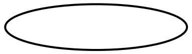
Limited-memory heuristic search

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

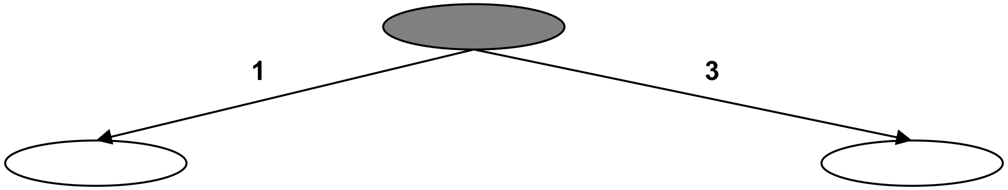


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

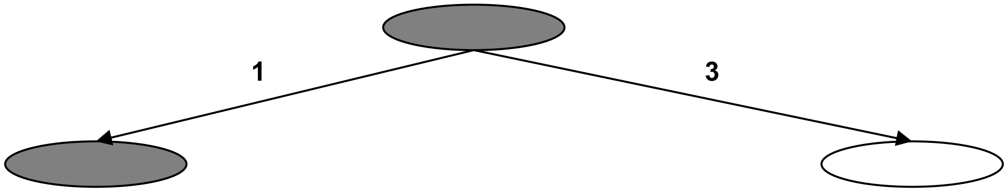


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

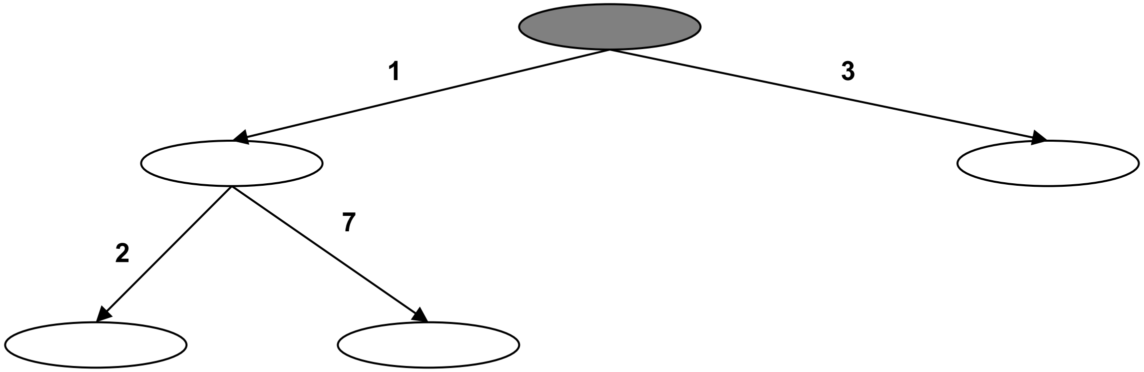


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

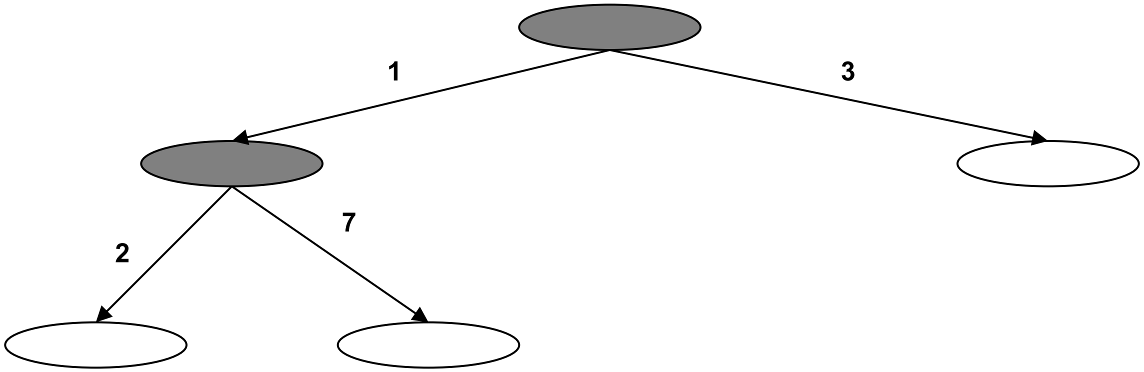


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

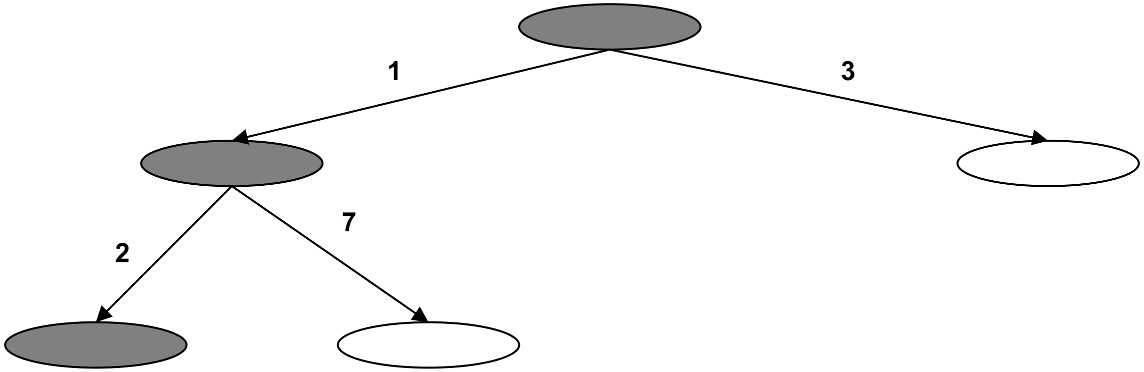


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

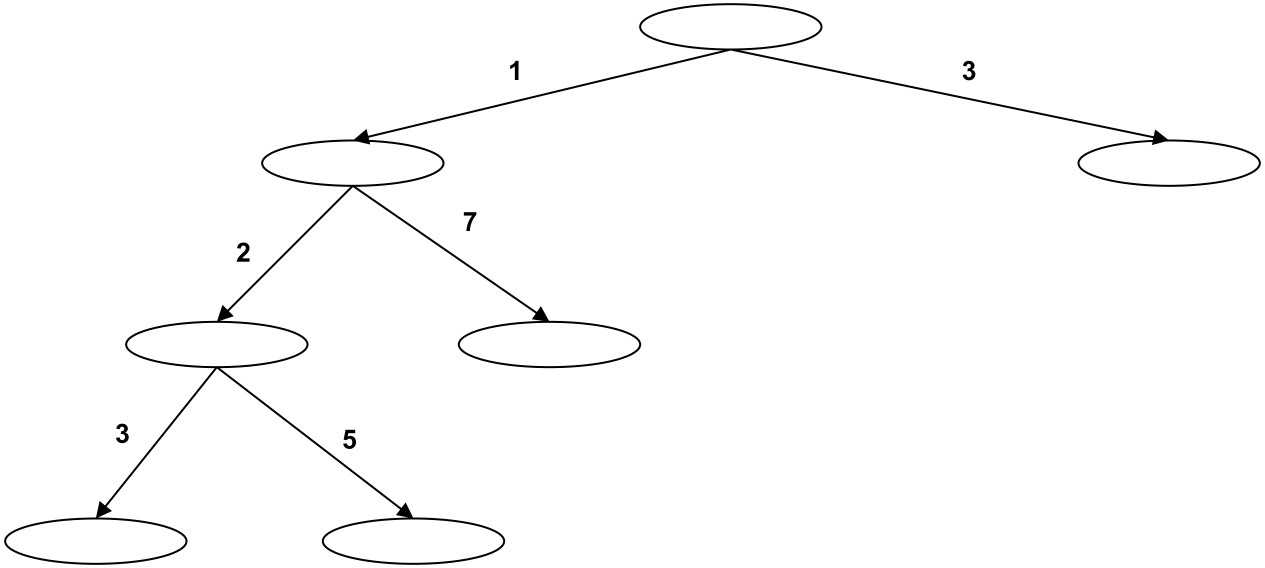


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

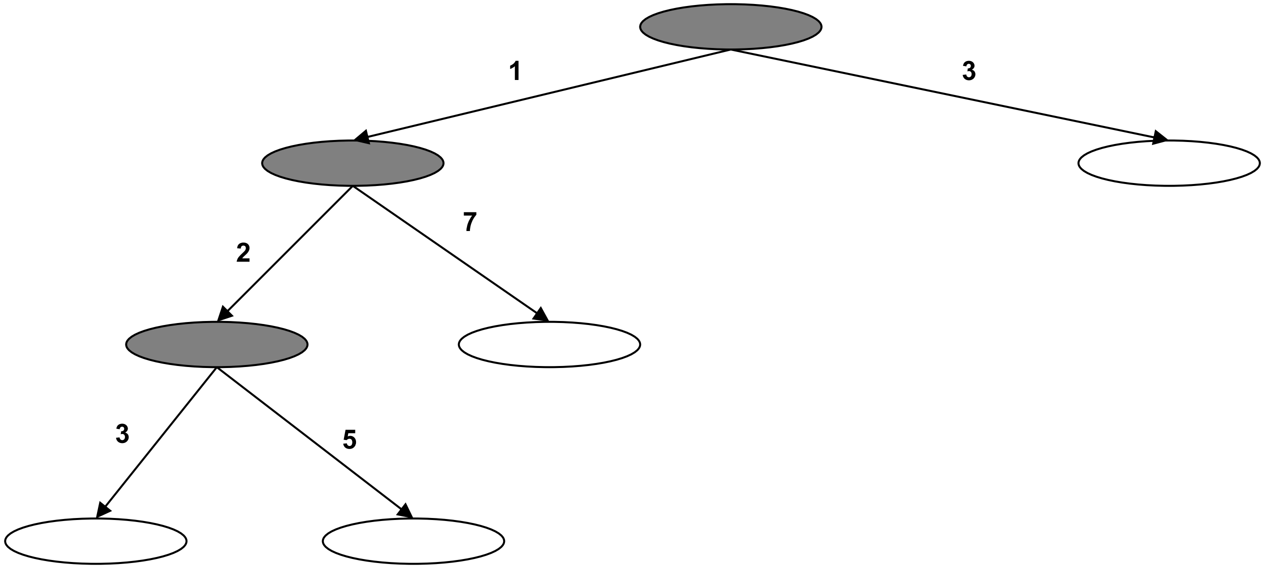


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

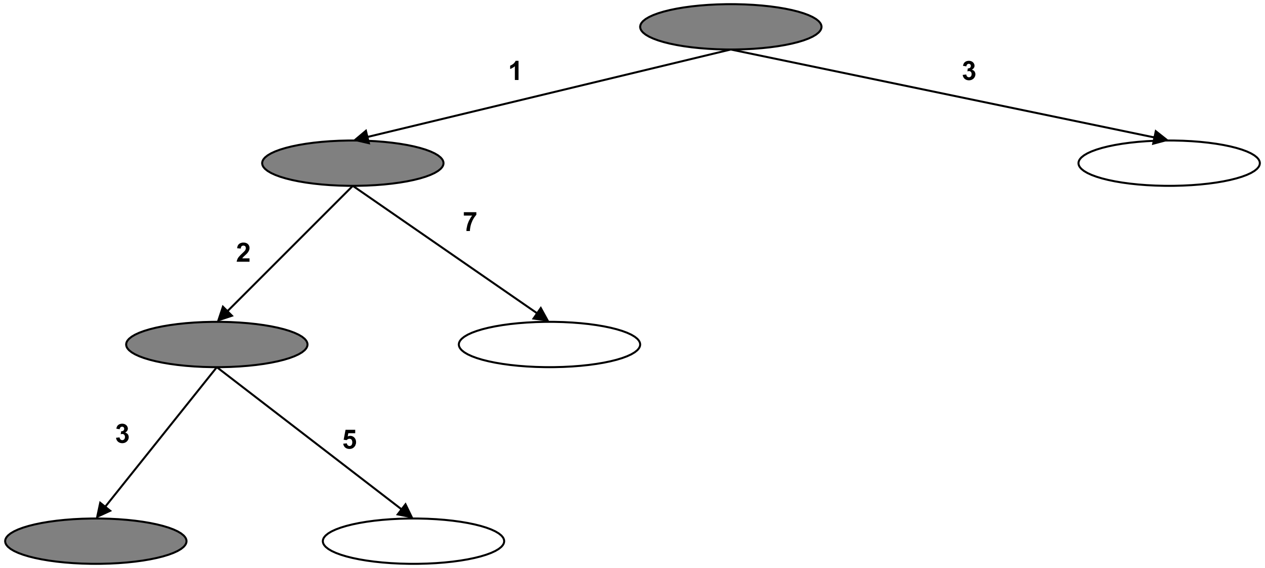


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

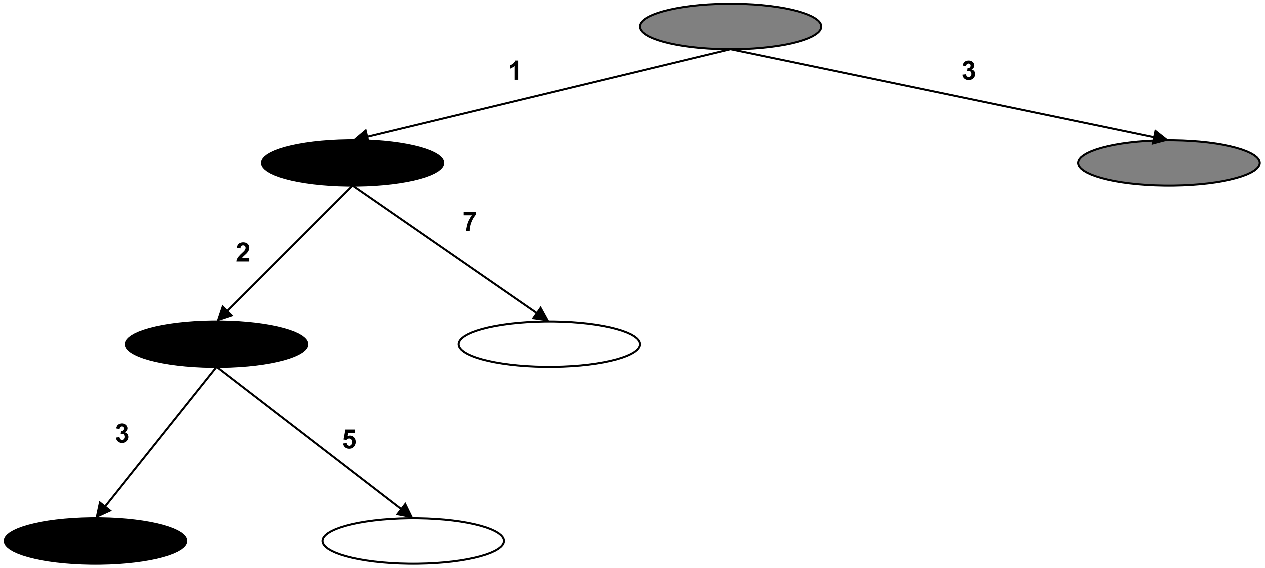


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

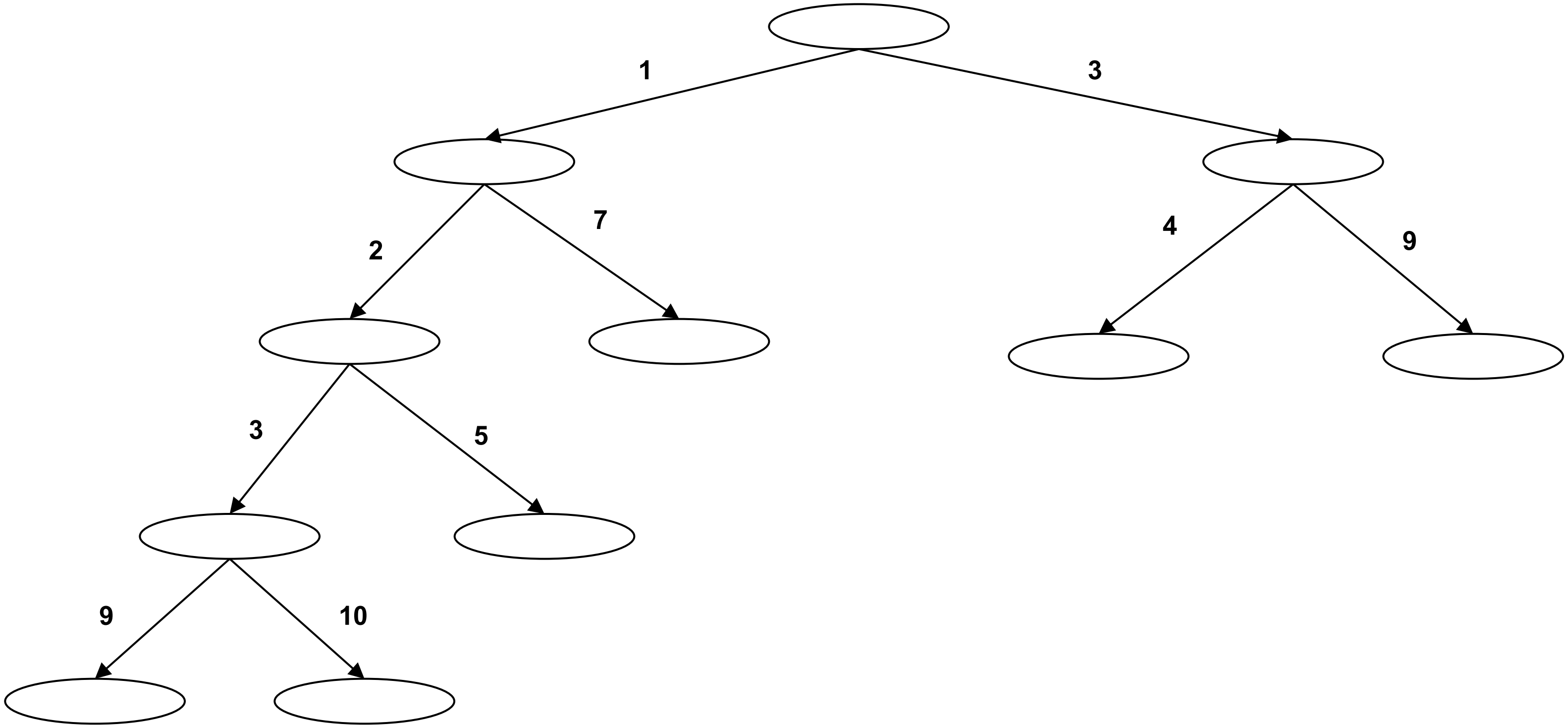


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

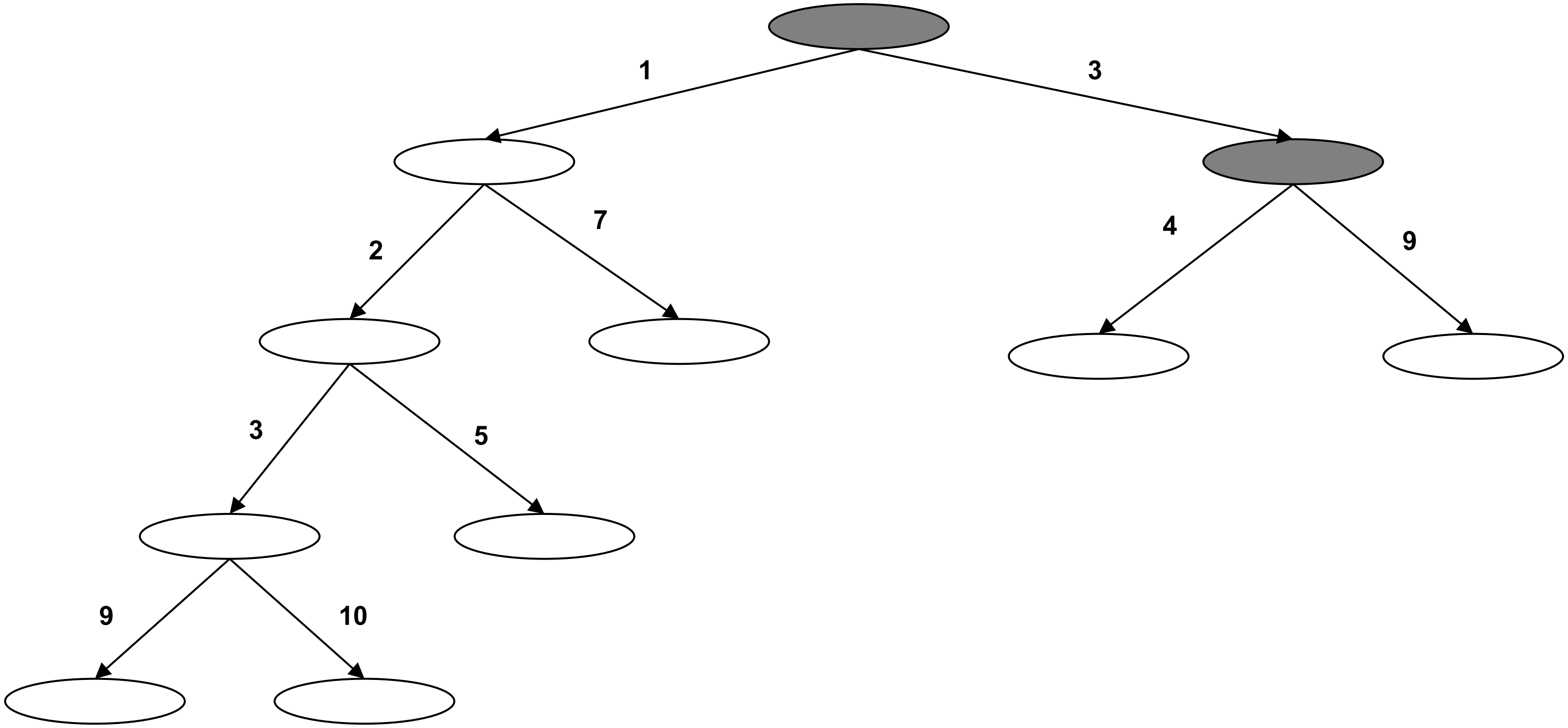


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

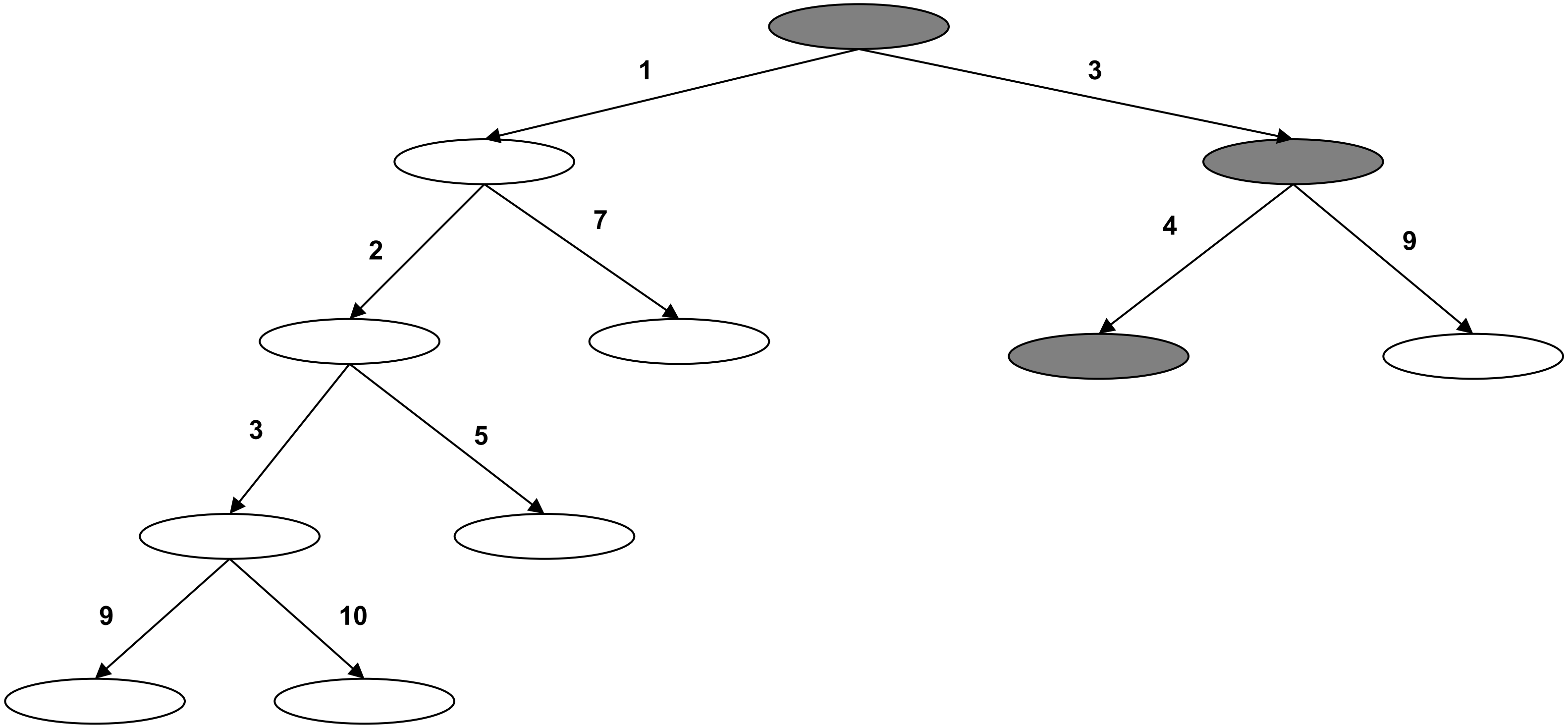


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

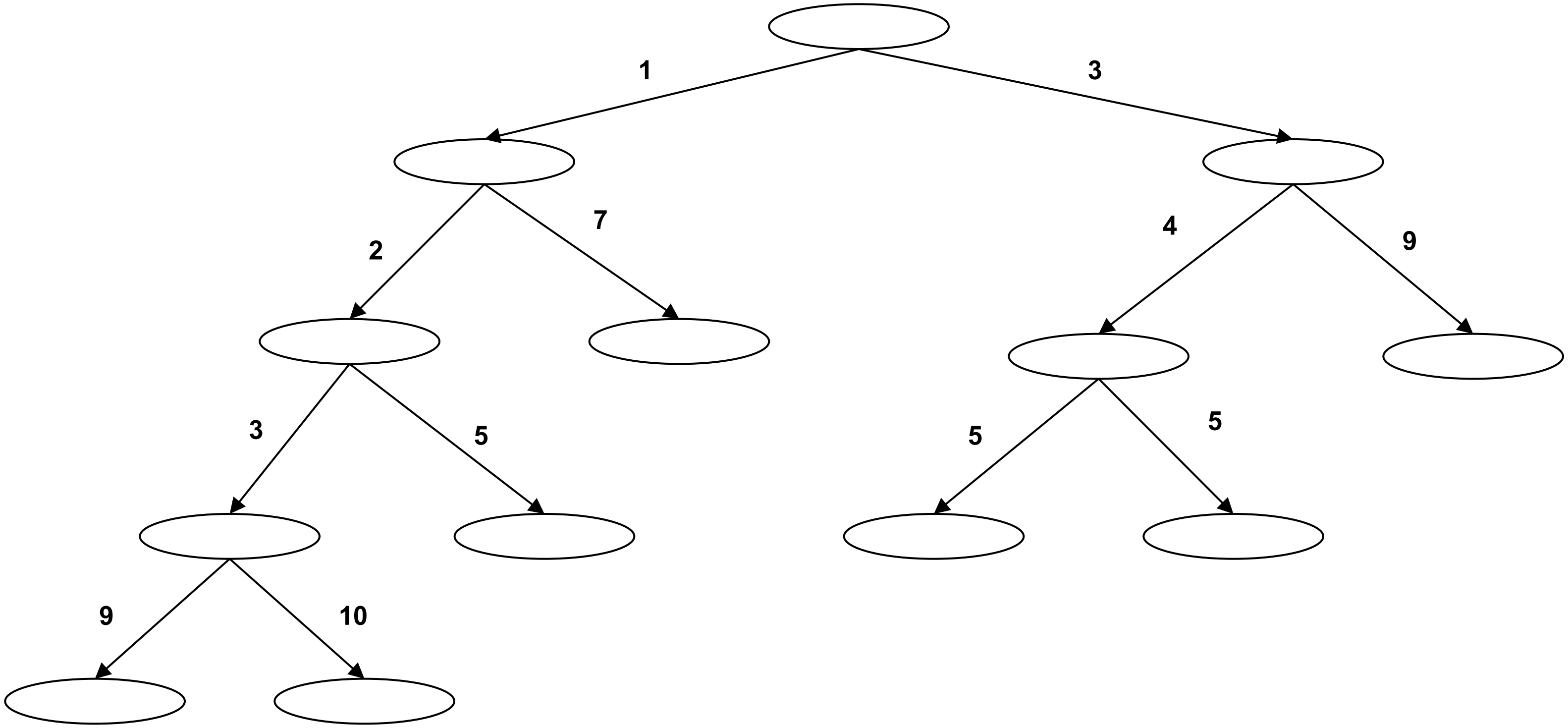


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

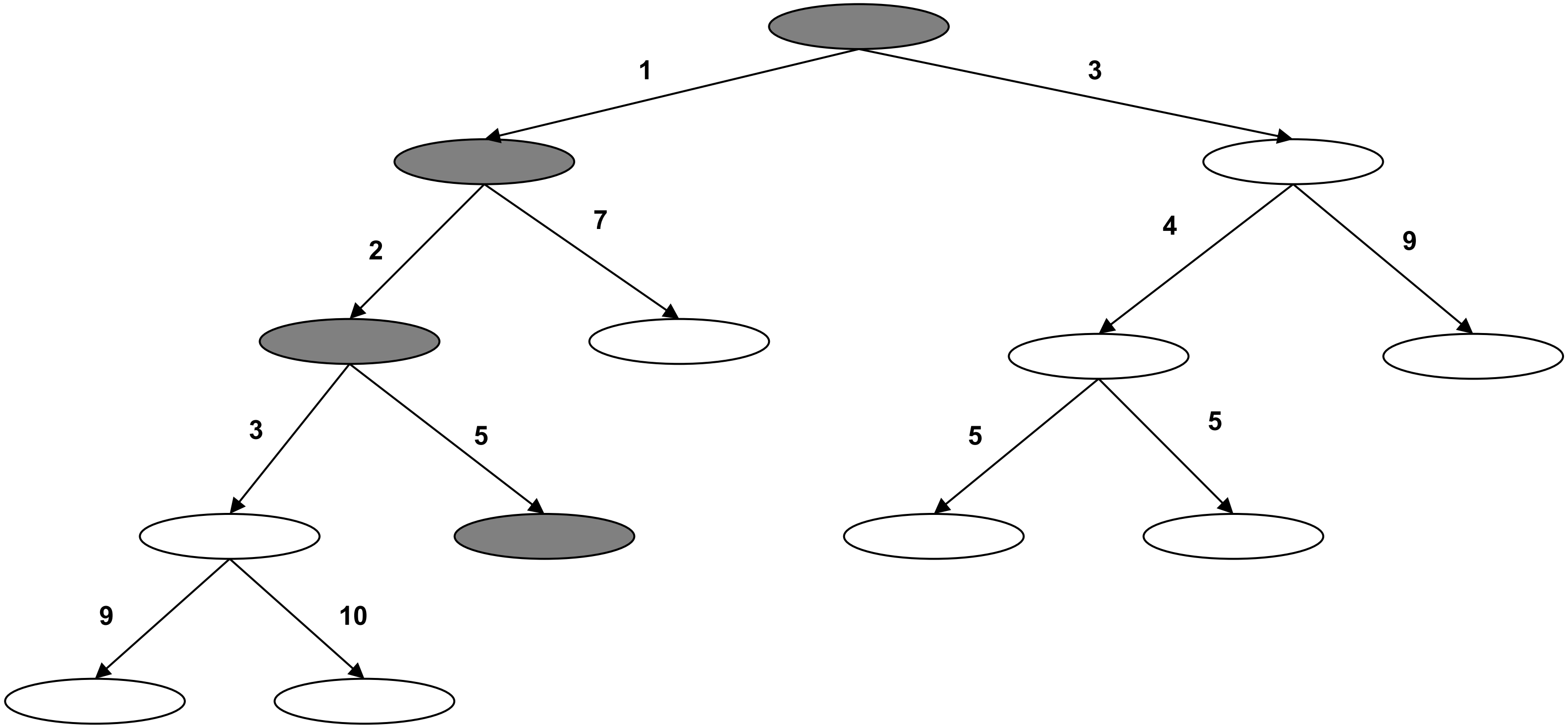


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

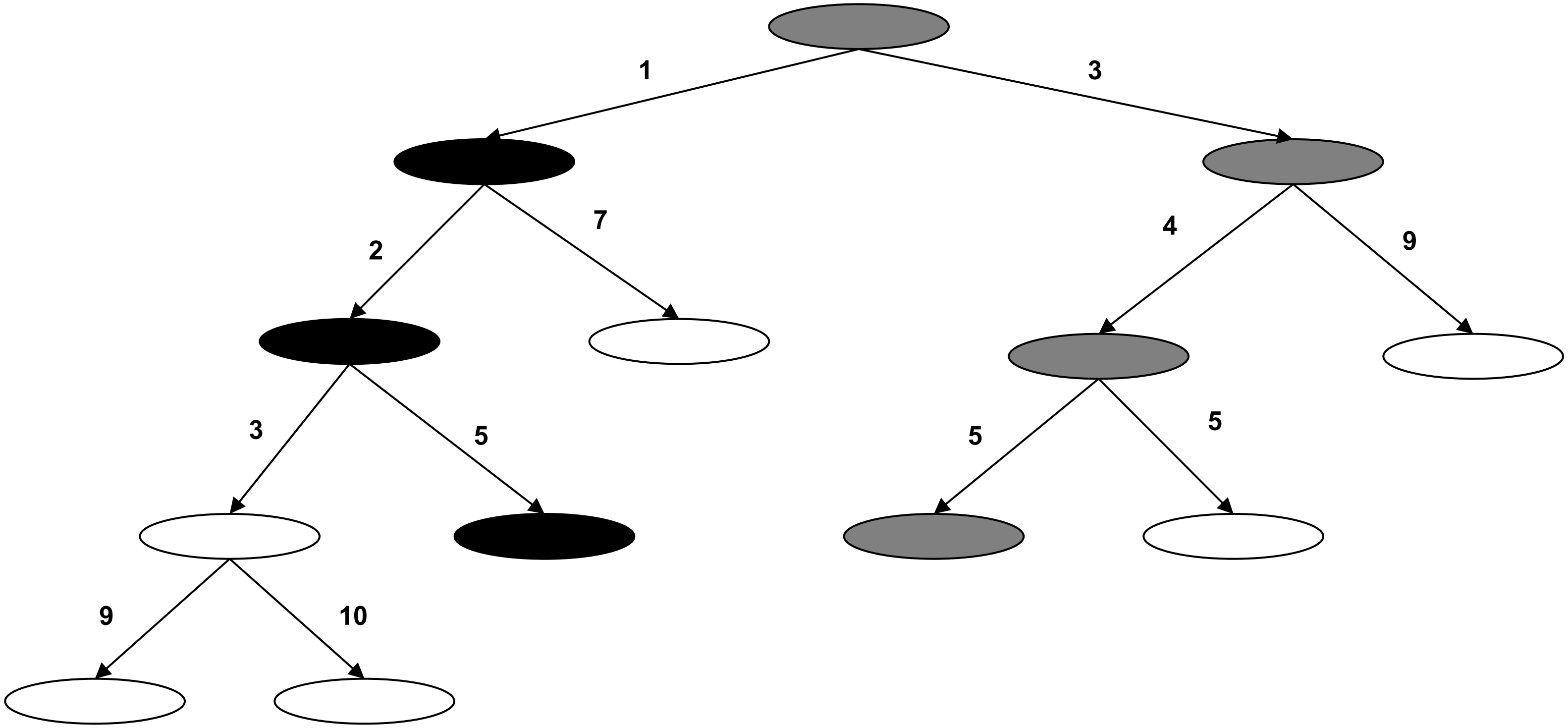


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.

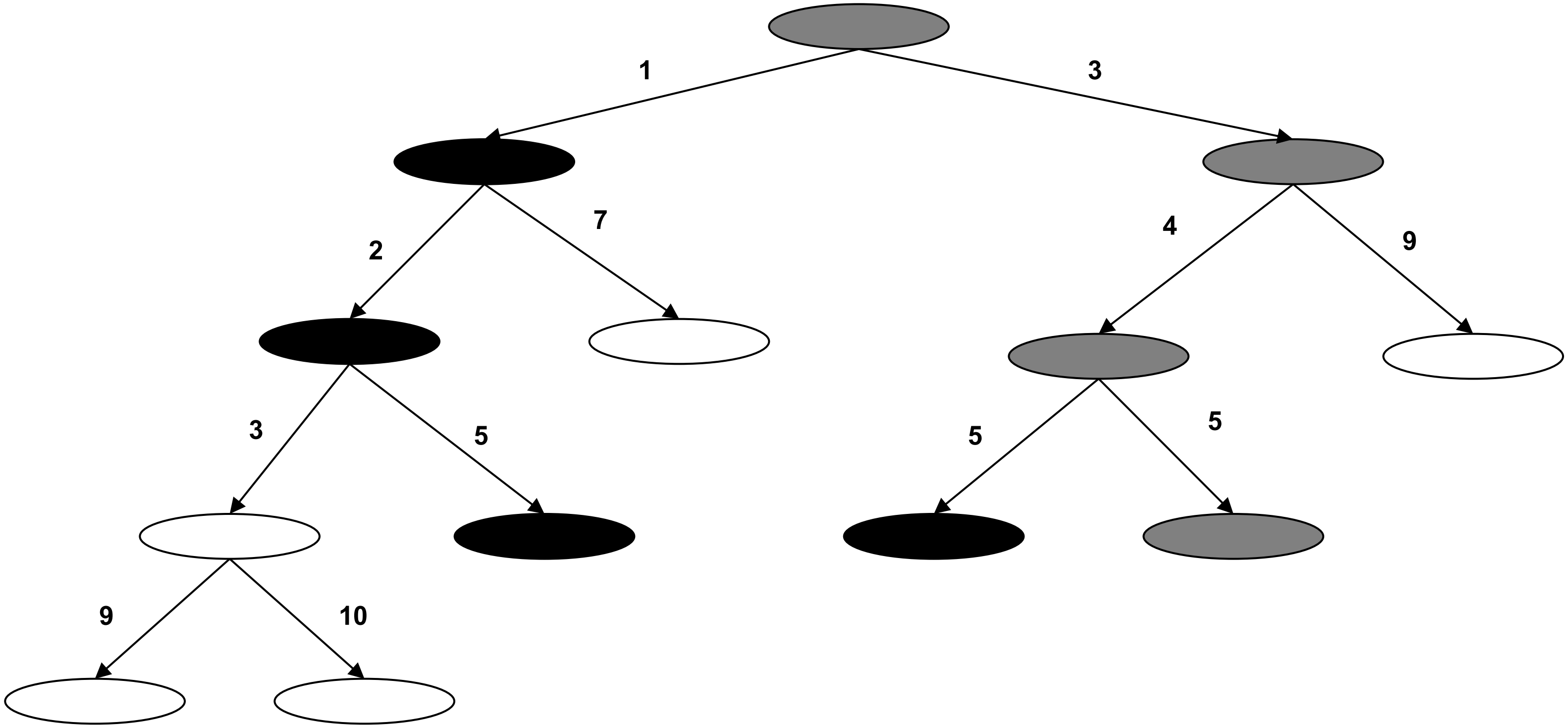


Search Heuristics with limited memory

Iterative–Deepening A* (IDA*)

The search is done as in the uninformed Iterative Deepening.

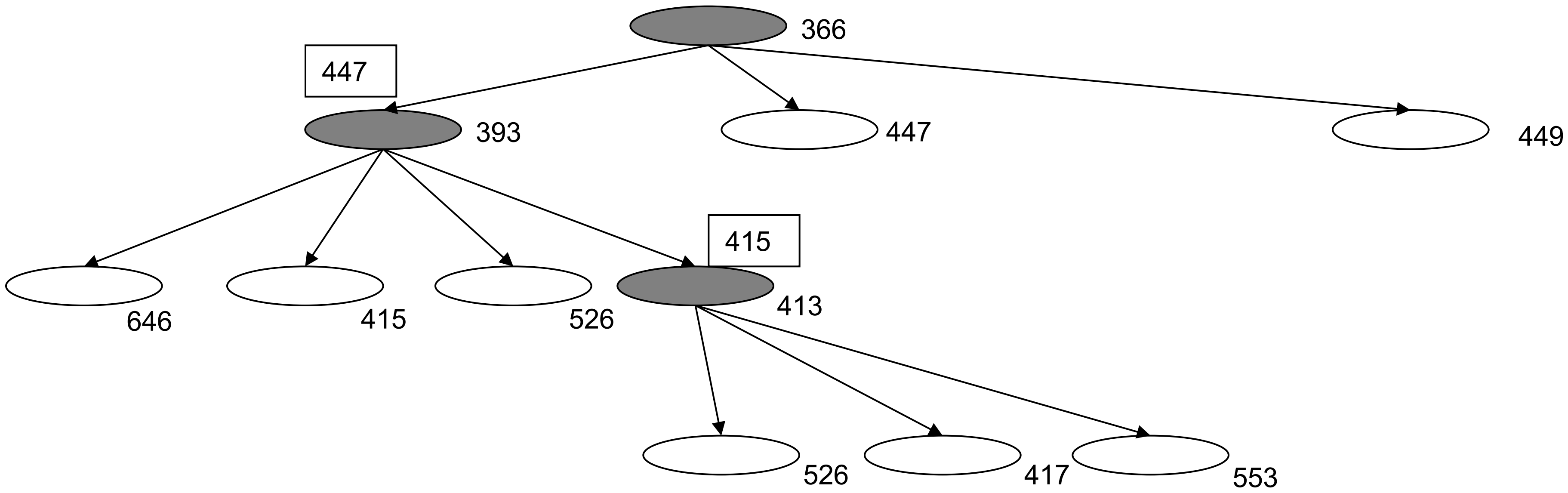
In IDA*, to determine the next node to expand, the node with minimum f(n) is chosen instead of the one with minimum depth.



Search Heuristics with limited memory

Recursive Best-First Search (RBFS)

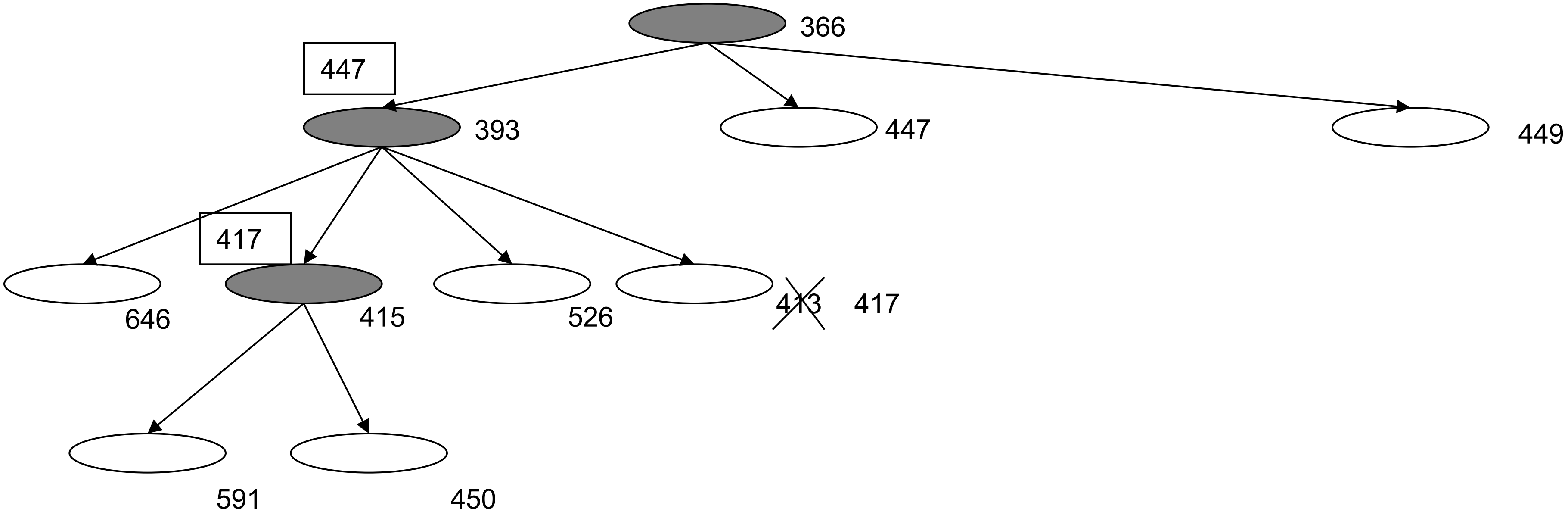
The f-value of the best alternative path starting from any of the current node's ancestors is tracked



Search Heuristics with limited memory

Recursive Best-First Search (RBFS)

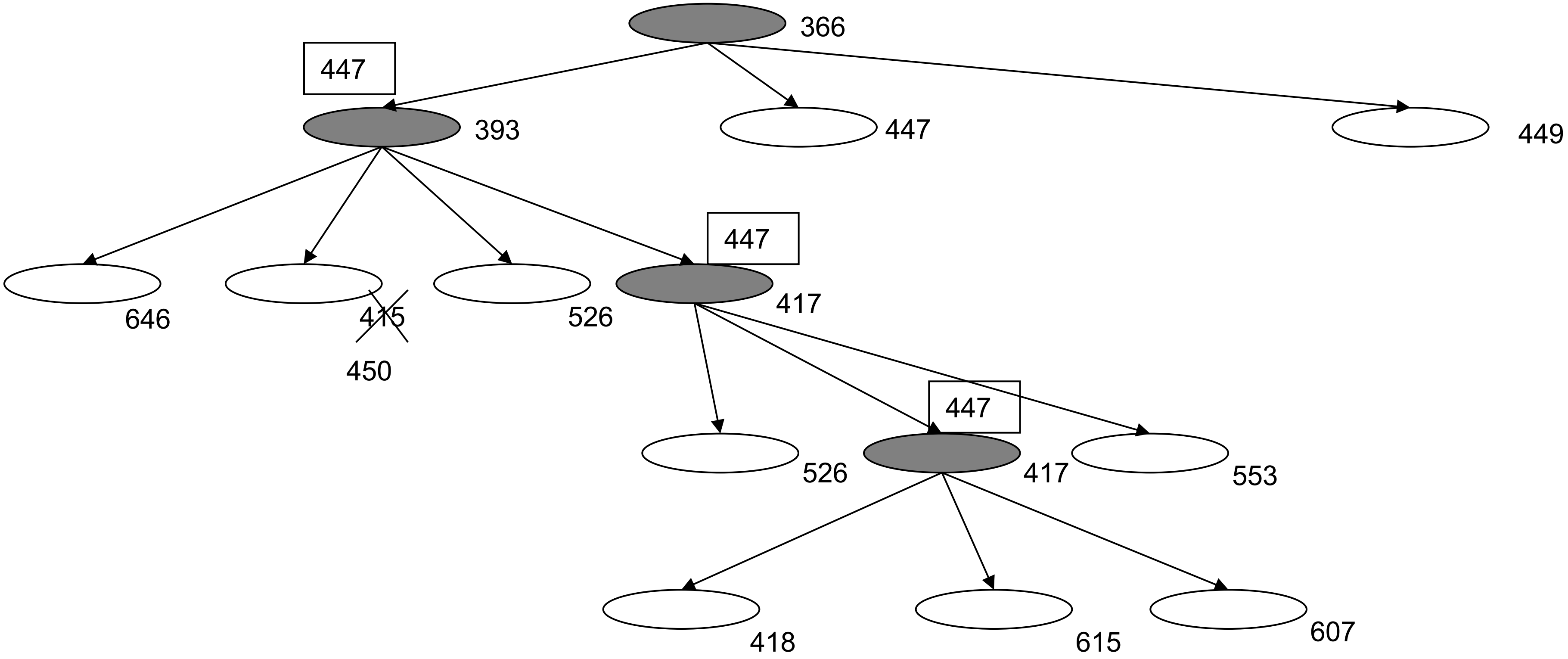
The f-value of the best alternative path starting from any of the current node's ancestors is tracked



Search Heuristics with limited memory

Recursive Best-First Search (RBFS)

The f-value of the best alternative path starting from any of the current node's ancestors is tracked



Search Heuristics with limited memory

(simplified) Memory-Bounded A* (SMA*)

It behaves like A* as long as it has memory available. When it runs out of memory, SMA* discards the worst node (smallest f-value) and continues expanding.

In the parent node of the discarded node it retains its value.

When all leaves have the same f-value, SMA* deletes the oldest leaf and expands the youngest.