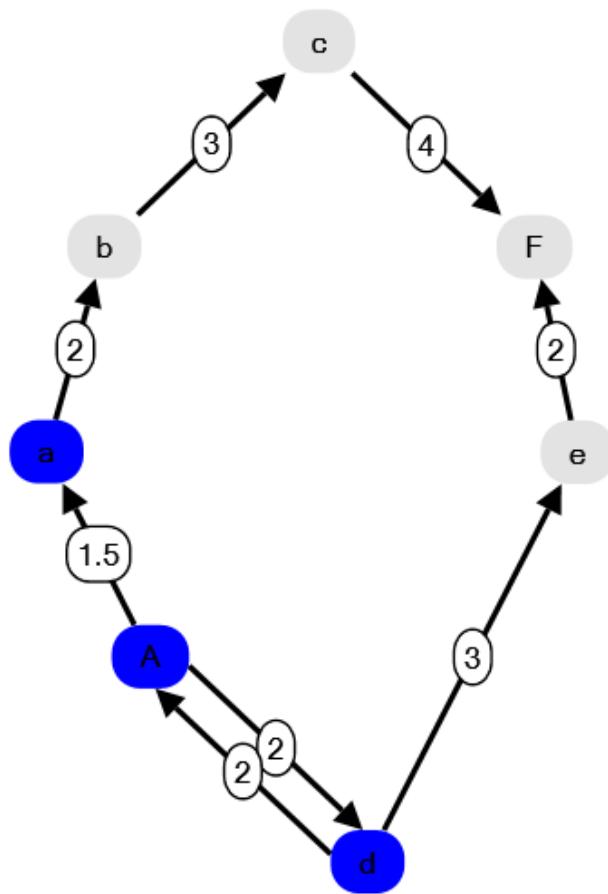
Visit

The algorithm has marked the node d as visited



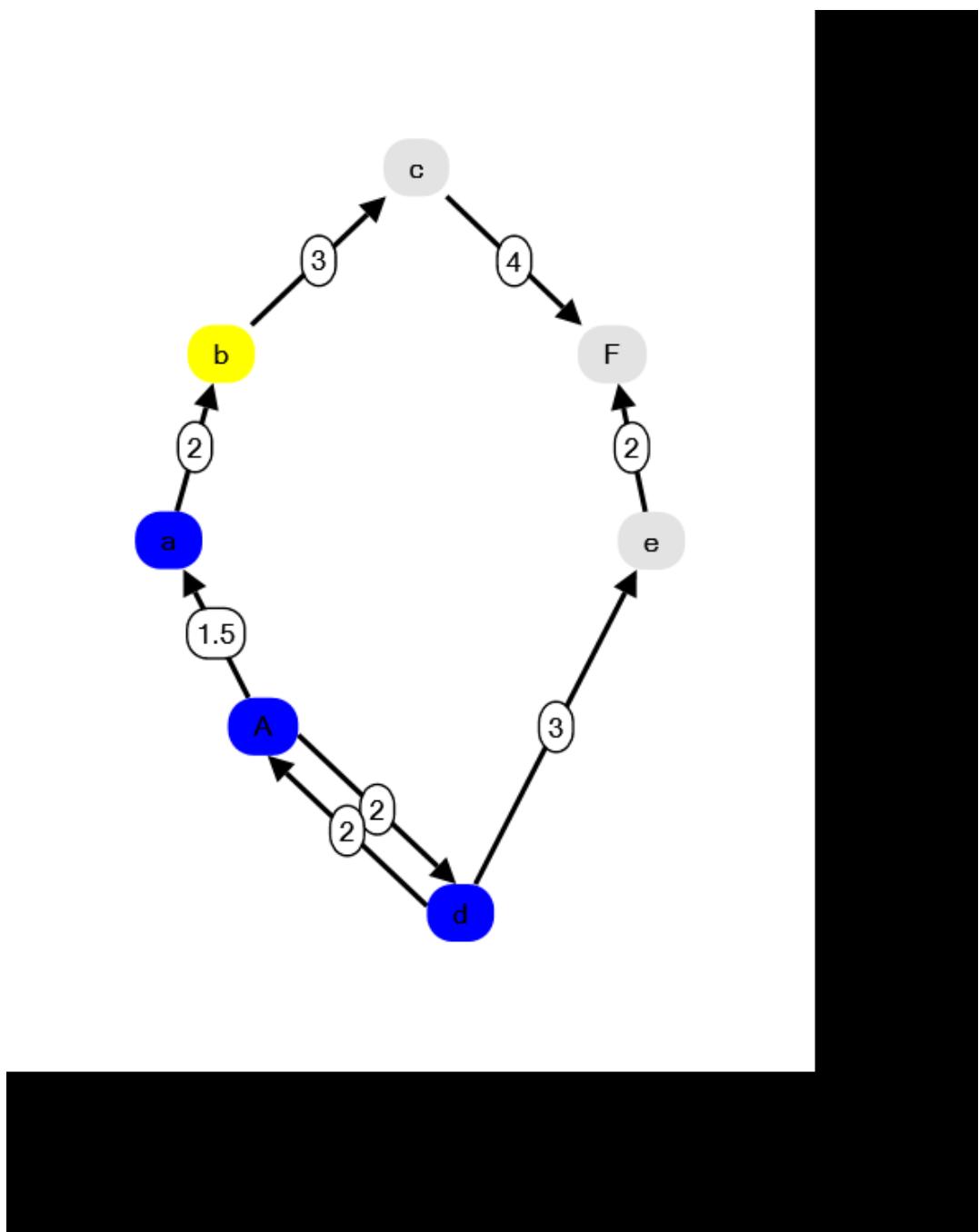
Post-Visit

The algorithm has closed the node d



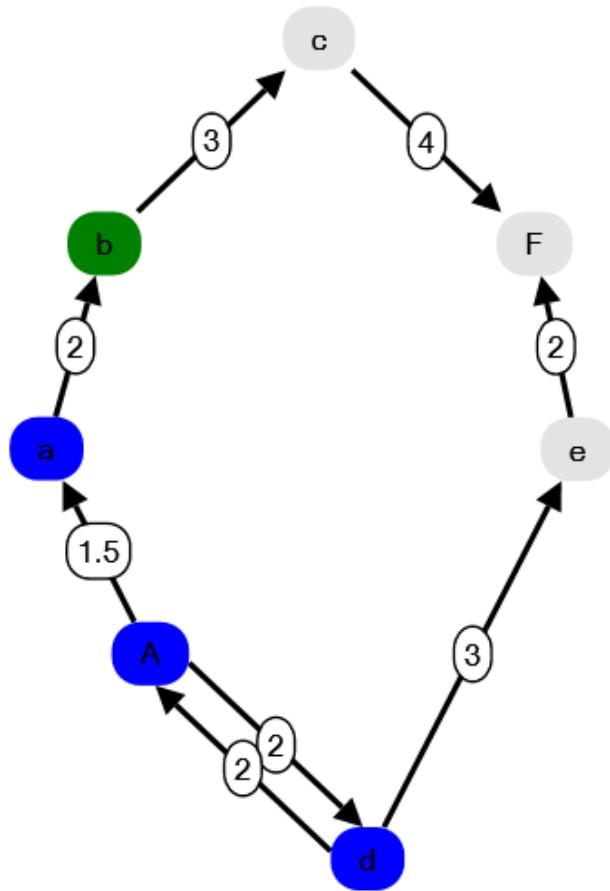
Pre-Visit

The algorithm has opened the node *b*



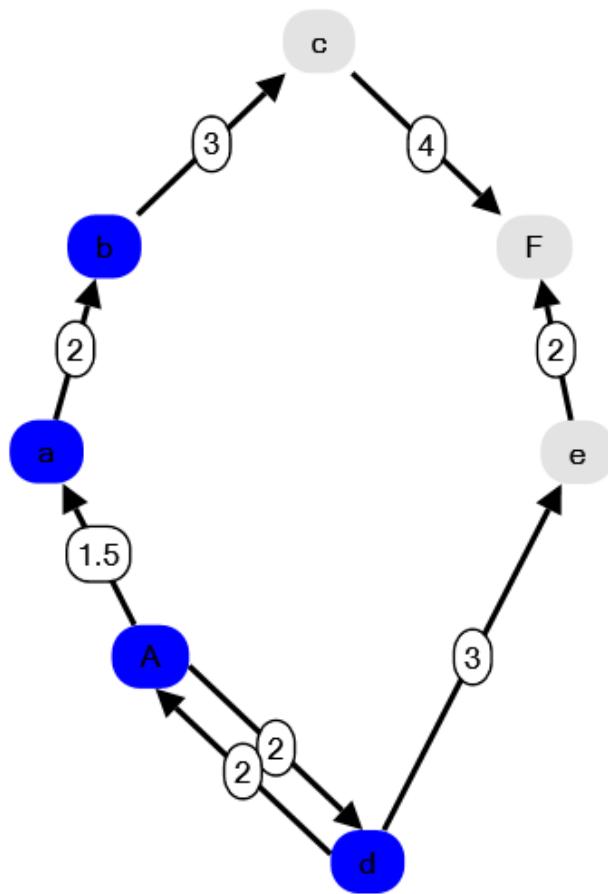
Visit

The algorithm has marked the node *b* as visited



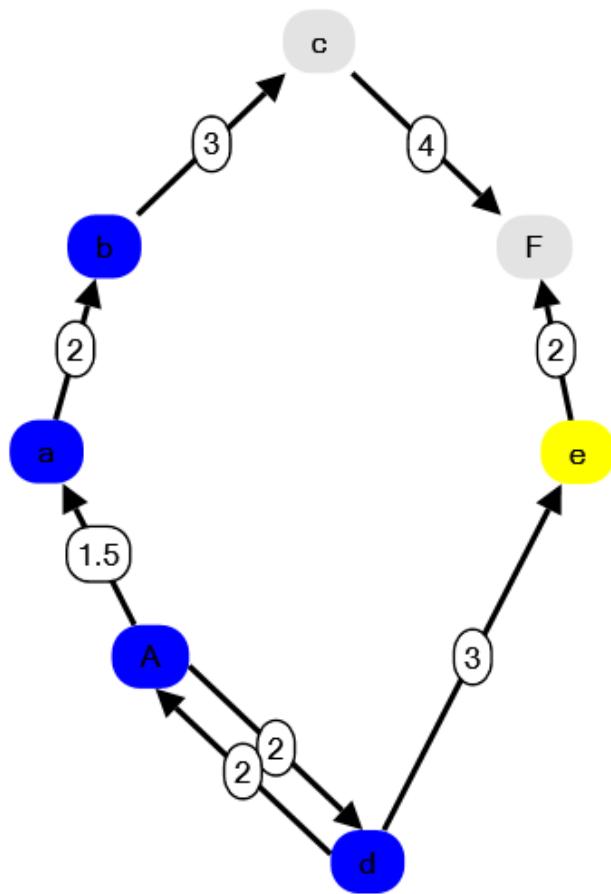
Post-Visit

The algorithm has closed the node b



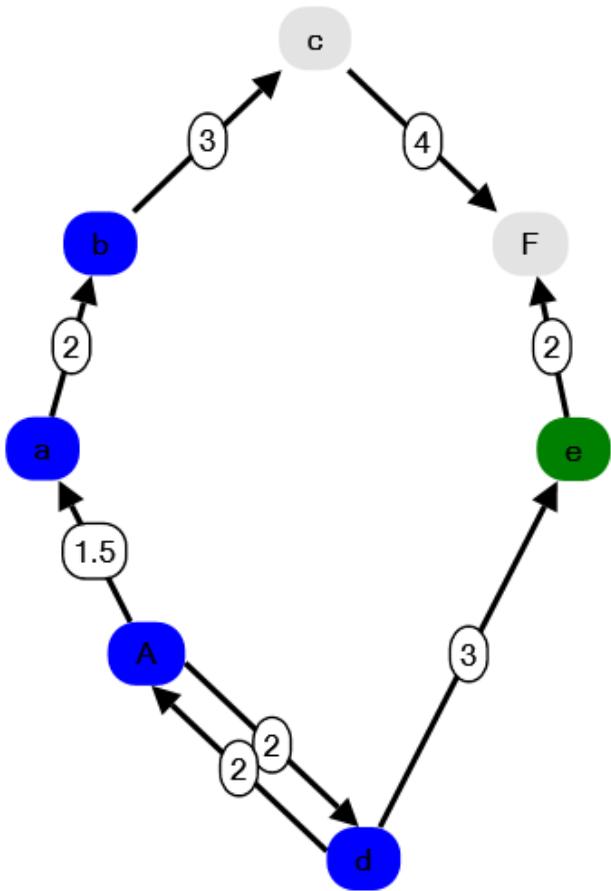
Pre-Visit

The algorithm has opened the node *e*



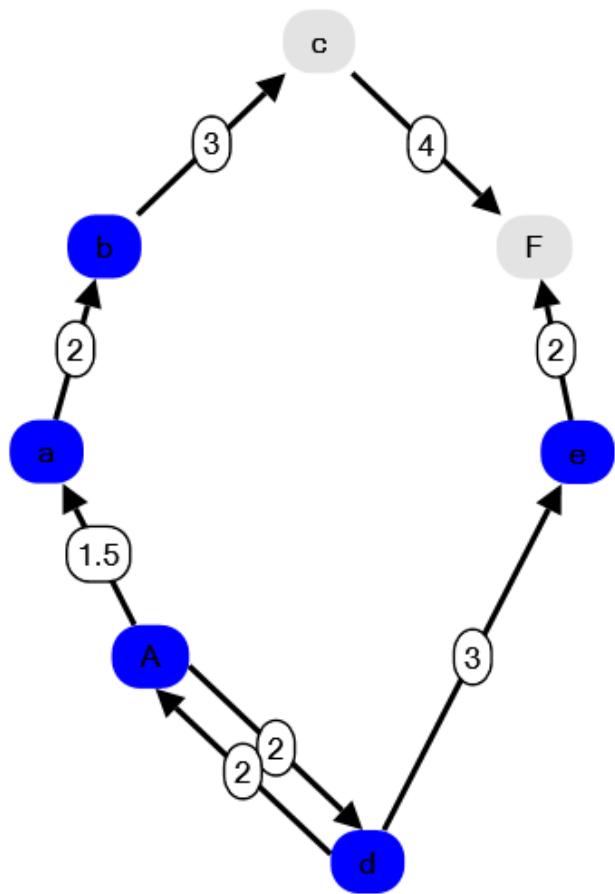
Visit

The algorithm has marked the node e as visited



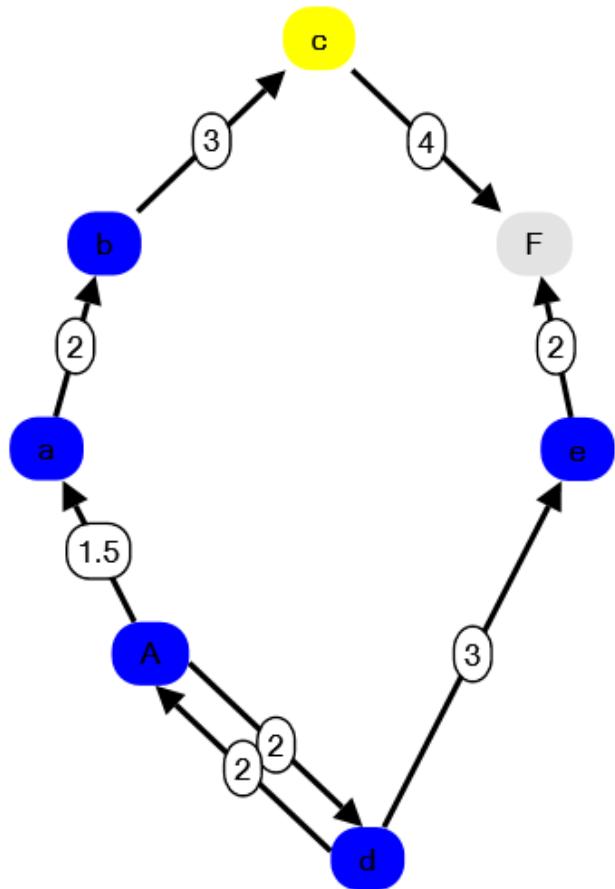
Post-Visit

The algorithm has closed the node e



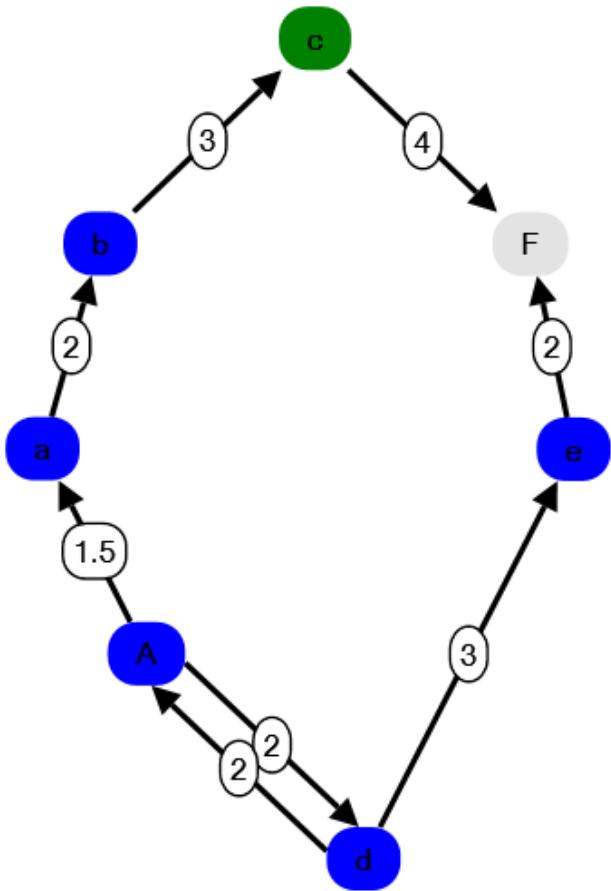
Pre-Visit

The algorithm has opened the node *c*



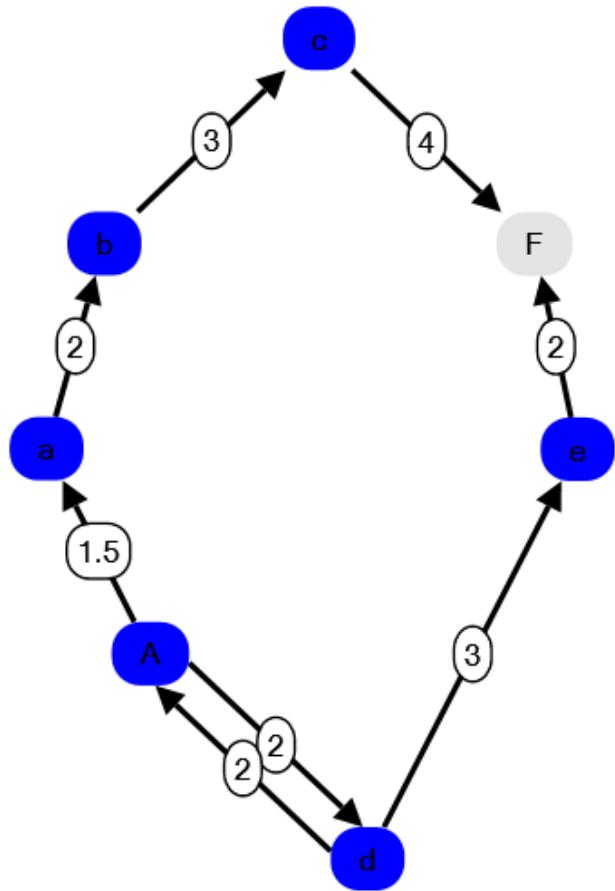
Visit

The algorithm has marked the node c as visited



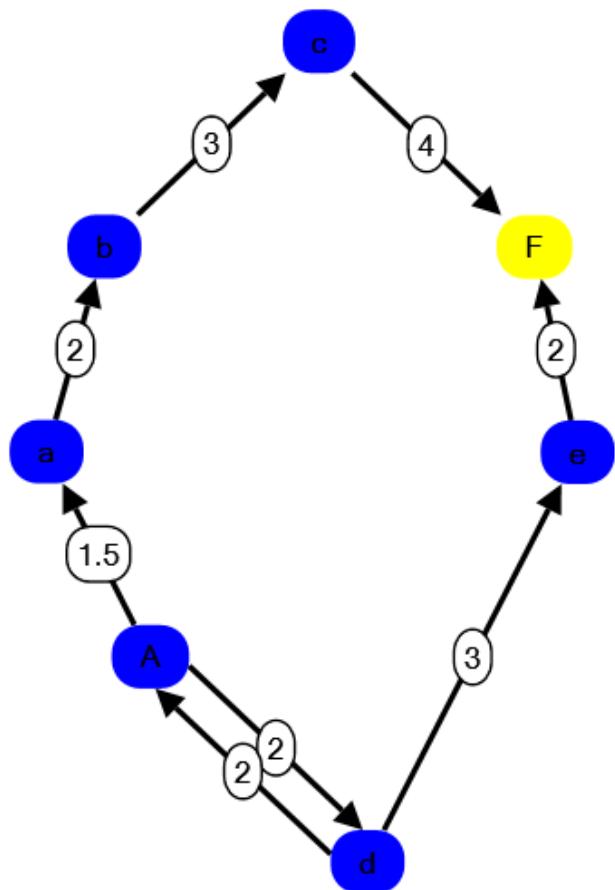
Post-Visit

The algorithm has closed the node c



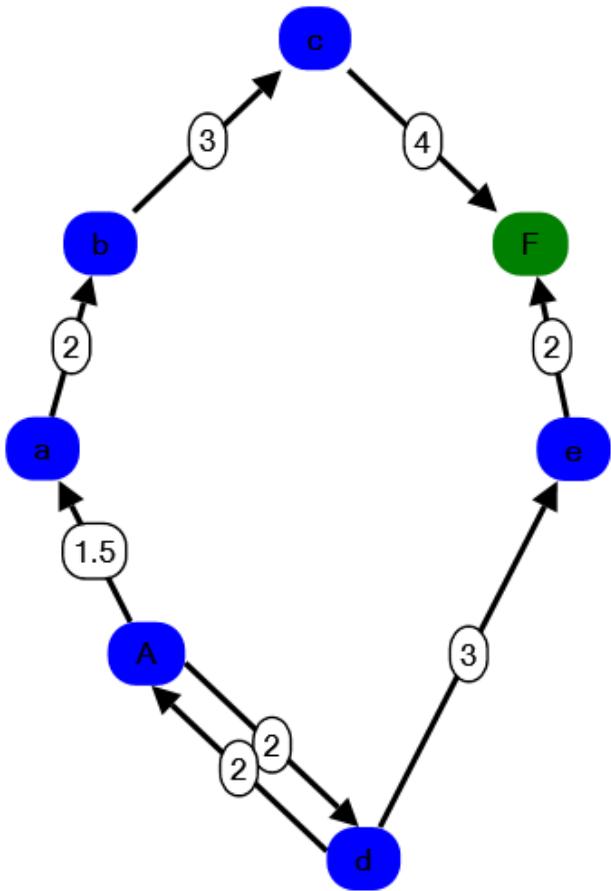
Pre-Visit

The algorithm has opened the node F



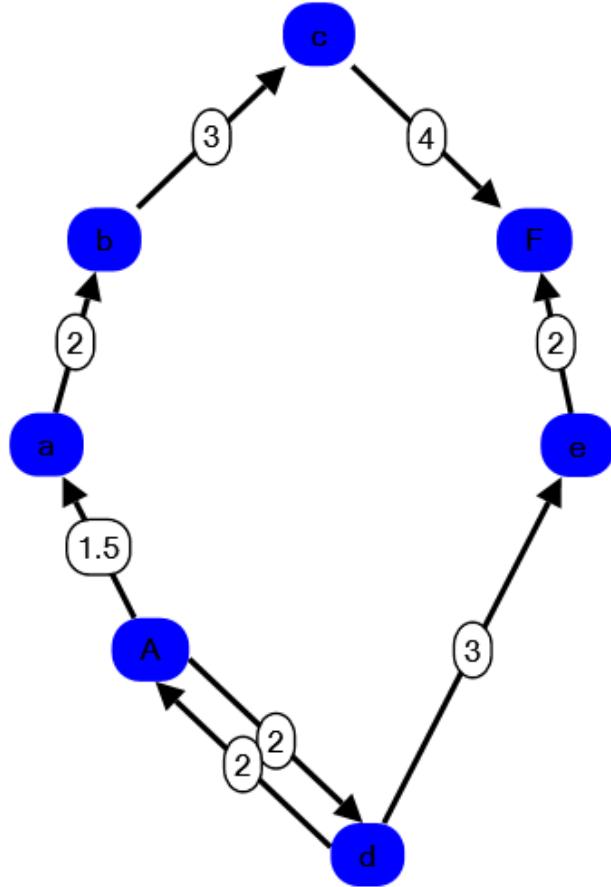
Visit

The algorithm has marked the node F as visited



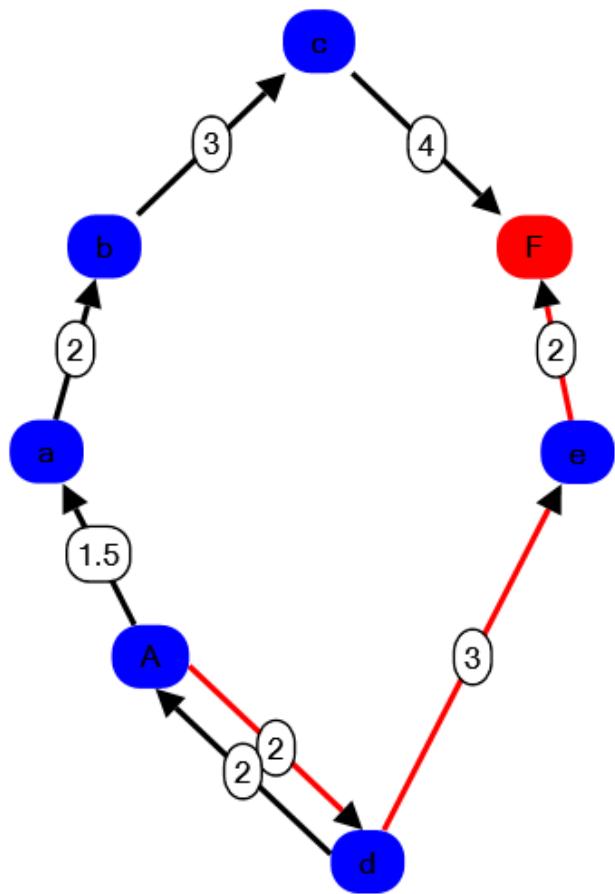
Post-Visit

The algorithm has closed the node F



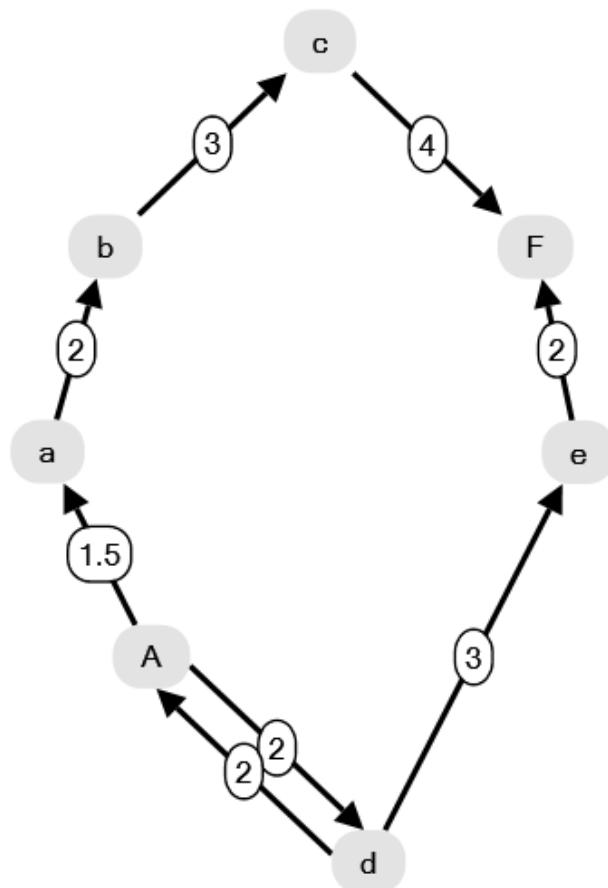
Result Found

The algorithm has found the target node F



Reset

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



Depth-first search

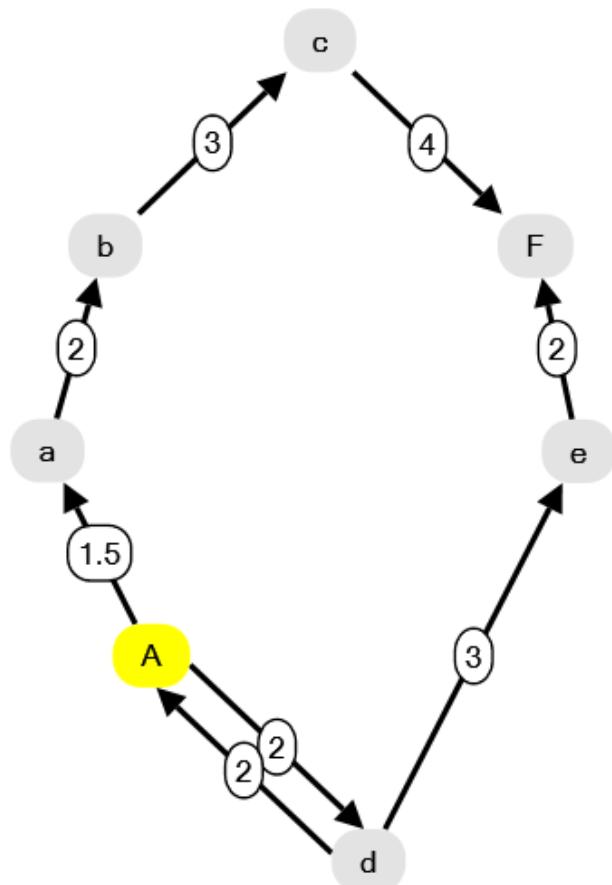
Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. One starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking.

The algorithm took 00:00:00.0005072 to find the goal.

The result path costs 10.5

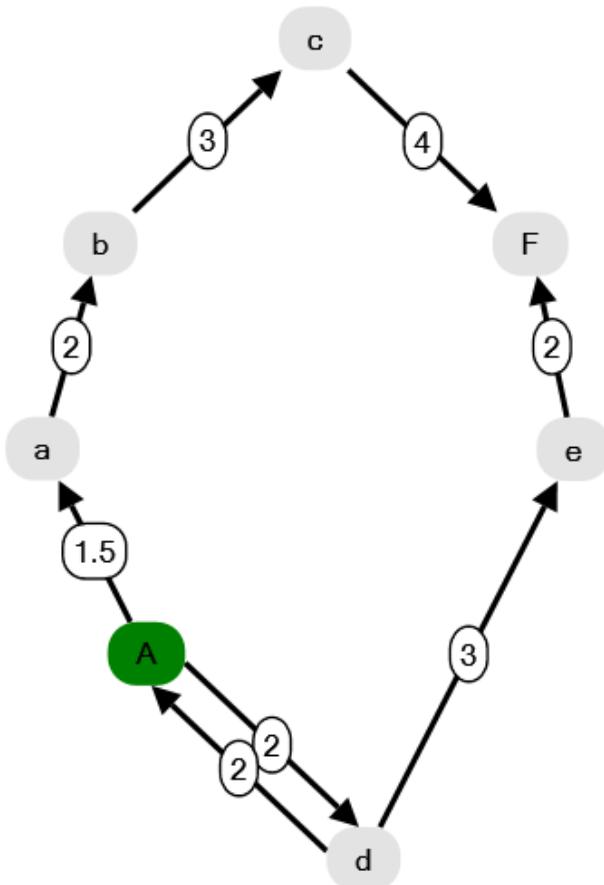
Pre-Visit

The algorithm has opened the node A



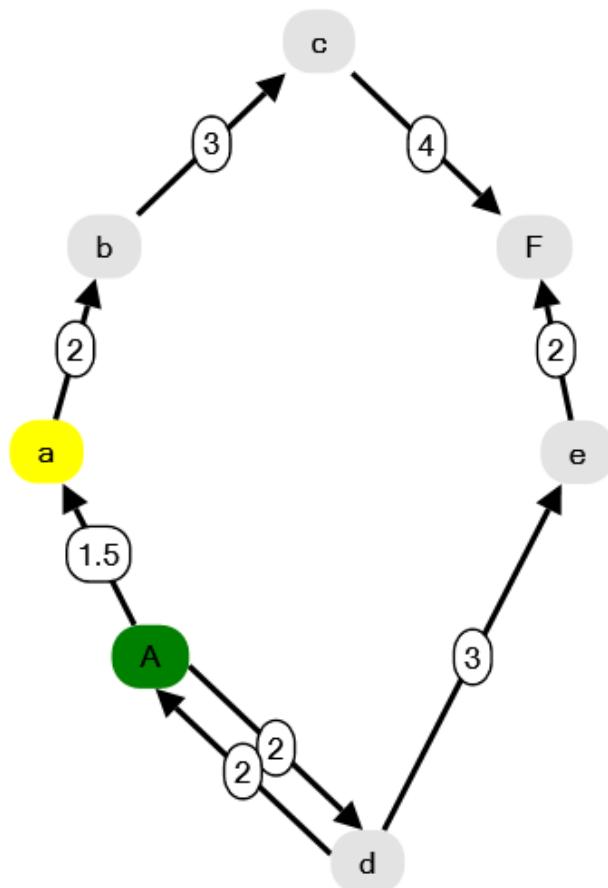
Visit

The algorithm has marked the node A as visited



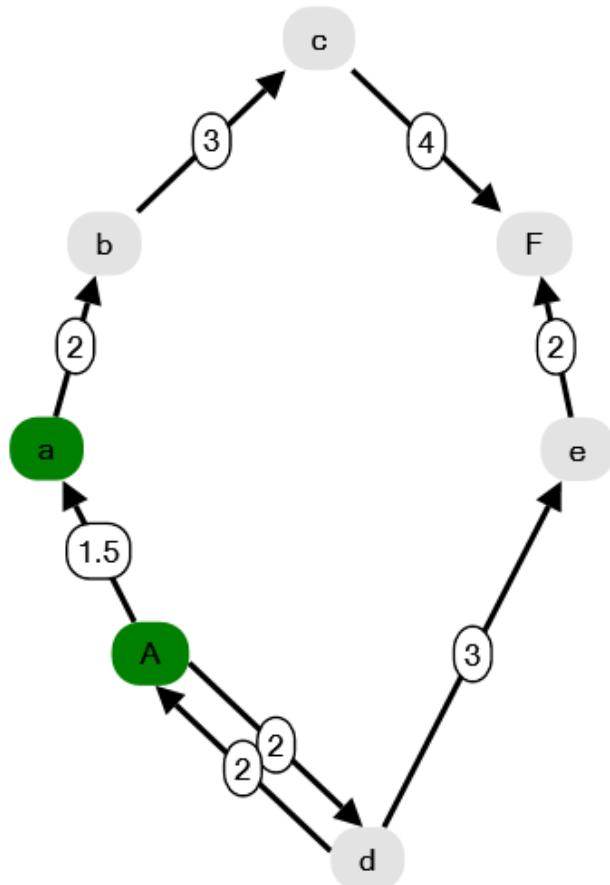
Pre-Visit

The algorithm has opened the node *a*



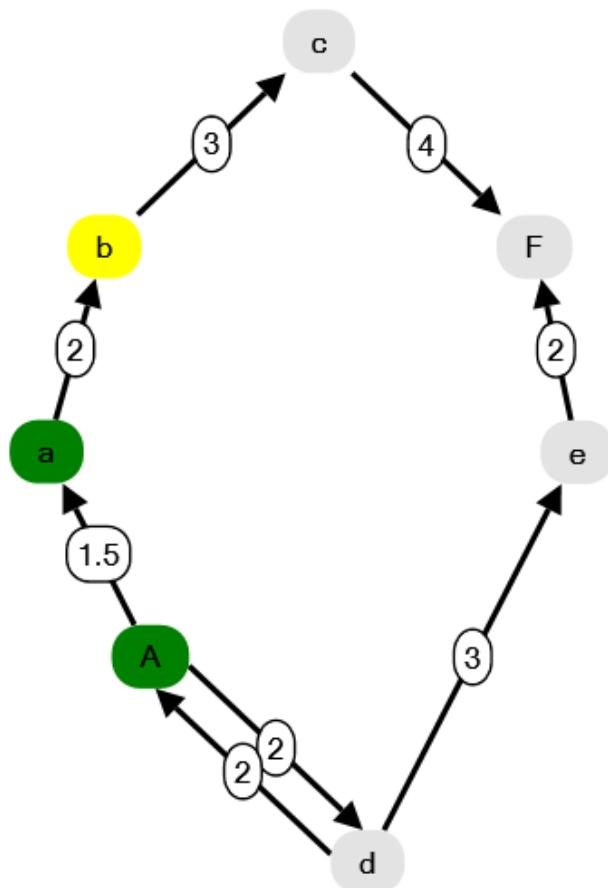
Visit

The algorithm has marked the node *a* as visited



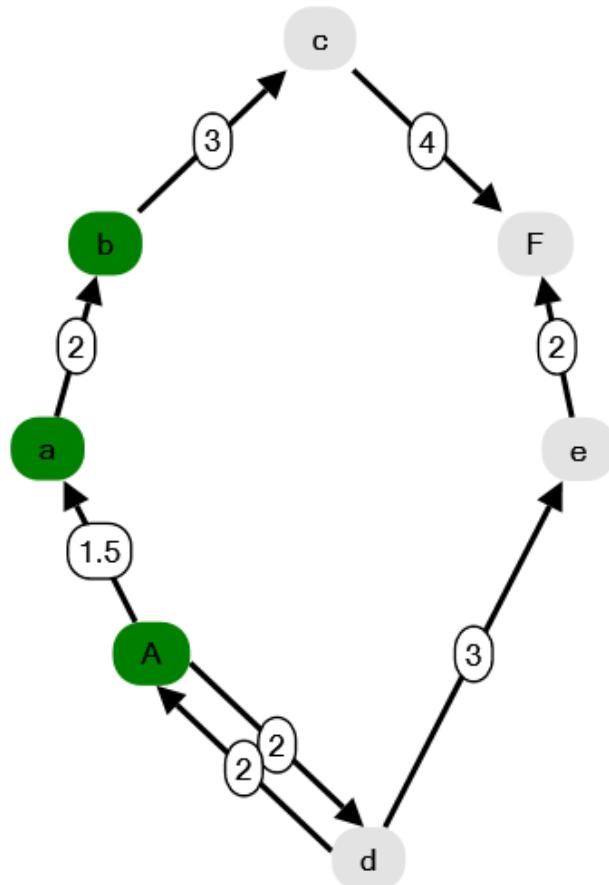
Pre-Visit

The algorithm has opened the node *b*



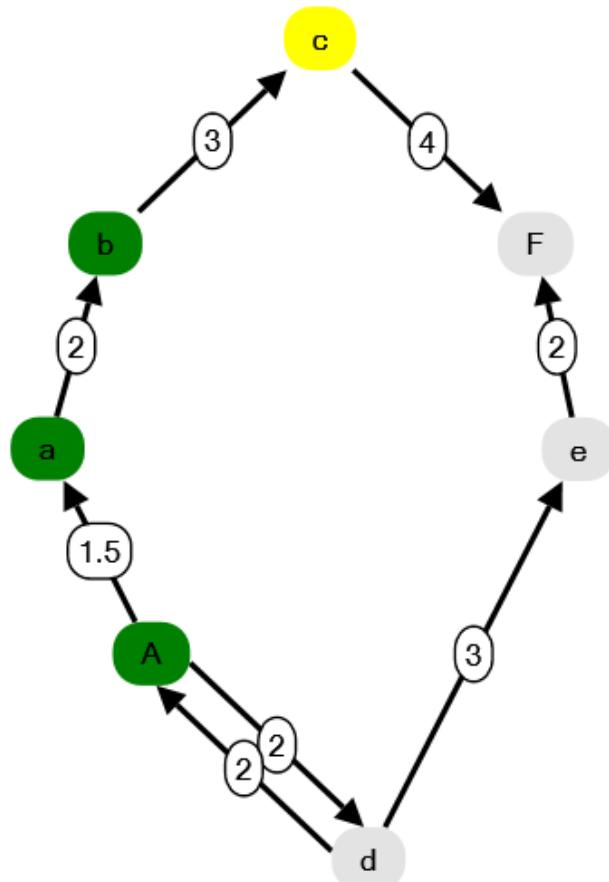
Visit

The algorithm has marked the node *b* as visited



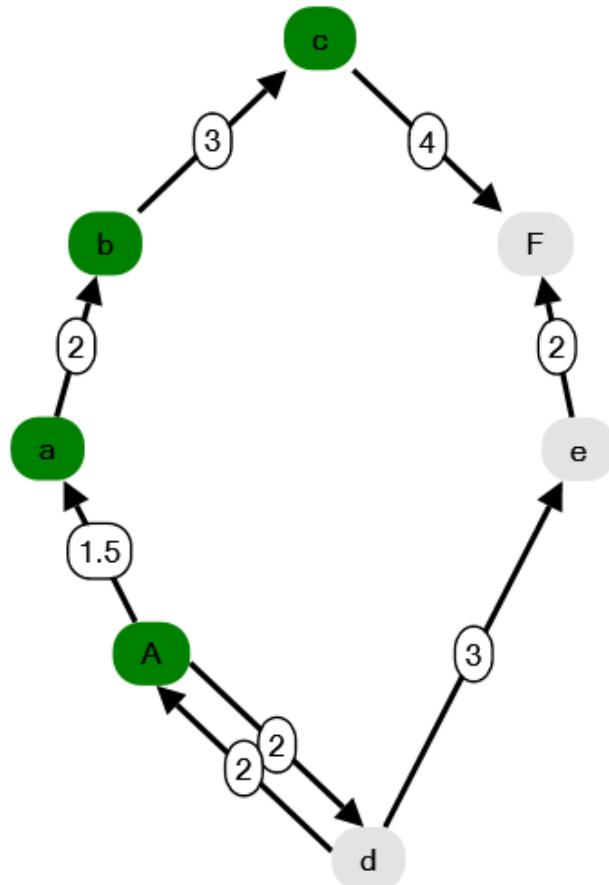
Pre-Visit

The algorithm has opened the node *c*



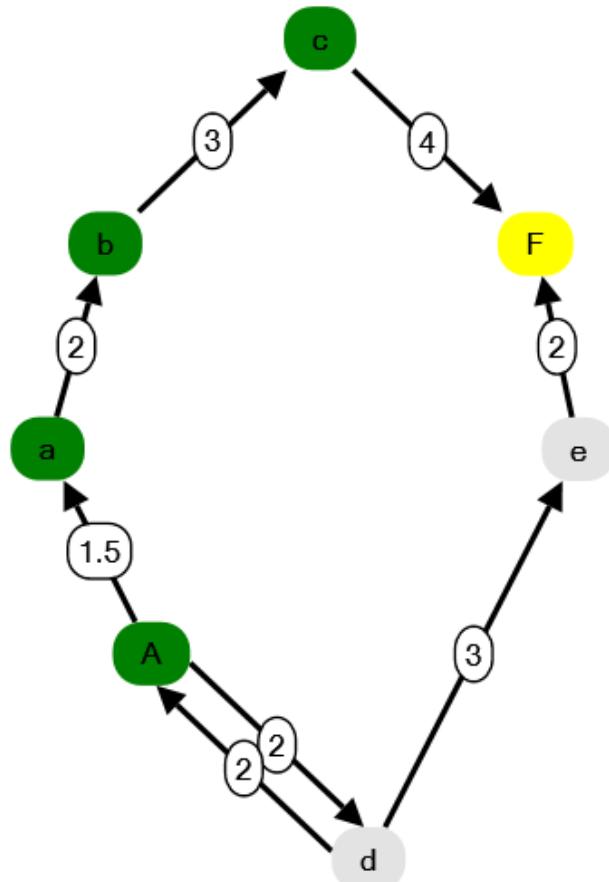
Visit

The algorithm has marked the node c as visited



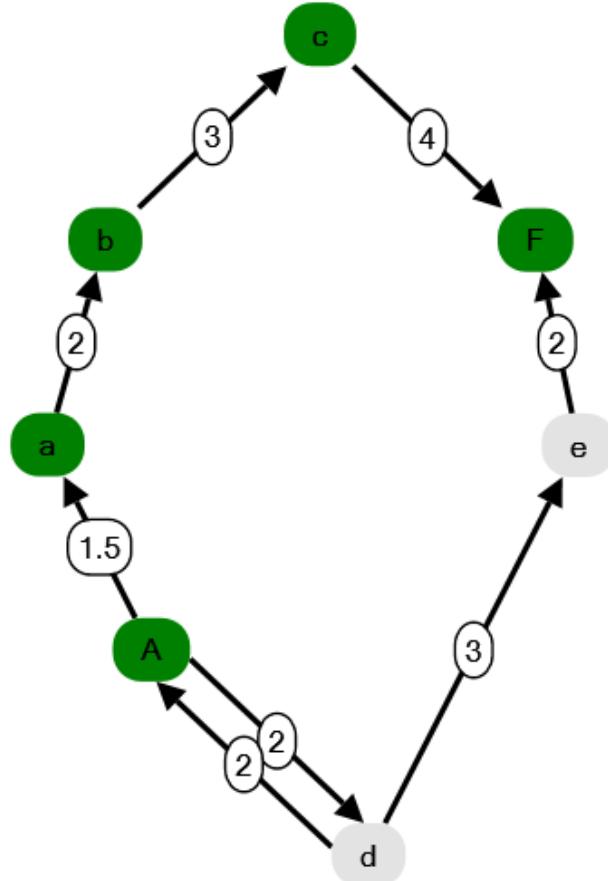
Pre-Visit

The algorithm has opened the node F



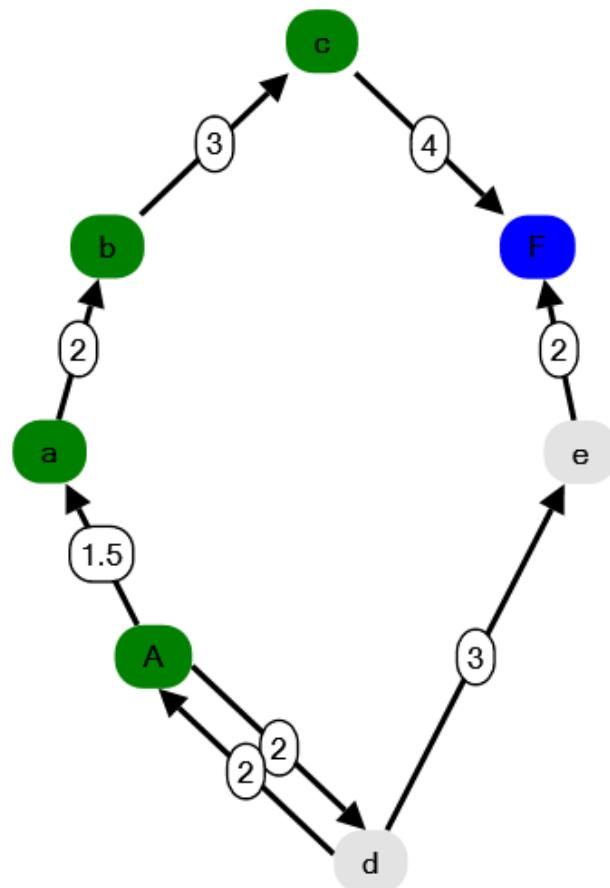
Visit

The algorithm has marked the node F as visited



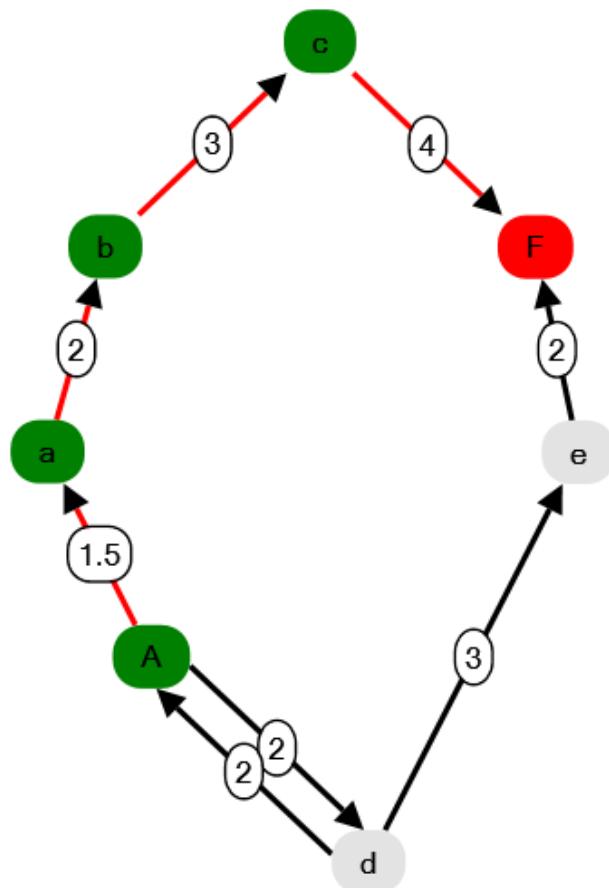
Post-Visit

The algorithm has closed the node F



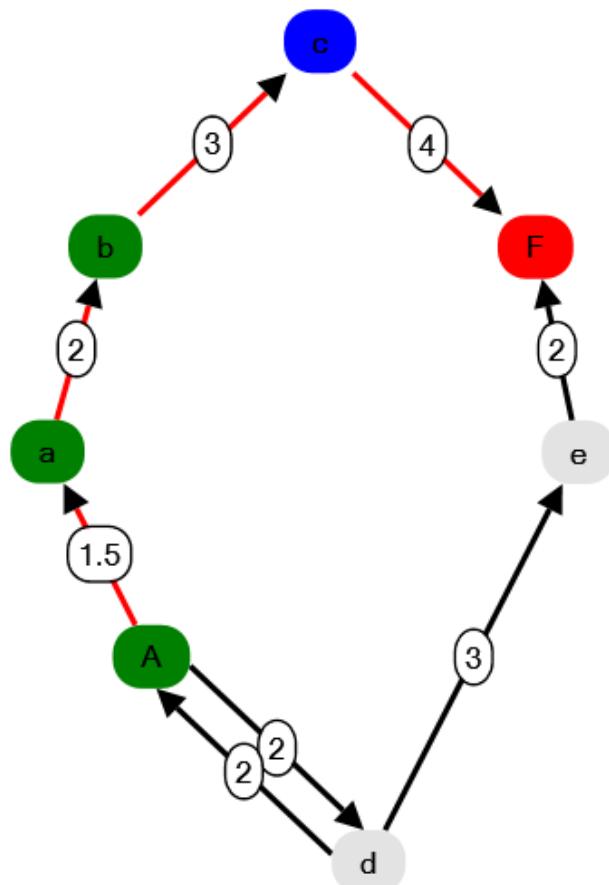
Result Found

The algorithm has found the target node F



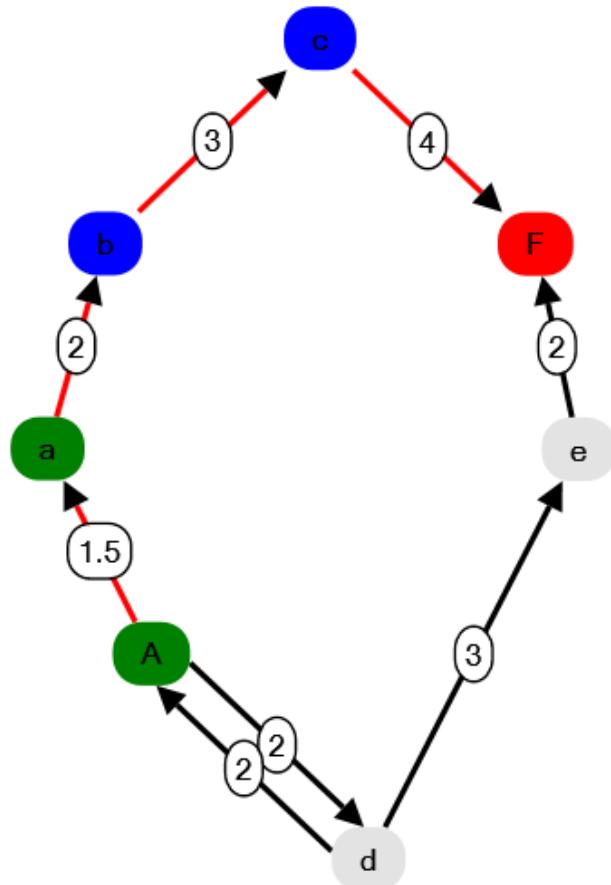
Post-Visit

The algorithm has closed the node c



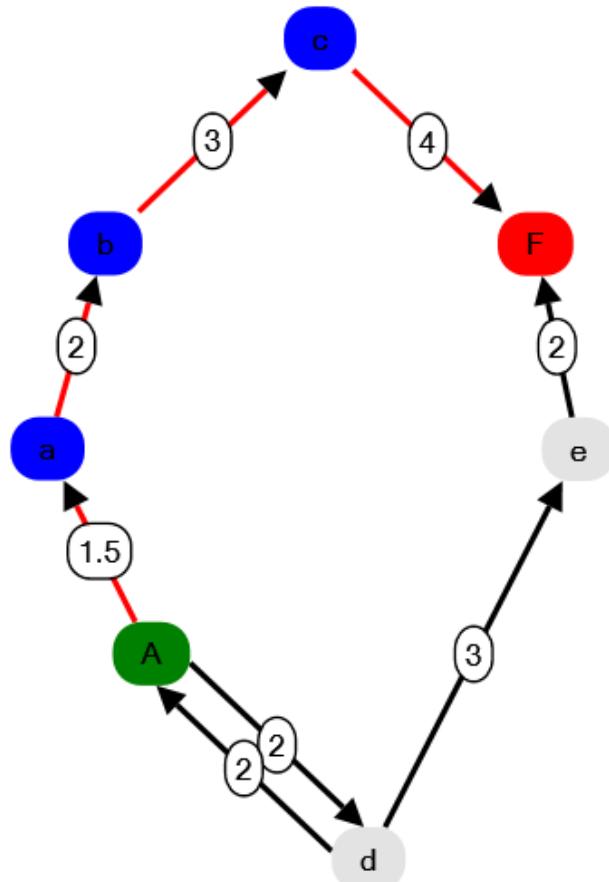
Post-Visit

The algorithm has closed the node b



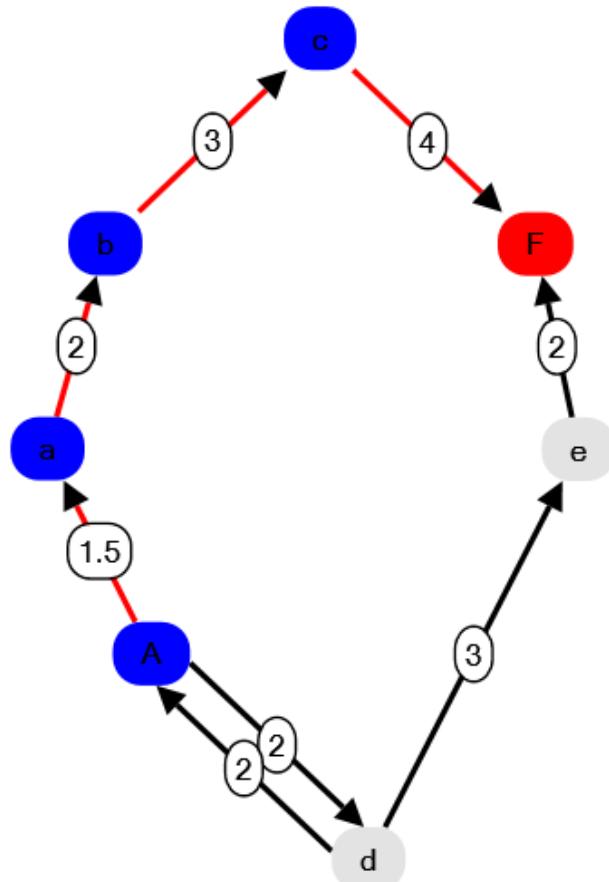
Post-Visit

The algorithm has closed the node *a*



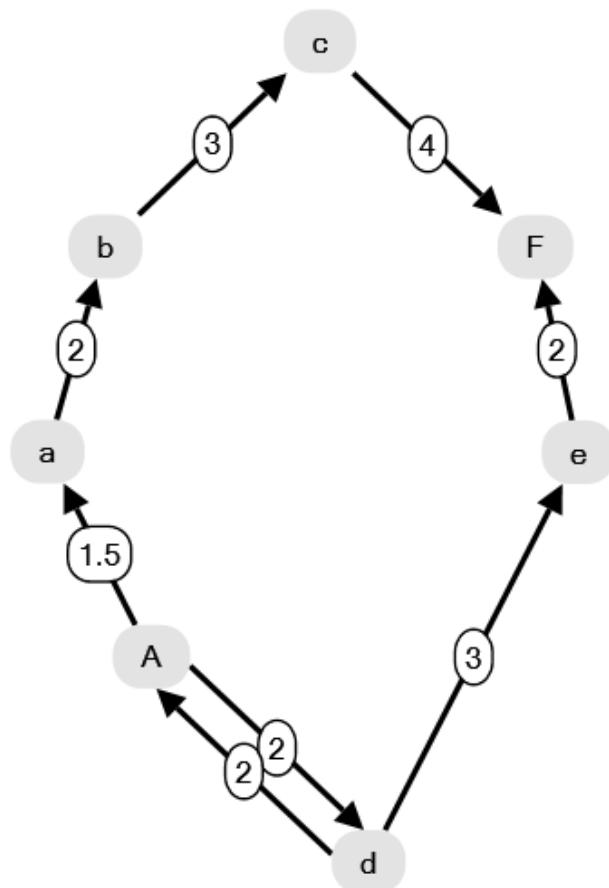
Post-Visit

The algorithm has closed the node A



Reset

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



Uniform Cost Search

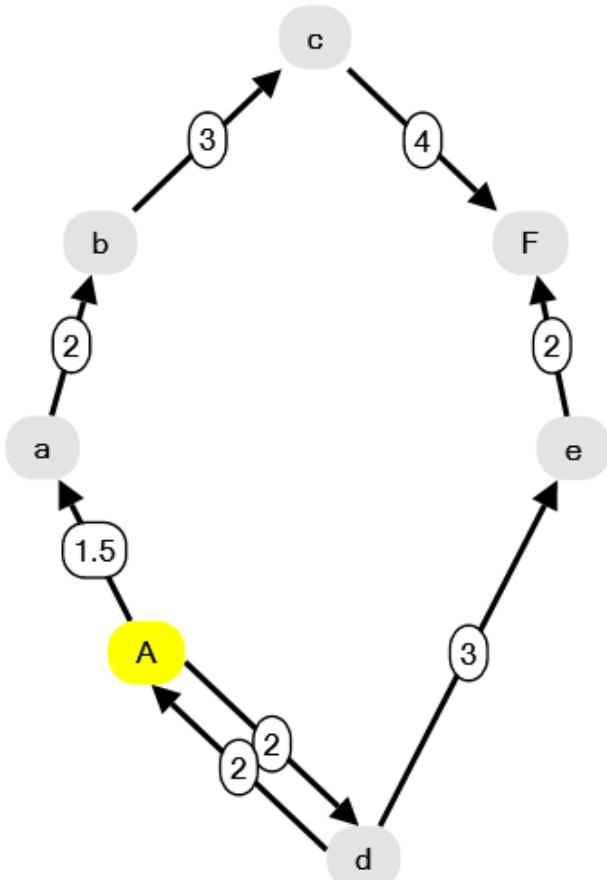
If all the edges in the search graph do not have the same cost then breadth-first search generalizes to uniform-cost search. Instead of expanding nodes in order of their depth from the root, uniform-cost search expands nodes in order of their cost from the root. At each step, the next step n to be expanded 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 00:00:00.0008861 to find the goal.

The result path costs 7

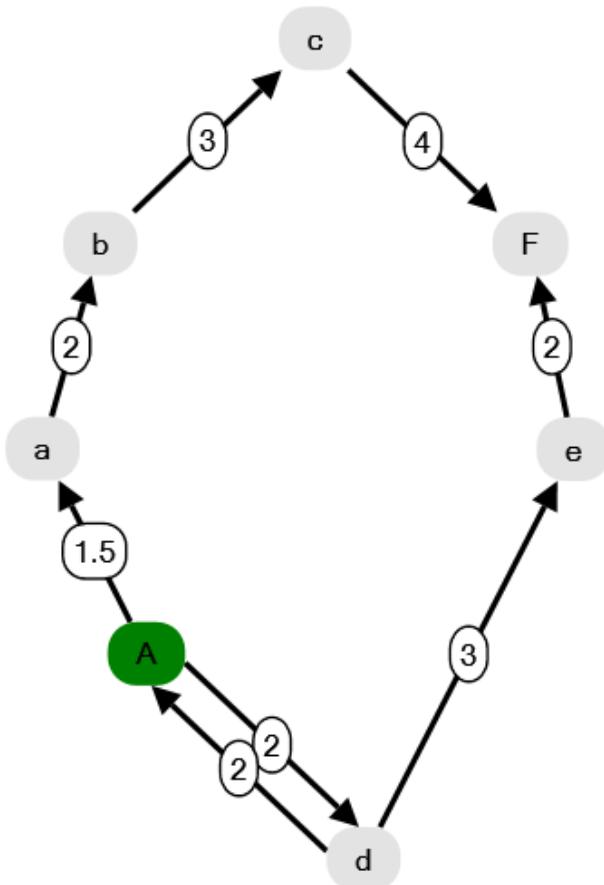
Pre-Visit

The algorithm has opened the node A



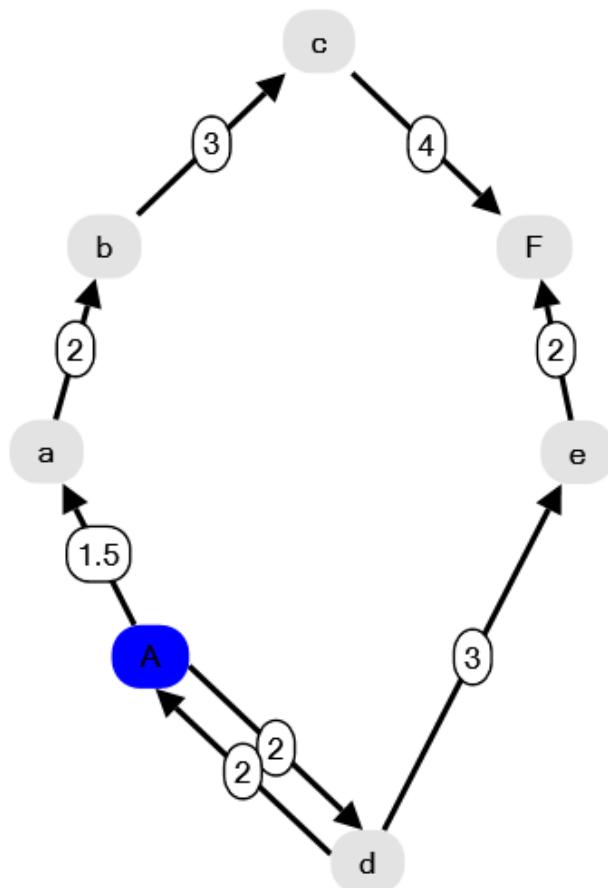
Visit

The algorithm has marked the node A as visited



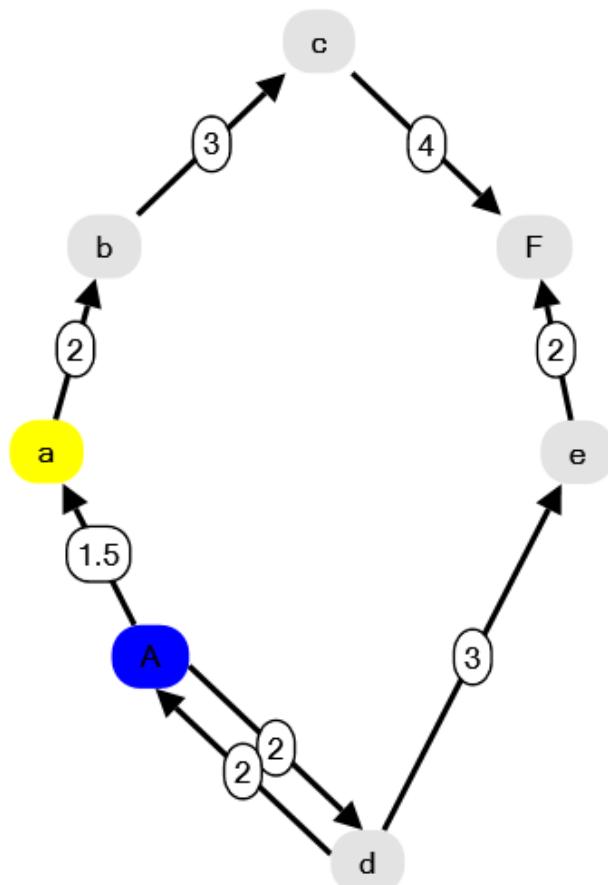
Post-Visit

The algorithm has closed the node A



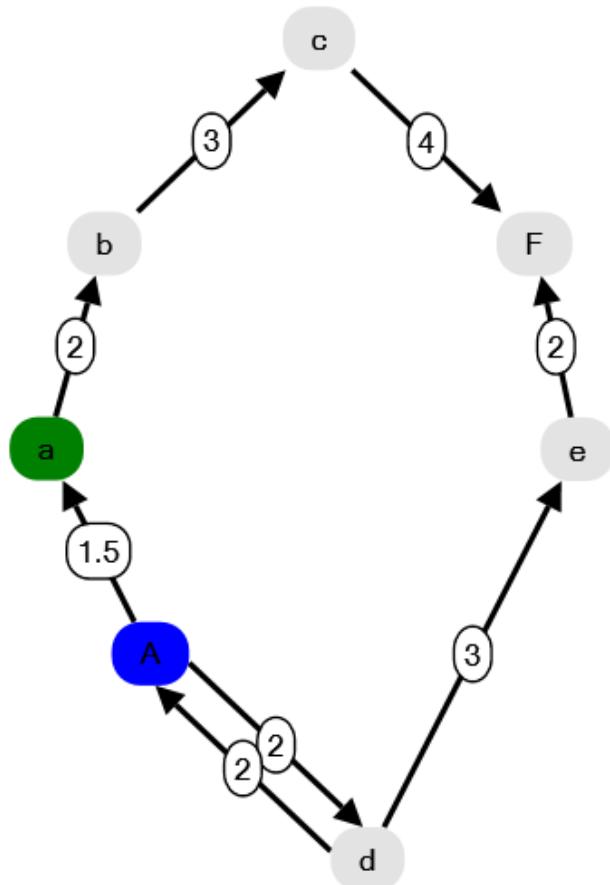
Pre-Visit

The algorithm has opened the node *a*



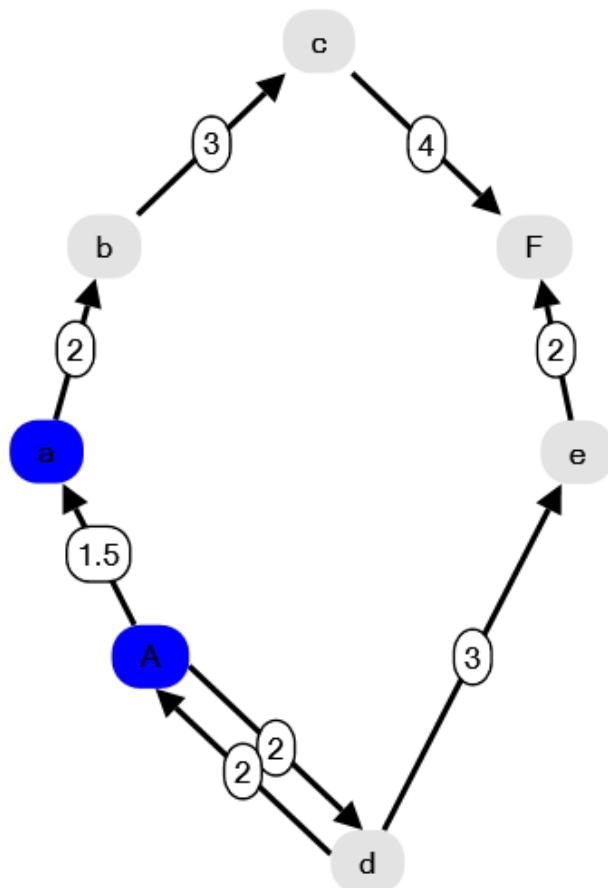
Visit

The algorithm has marked the node *a* as visited



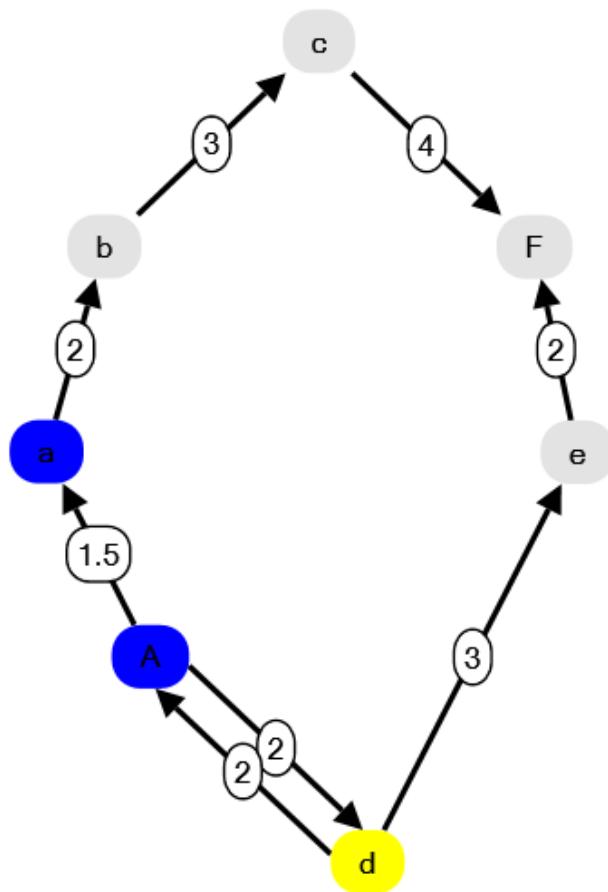
Post-Visit

The algorithm has closed the node *a*



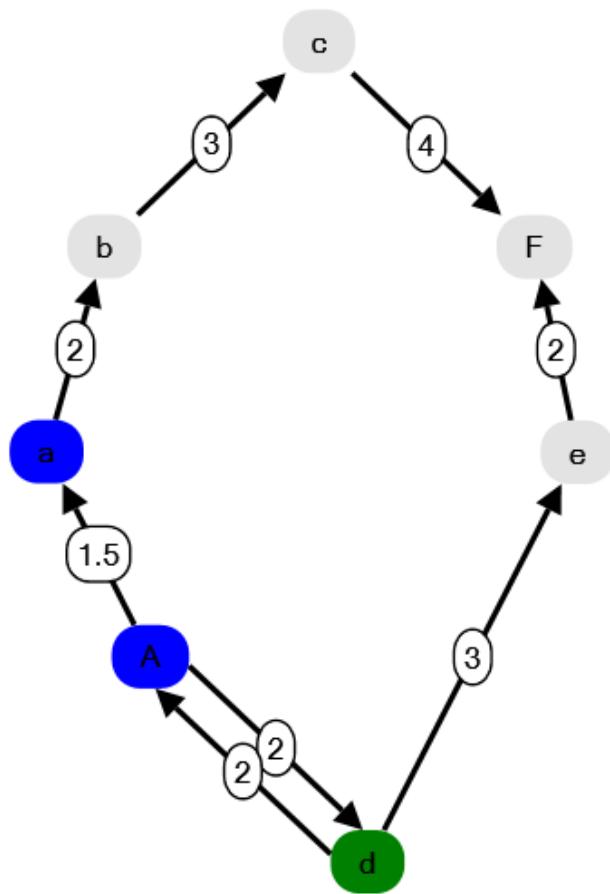
Pre-Visit

The algorithm has opened the node *d*



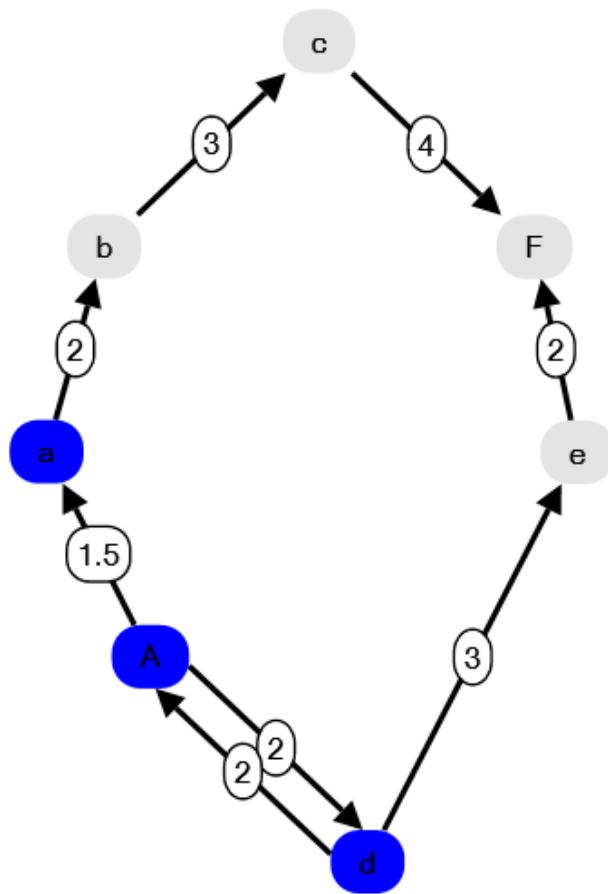
Visit

The algorithm has marked the node d as visited



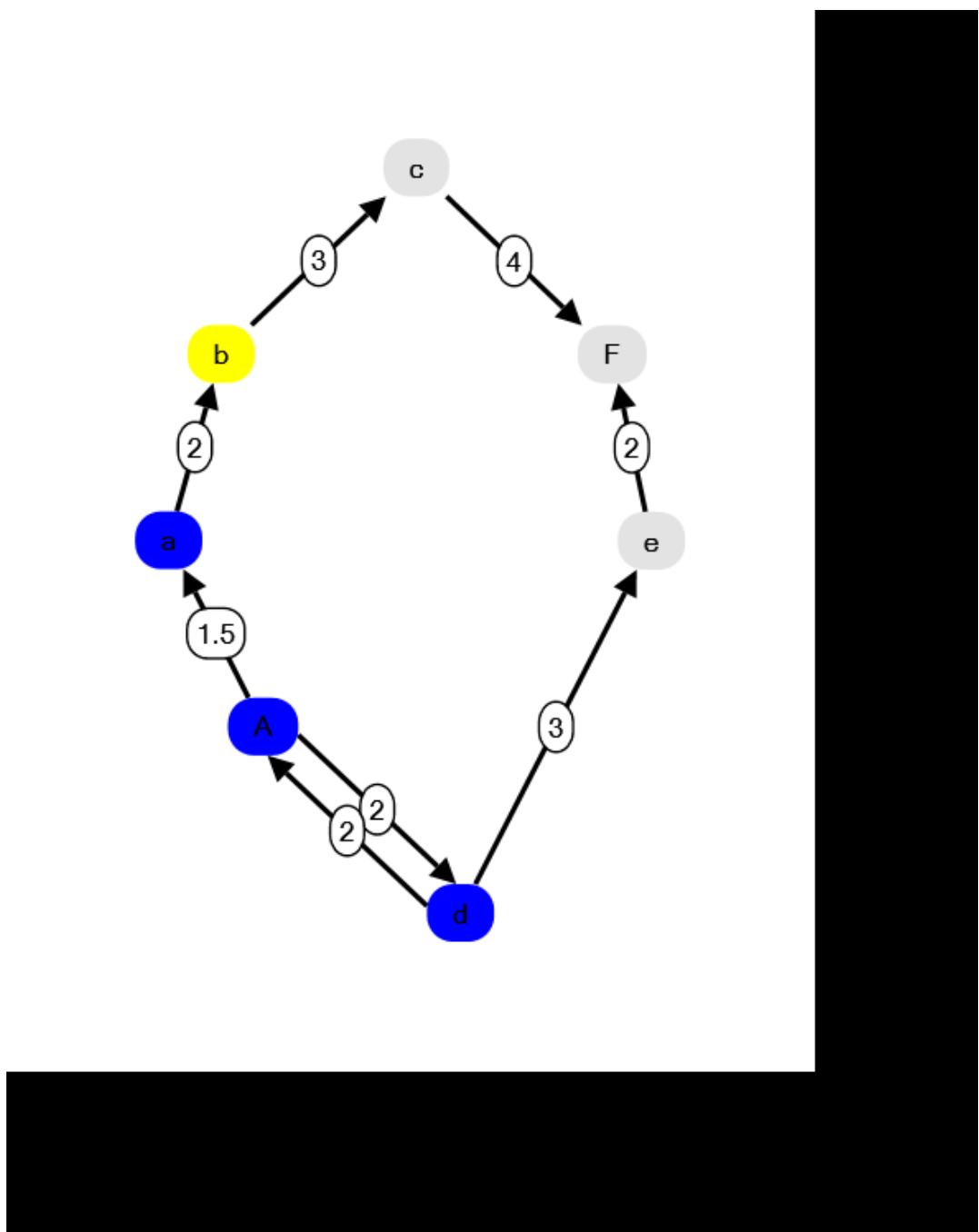
Post-Visit

The algorithm has closed the node d



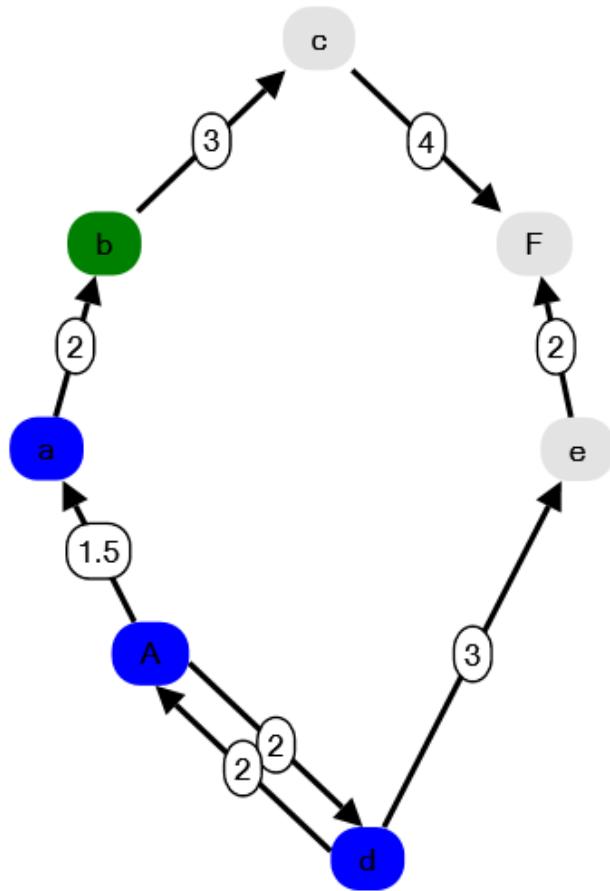
Pre-Visit

The algorithm has opened the node *b*



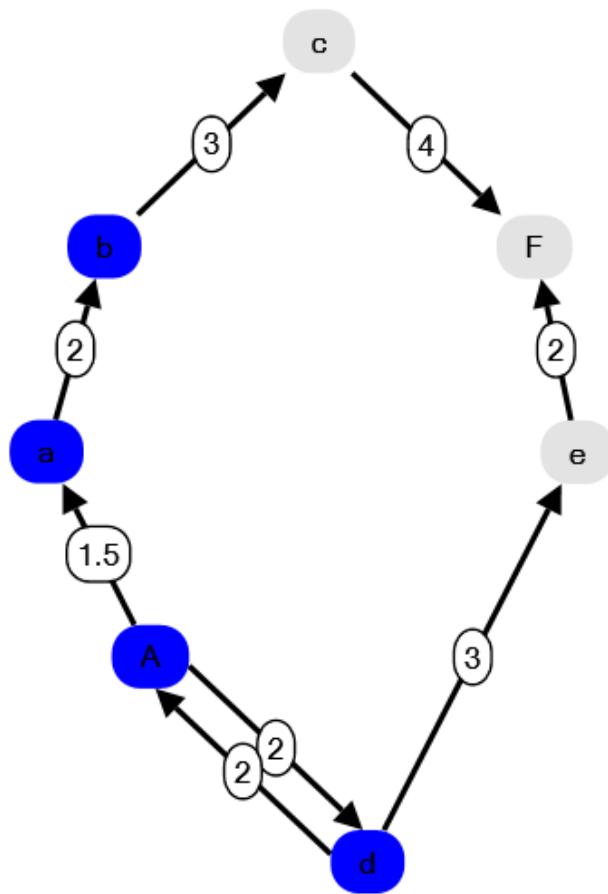
Visit

The algorithm has marked the node *b* as visited



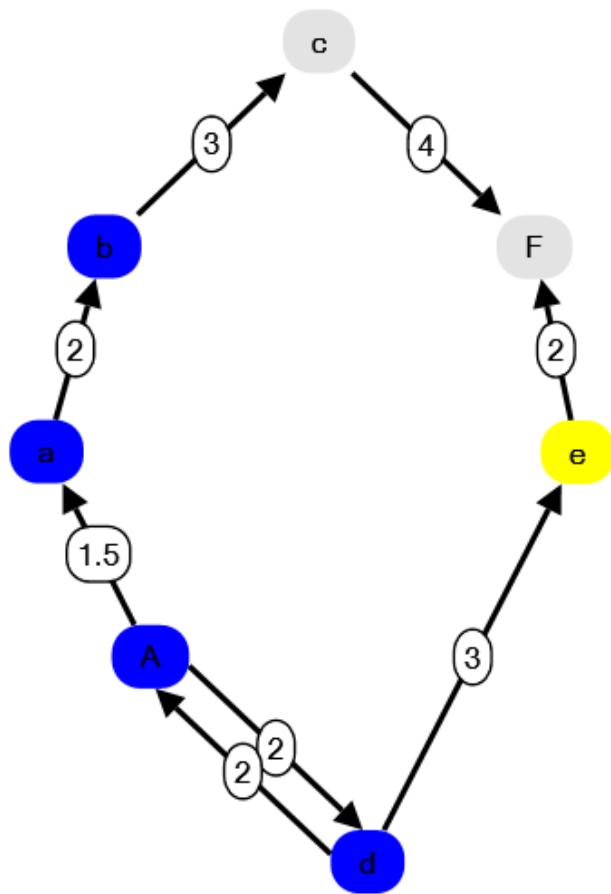
Post-Visit

The algorithm has closed the node b



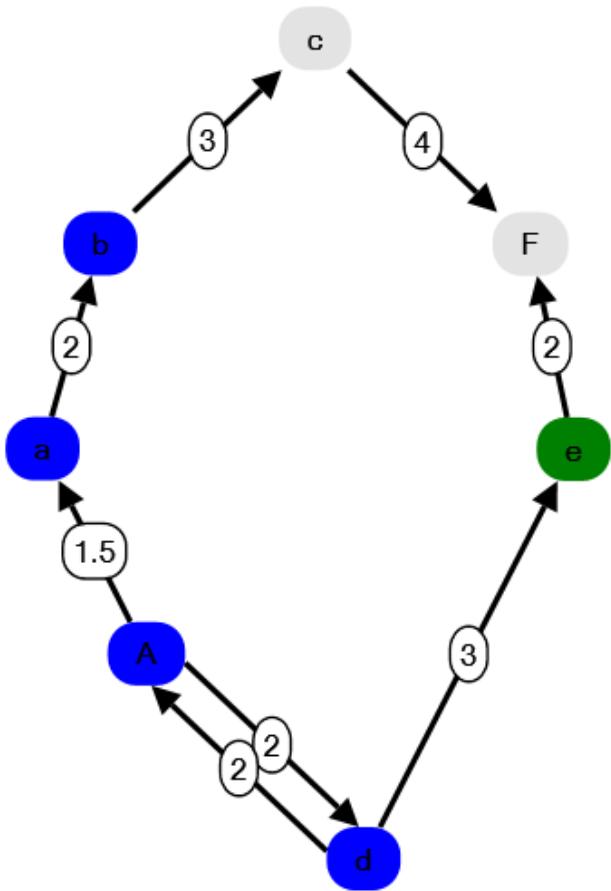
Pre-Visit

The algorithm has opened the node *e*



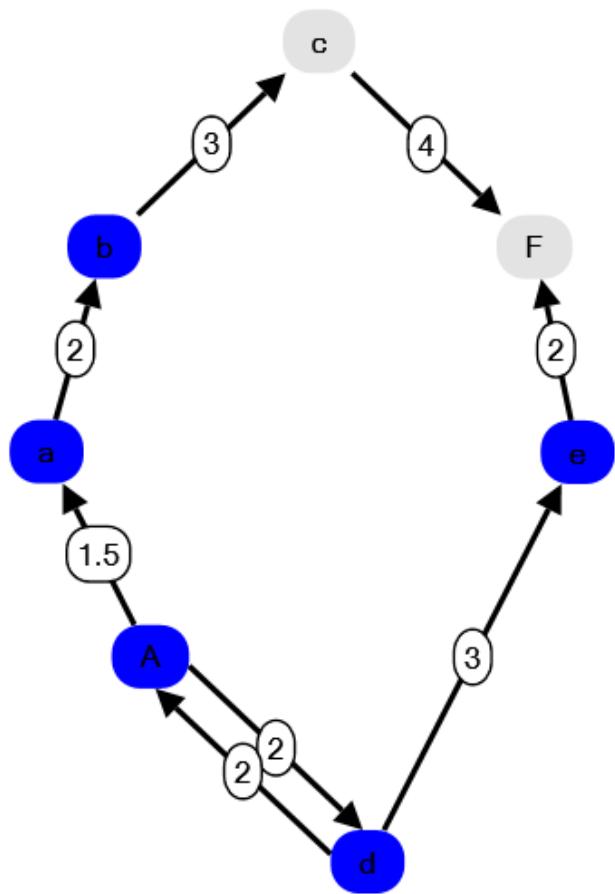
Visit

The algorithm has marked the node e as visited



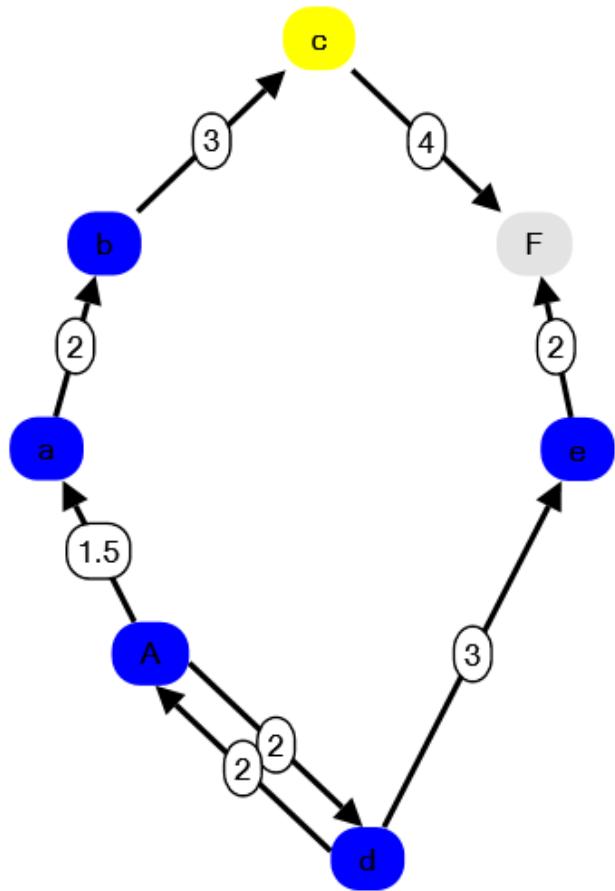
Post-Visit

The algorithm has closed the node e



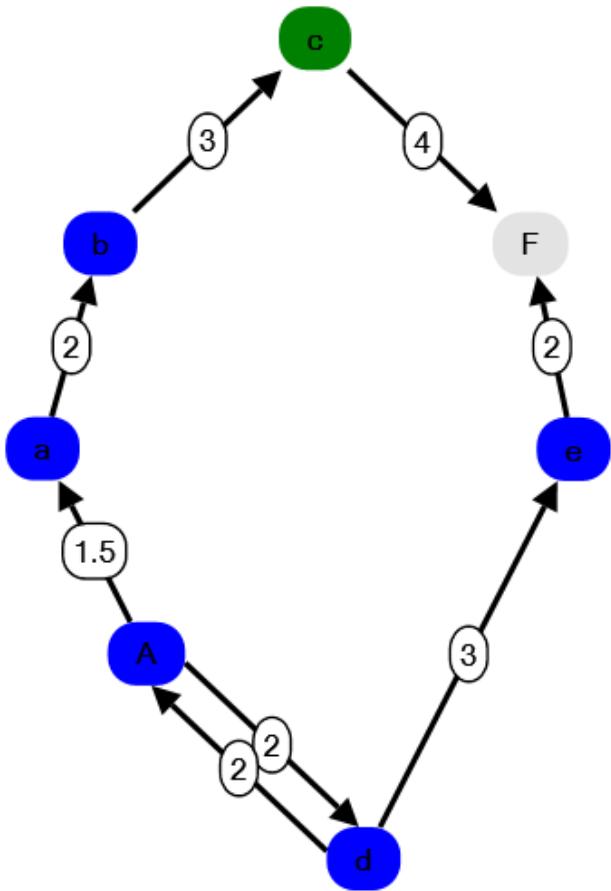
Pre-Visit

The algorithm has opened the node *c*



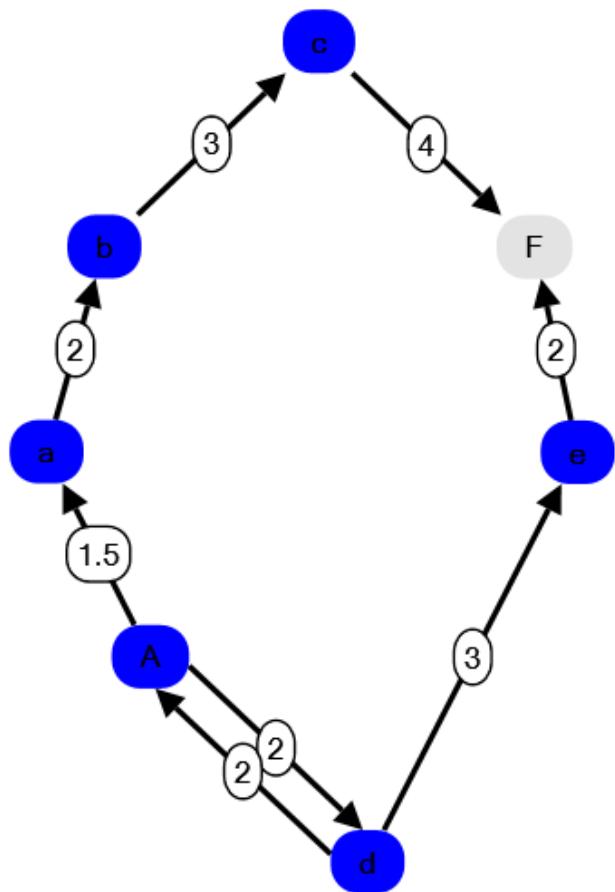
Visit

The algorithm has marked the node c as visited



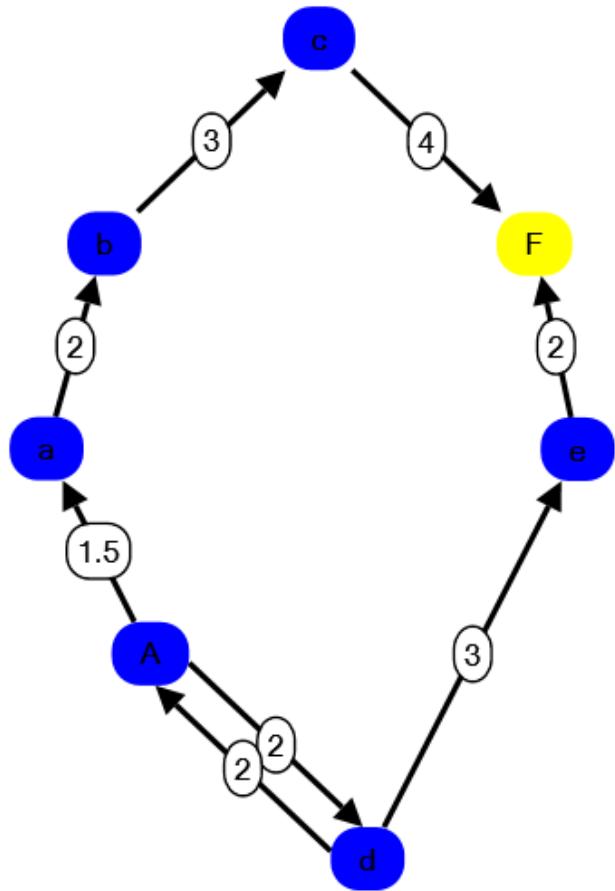
Post-Visit

The algorithm has closed the node c



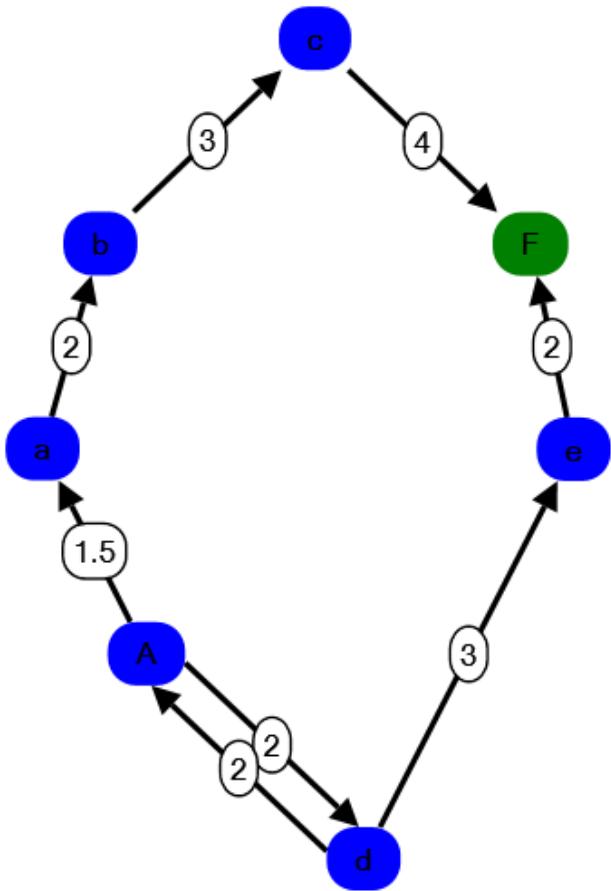
Pre-Visit

The algorithm has opened the node F



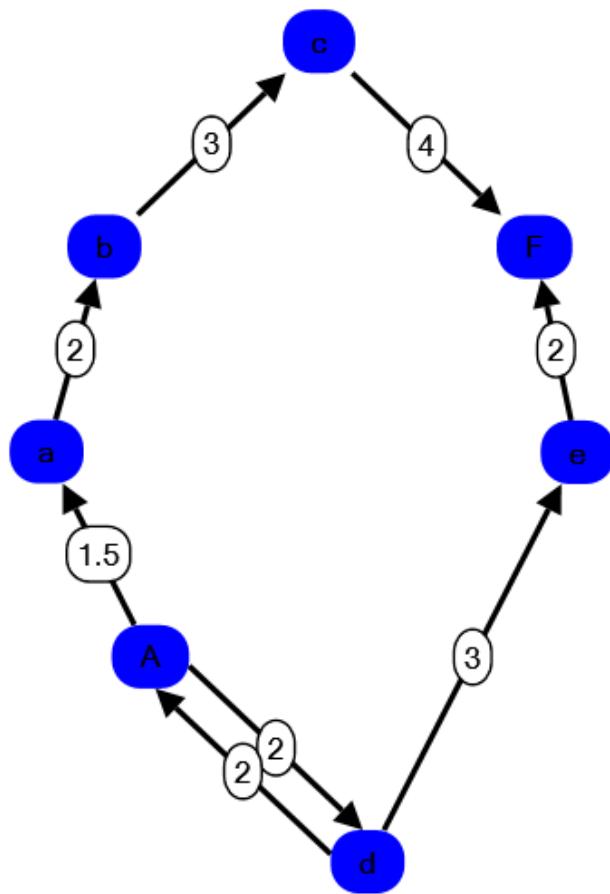
Visit

The algorithm has marked the node F as visited



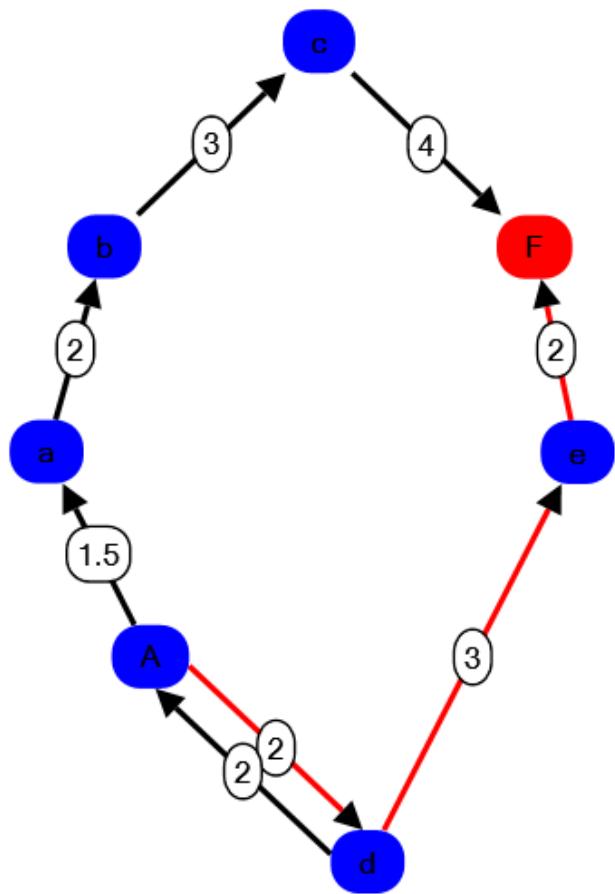
Post-Visit

The algorithm has closed the node F



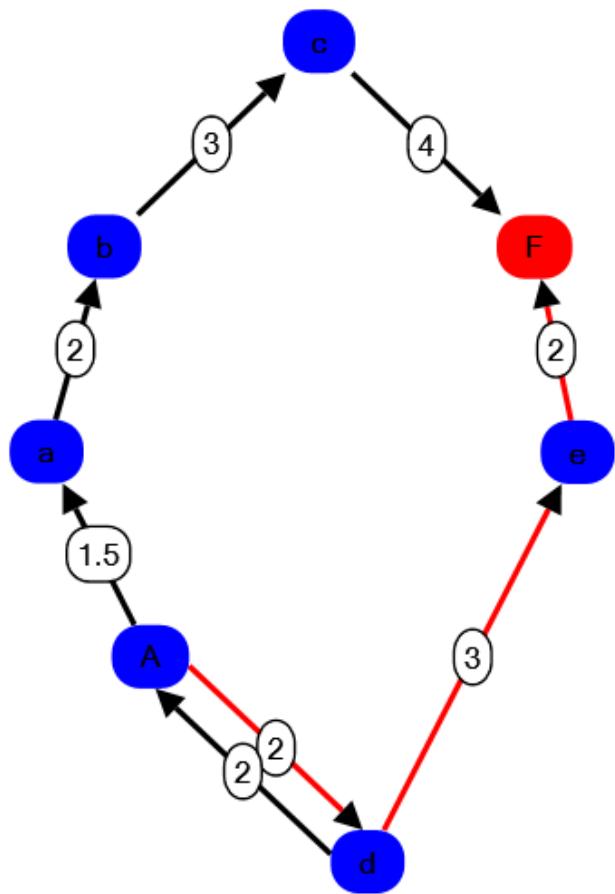
Result Found

The algorithm has found the target node F



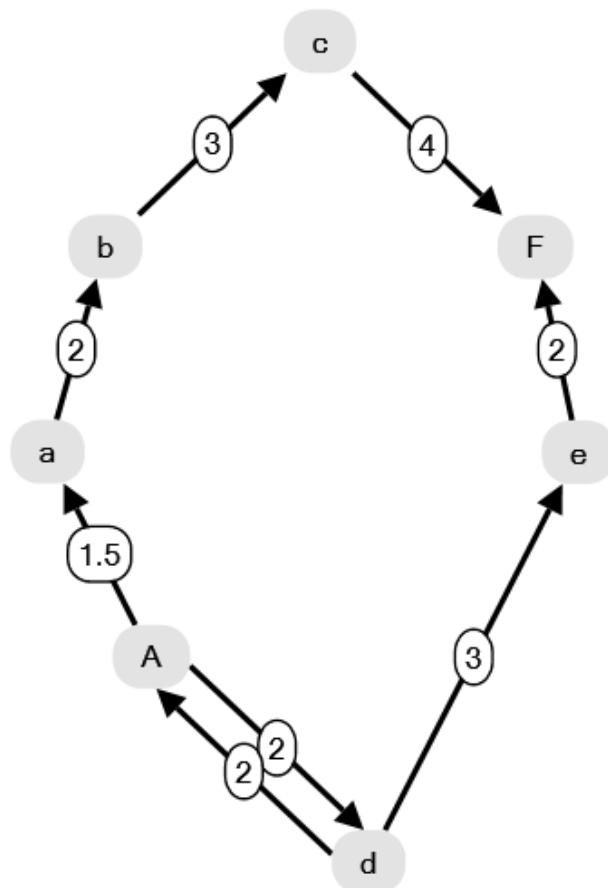
Result Found

The algorithm has found the target node F



Reset

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



Iterative Depth limited search

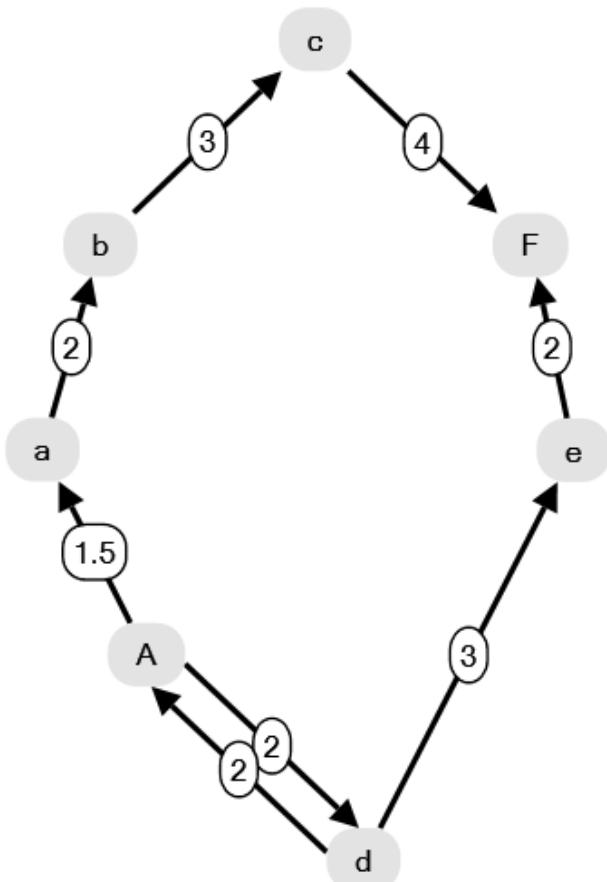
In computer science, iterative deepening search or more specifically iterative deepening depth-first search (IDS or IDDFS) is a state space/graph search strategy in which a depth-limited version of depth-first search is run repeatedly with increasing depth limits until the goal is found. IDDFS is equivalent 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 00:00:00.0007905 to find the goal.

The result path costs 7

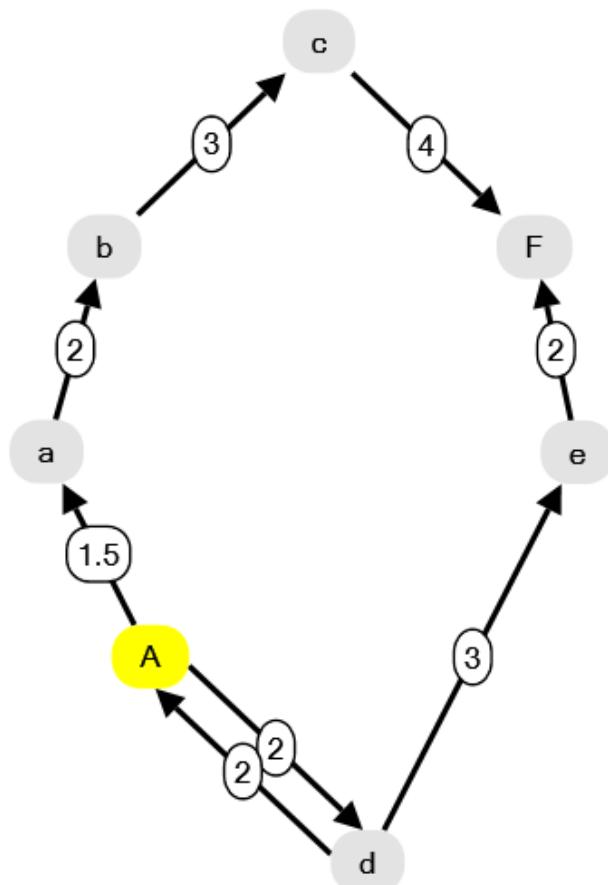
Reset

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



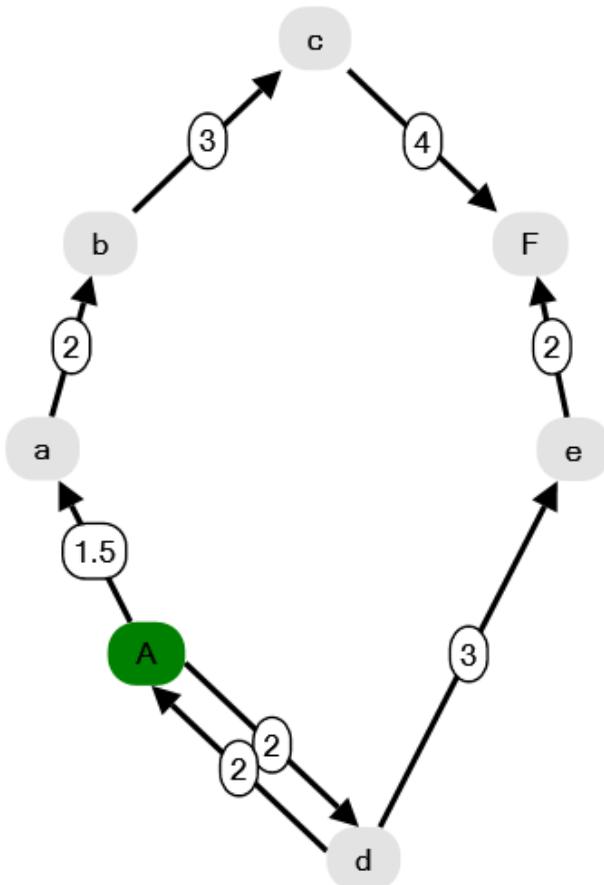
Pre-Visit

The algorithm has opened the node A



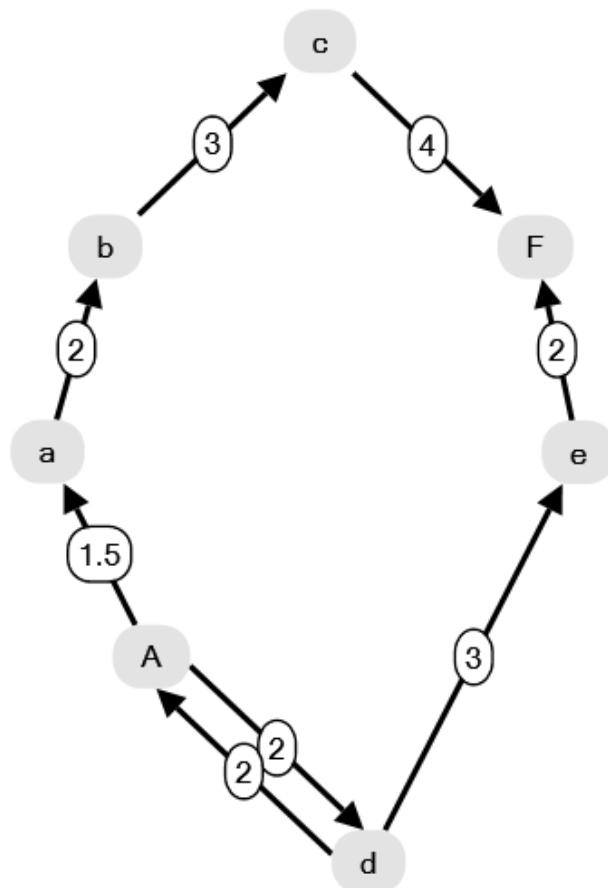
Visit

The algorithm has marked the node A as visited



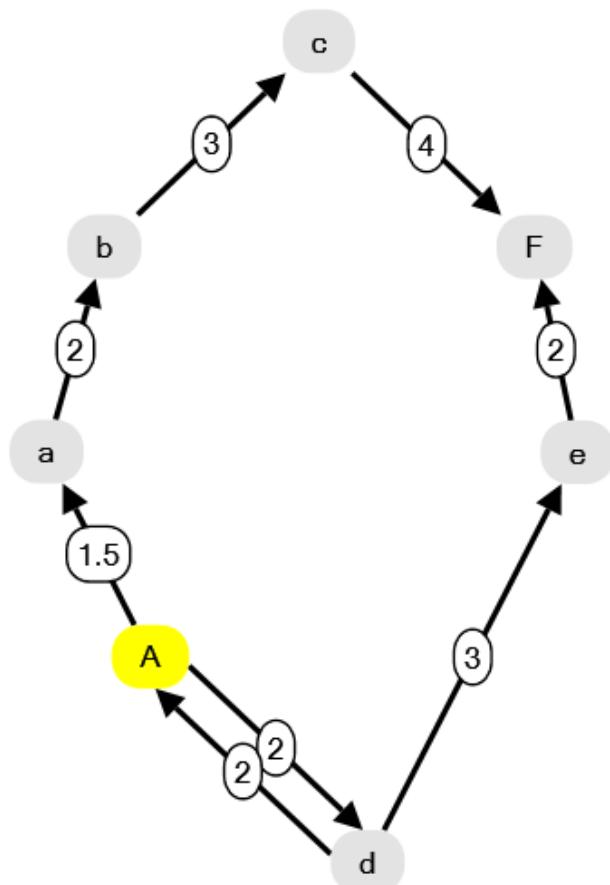
Reset

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



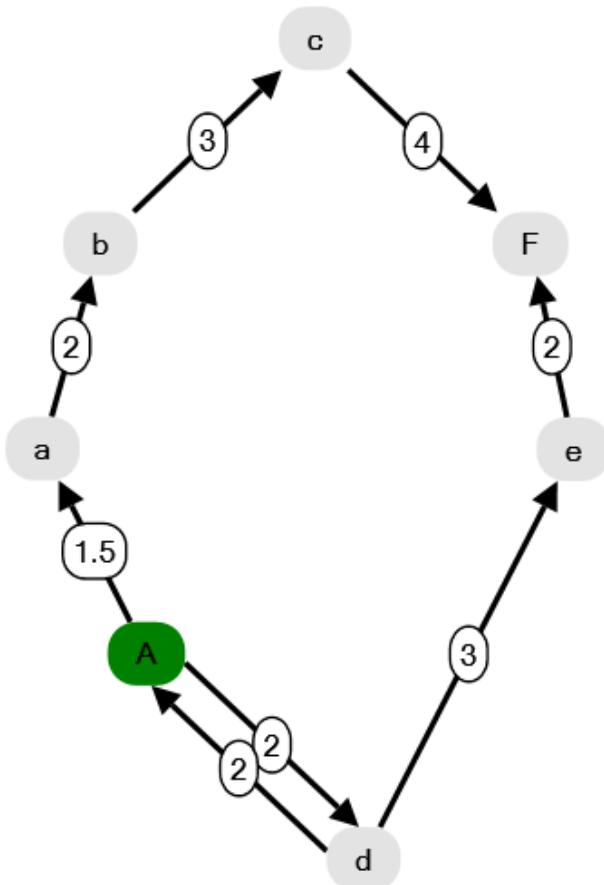
Pre-Visit

The algorithm has opened the node A



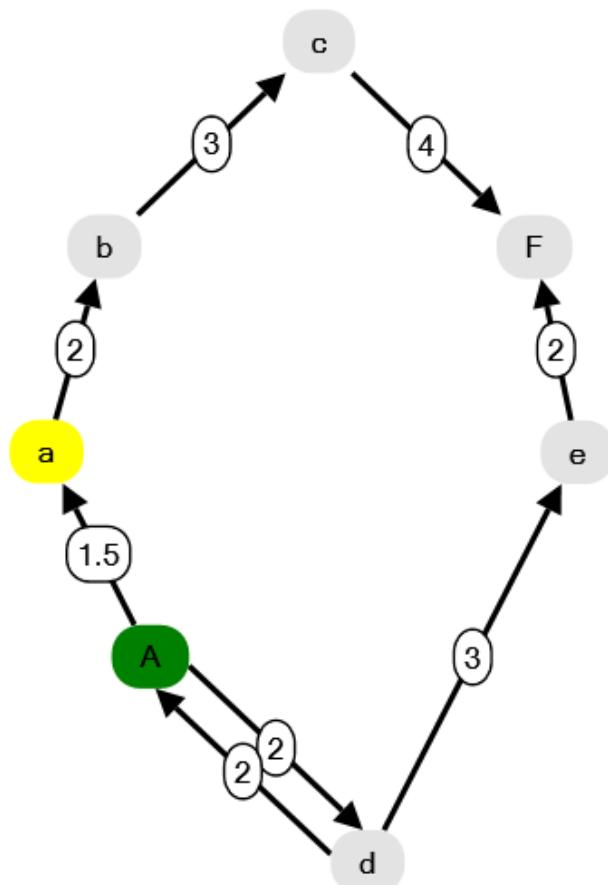
Visit

The algorithm has marked the node A as visited



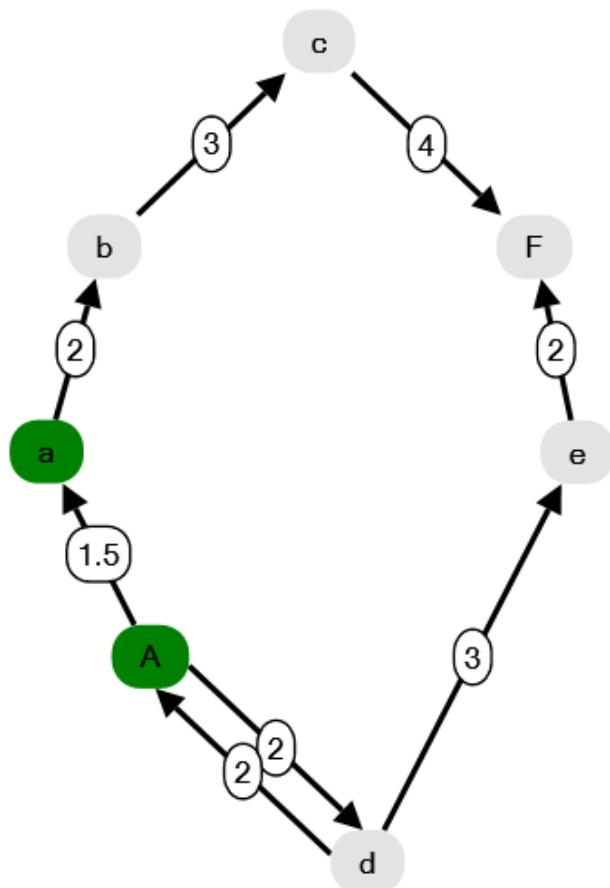
Pre-Visit

The algorithm has opened the node *a*



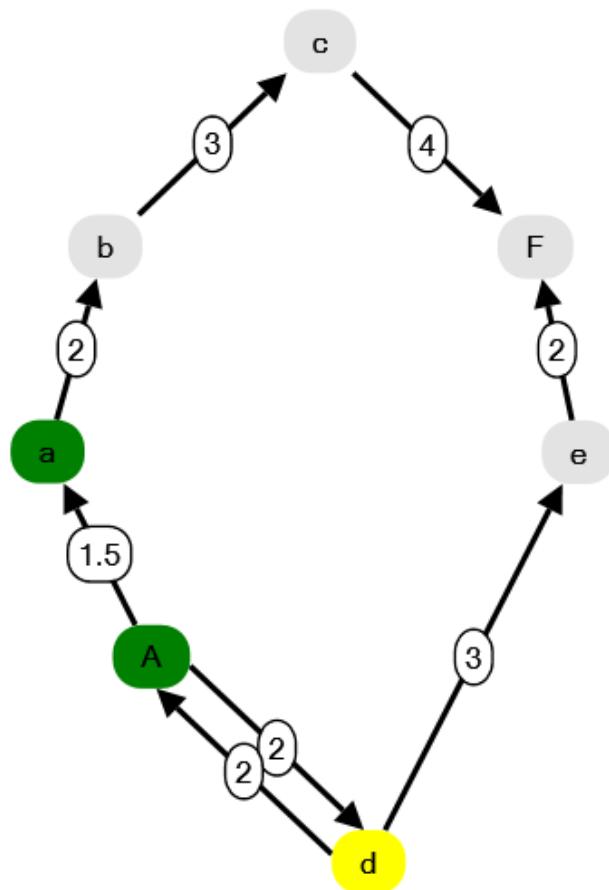
Visit

The algorithm has marked the node *a* as visited



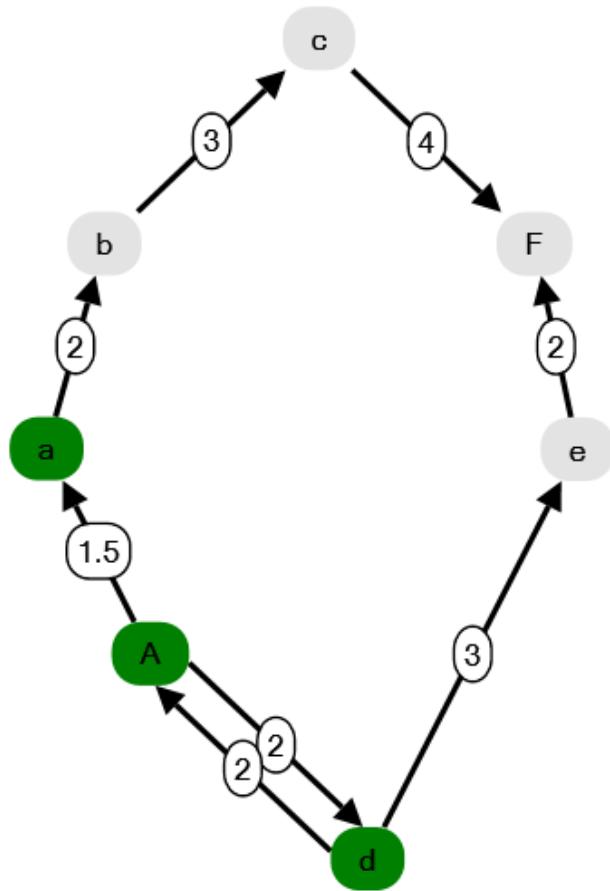
Pre-Visit

The algorithm has opened the node *d*



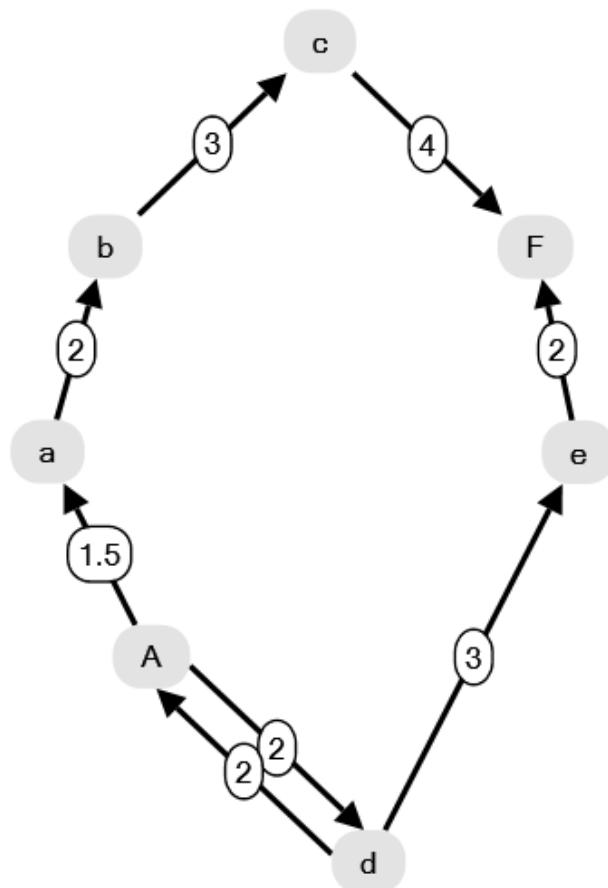
Visit

The algorithm has marked the node d as visited



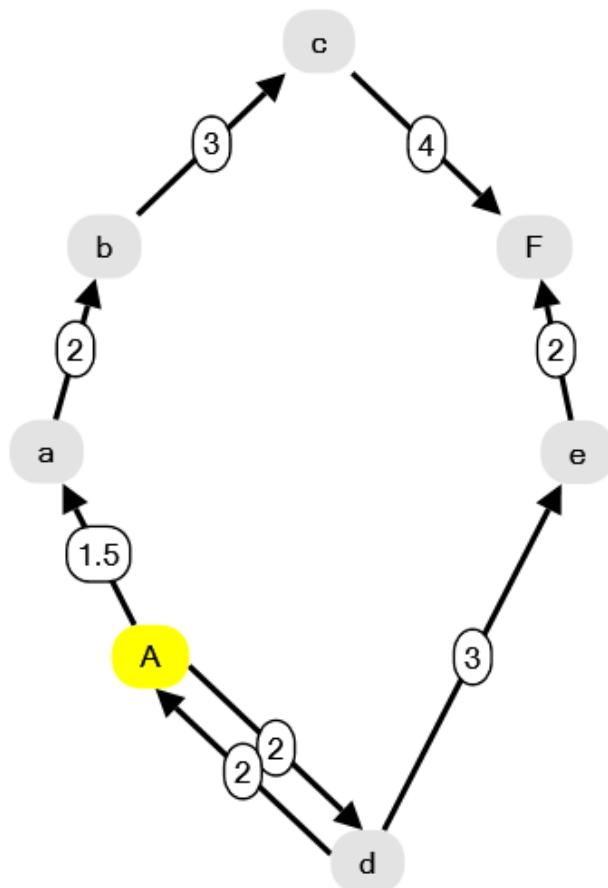
Reset

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



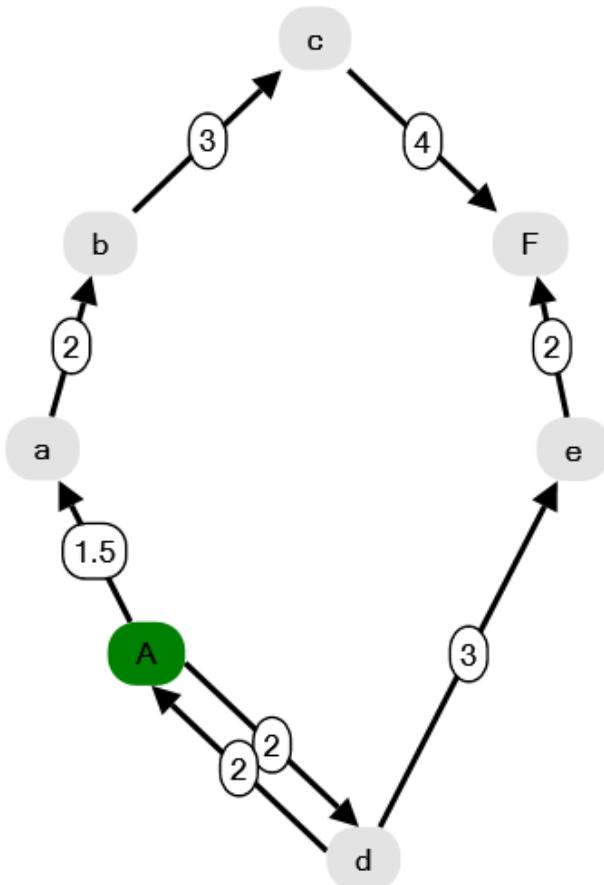
Pre-Visit

The algorithm has opened the node A



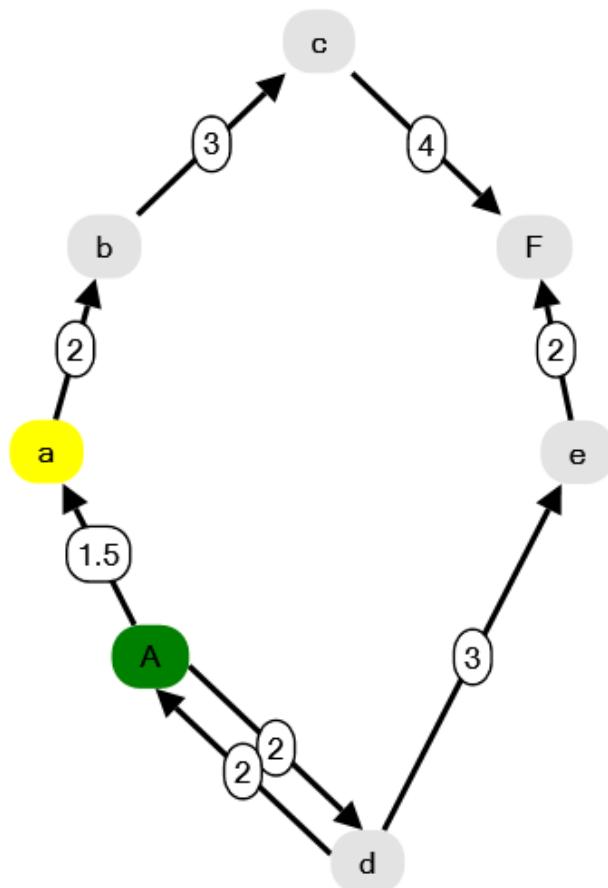
Visit

The algorithm has marked the node A as visited



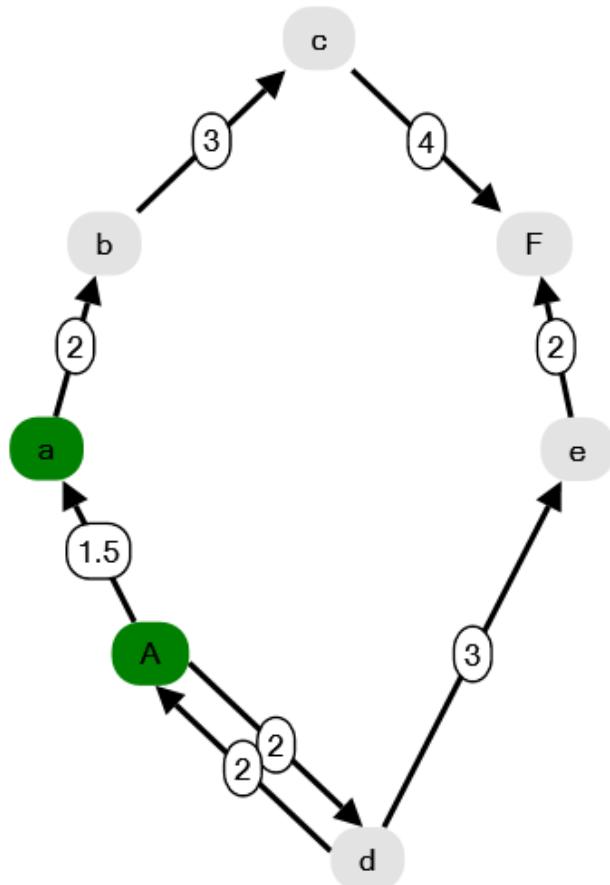
Pre-Visit

The algorithm has opened the node *a*



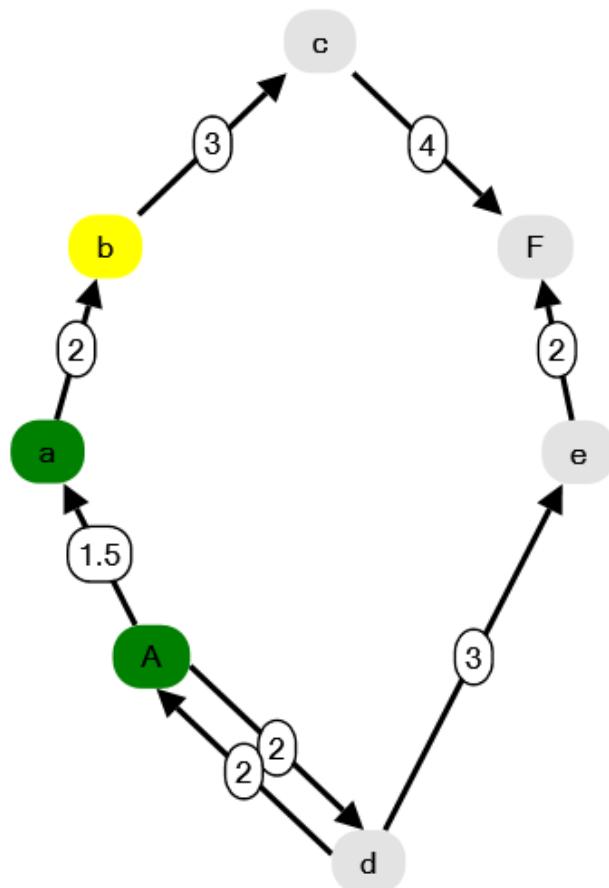
Visit

The algorithm has marked the node *a* as visited



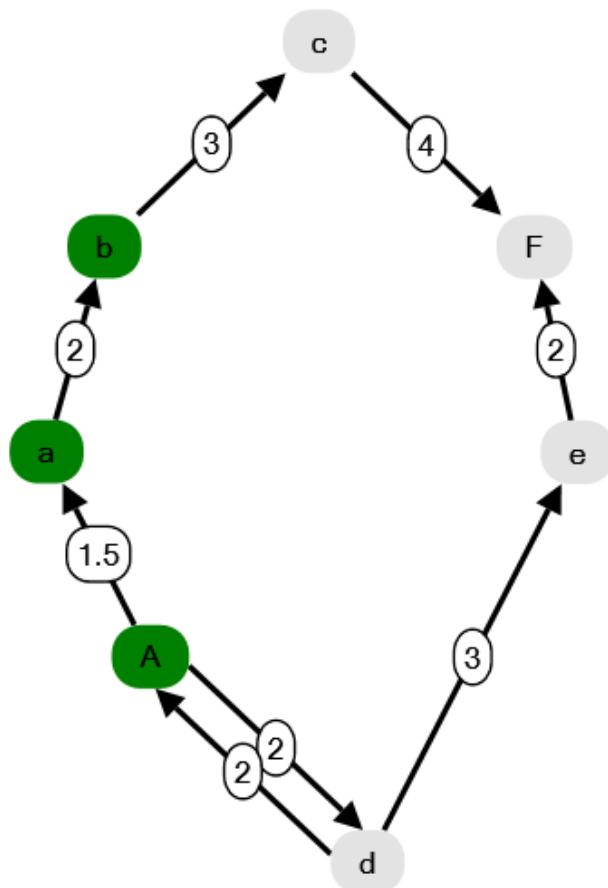
Pre-Visit

The algorithm has opened the node *b*



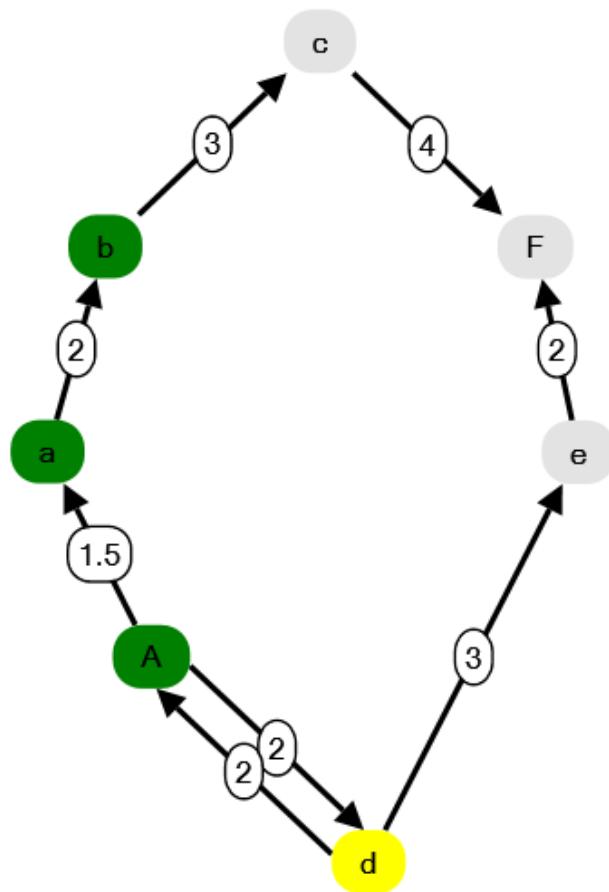
Visit

The algorithm has marked the node *b* as visited



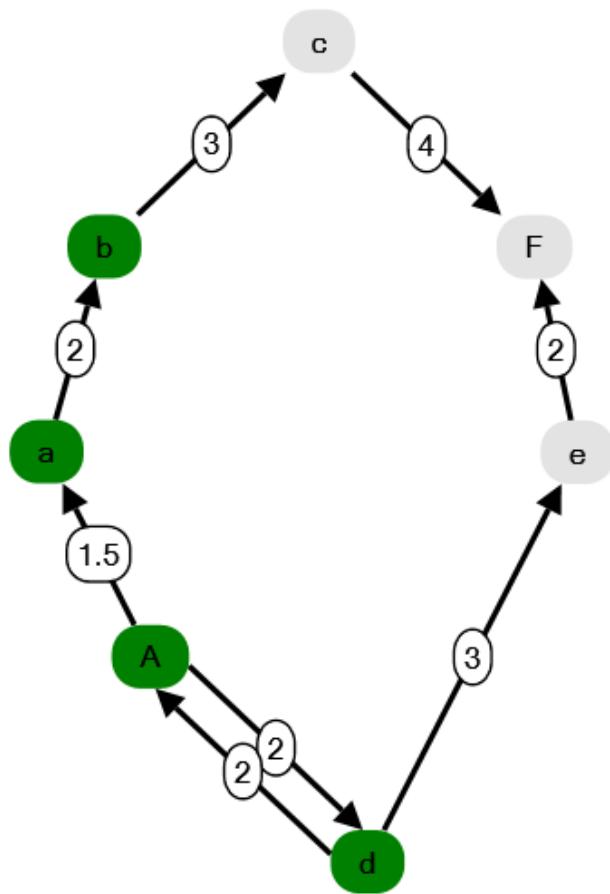
Pre-Visit

The algorithm has opened the node *d*



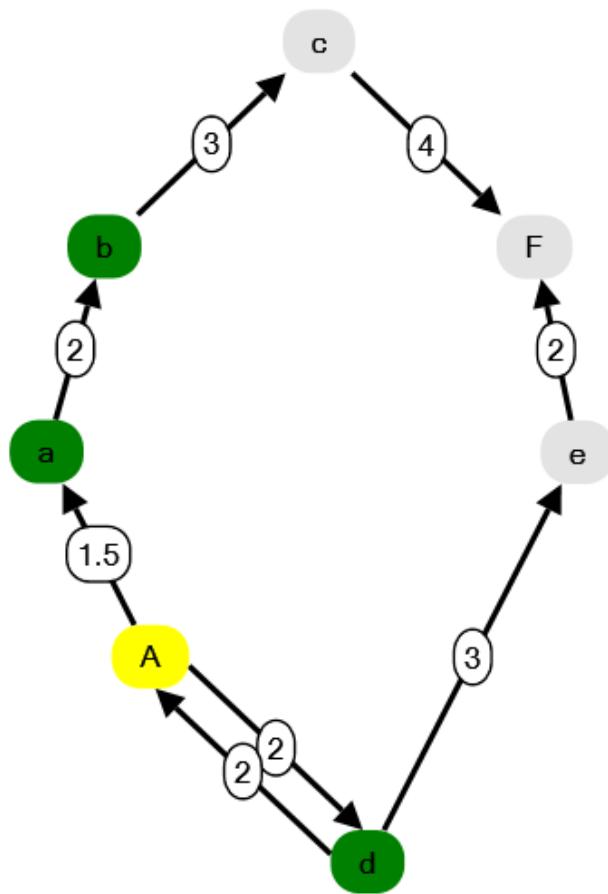
Visit

The algorithm has marked the node d as visited



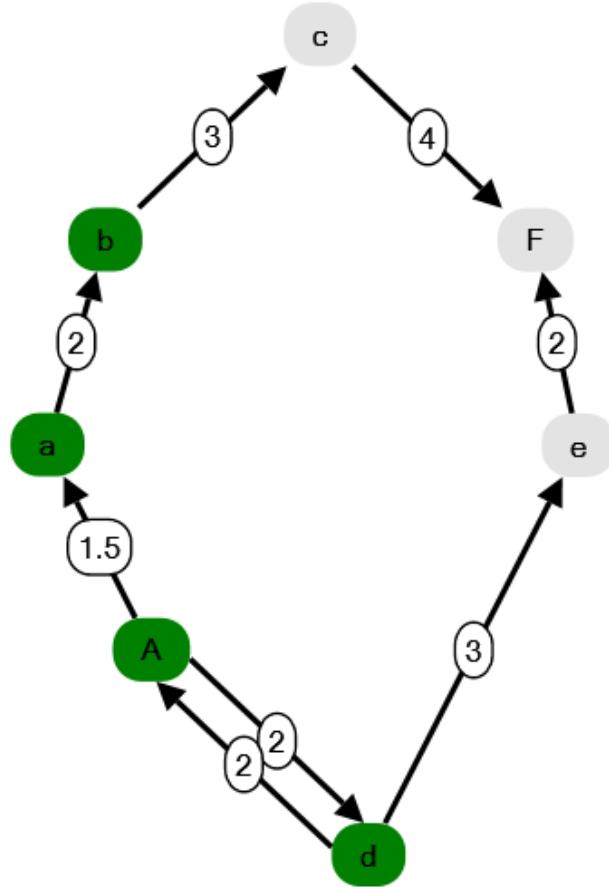
Pre-Visit

The algorithm has opened the node A



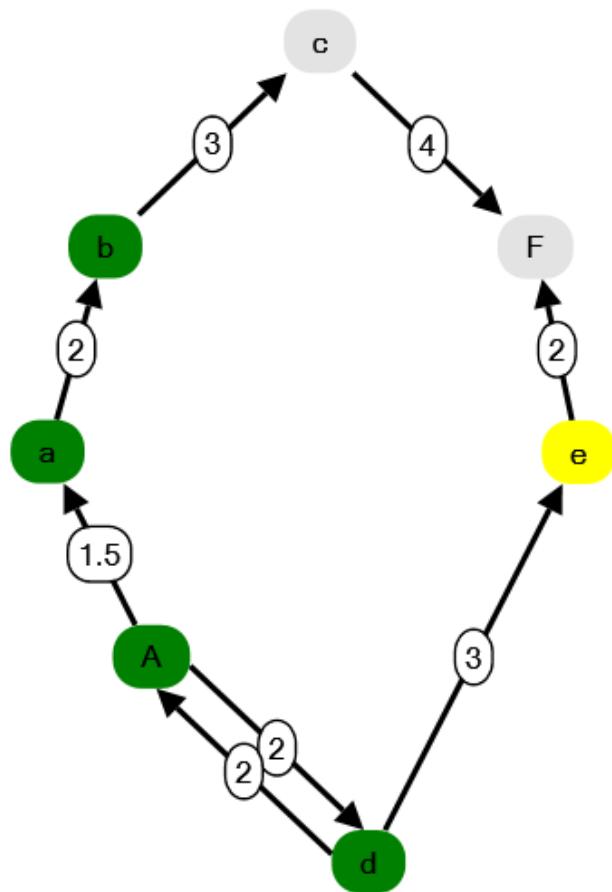
Visit

The algorithm has marked the node A as visited



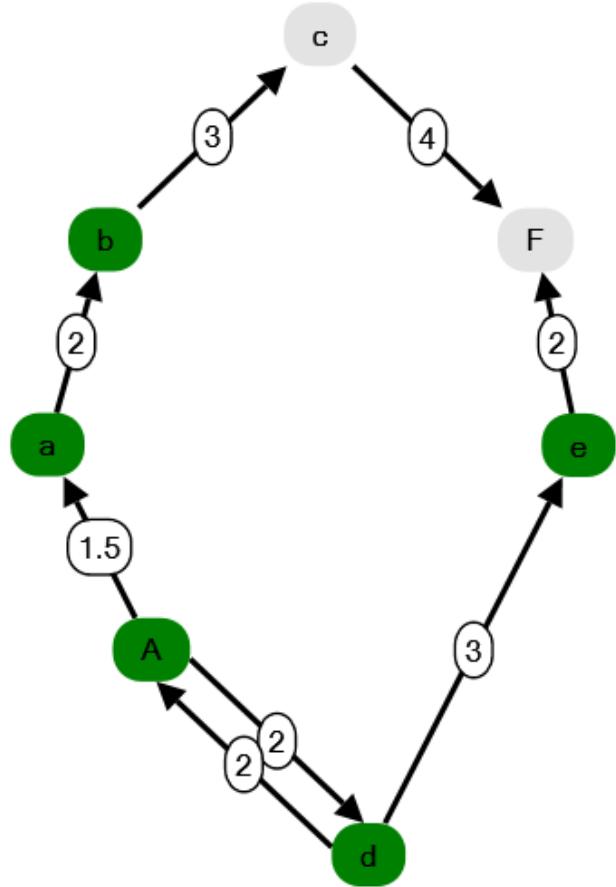
Pre-Visit

The algorithm has opened the node *e*



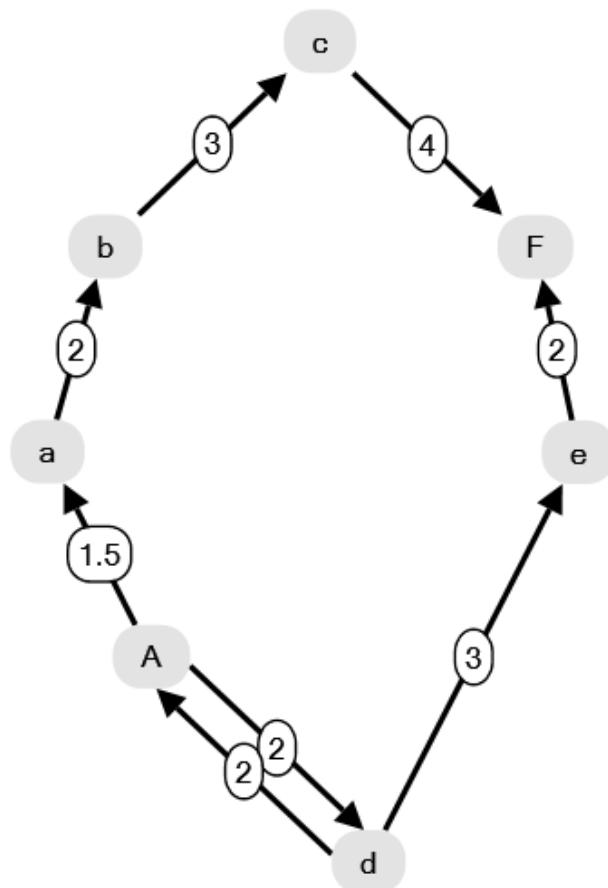
Visit

The algorithm has marked the node e as visited



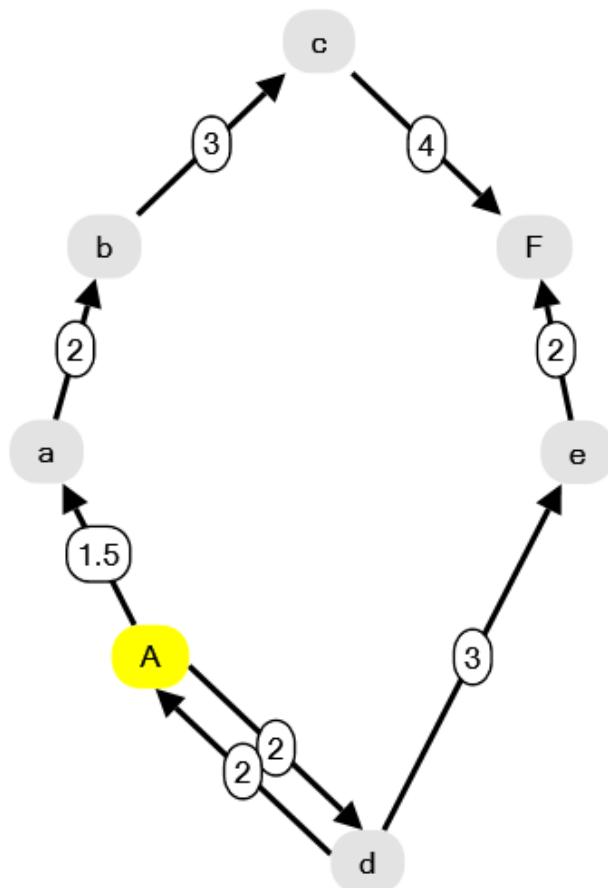
Reset

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



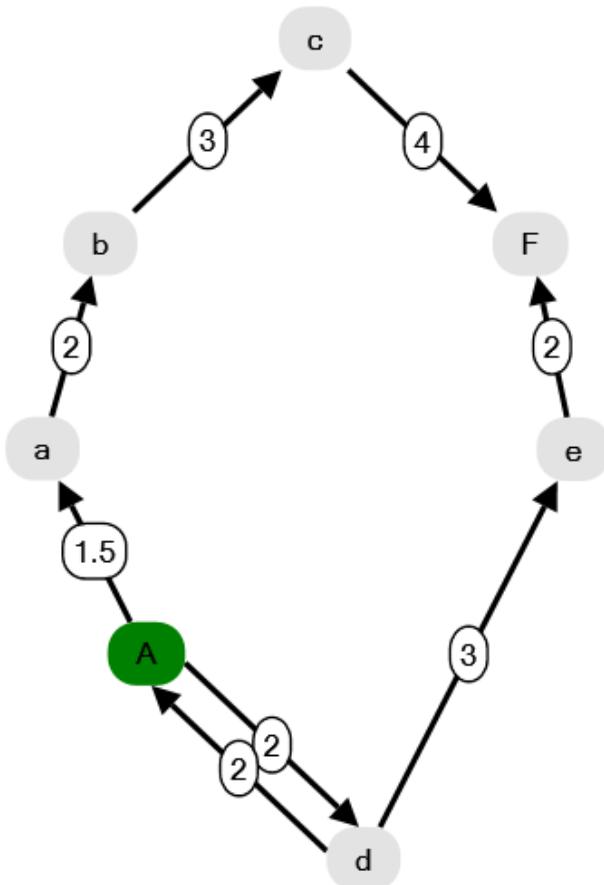
Pre-Visit

The algorithm has opened the node A



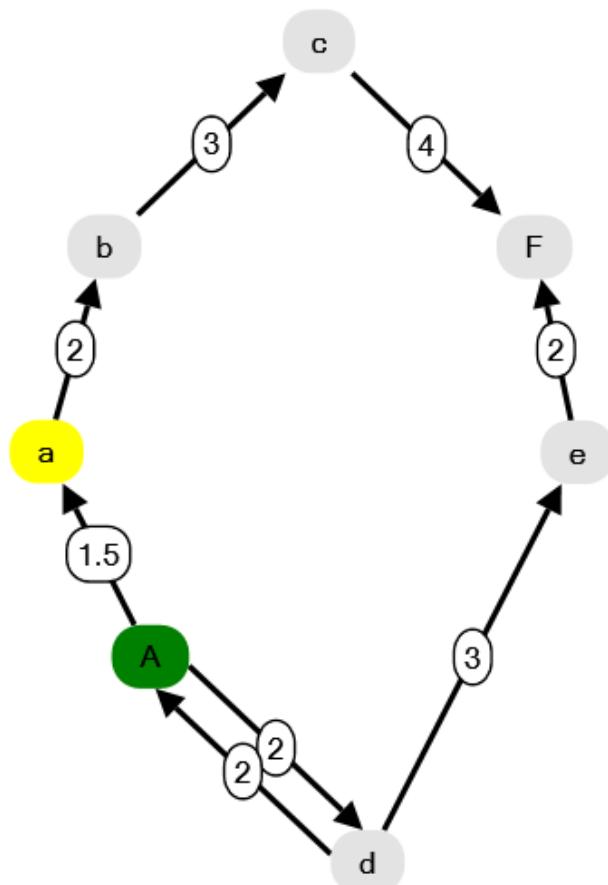
Visit

The algorithm has marked the node A as visited



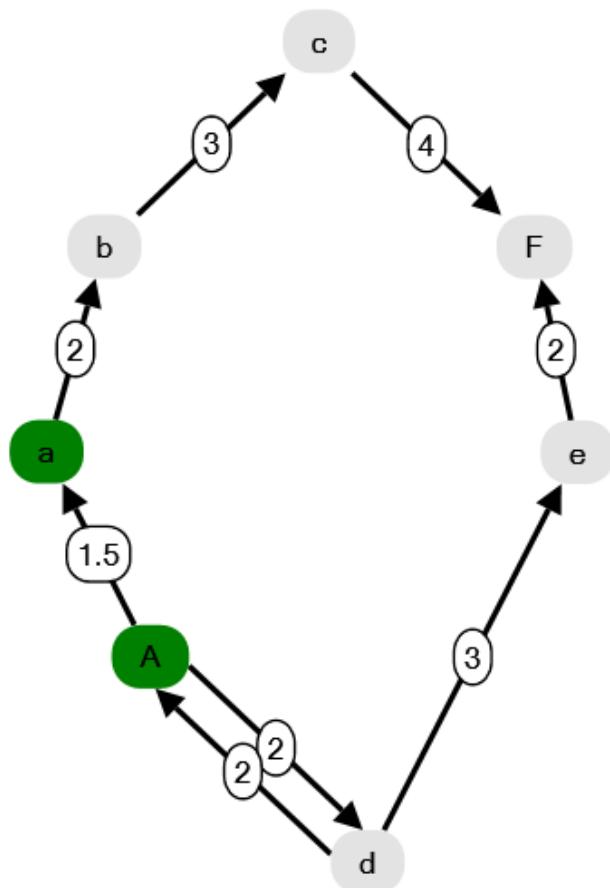
Pre-Visit

The algorithm has opened the node *a*



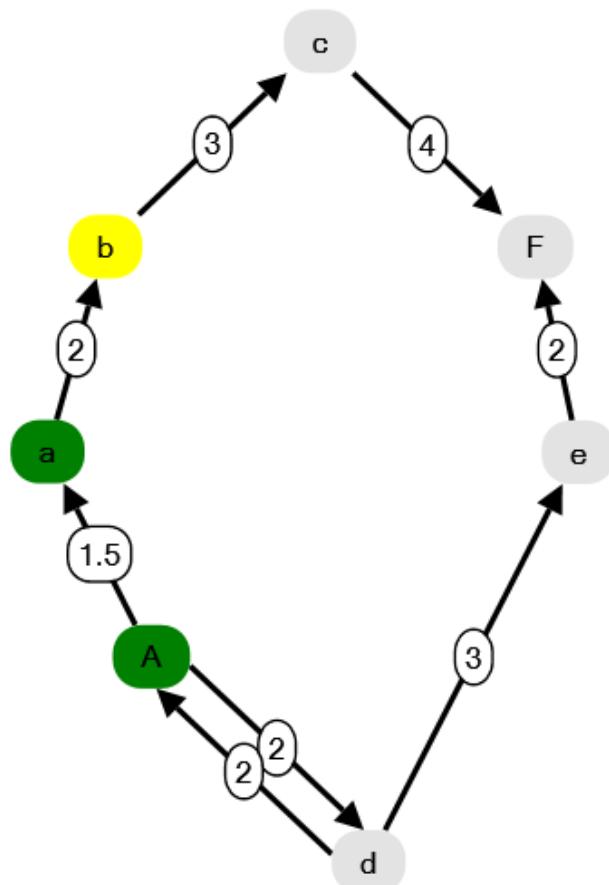
Visit

The algorithm has marked the node *a* as visited



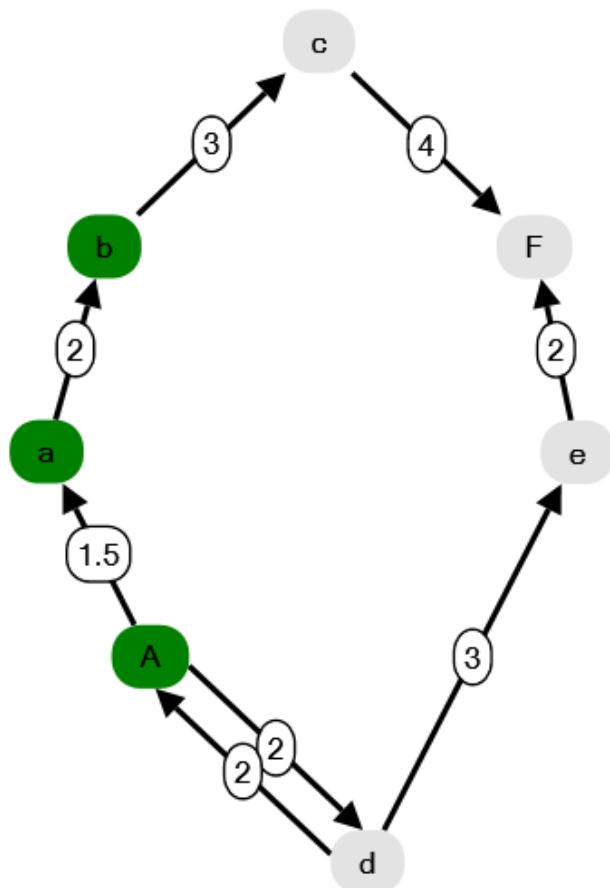
Pre-Visit

The algorithm has opened the node *b*



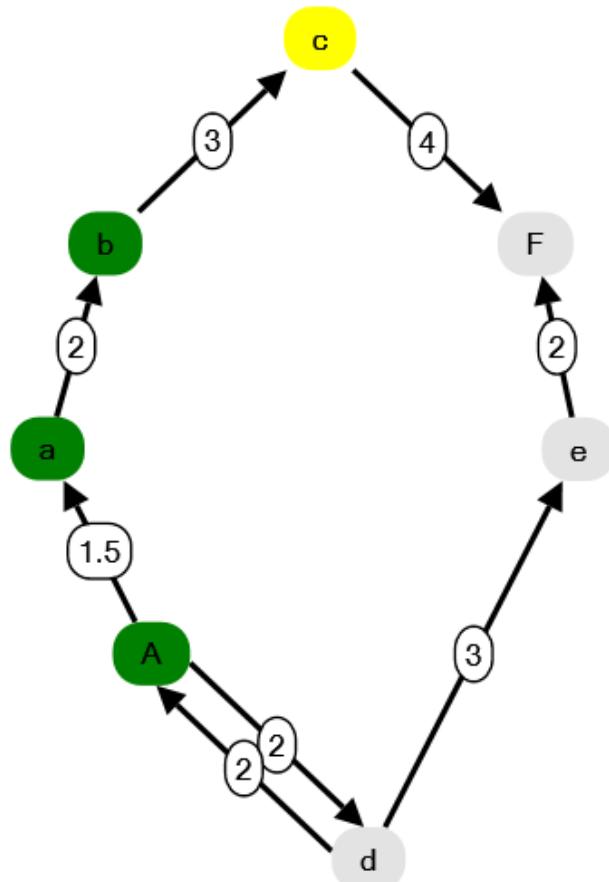
Visit

The algorithm has marked the node *b* as visited



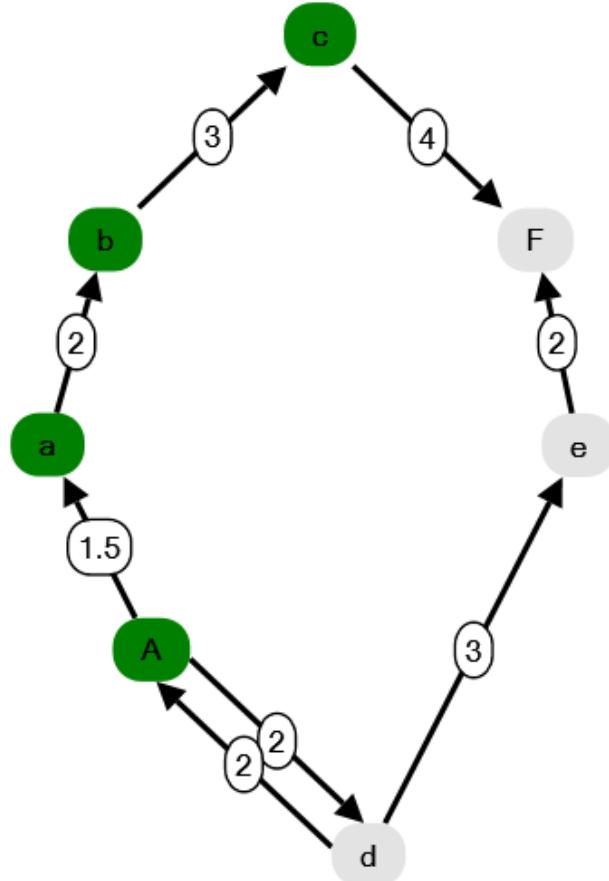
Pre-Visit

The algorithm has opened the node *c*



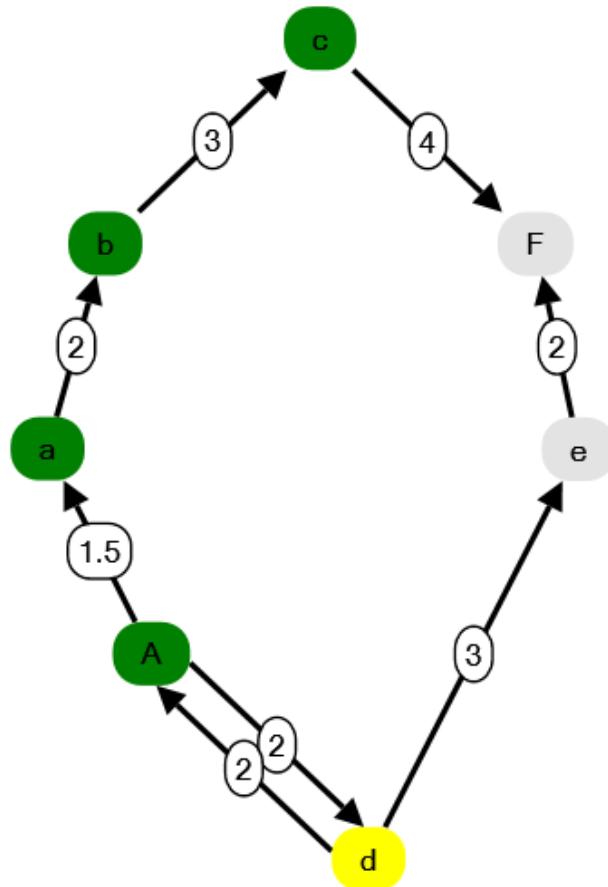
Visit

The algorithm has marked the node c as visited



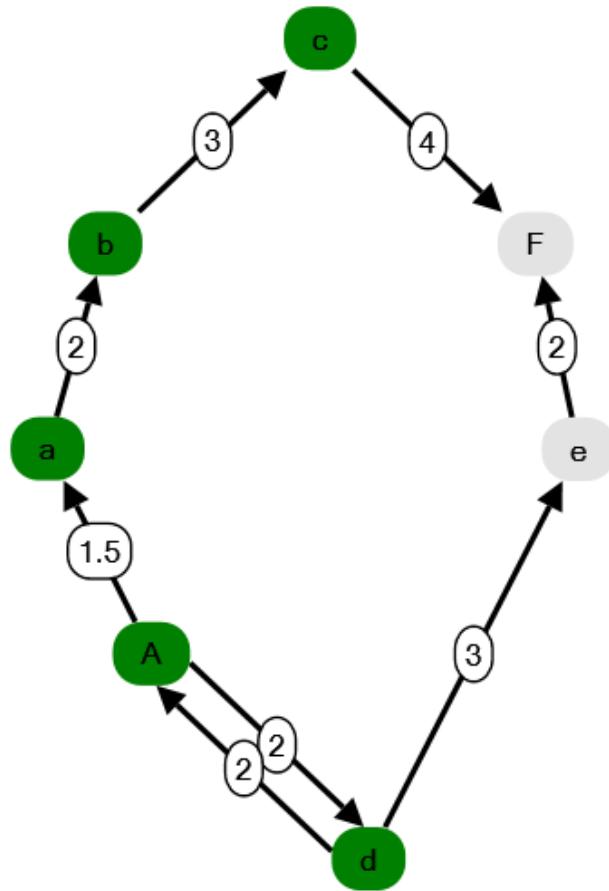
Pre-Visit

The algorithm has opened the node *d*



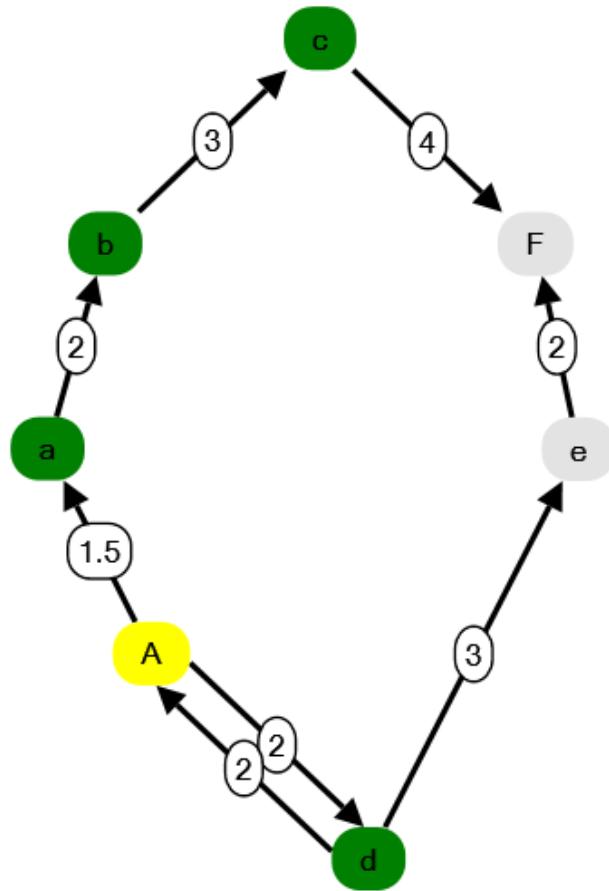
Visit

The algorithm has marked the node d as visited



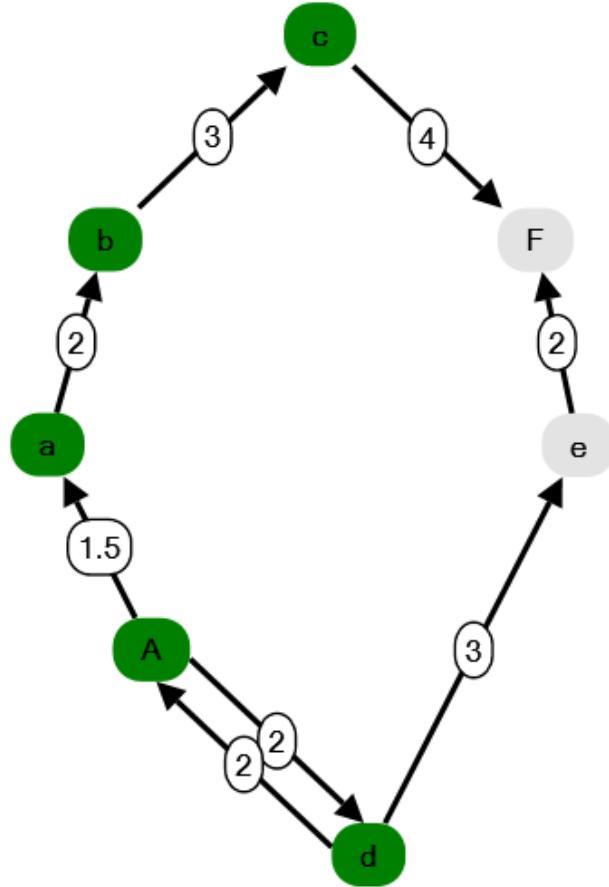
Pre-Visit

The algorithm has opened the node A



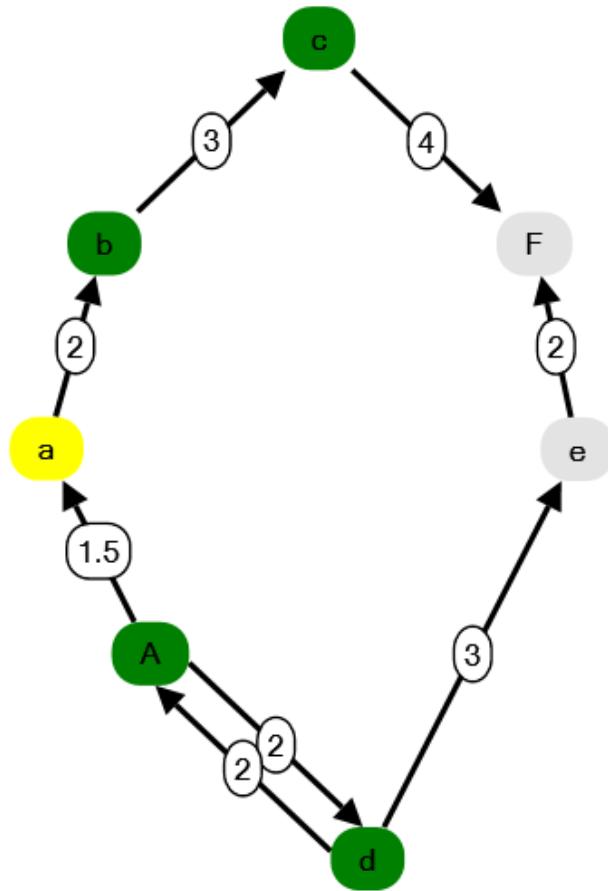
Visit

The algorithm has marked the node A as visited



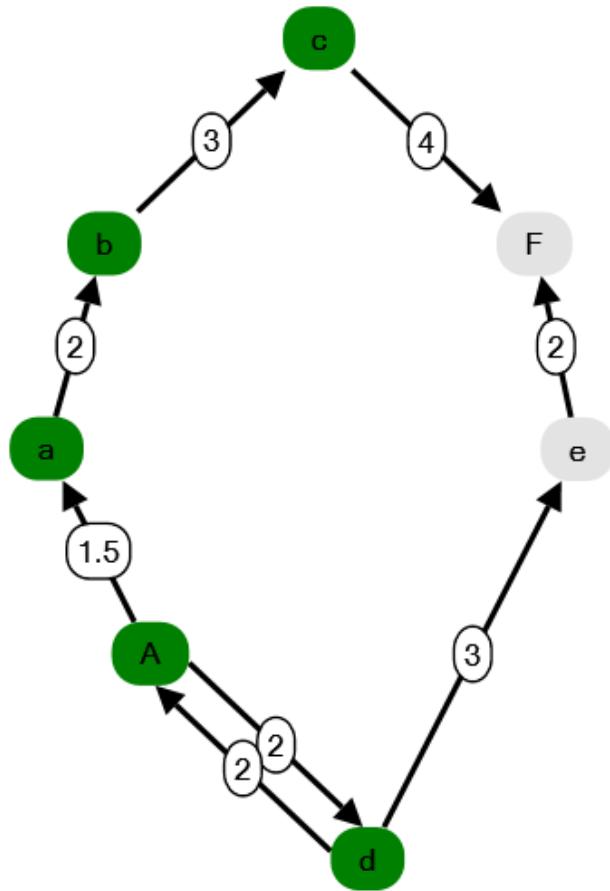
Pre-Visit

The algorithm has opened the node *a*



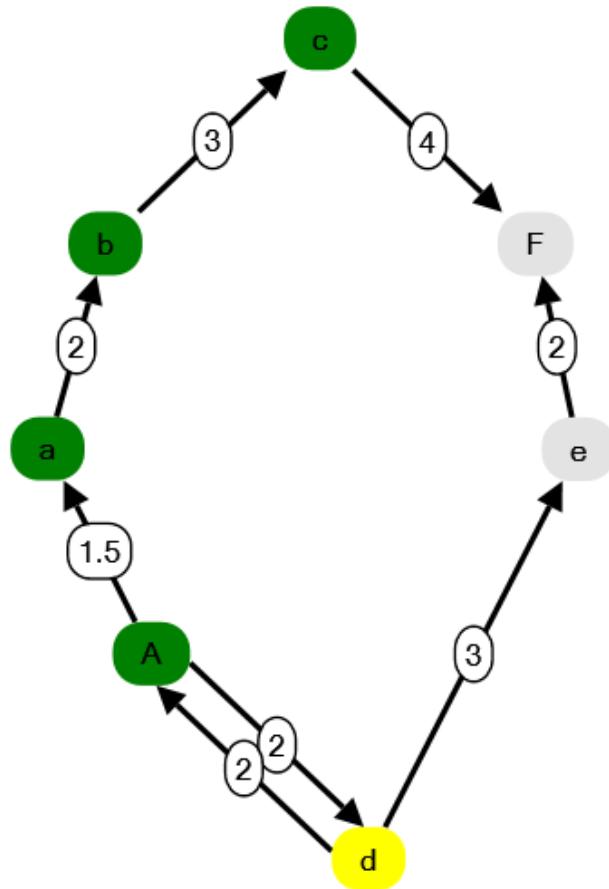
Visit

The algorithm has marked the node *a* as visited



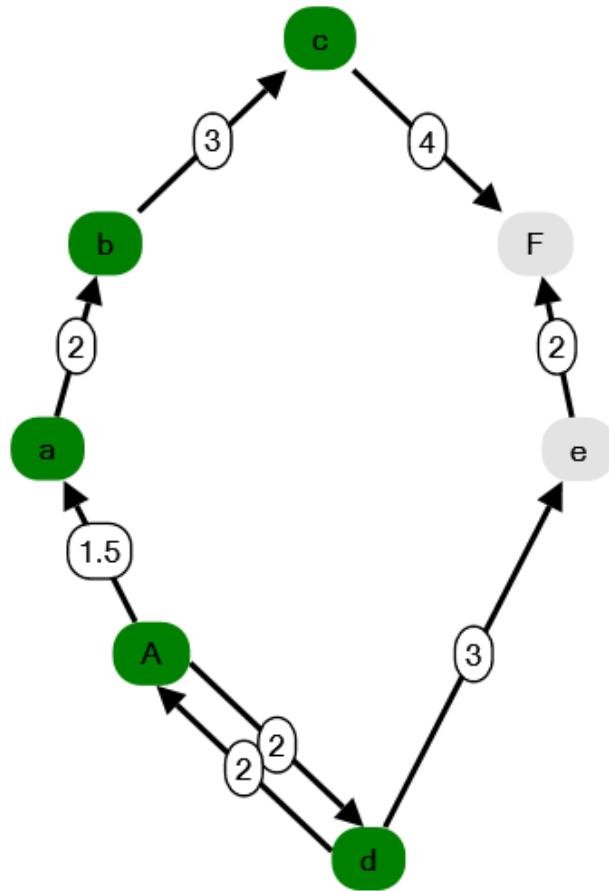
Pre-Visit

The algorithm has opened the node *d*



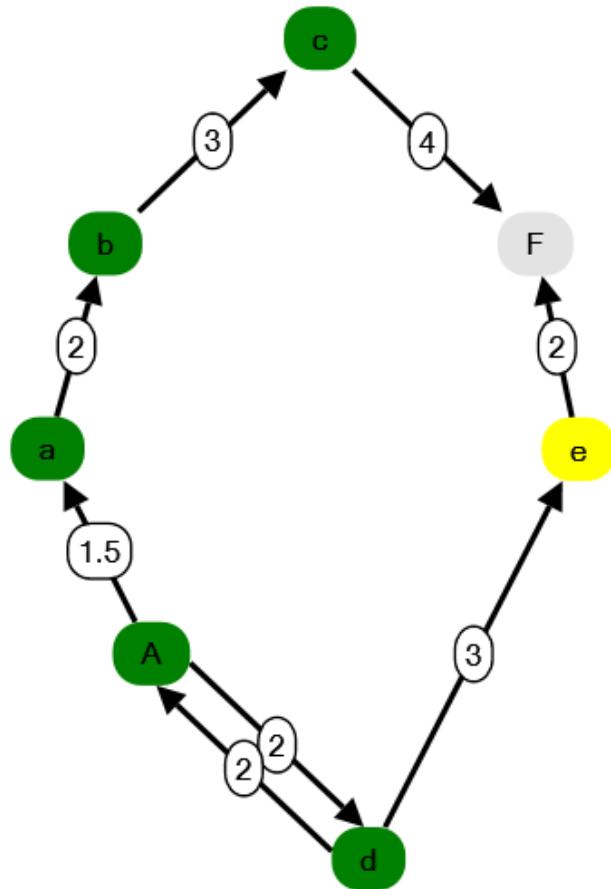
Visit

The algorithm has marked the node d as visited



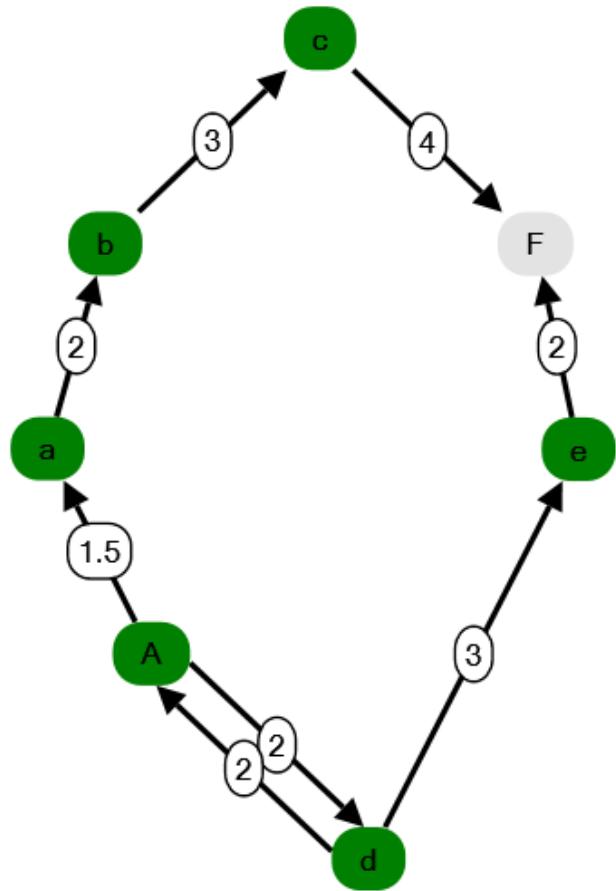
Pre-Visit

The algorithm has opened the node *e*



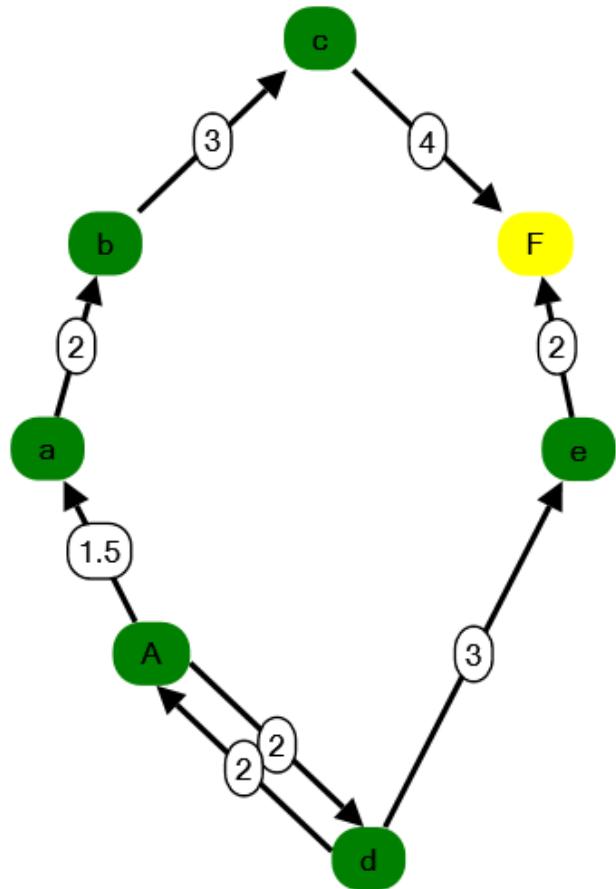
Visit

The algorithm has marked the node e as visited



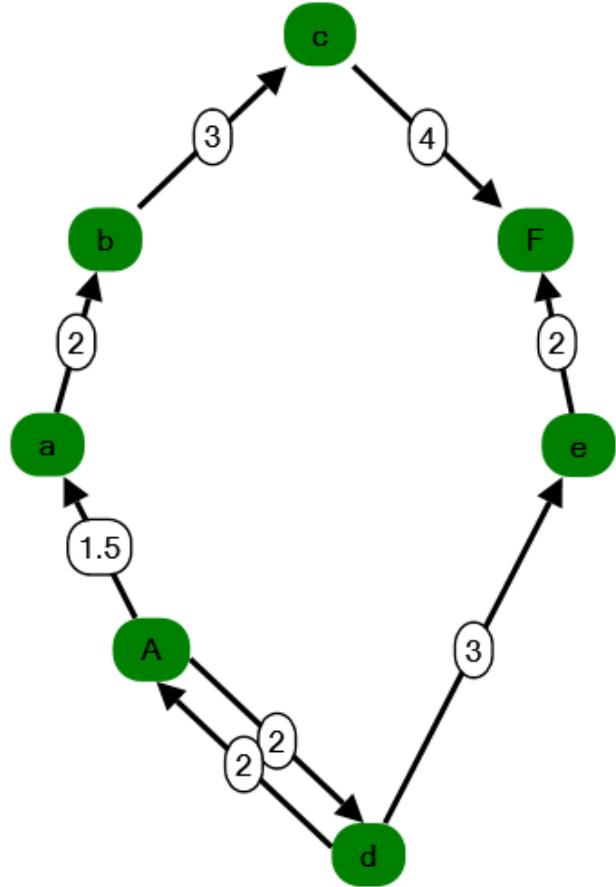
Pre-Visit

The algorithm has opened the node F



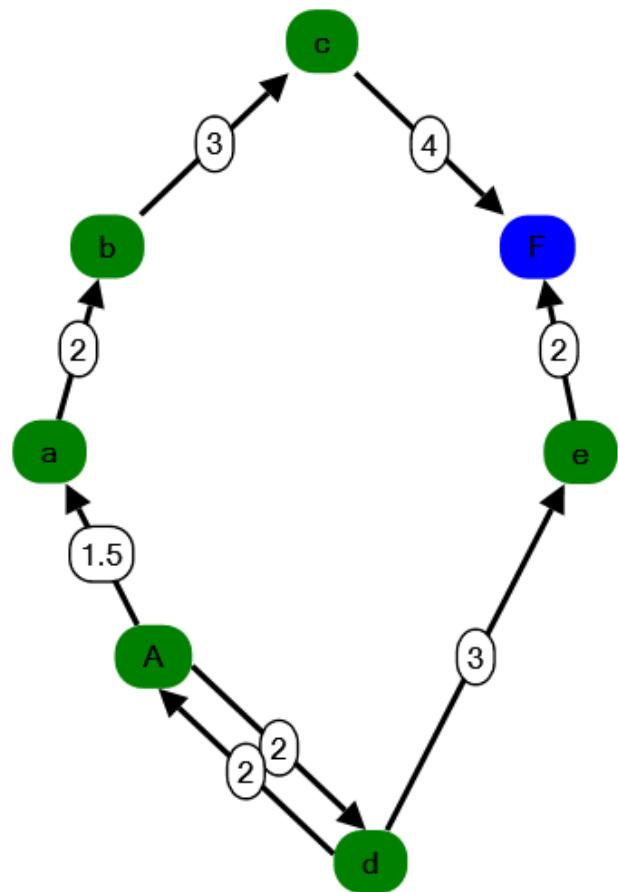
Visit

The algorithm has marked the node F as visited



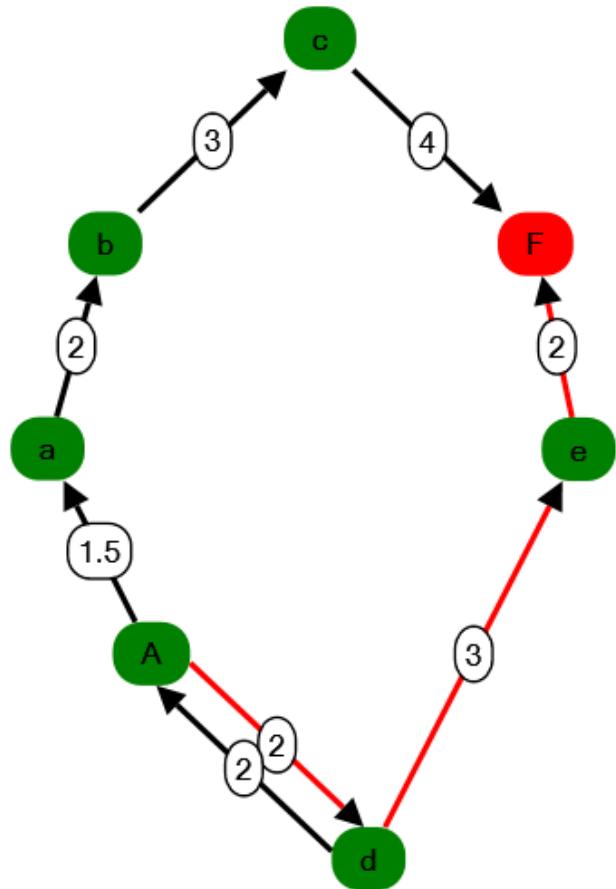
Post-Visit

The algorithm has closed the node F



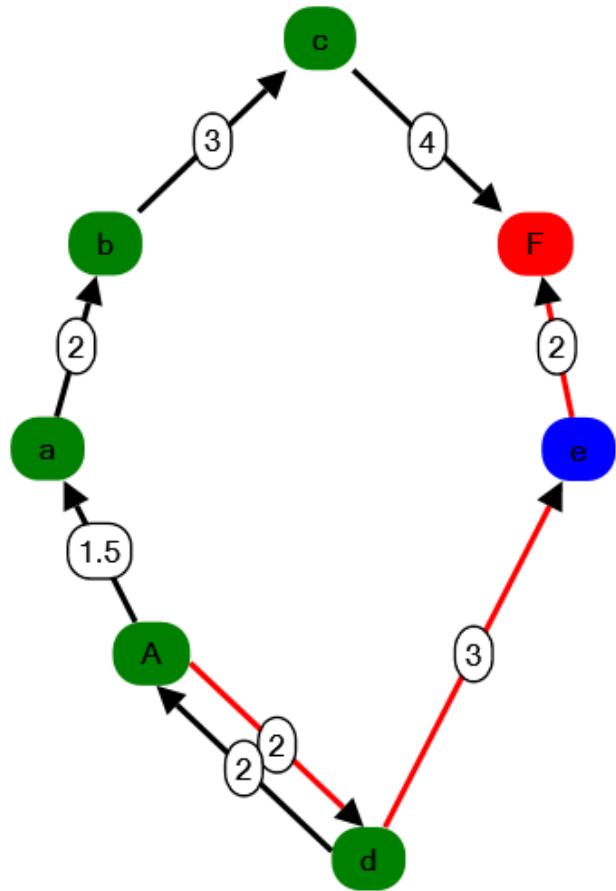
Result Found

The algorithm has found the target node F



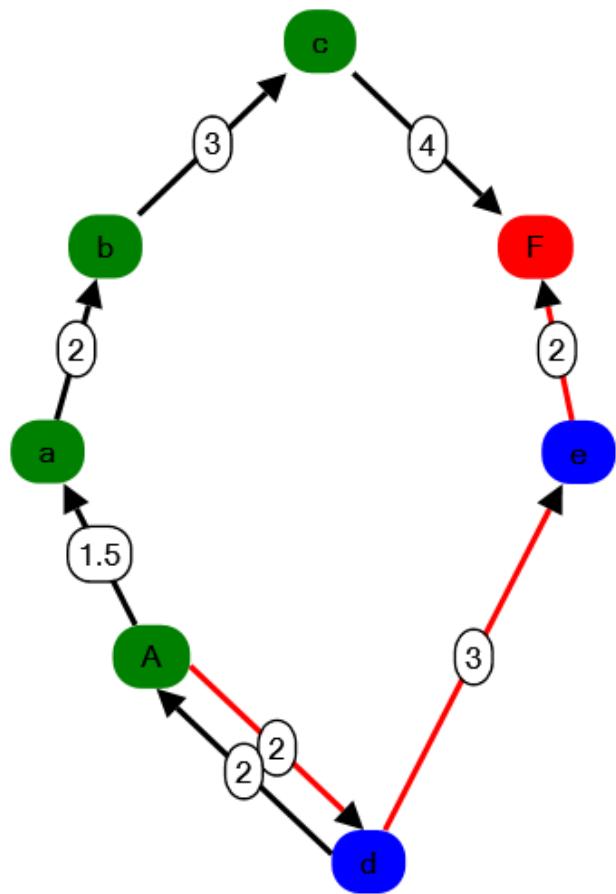
Post-Visit

The algorithm has closed the node e



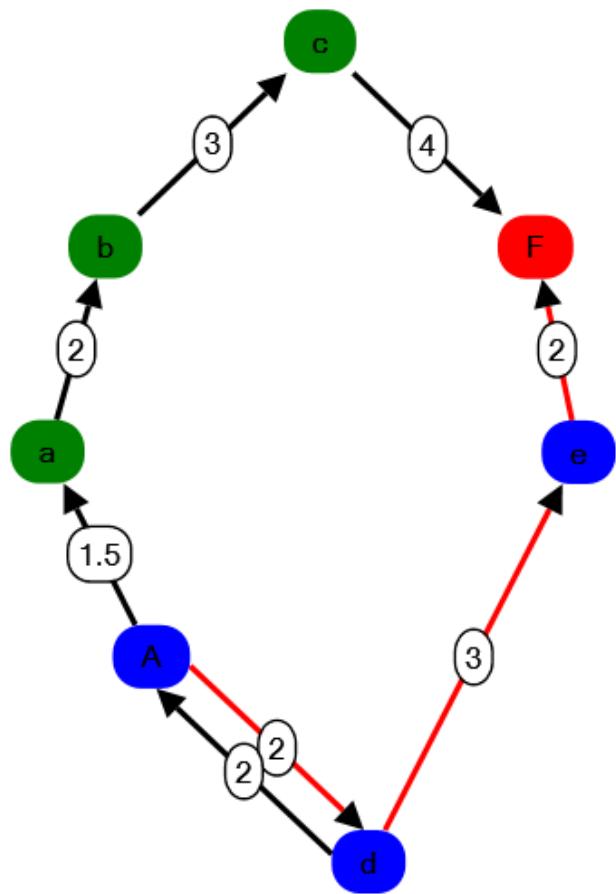
Post-Visit

The algorithm has closed the node d



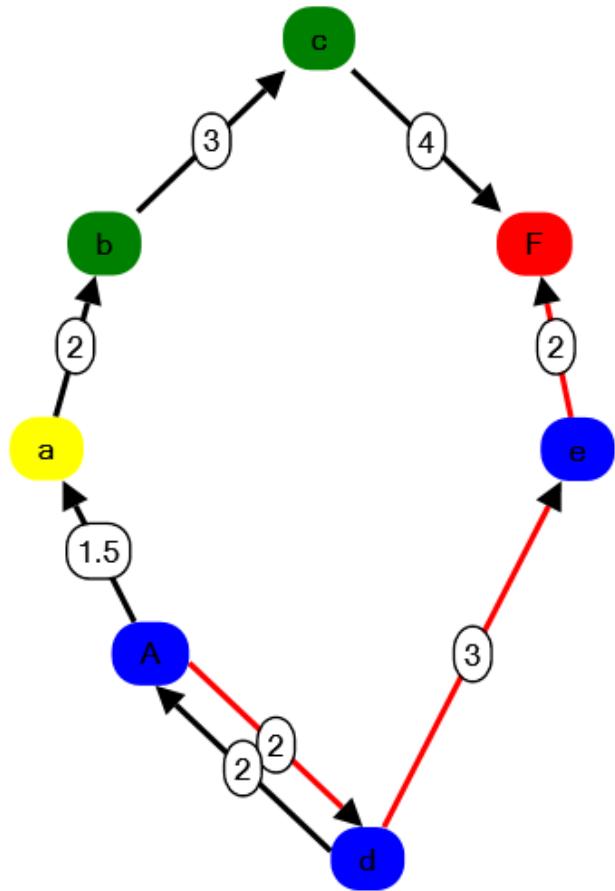
Post-Visit

The algorithm has closed the node A



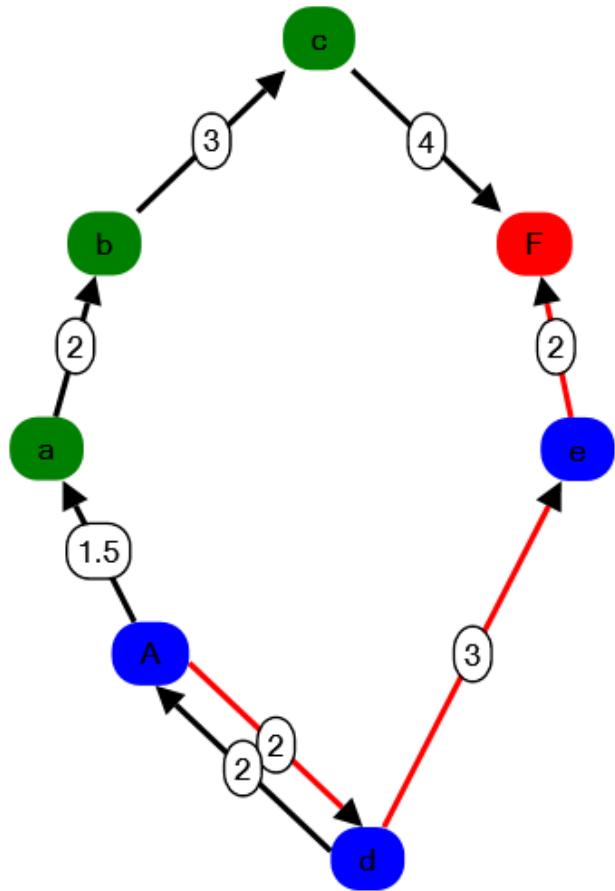
Pre-Visit

The algorithm has opened the node *a*



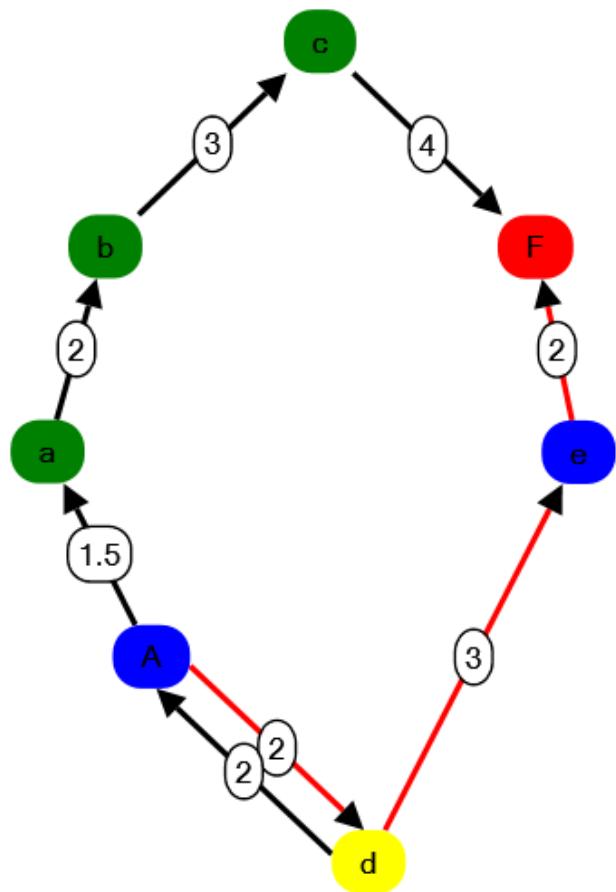
Visit

The algorithm has marked the node *a* as visited



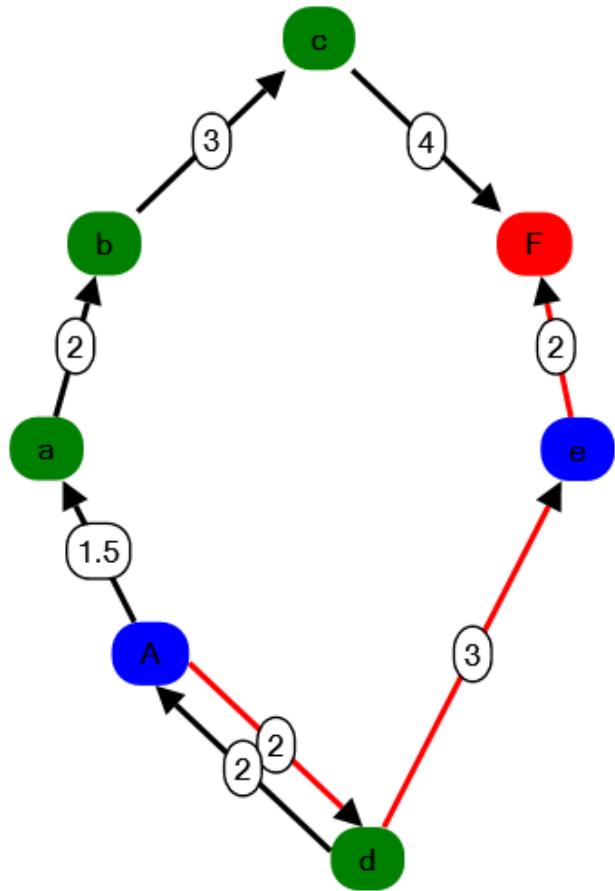
Pre-Visit

The algorithm has opened the node *d*



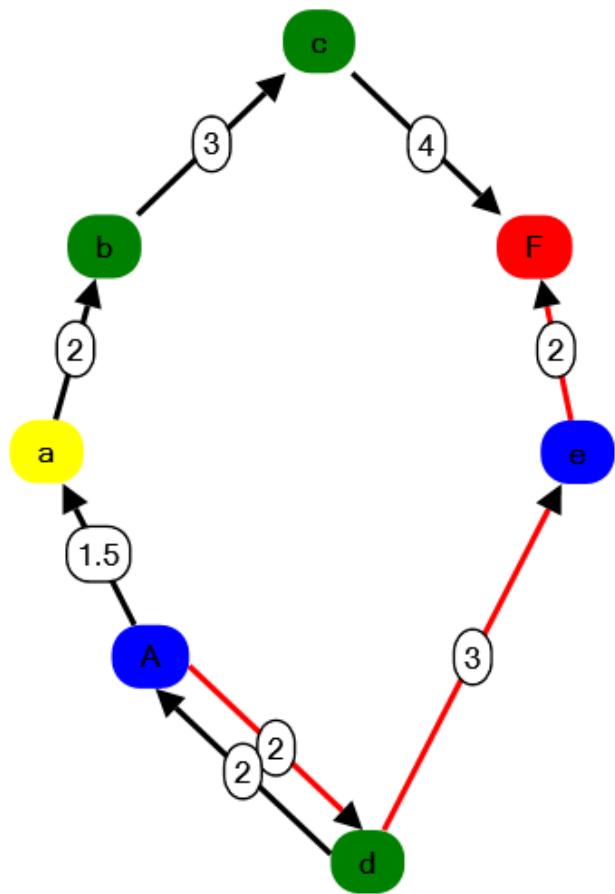
Visit

The algorithm has marked the node d as visited



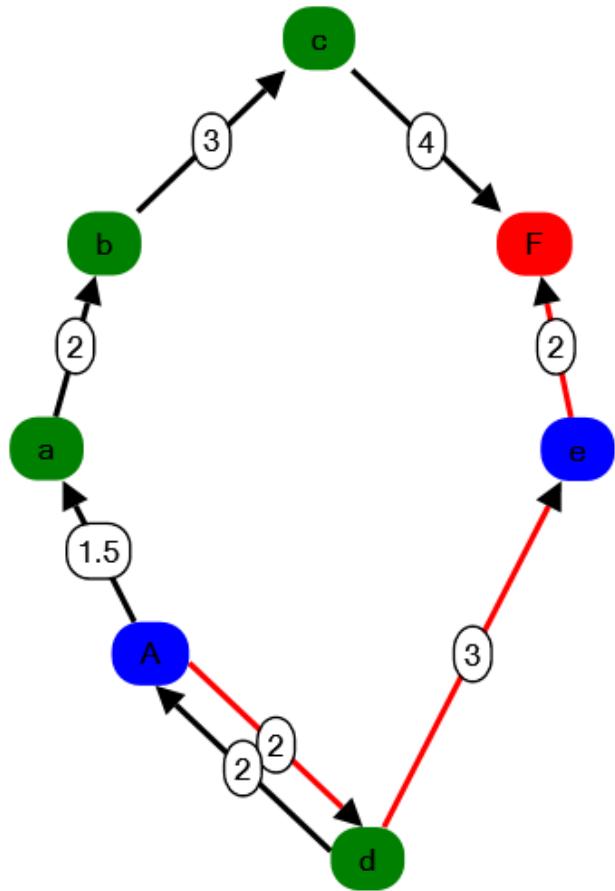
Pre-Visit

The algorithm has opened the node *a*



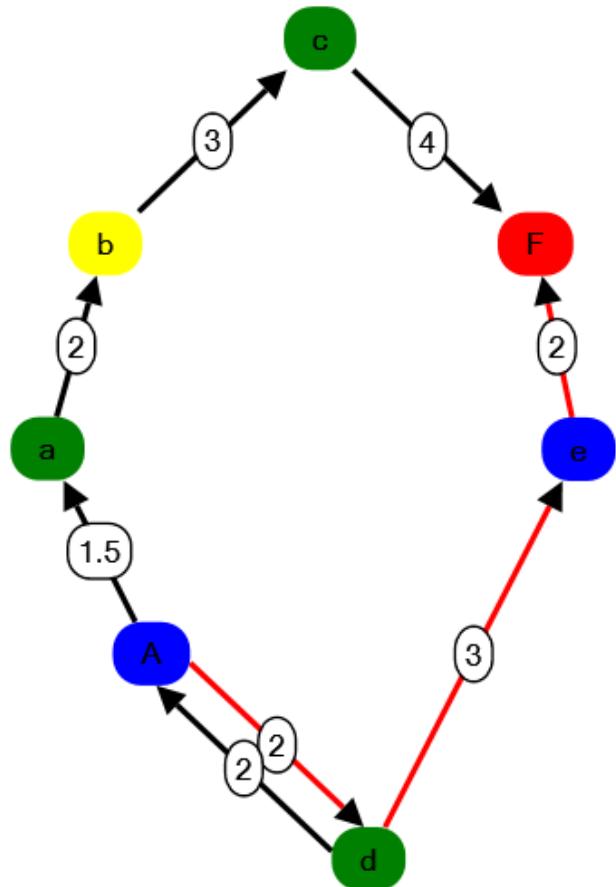
Visit

The algorithm has marked the node *a* as visited



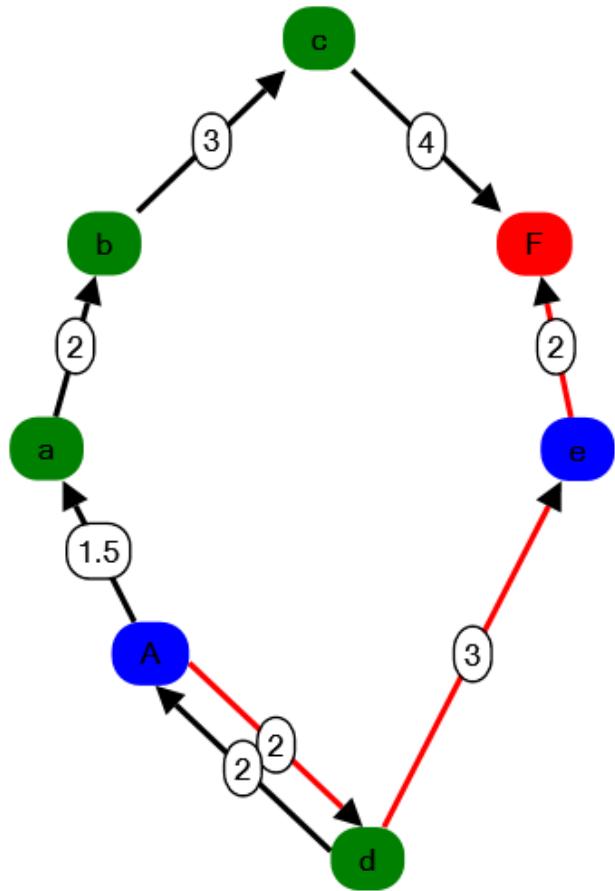
Pre-Visit

The algorithm has opened the node *b*



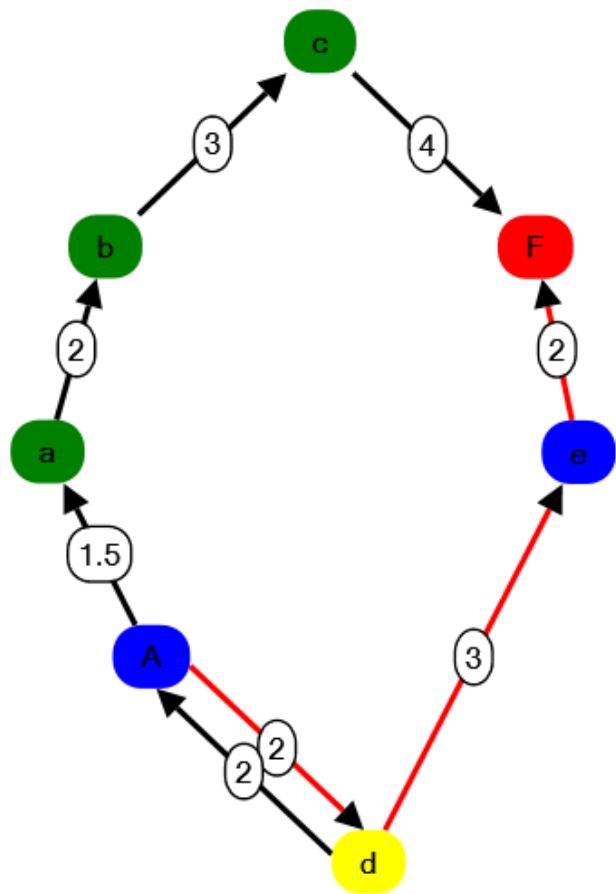
Visit

The algorithm has marked the node b as visited



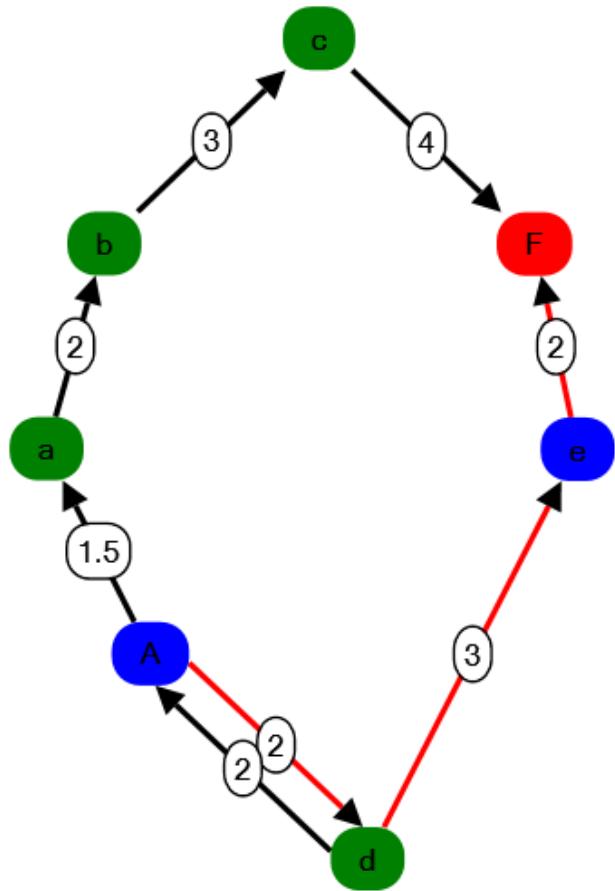
Pre-Visit

The algorithm has opened the node *d*



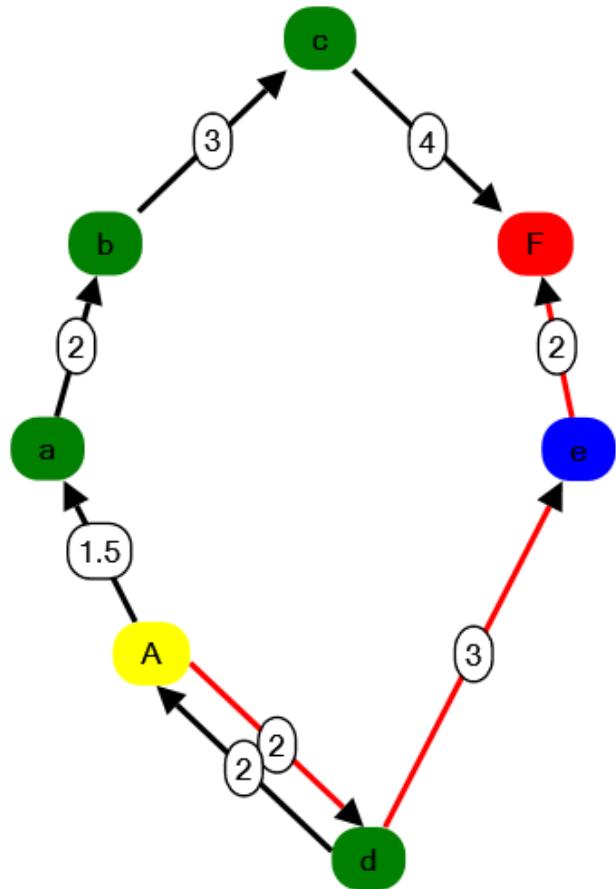
Visit

The algorithm has marked the node d as visited



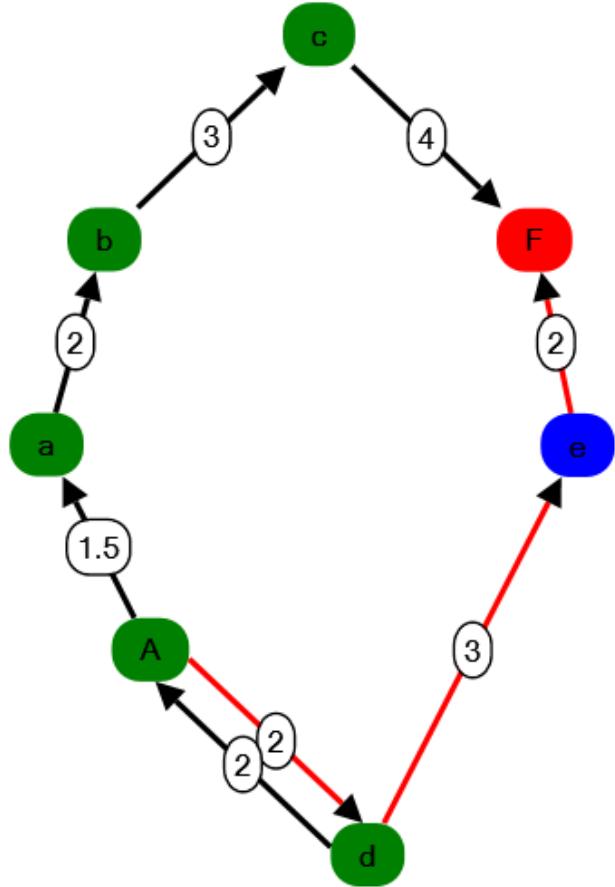
Pre-Visit

The algorithm has opened the node A



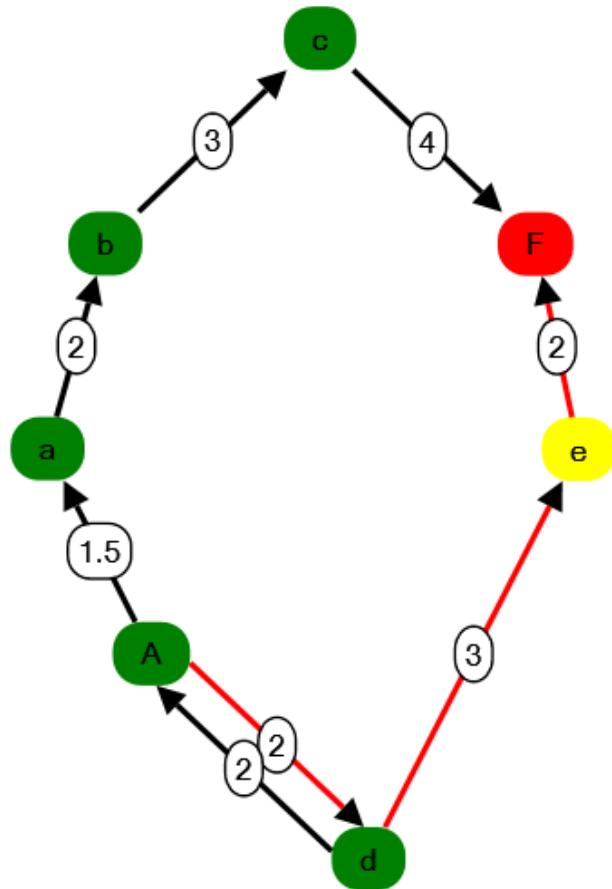
Visit

The algorithm has marked the node A as visited



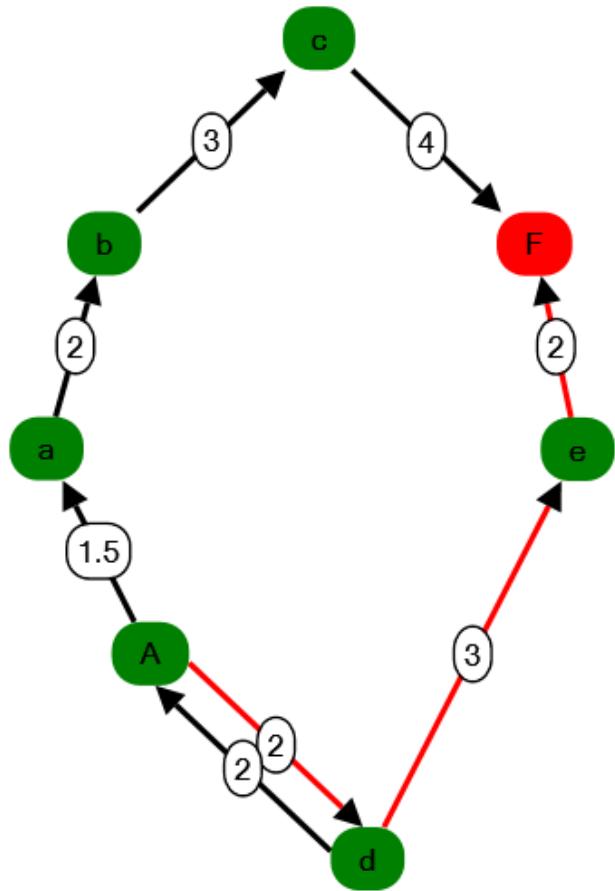
Pre-Visit

The algorithm has opened the node *e*



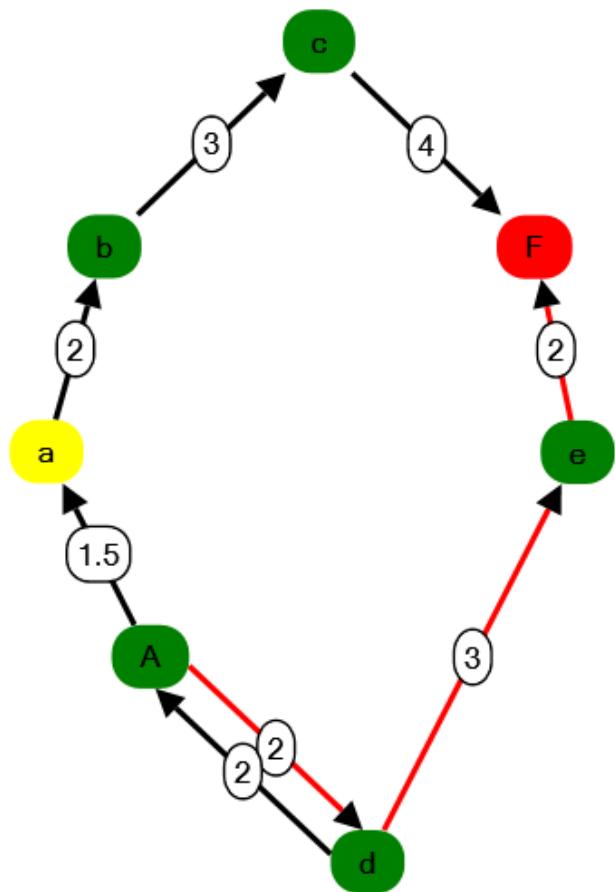
Visit

The algorithm has marked the node e as visited



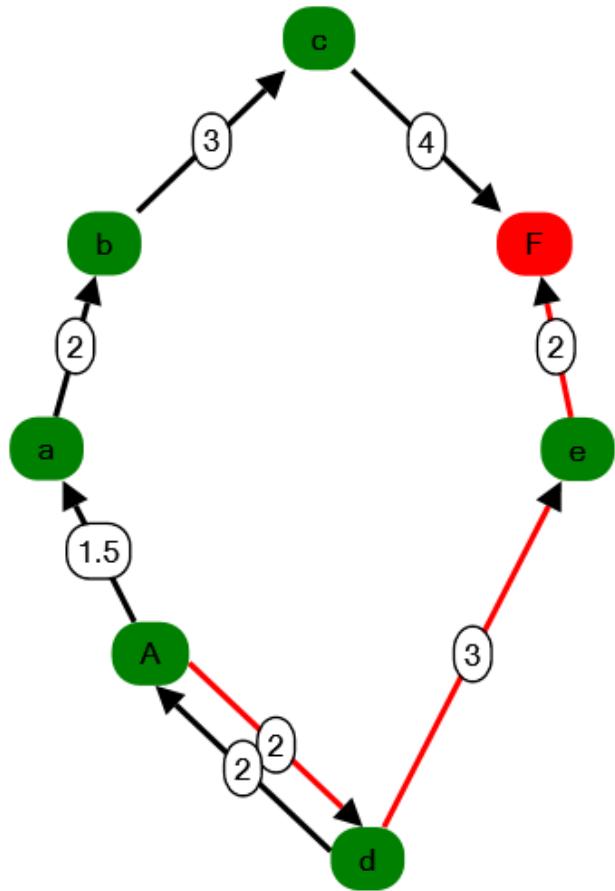
Pre-Visit

The algorithm has opened the node *a*



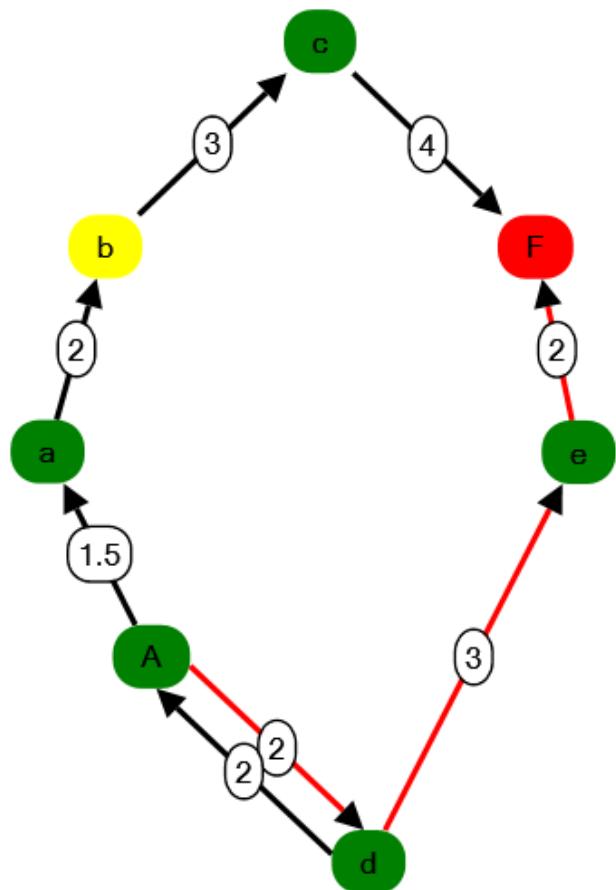
Visit

The algorithm has marked the node *a* as visited



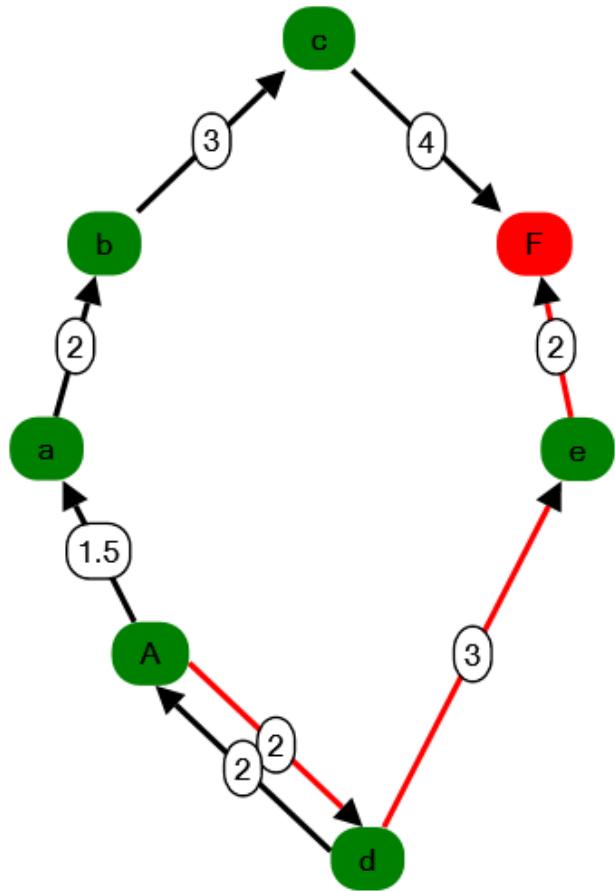
Pre-Visit

The algorithm has opened the node *b*



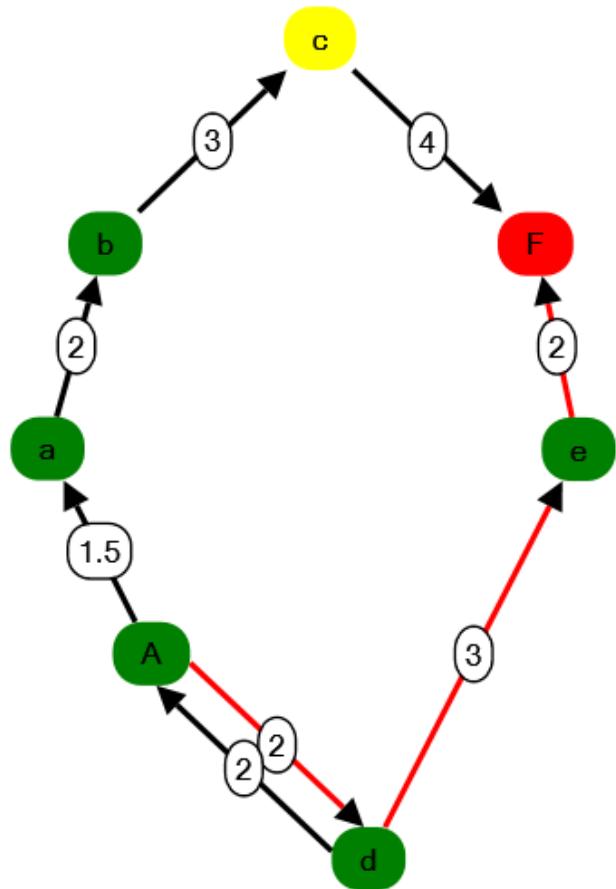
Visit

The algorithm has marked the node b as visited



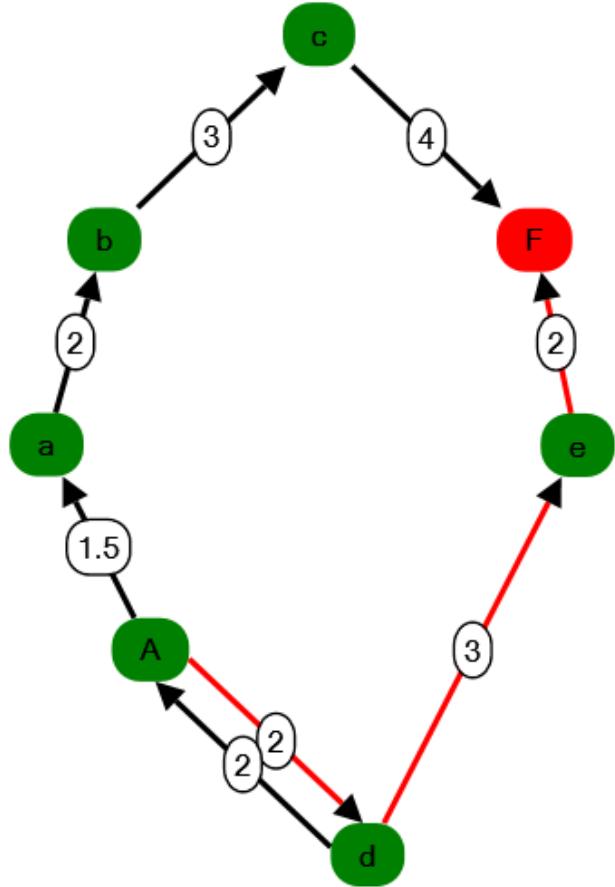
Pre-Visit

The algorithm has opened the node *c*



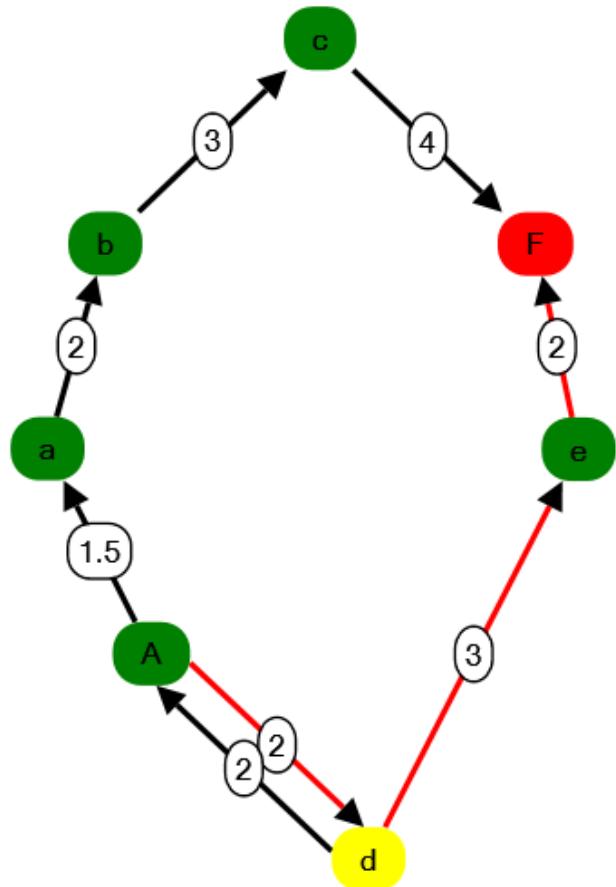
Visit

The algorithm has marked the node c as visited



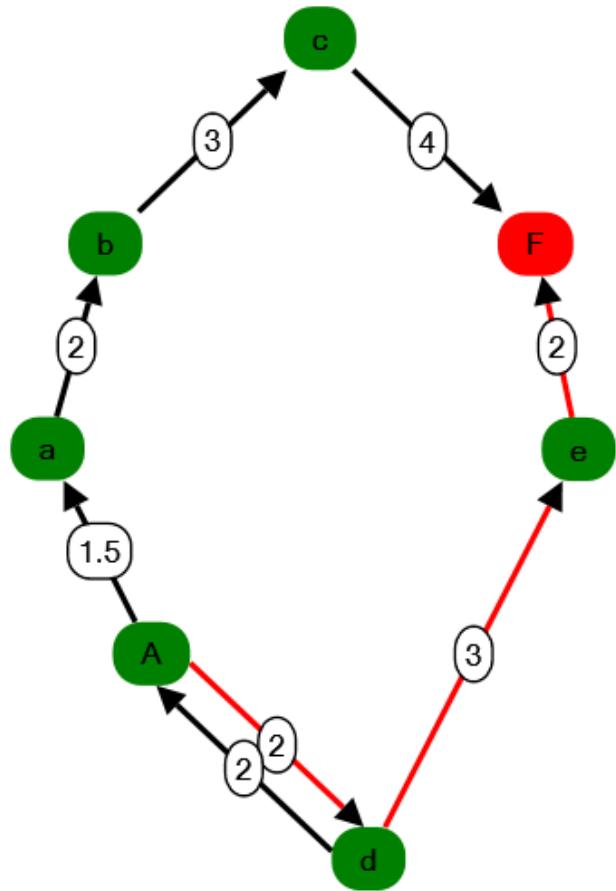
Pre-Visit

The algorithm has opened the node *d*



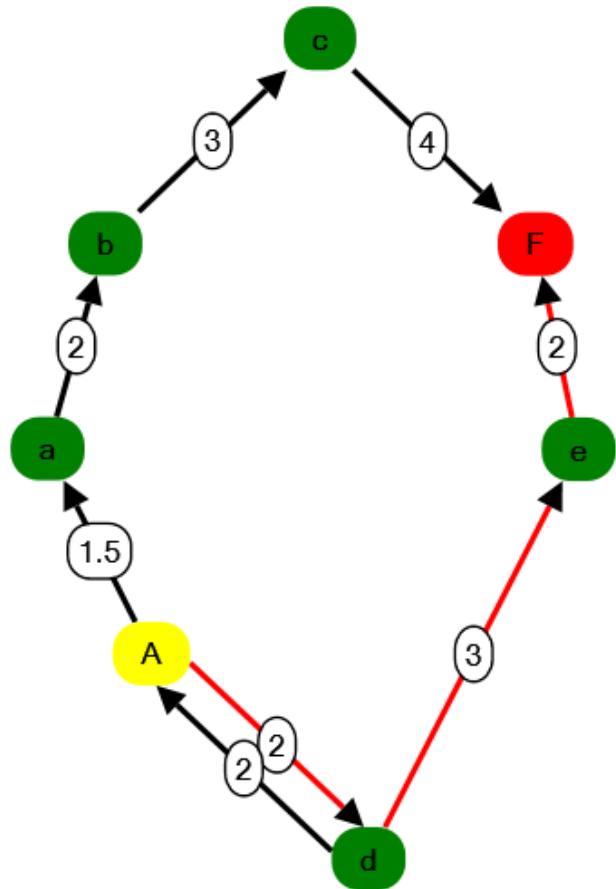
Visit

The algorithm has marked the node d as visited



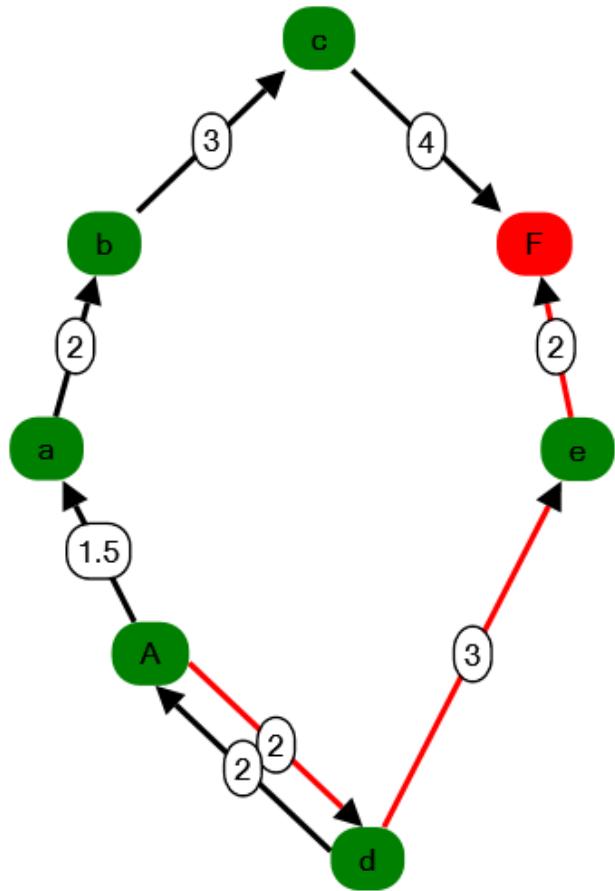
Pre-Visit

The algorithm has opened the node A



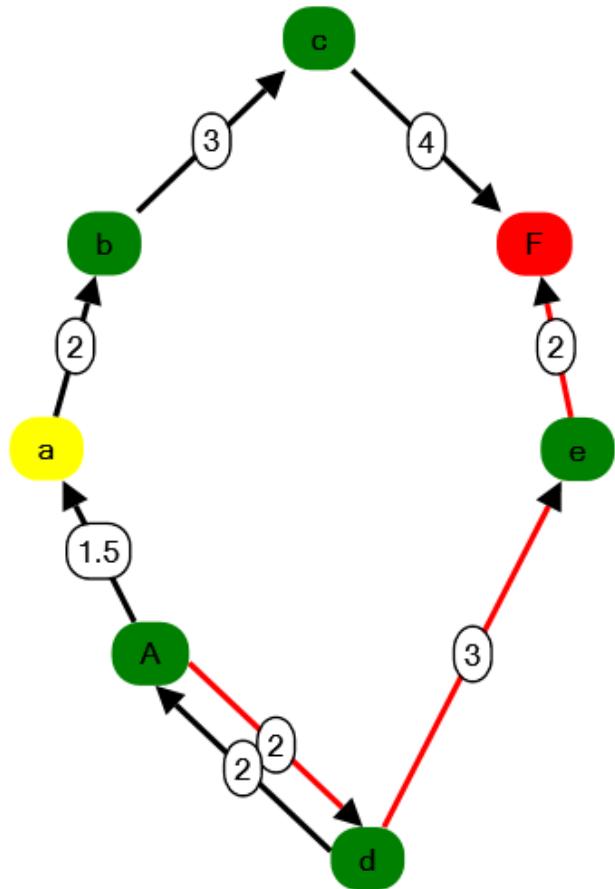
Visit

The algorithm has marked the node A as visited



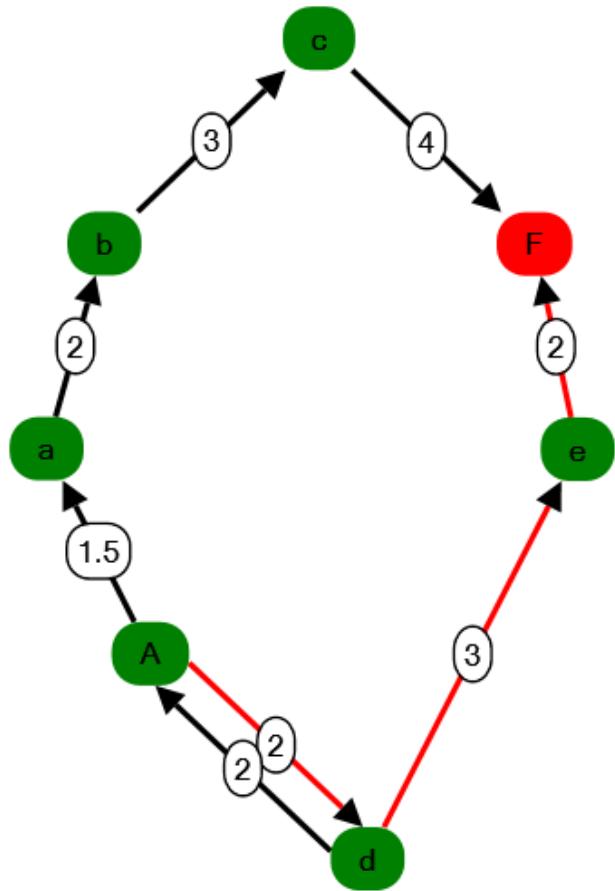
Pre-Visit

The algorithm has opened the node *a*



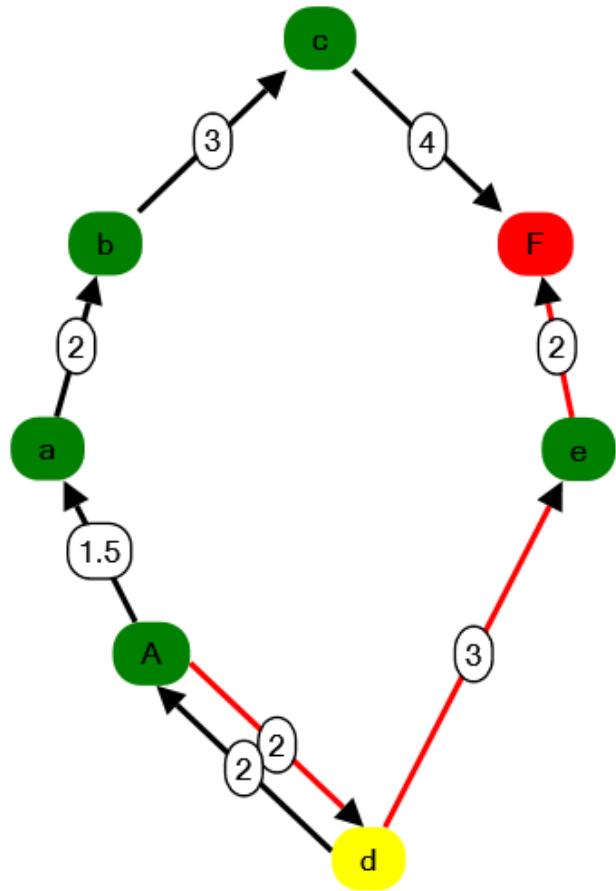
Visit

The algorithm has marked the node *a* as visited



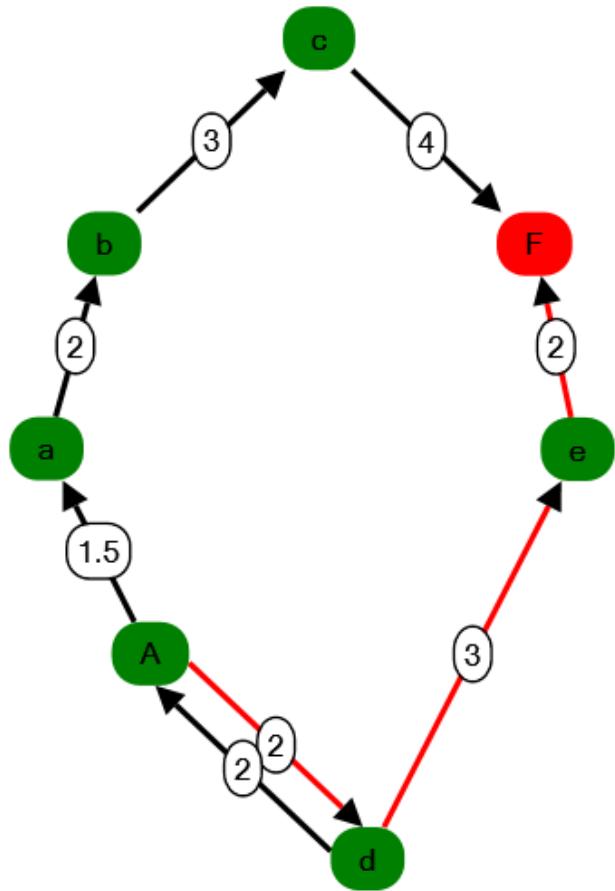
Pre-Visit

The algorithm has opened the node *d*



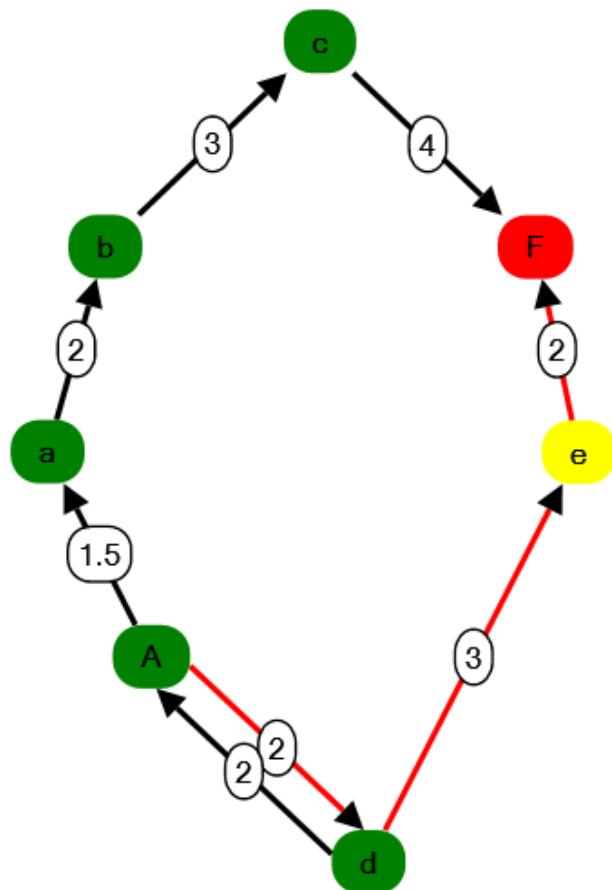
Visit

The algorithm has marked the node d as visited



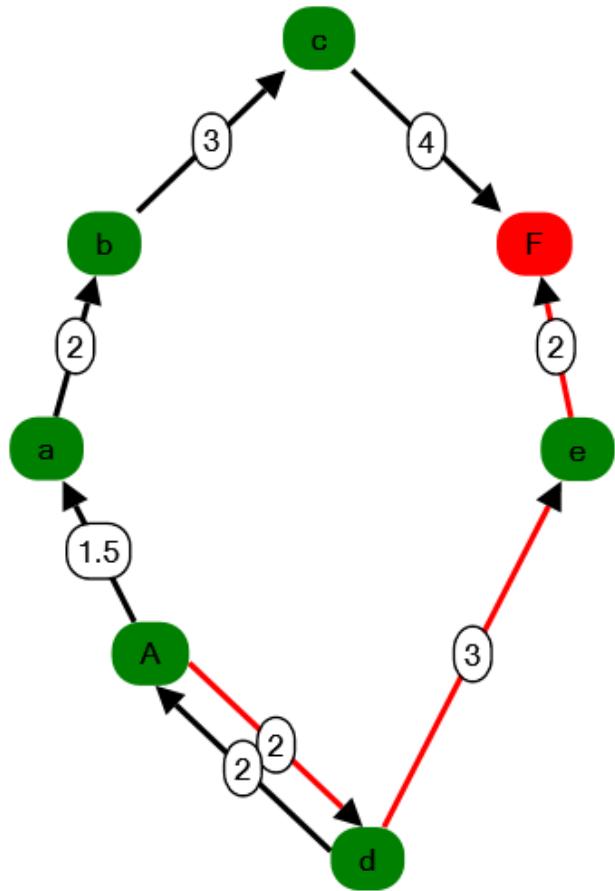
Pre-Visit

The algorithm has opened the node *e*



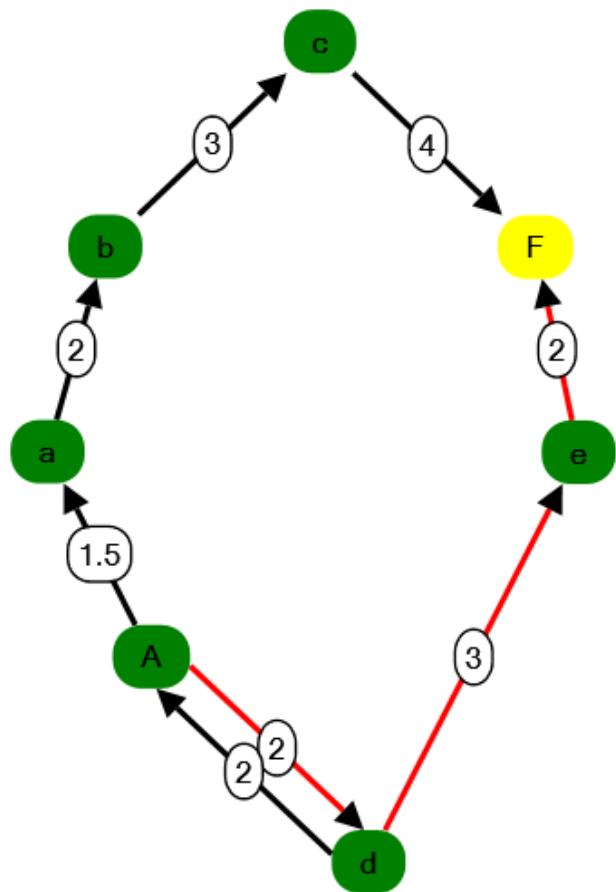
Visit

The algorithm has marked the node e as visited



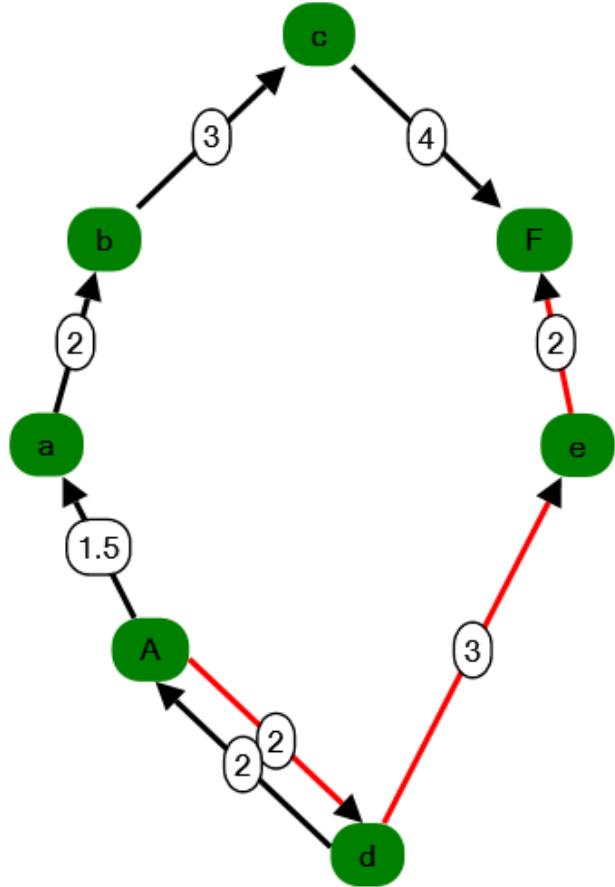
Pre-Visit

The algorithm has opened the node F



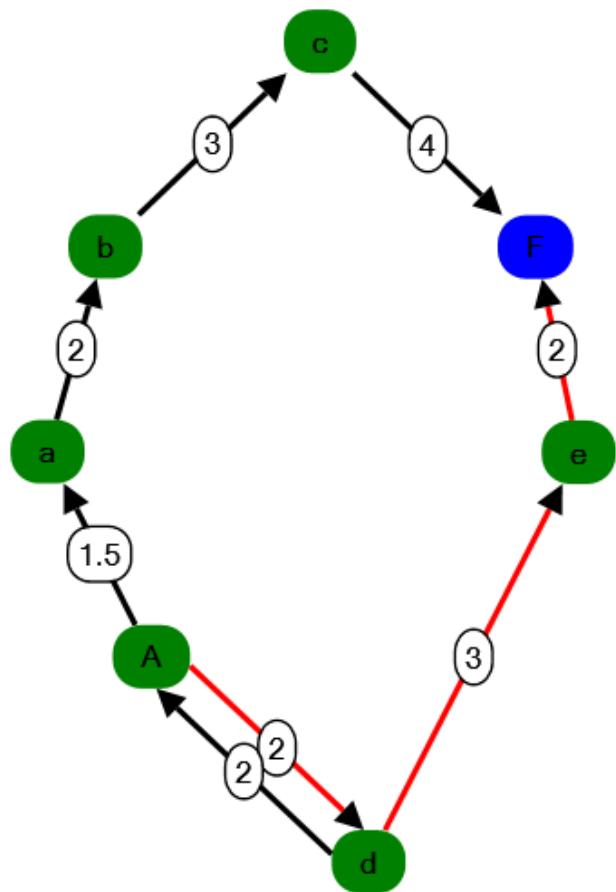
Visit

The algorithm has marked the node F as visited



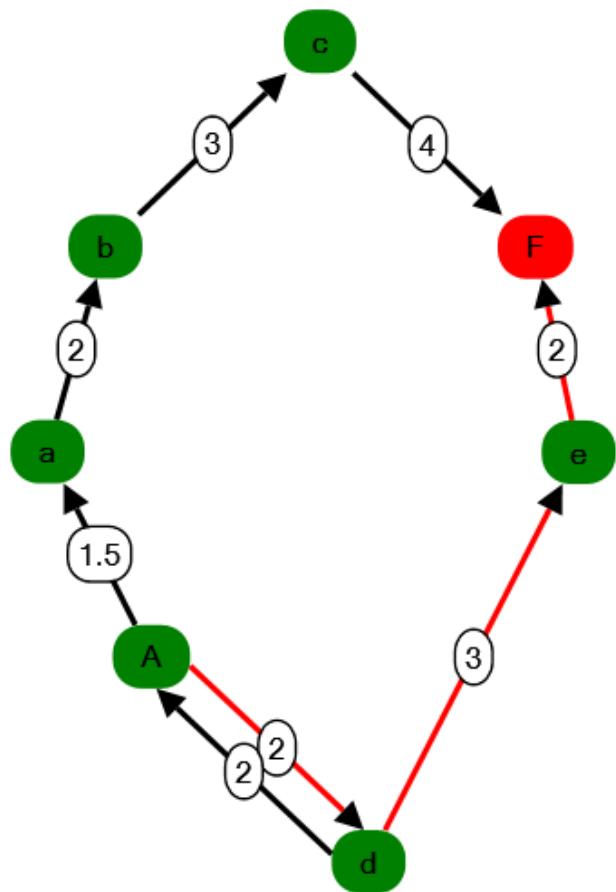
Post-Visit

The algorithm has closed the node F



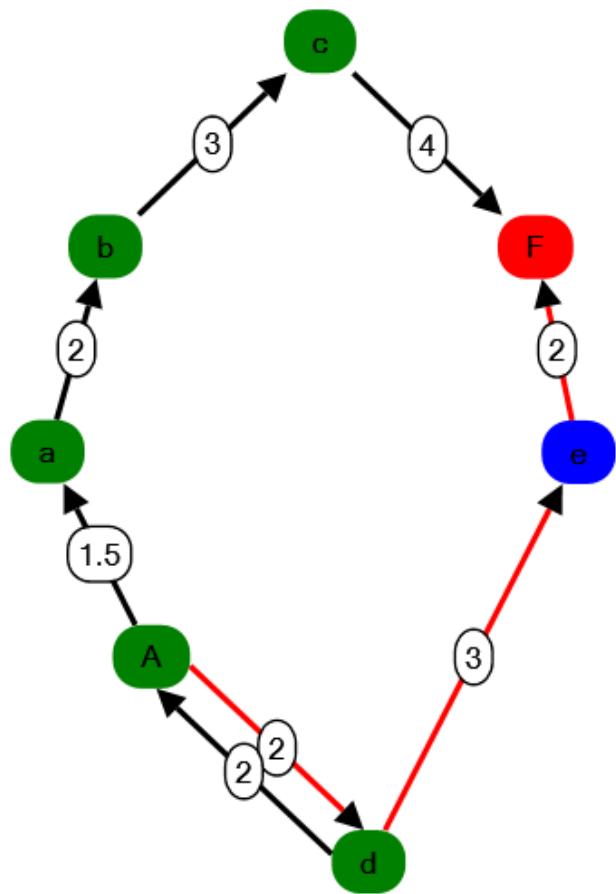
Result Found

The algorithm has found the target node F



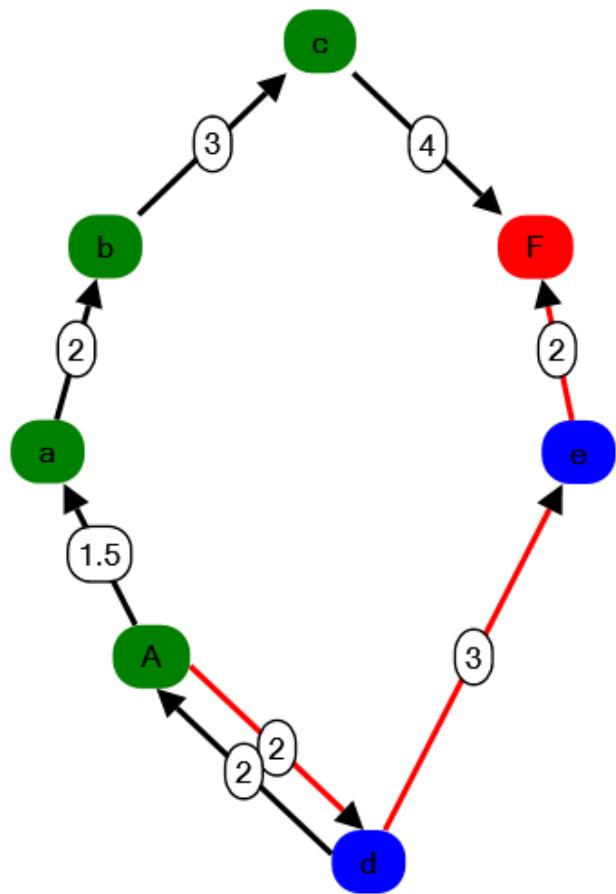
Post-Visit

The algorithm has closed the node e



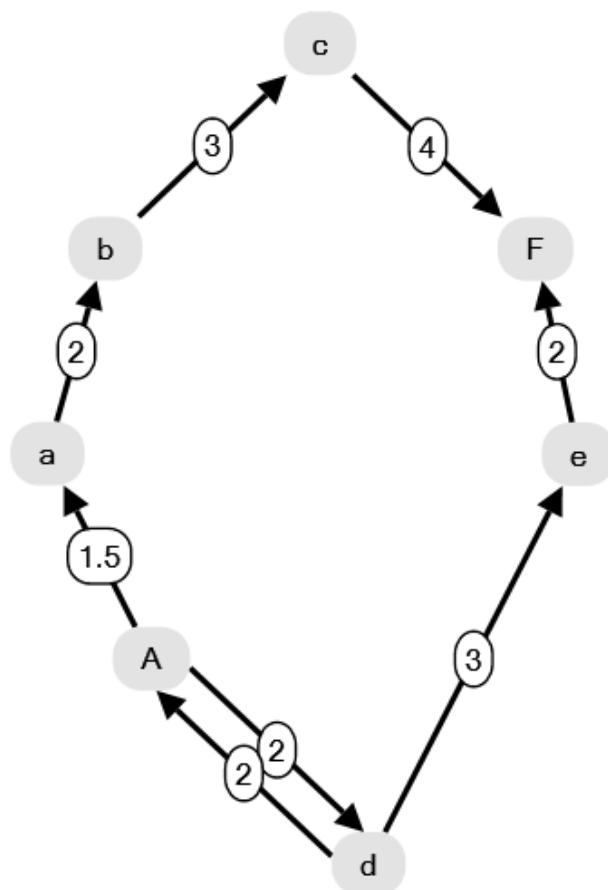
Post-Visit

The algorithm has closed the node d



Reset

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



A*

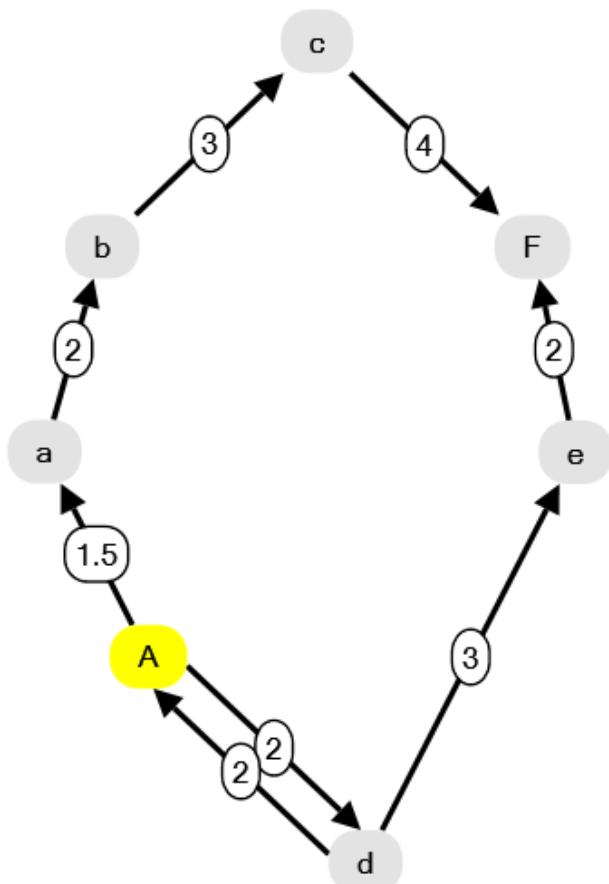
In computer science, A* (pronounced as "A star") is a computer algorithm that is widely used in pathfinding and graph traversal, the process of plotting an efficiently directed path between multiple points, called nodes. It enjoys widespread use due to its performance and accuracy. However, in practical 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 00:00:00.0012005 to find the goal.

The result path costs 7

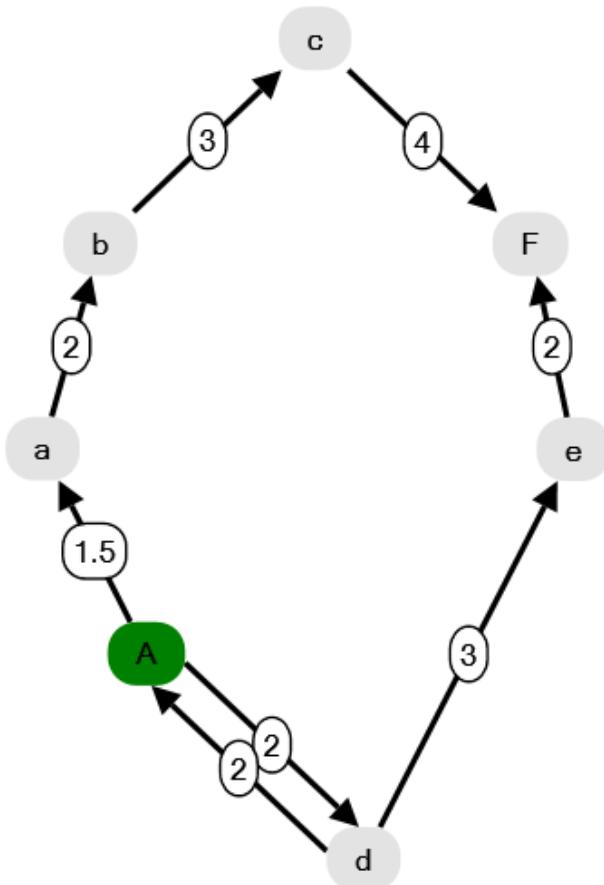
Pre-Visit

The algorithm has opened the node A



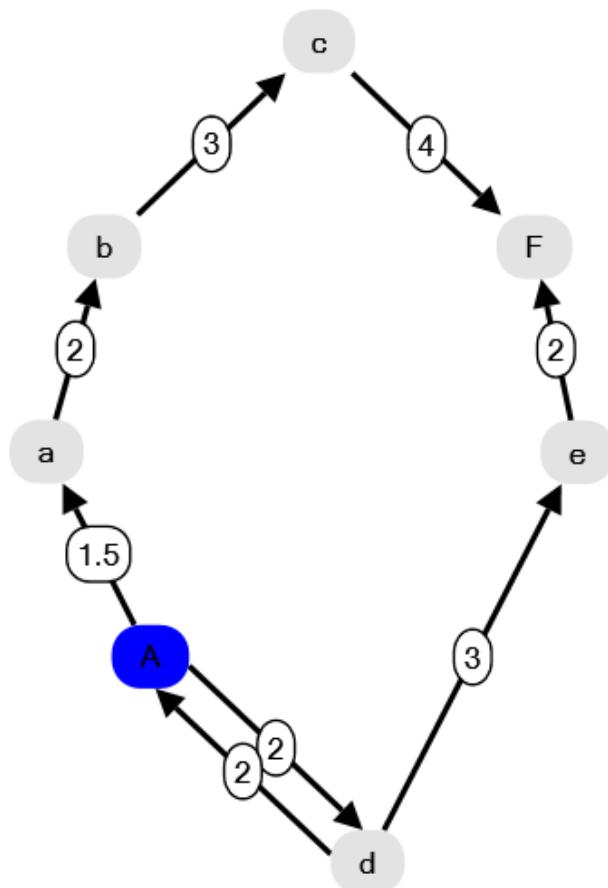
Visit

The algorithm has marked the node A as visited



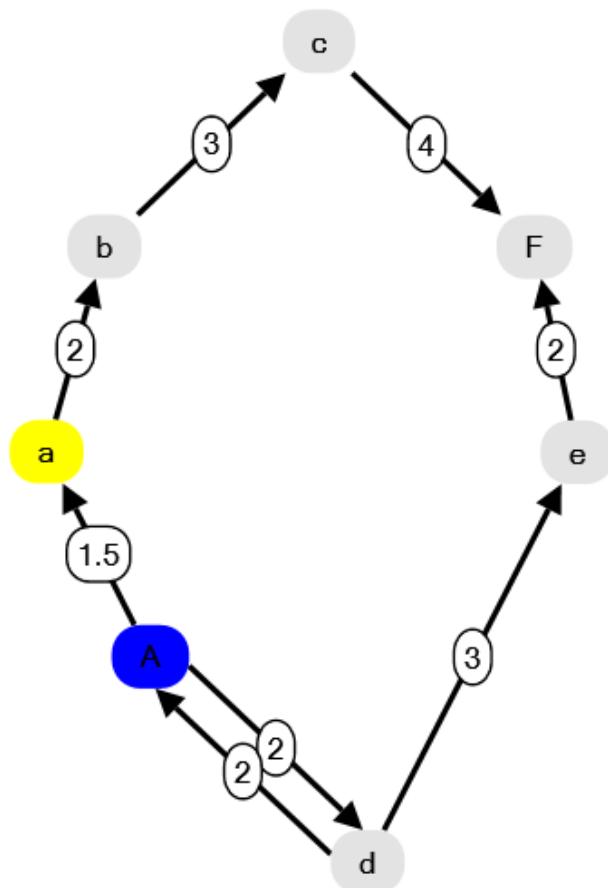
Post-Visit

The algorithm has closed the node A



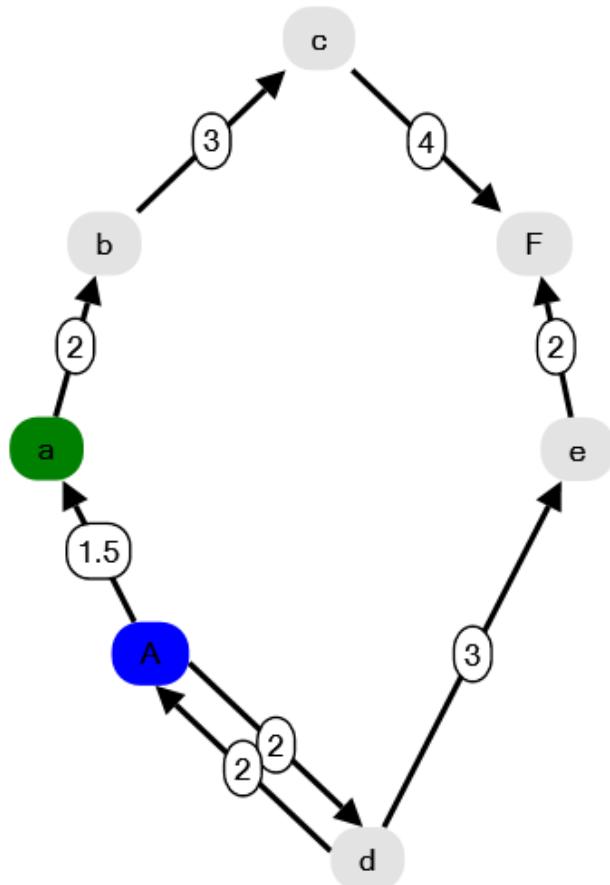
Pre-Visit

The algorithm has opened the node *a*



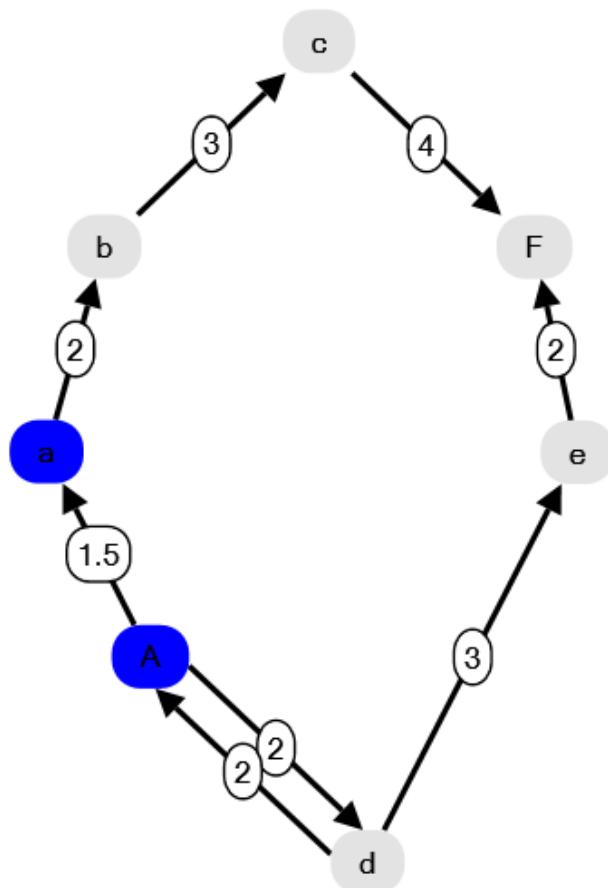
Visit

The algorithm has marked the node *a* as visited



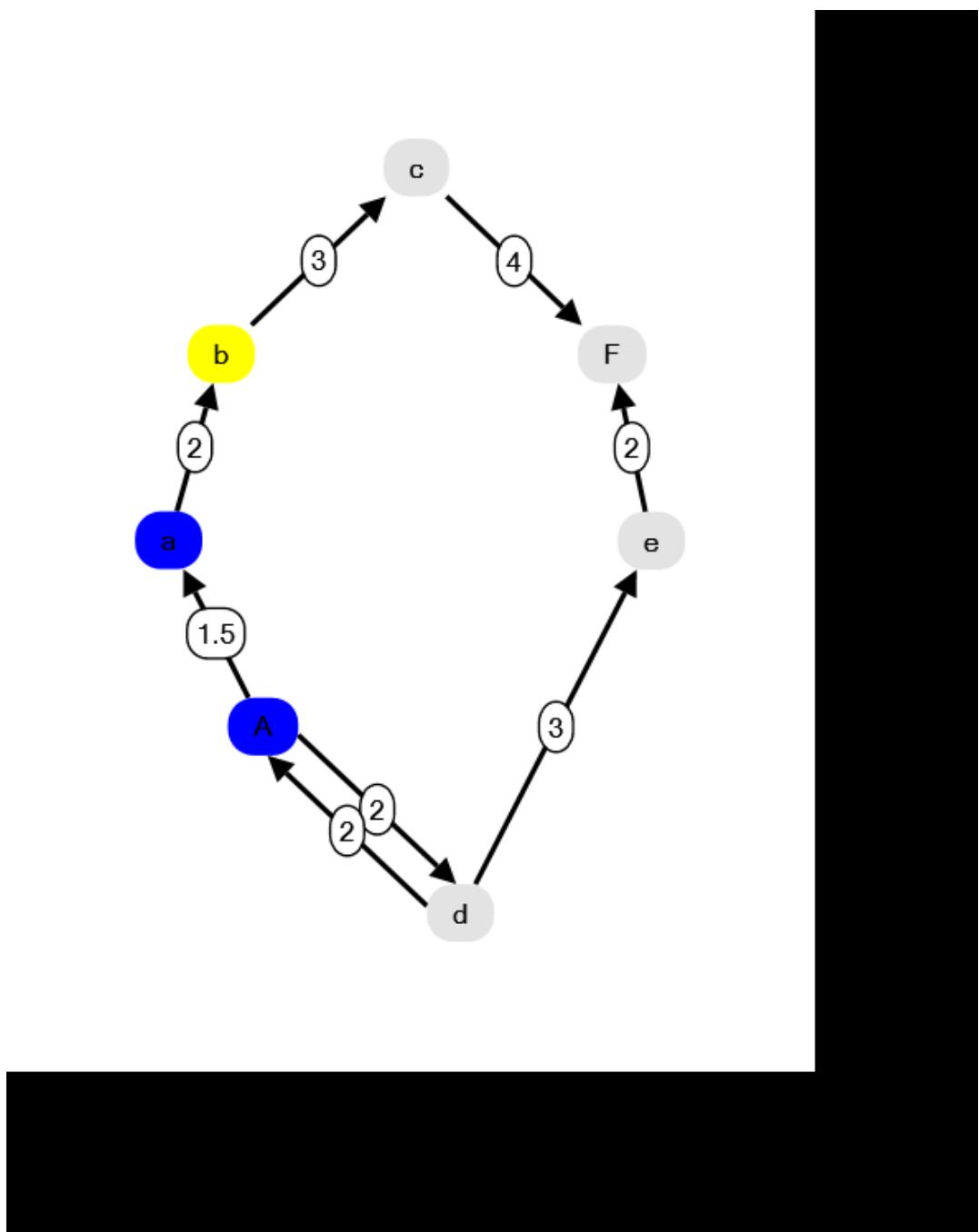
Post-Visit

The algorithm has closed the node *a*



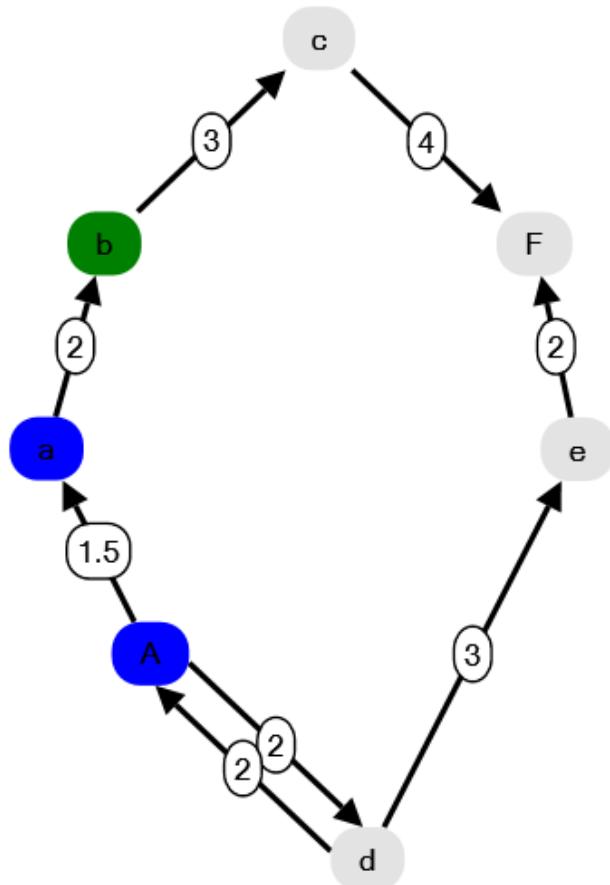
Pre-Visit

The algorithm has opened the node *b*



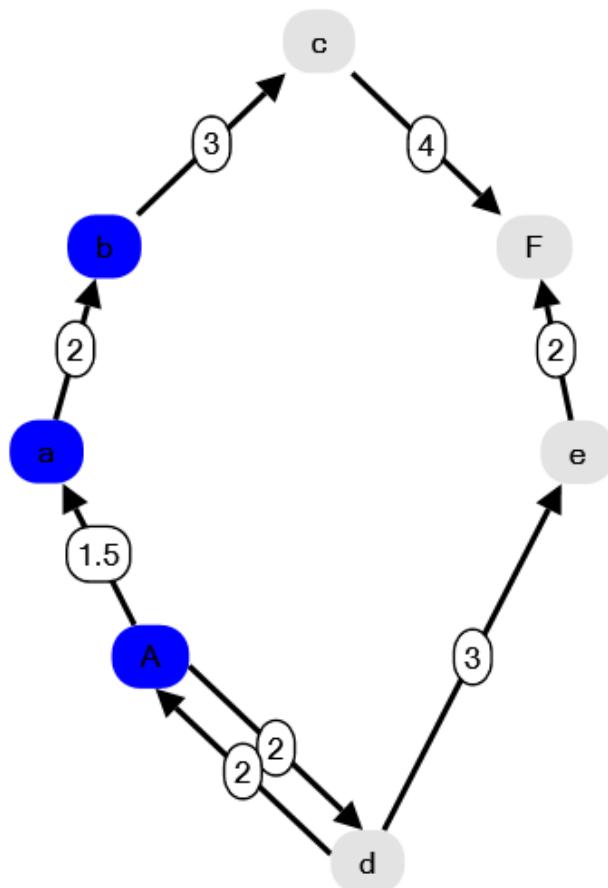
Visit

The algorithm has marked the node *b* as visited



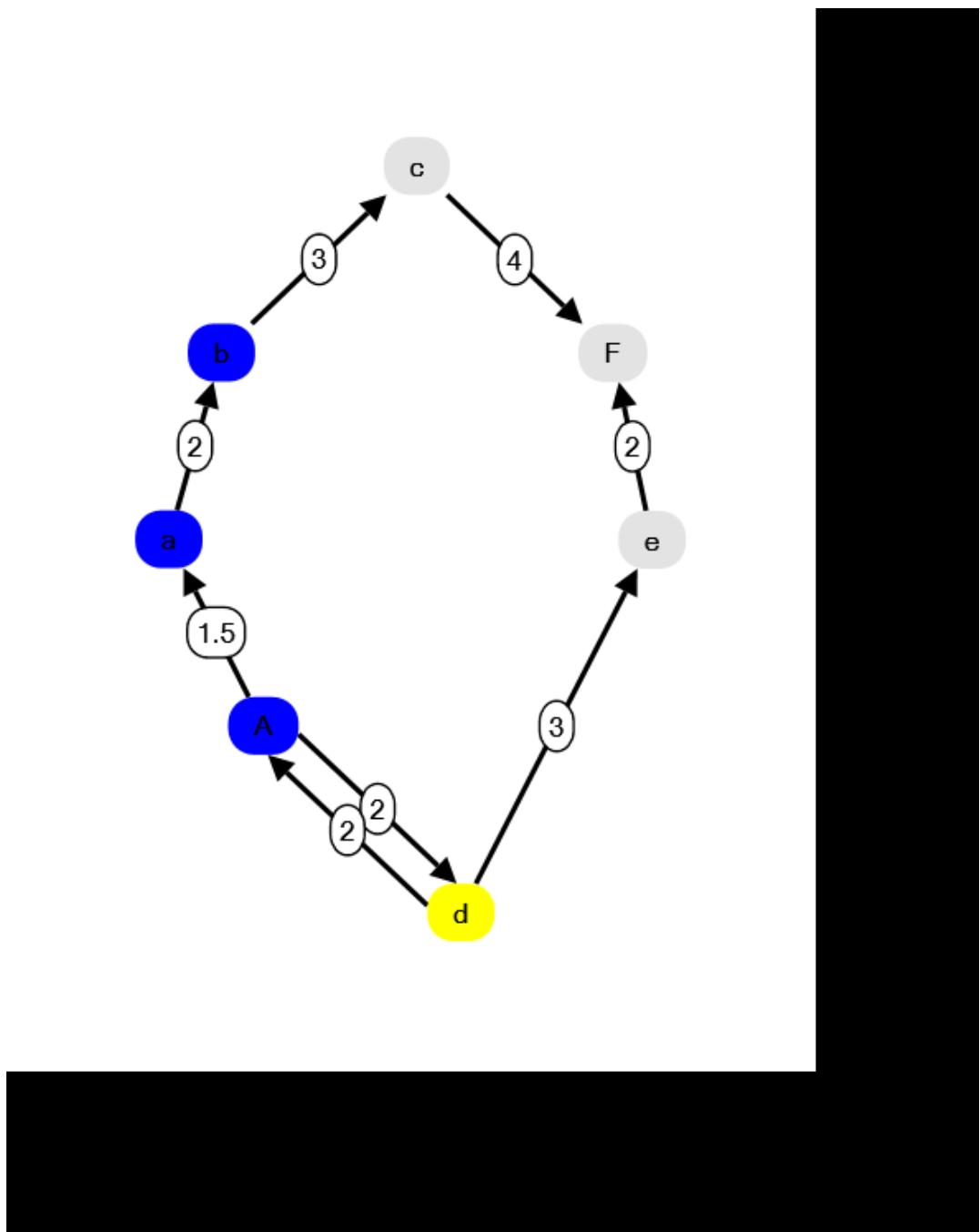
Post-Visit

The algorithm has closed the node b



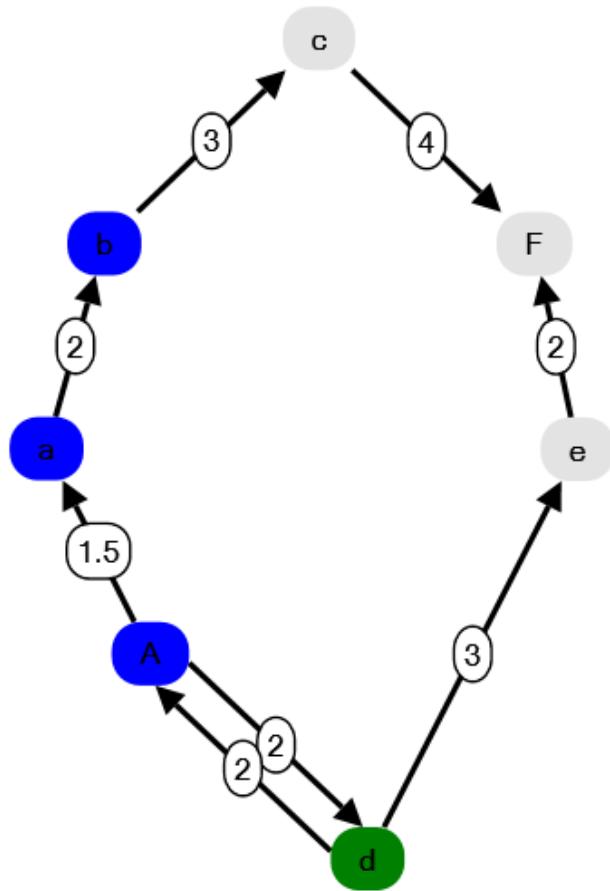
Pre-Visit

The algorithm has opened the node *d*



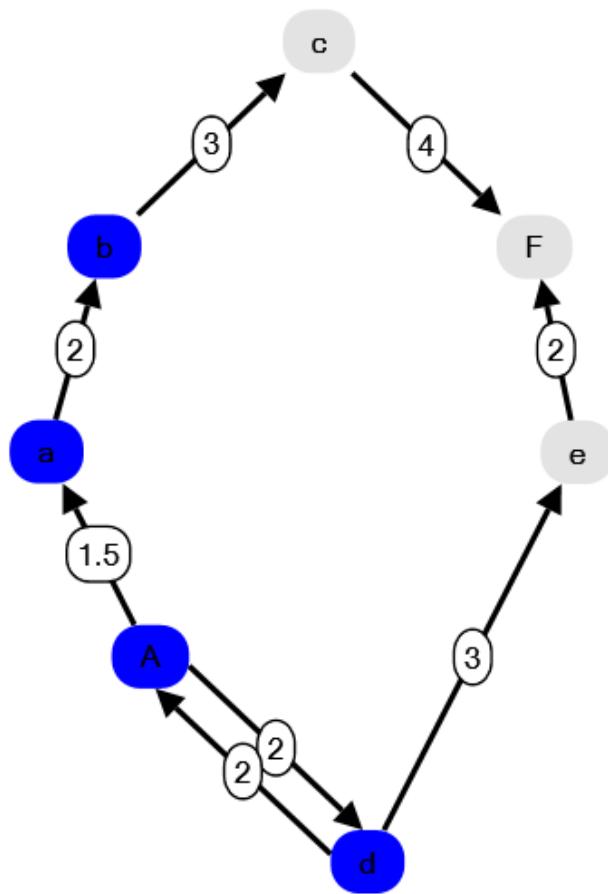
Visit

The algorithm has marked the node d as visited



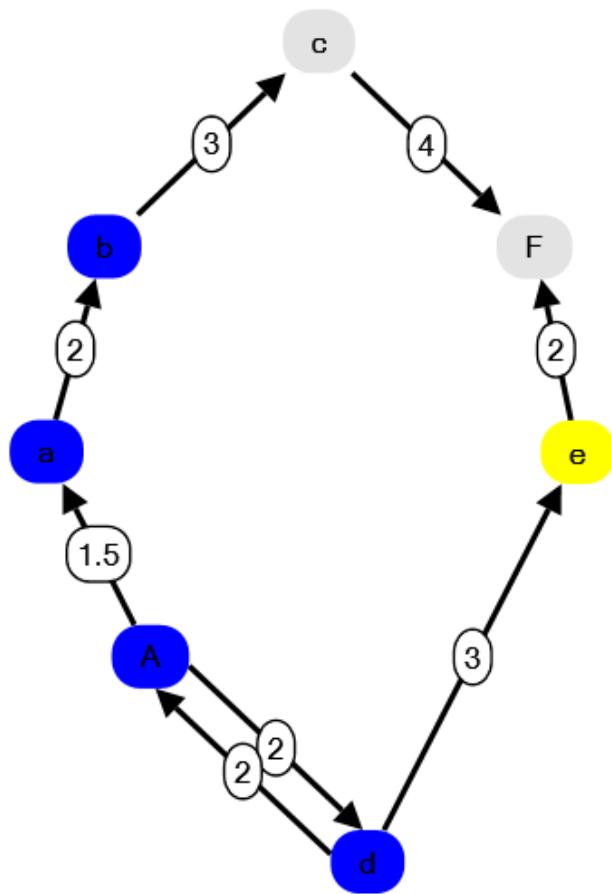
Post-Visit

The algorithm has closed the node d



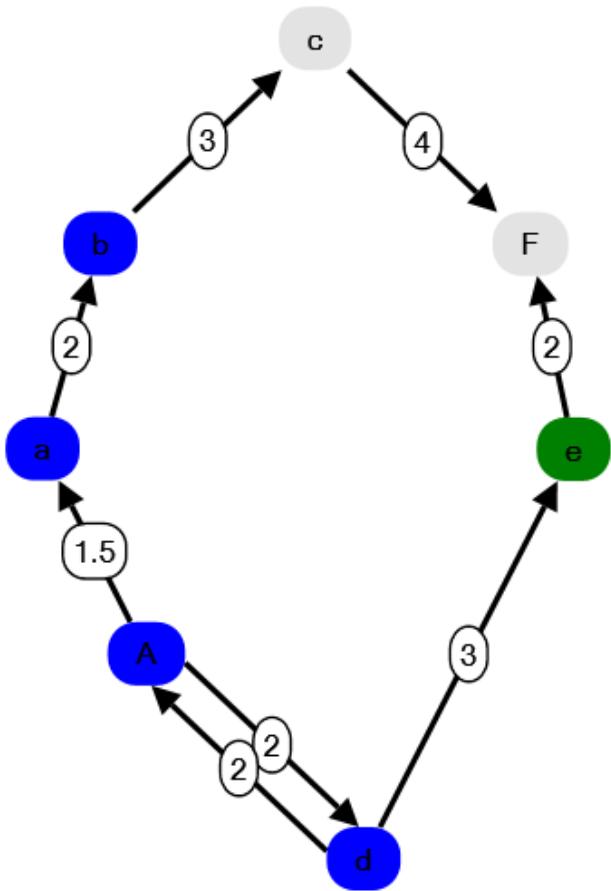
Pre-Visit

The algorithm has opened the node *e*



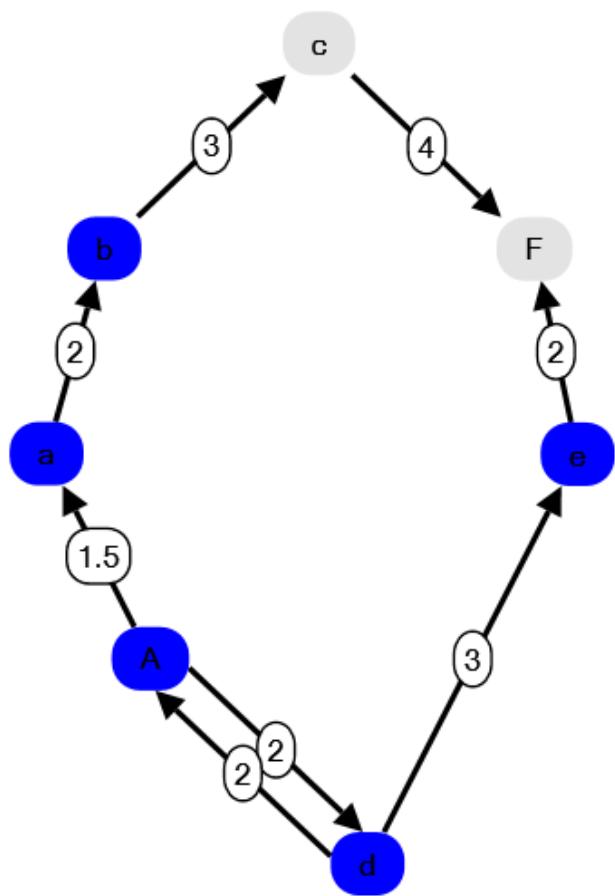
Visit

The algorithm has marked the node e as visited



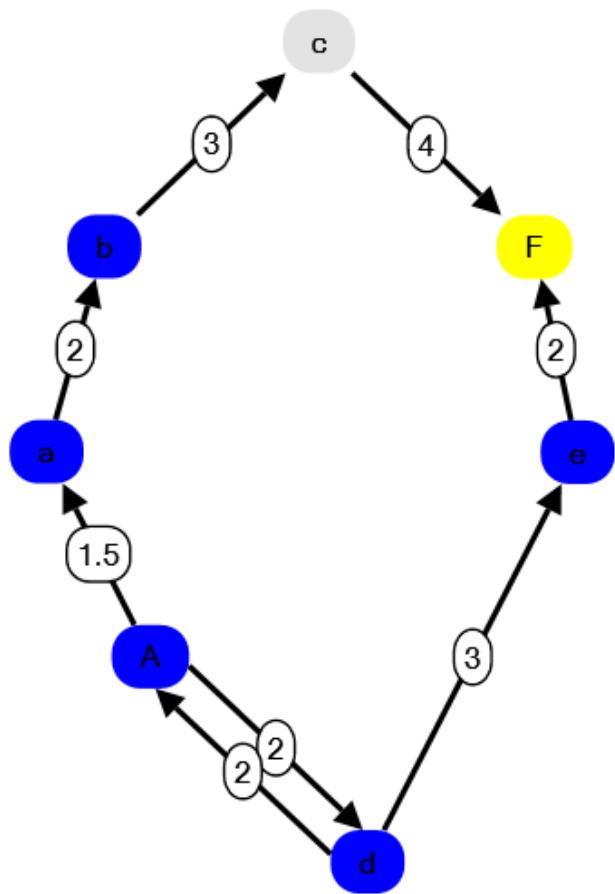
Post-Visit

The algorithm has closed the node e



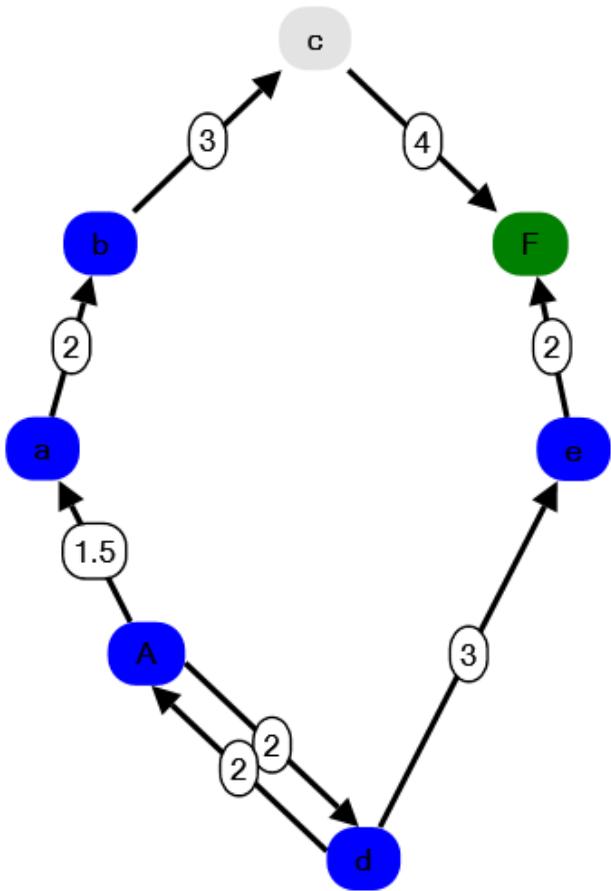
Pre-Visit

The algorithm has opened the node F



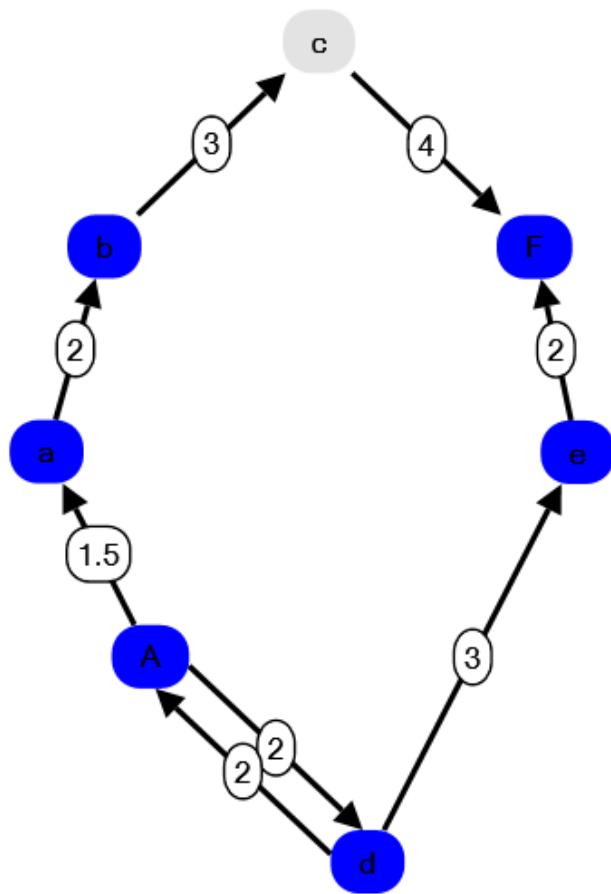
Visit

The algorithm has marked the node F as visited



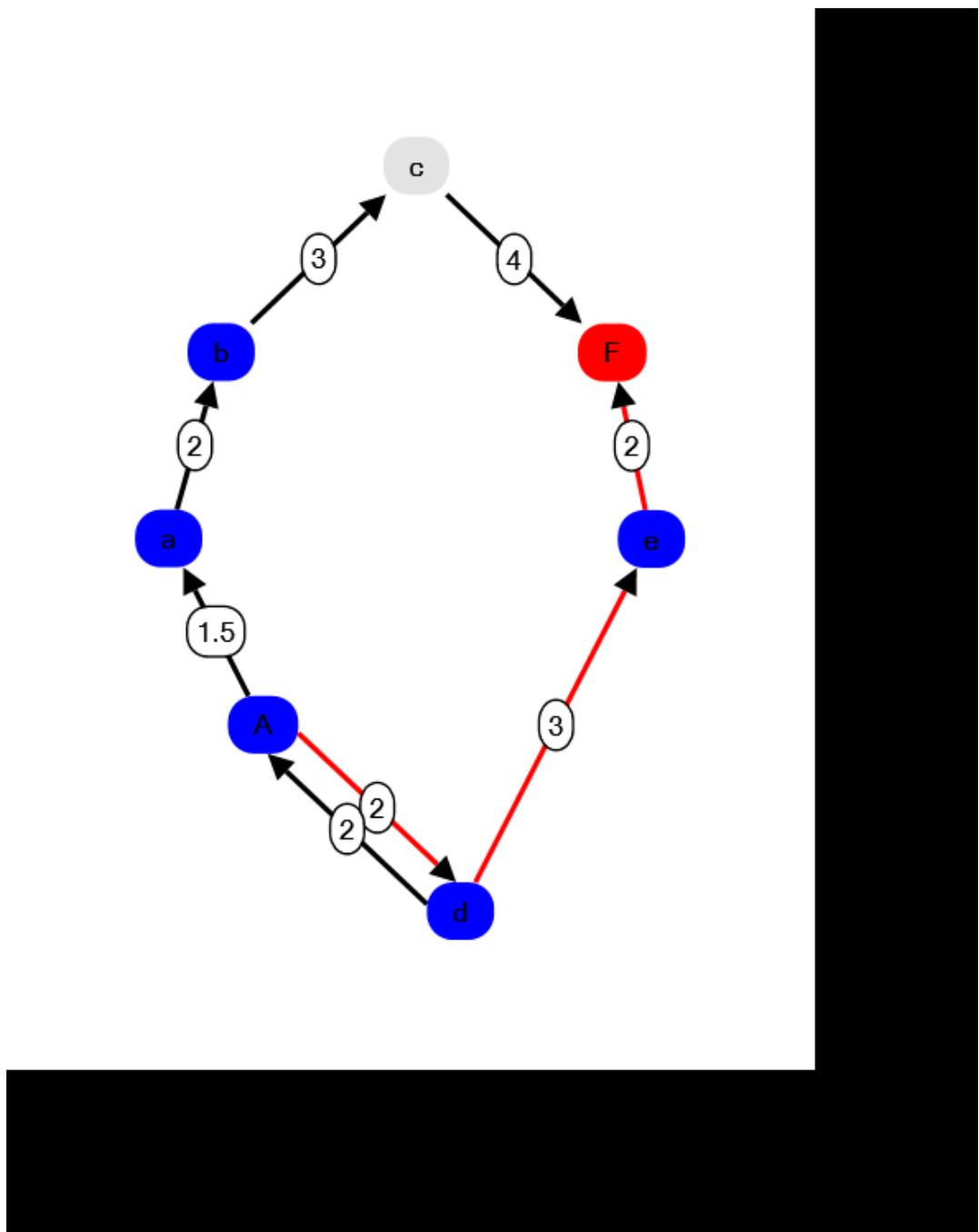
Post-Visit

The algorithm has closed the node F



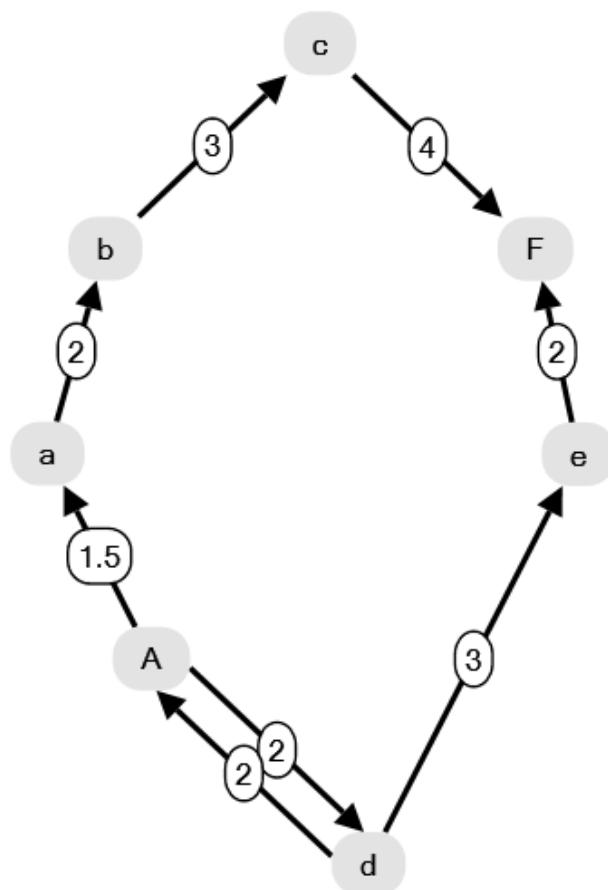
Result Found

The algorithm has found the target node F



Reset

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



Greedy Best-first search

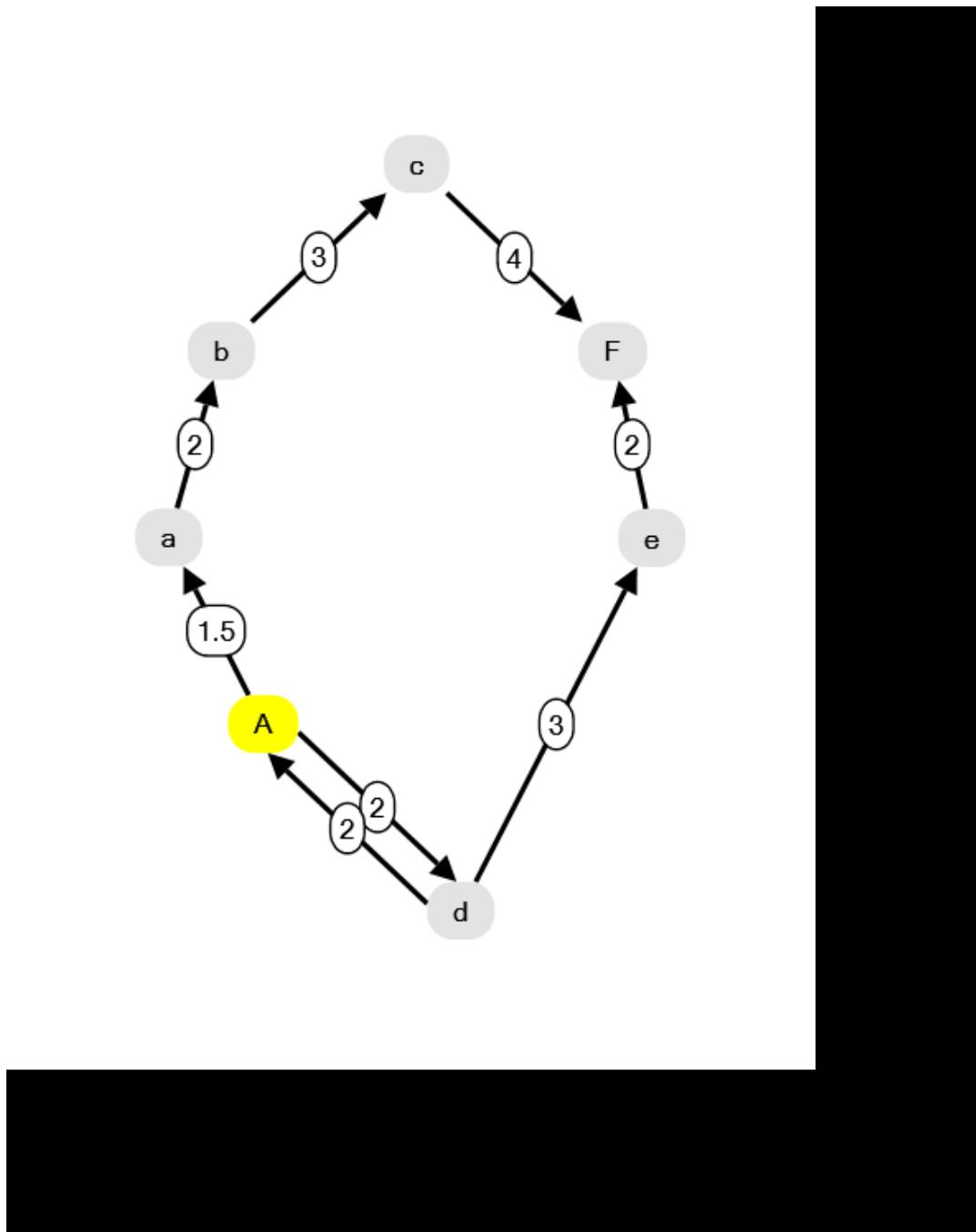
Best-first search is a search algorithm which explores a graph by expanding the most promising node chosen according to a specified rule.

The algorithm took 00:00:00.0008568 to find the goal.

The result path costs 10.5

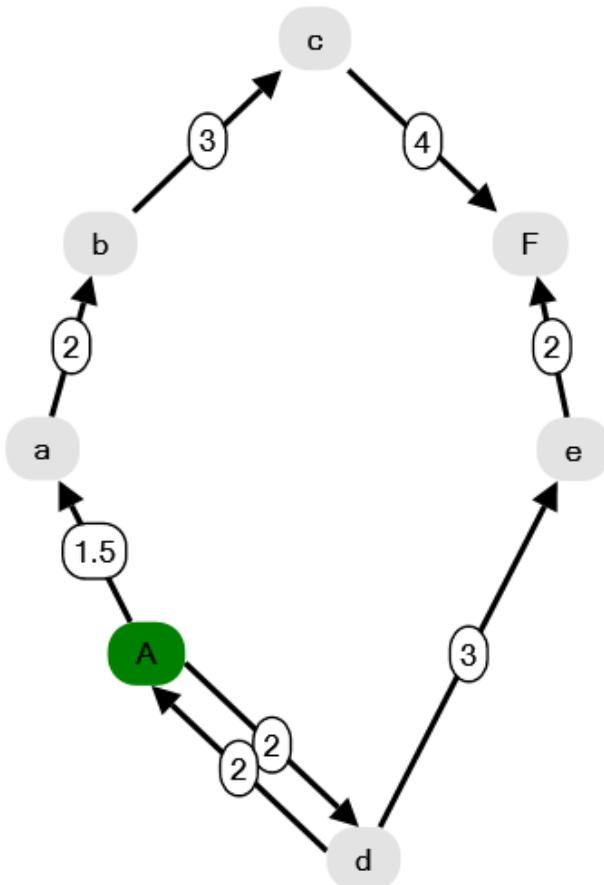
Pre-Visit

The algorithm has opened the node A



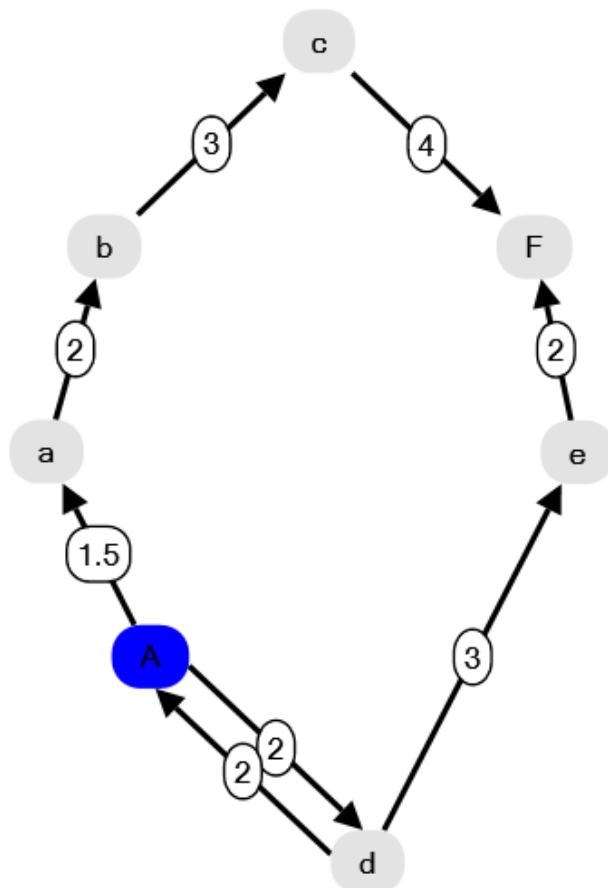
Visit

The algorithm has marked the node A as visited



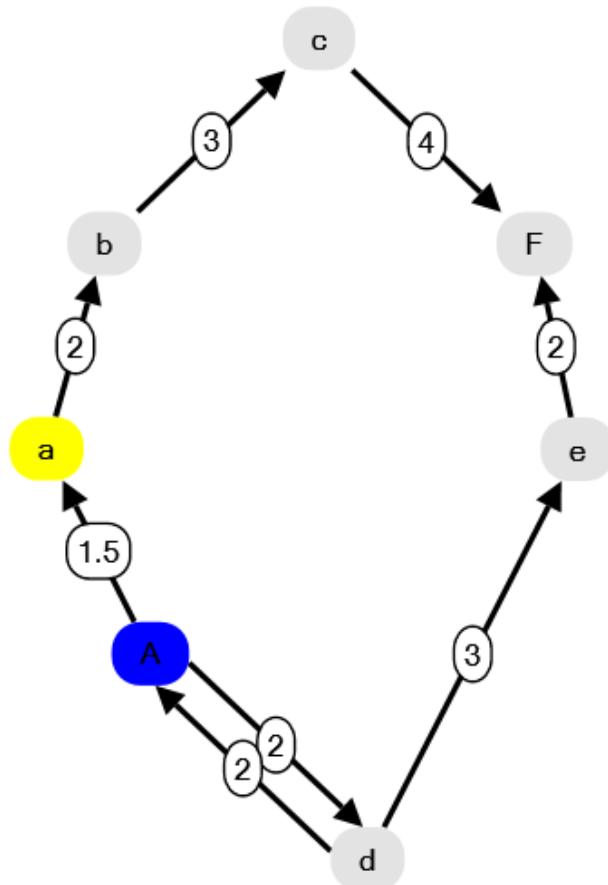
Post-Visit

The algorithm has closed the node A



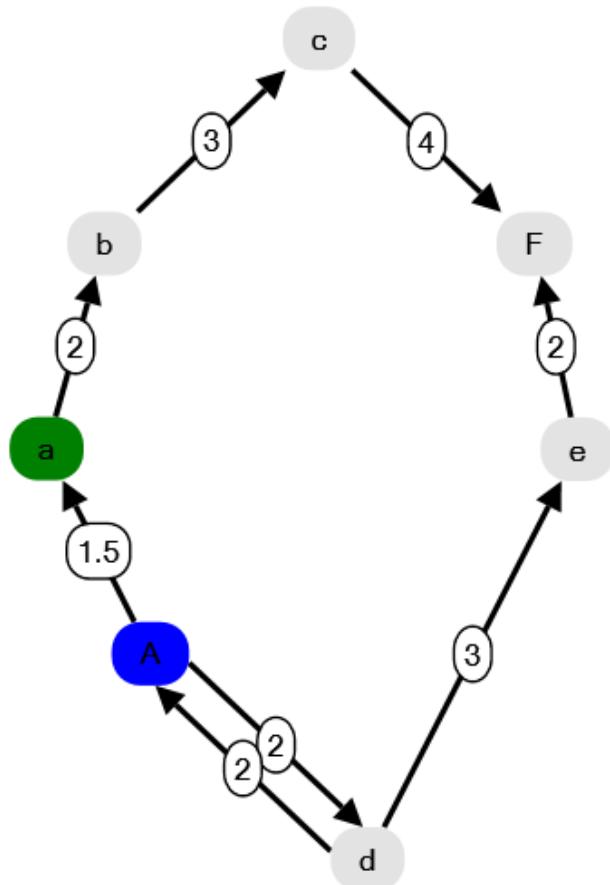
Pre-Visit

The algorithm has opened the node *a*



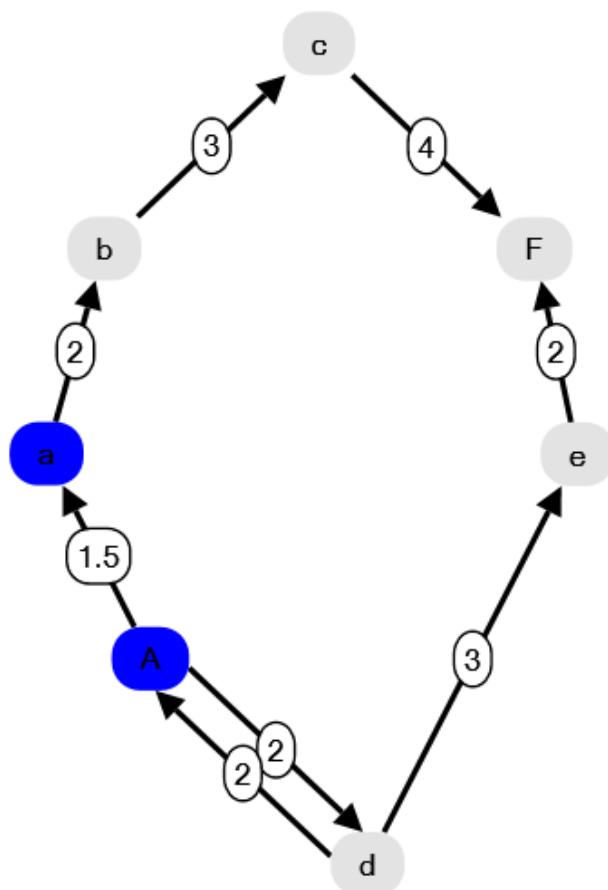
Visit

The algorithm has marked the node *a* as visited



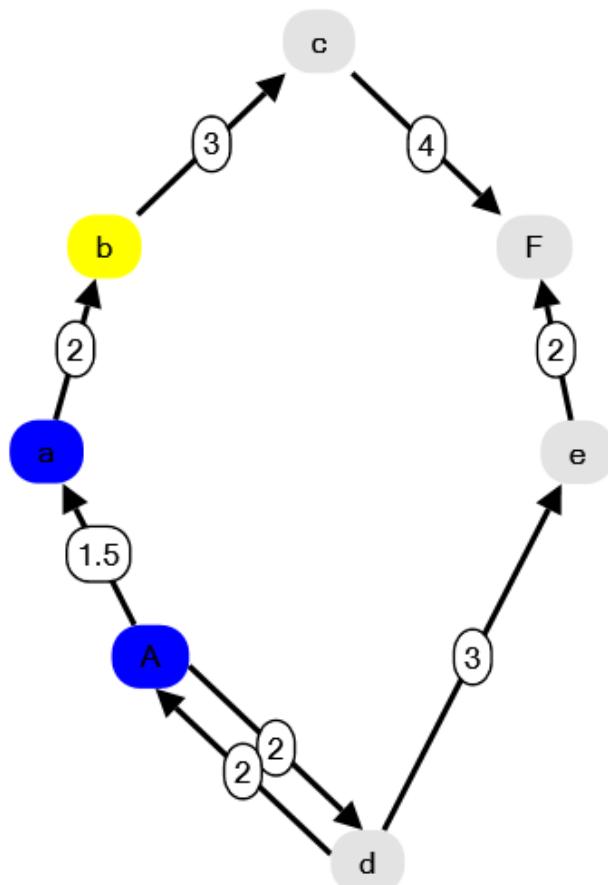
Post-Visit

The algorithm has closed the node *a*



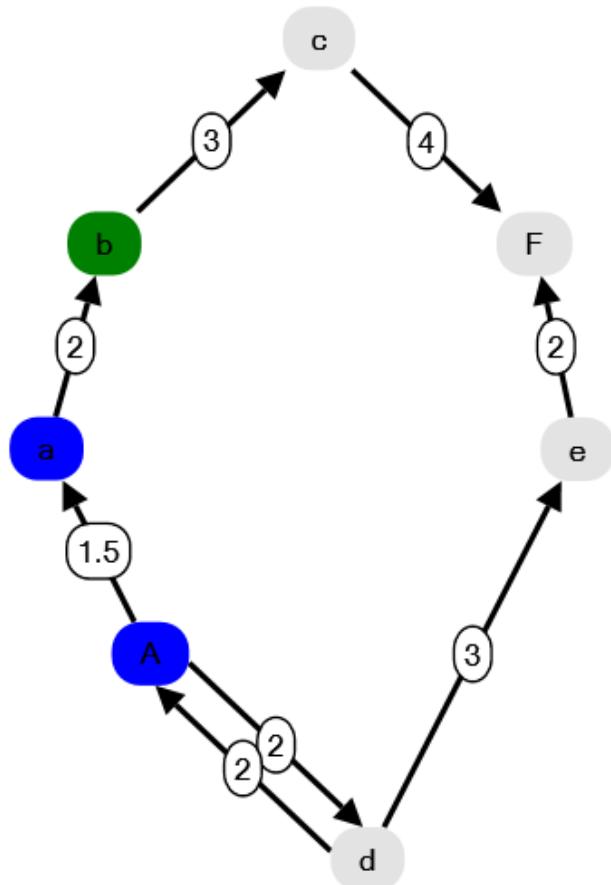
Pre-Visit

The algorithm has opened the node *b*



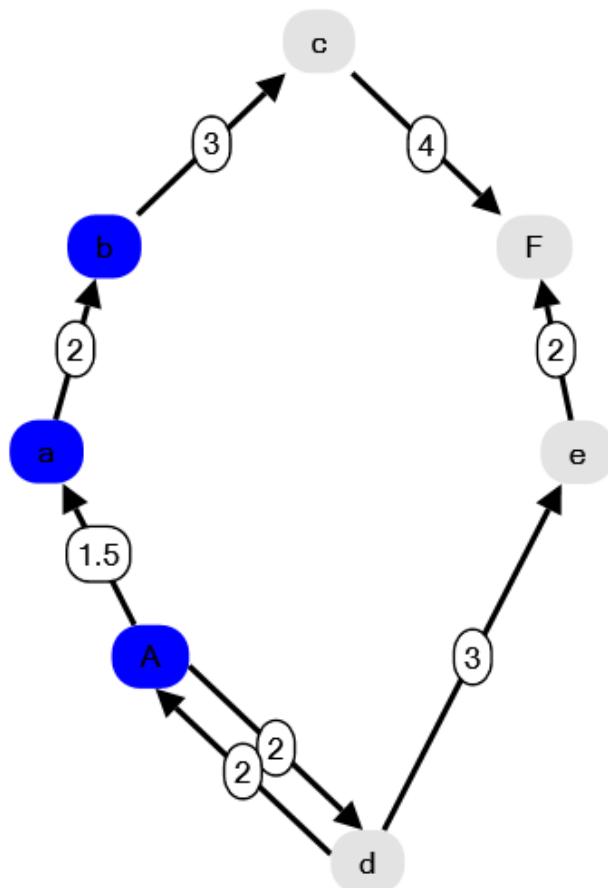
Visit

The algorithm has marked the node b as visited



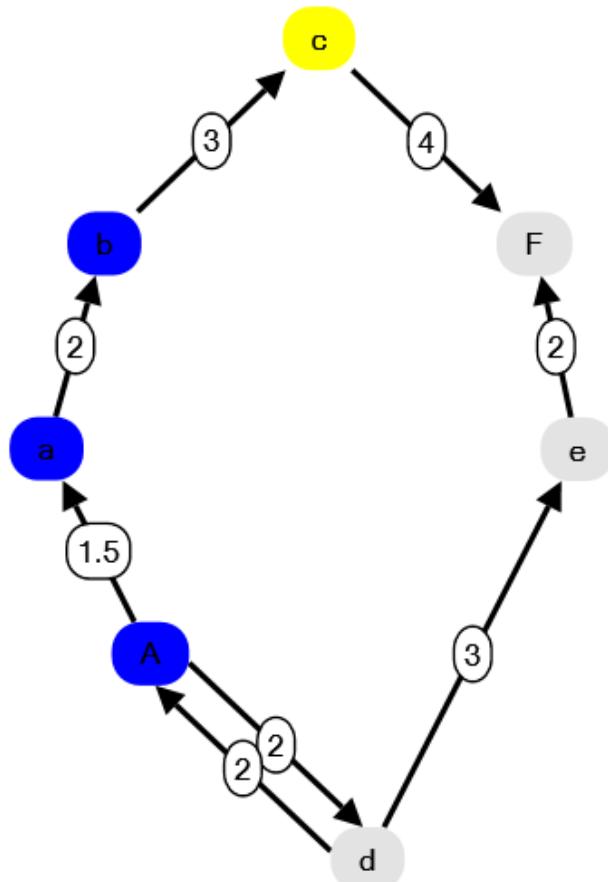
Post-Visit

The algorithm has closed the node b



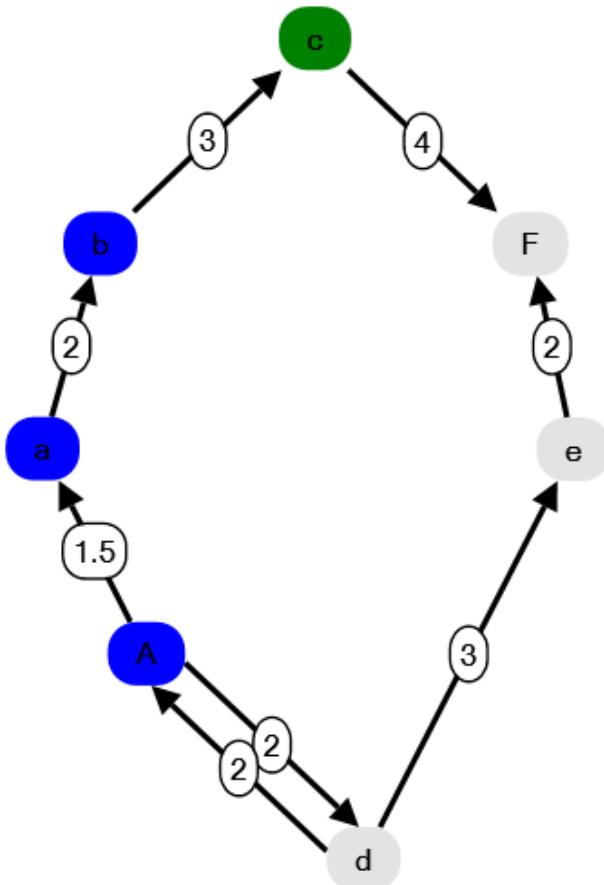
Pre-Visit

The algorithm has opened the node *c*



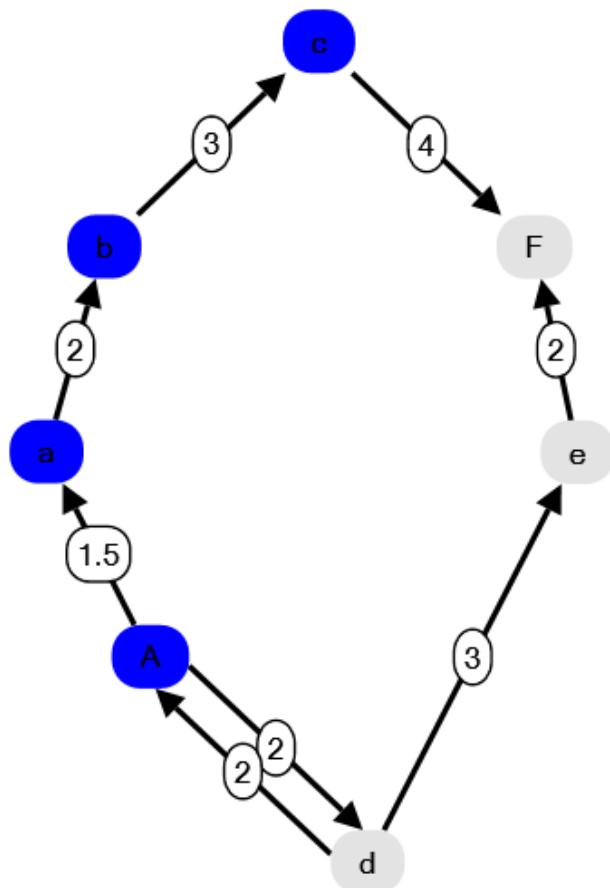
Visit

The algorithm has marked the node c as visited



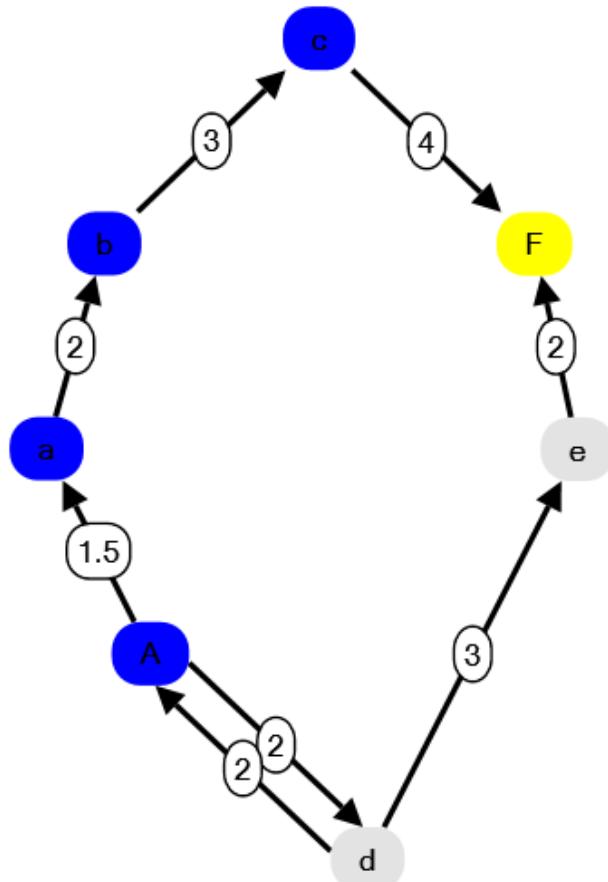
Post-Visit

The algorithm has closed the node c



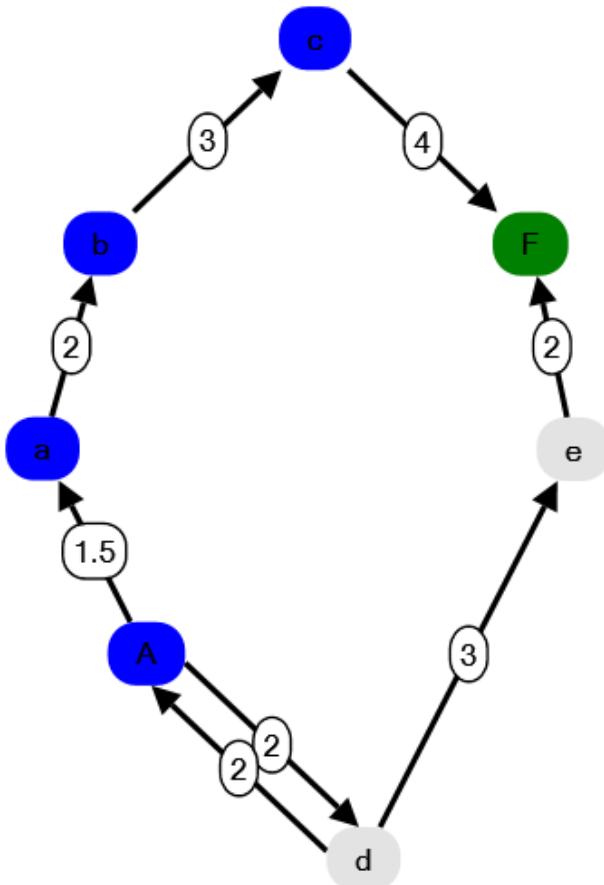
Pre-Visit

The algorithm has opened the node F



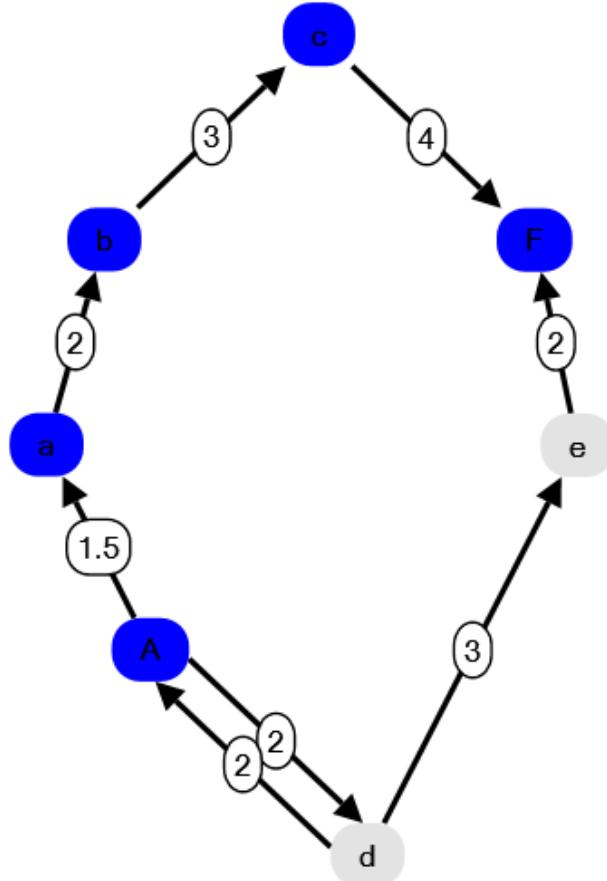
Visit

The algorithm has marked the node F as visited



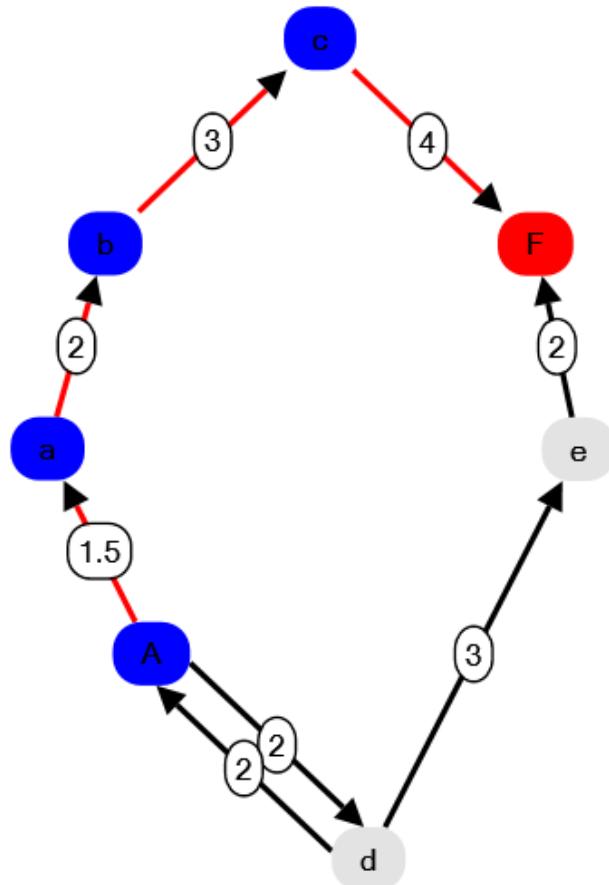
Post-Visit

The algorithm has closed the node F



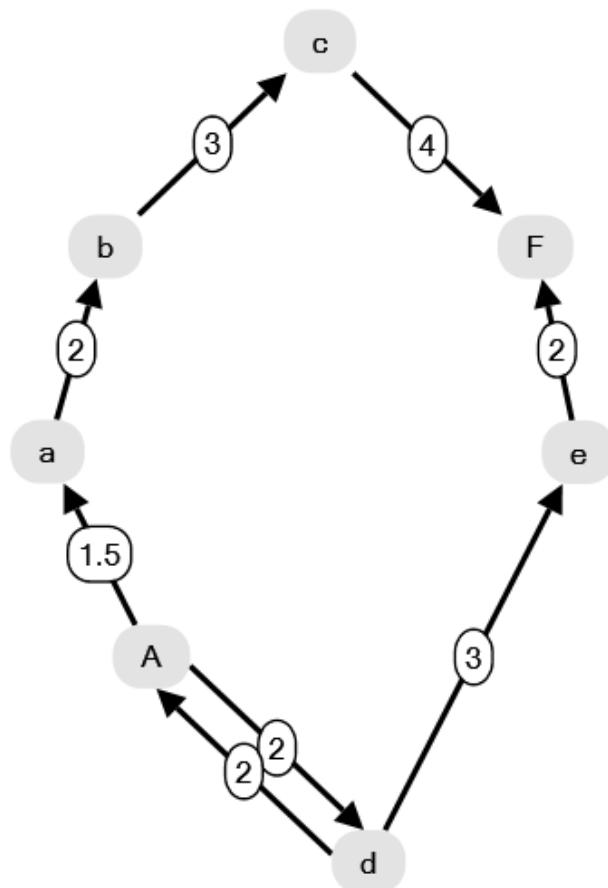
Result Found

The algorithm has found the target node F



Reset

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



Iterative deepening A*

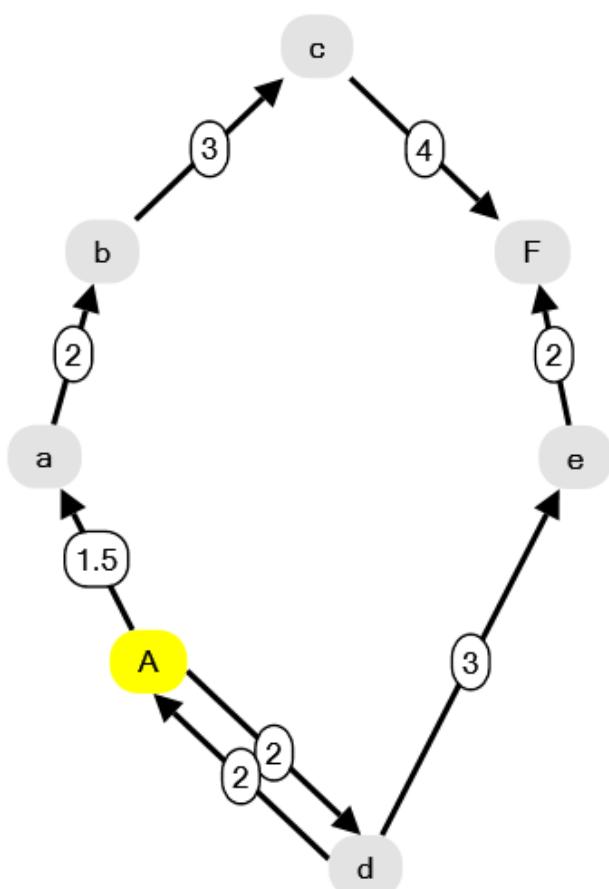
Iterative deepening A* (IDA*) is a graph traversal and path search algorithm that can find the shortest path between a designated start node and any member of a set of goal nodes in a weighted graph. It is a variant of iterative deepening depth-first search that borrows the idea to use a heuristic function 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 00:00:00.0007336 to find the goal.

The result path costs 7

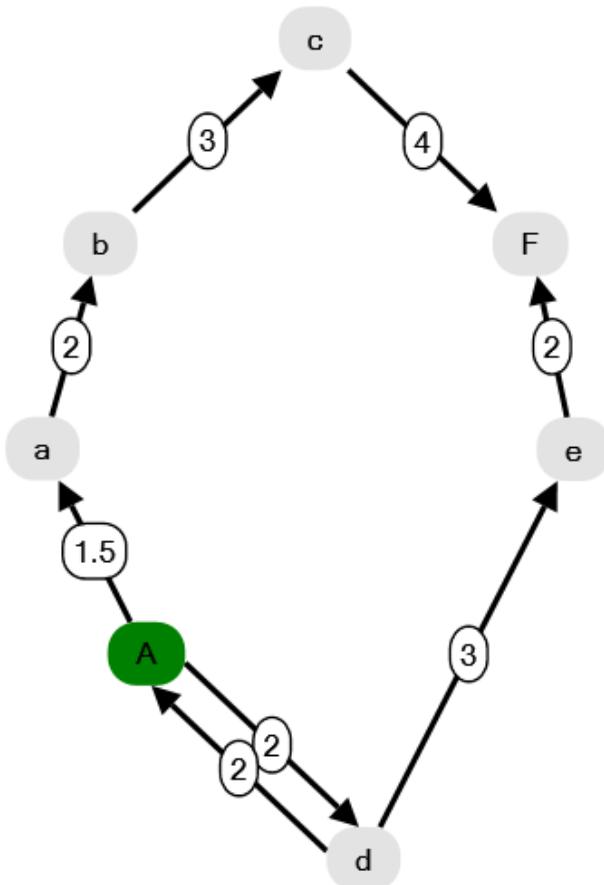
Pre-Visit

The algorithm has opened the node A



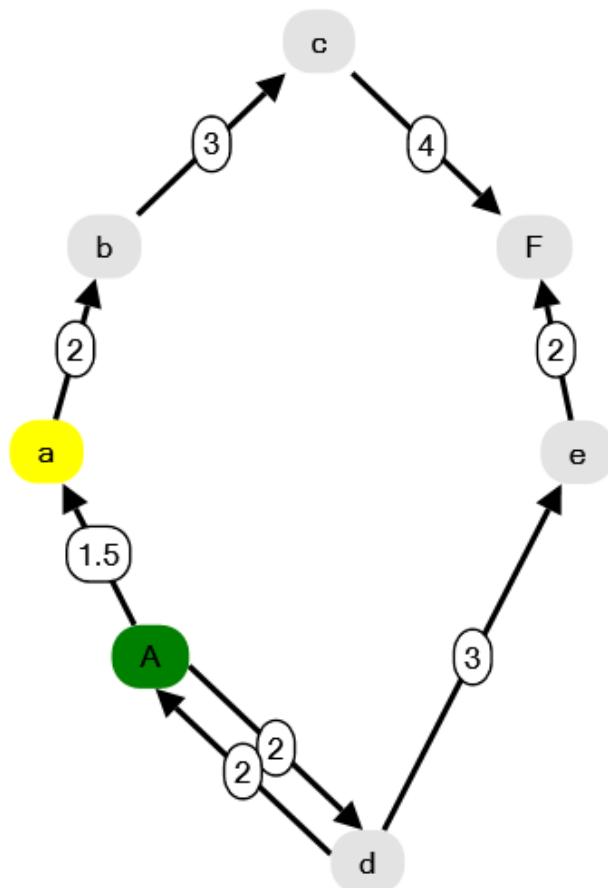
Visit

The algorithm has marked the node A as visited



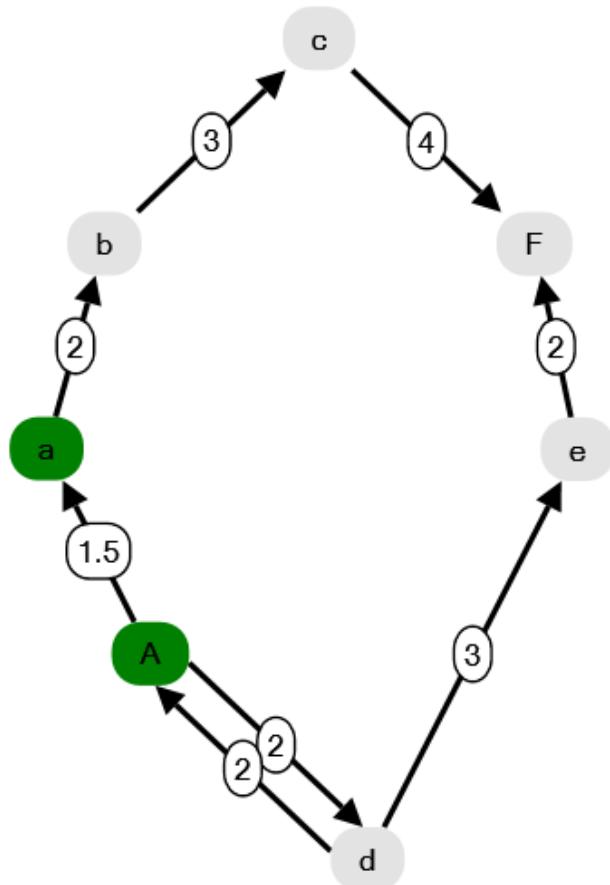
Pre-Visit

The algorithm has opened the node *a*



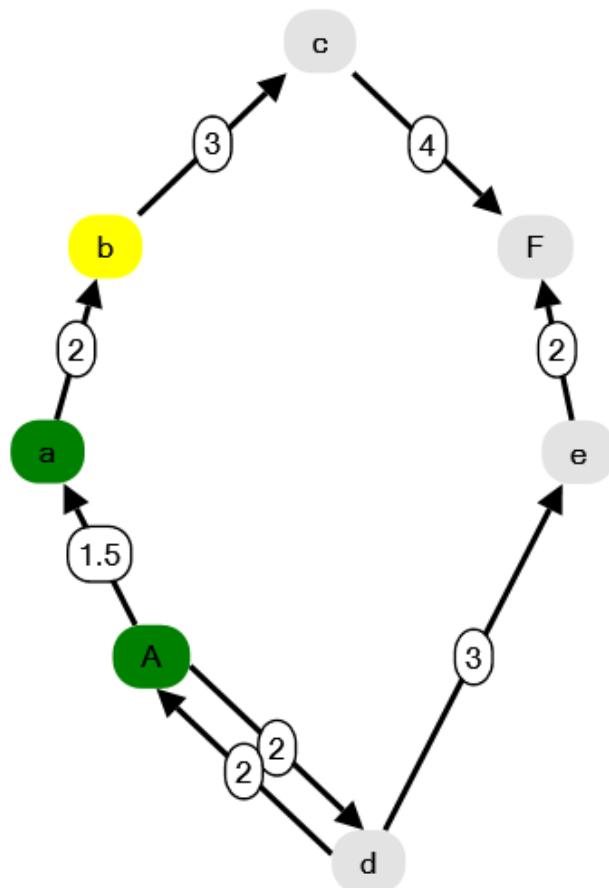
Visit

The algorithm has marked the node *a* as visited



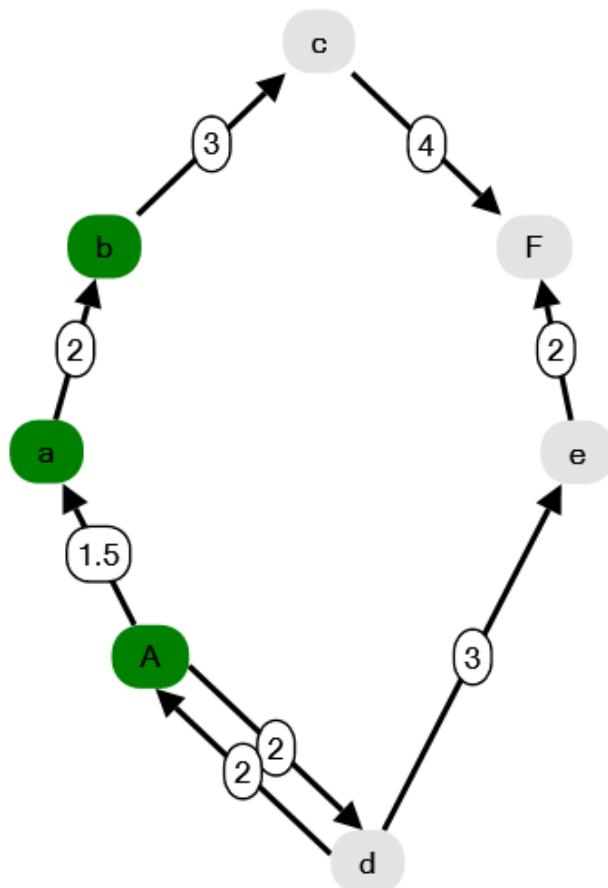
Pre-Visit

The algorithm has opened the node *b*



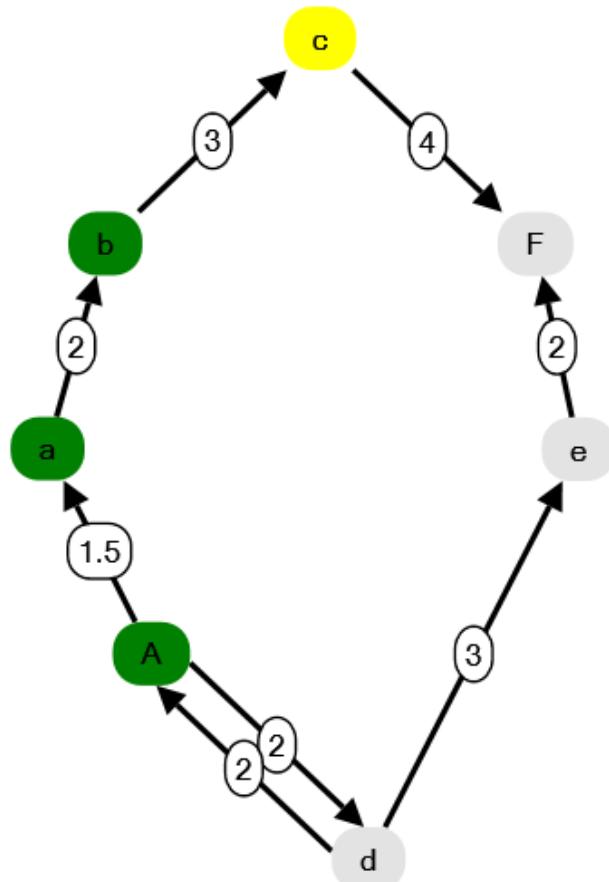
Visit

The algorithm has marked the node *b* as visited



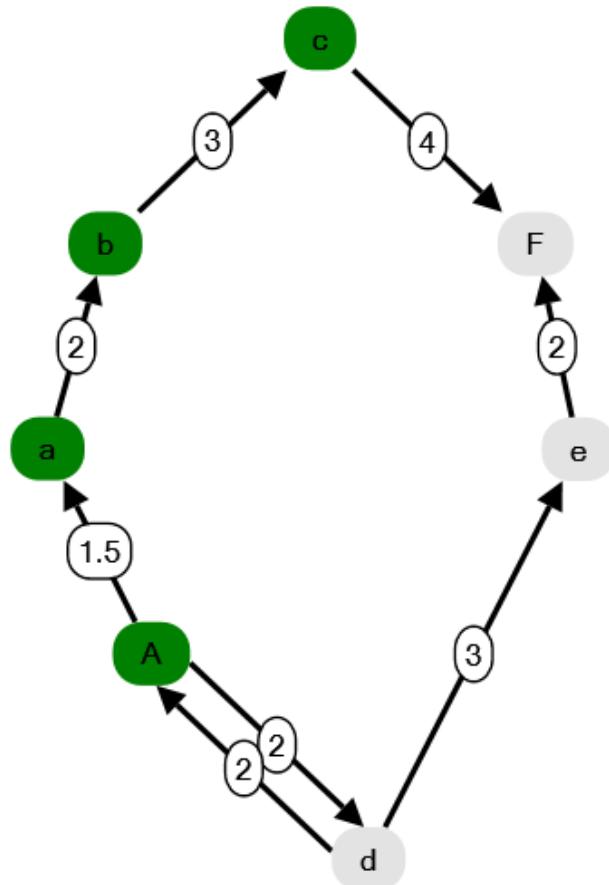
Pre-Visit

The algorithm has opened the node *c*



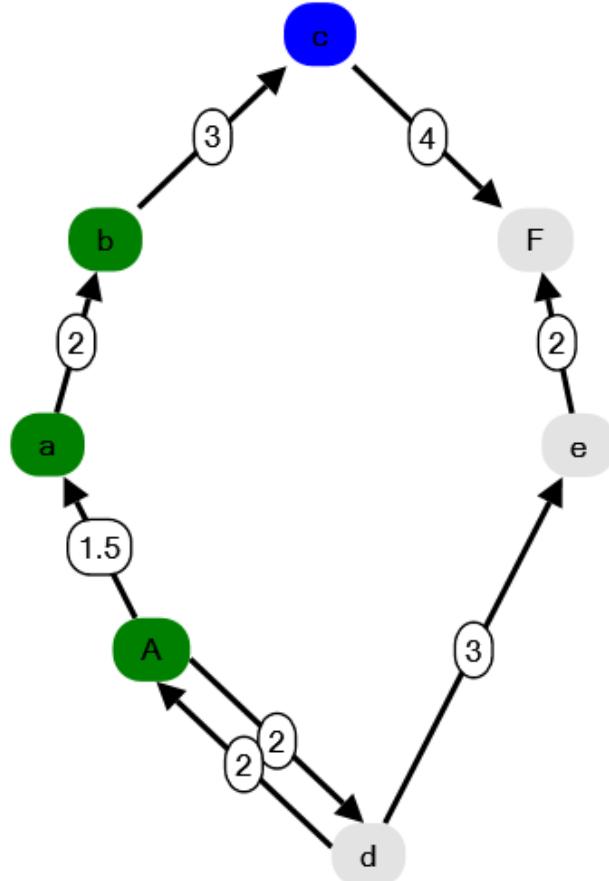
Visit

The algorithm has marked the node c as visited



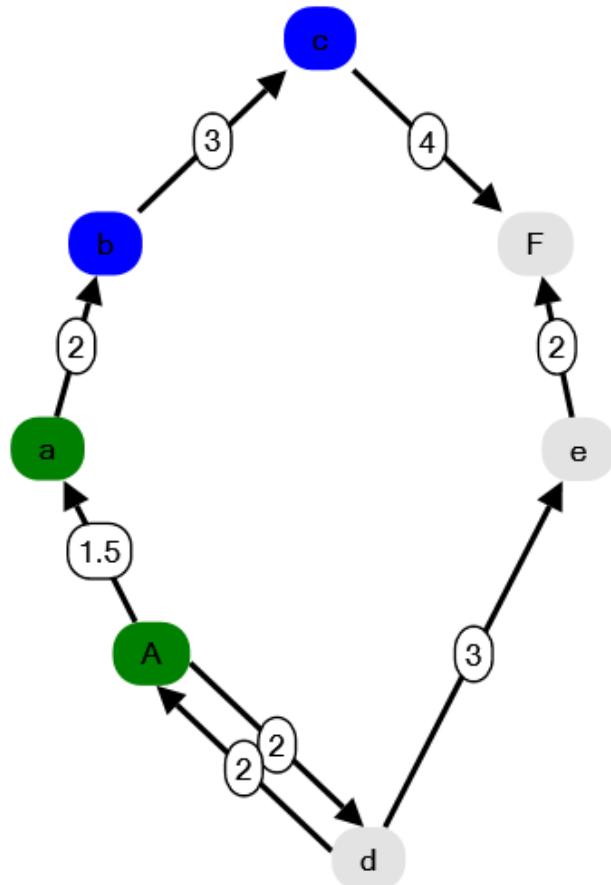
Post-Visit

The algorithm has closed the node c



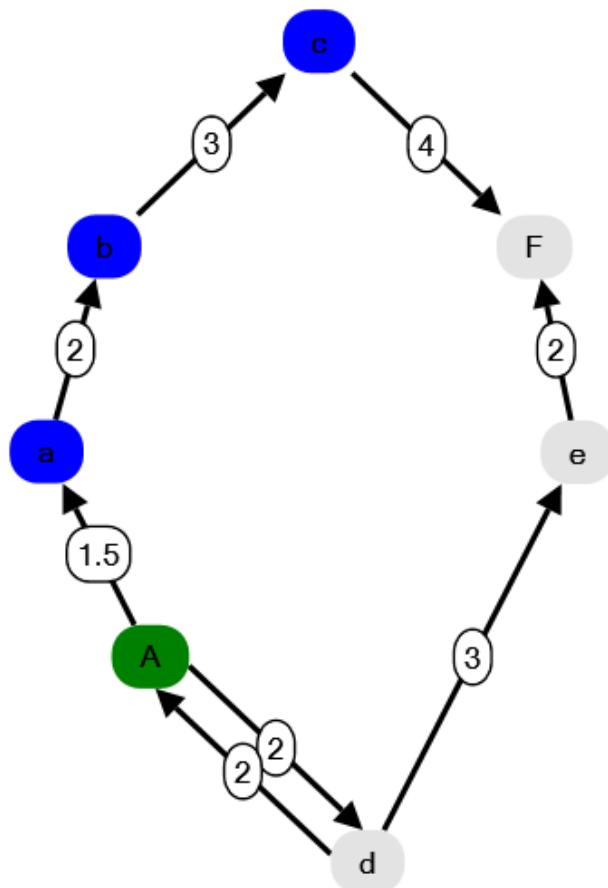
Post-Visit

The algorithm has closed the node b



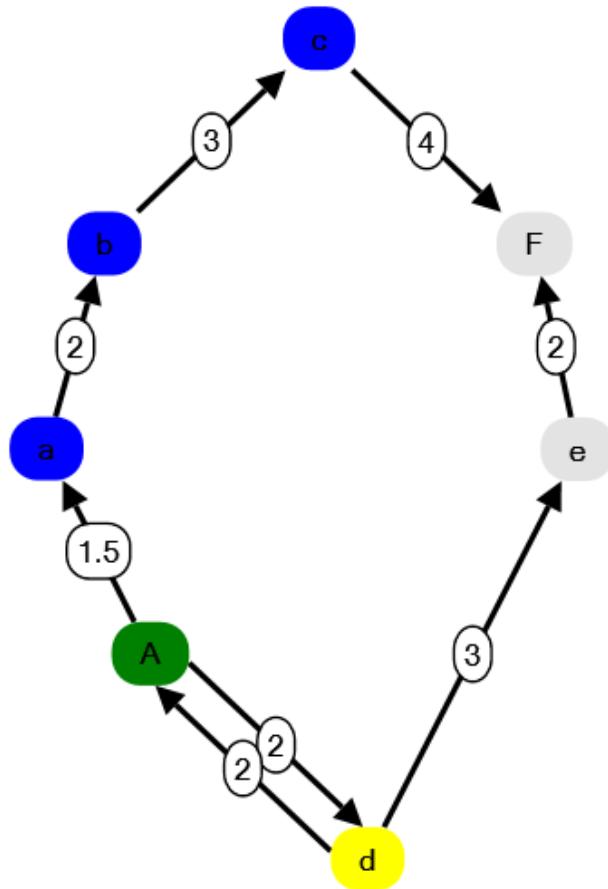
Post-Visit

The algorithm has closed the node *a*



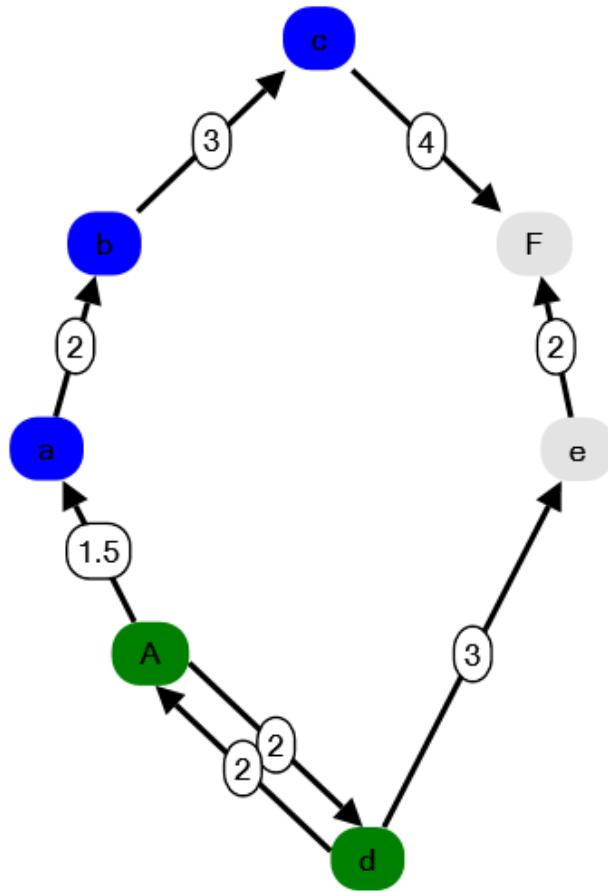
Pre-Visit

The algorithm has opened the node *d*



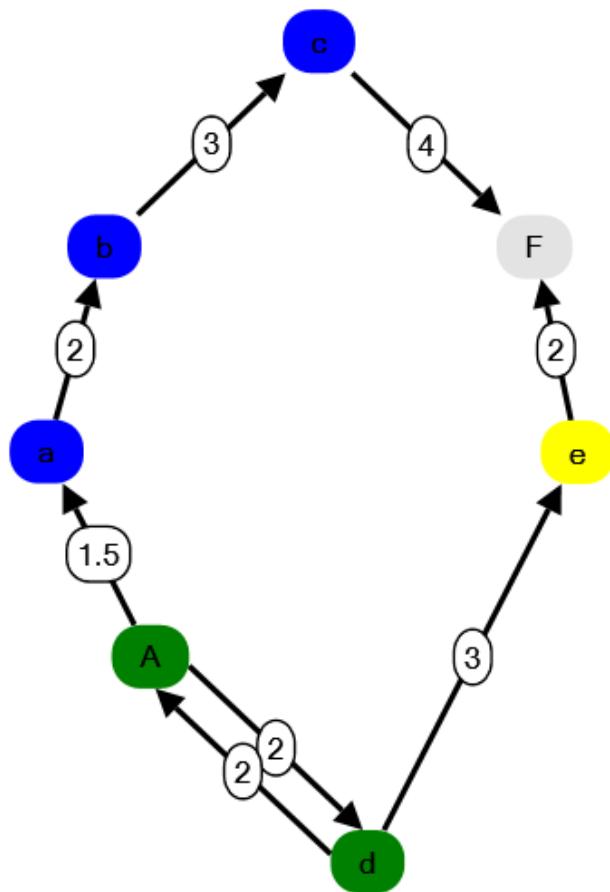
Visit

The algorithm has marked the node d as visited



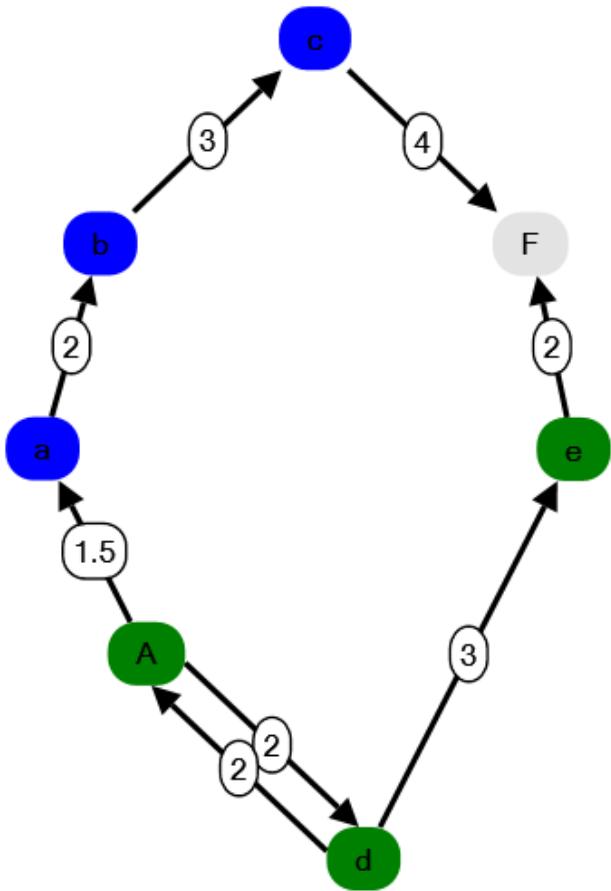
Pre-Visit

The algorithm has opened the node *e*



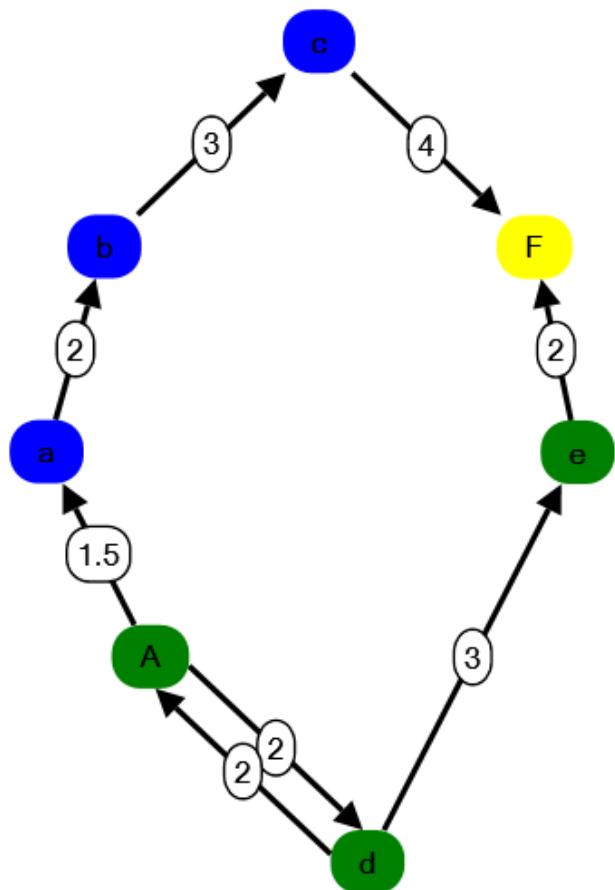
Visit

The algorithm has marked the node e as visited



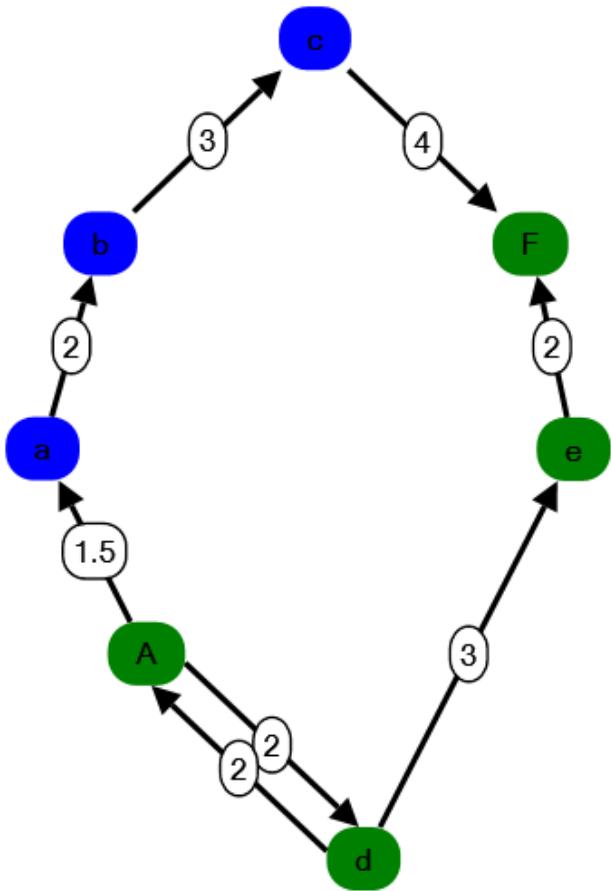
Pre-Visit

The algorithm has opened the node F



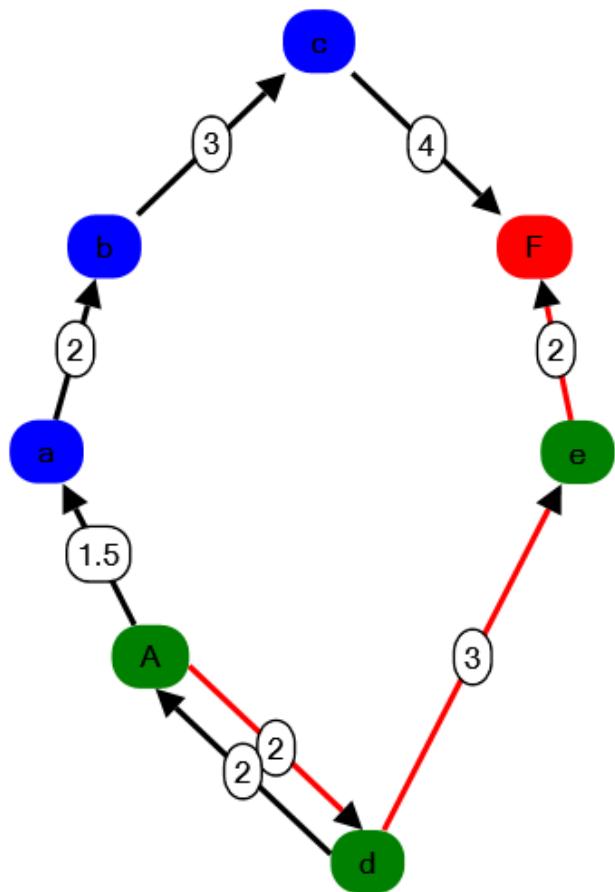
Visit

The algorithm has marked the node F as visited



Result Found

The algorithm has found the target node F



Reset

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

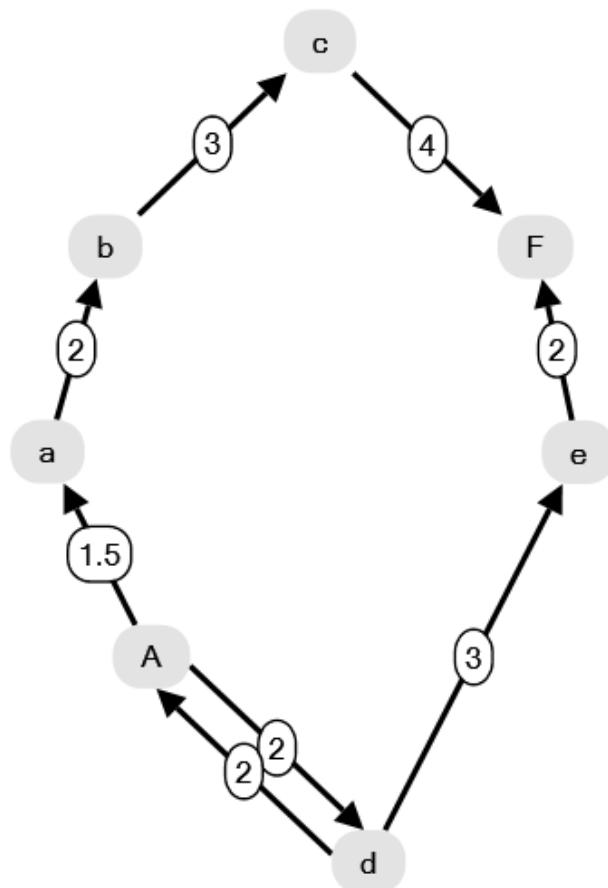


Table of Contents

Breadth-first search.....	4
Depth-first search.....	27
Uniform Cost Search.....	44
Iterative Depth limited search.....	68
A*	154
Greedy Best-first search.....	174
Iterative deepening A*	191