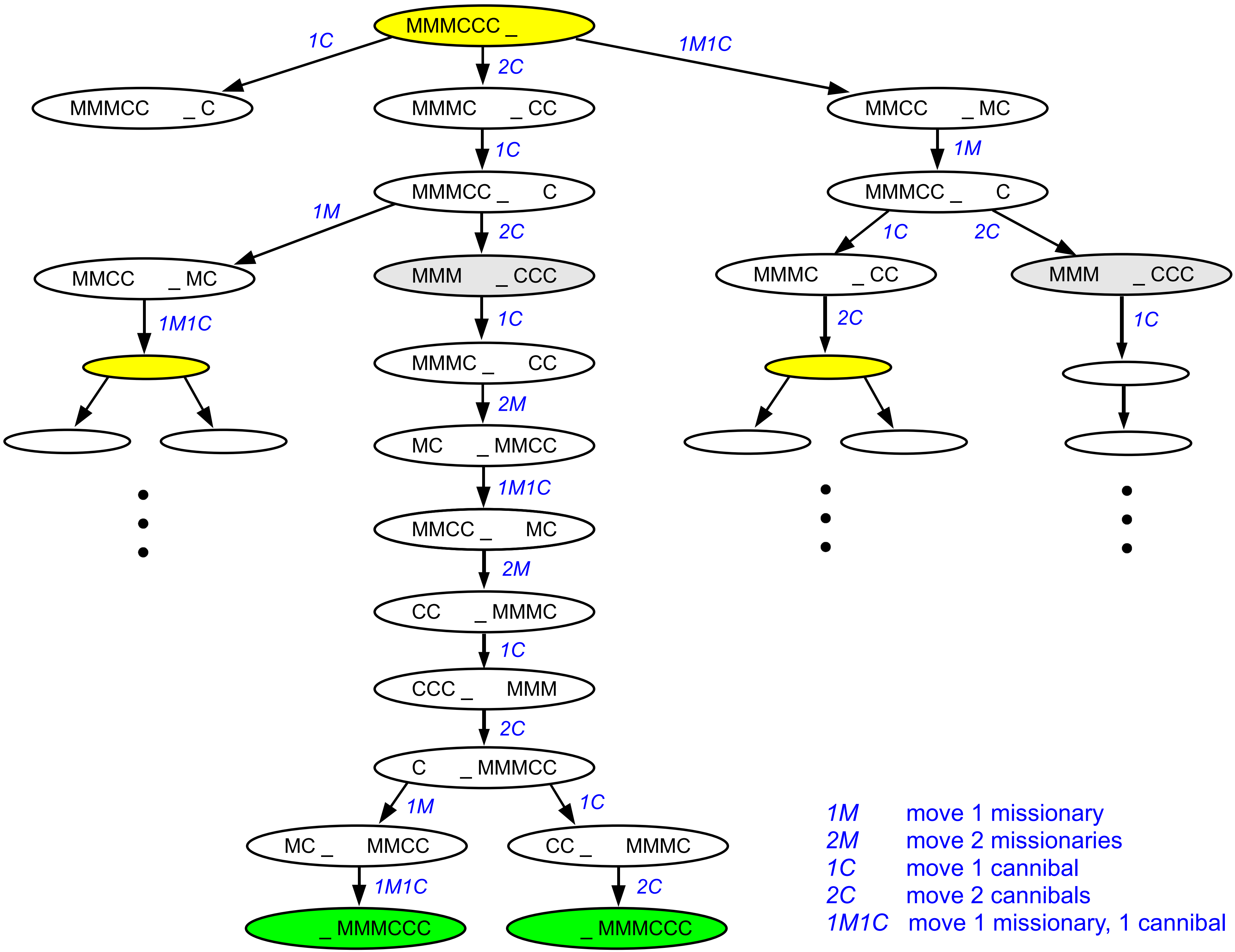
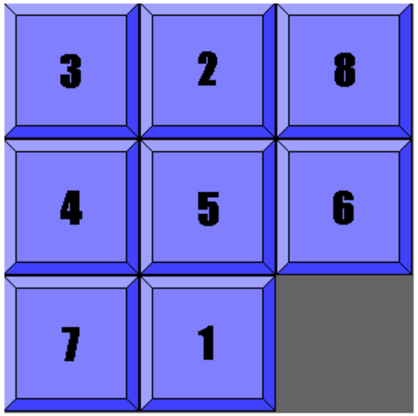
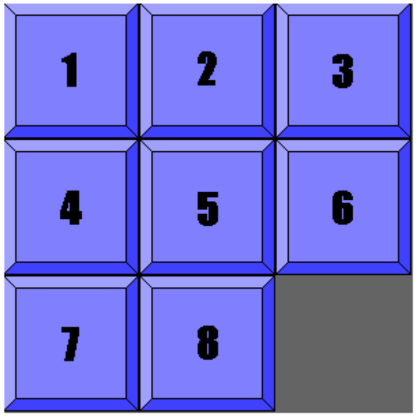


MMMCCC _
1C
2C
1M1C
MMMCC _ C
MMMC _ CC
MMCC _ MC
1C
1M
MMMCC _ C
MMMCC _ C
1M
2C
1C
2C
MMCC _ MC
MMM _ CCC
MMMC _ CC
MMM _ CCC
1M1C
1C
2C
1C
MMMC _ CC
2M
MC _ MMCC
1M1C
MMCC _ MC
2M
CC _ MMMC
1C
CCC _ MMM
2C
C _ MMMCC
1M
1C
MC _ MMCC
CC _ MMMC
1M1C
2C
_ MMMCCC
_ MMMCCC
1M move 1 missionary
2M move 2 missionaries
1C move 1 cannibal
2C move 2 cannibals
1M1C move 1 missionary, 1 cannibal

The Eight-Puzzle

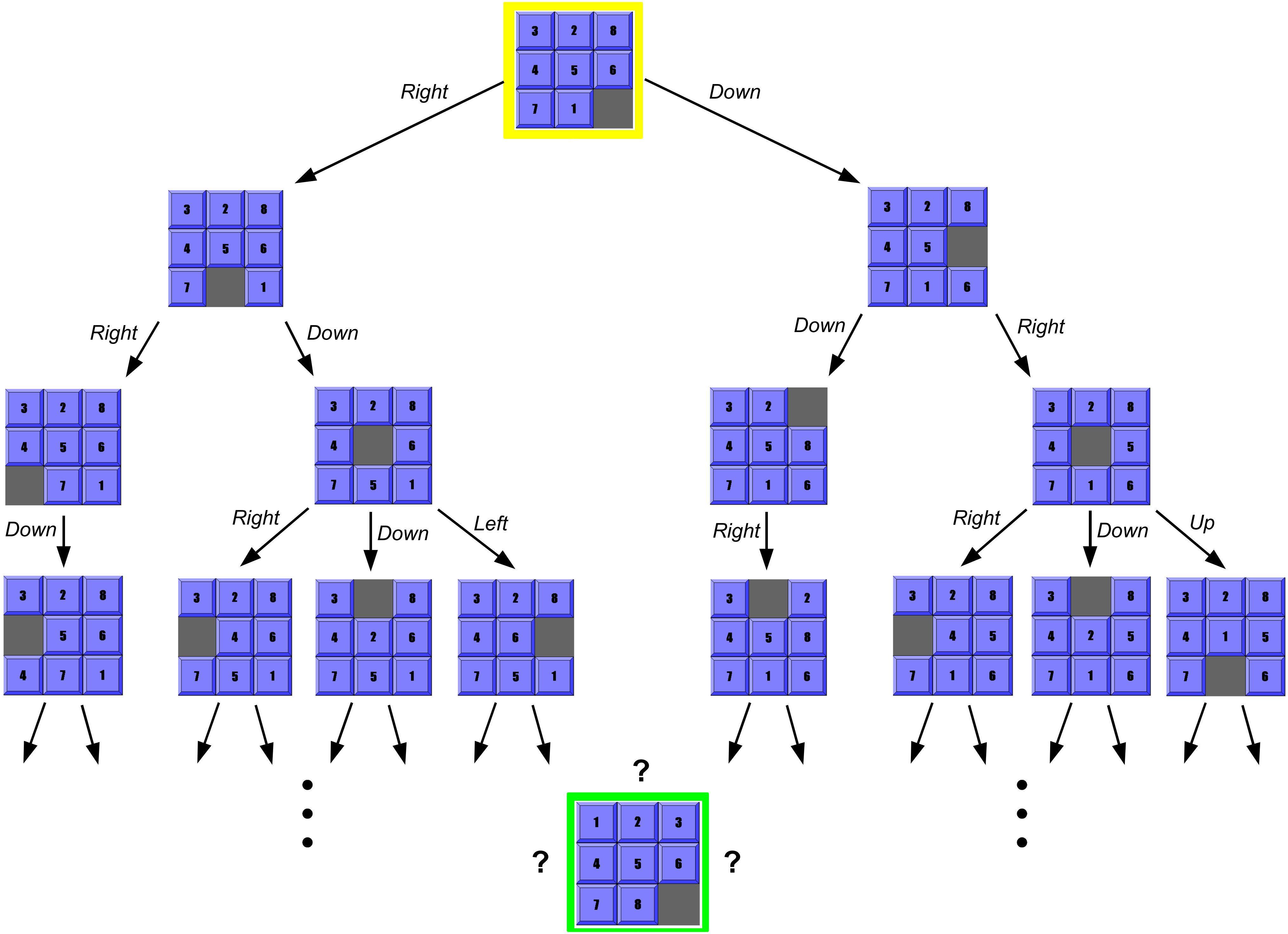


Initial state



Goal state

- State representation: **(3 2 8 4 5 6 7 1 _)**
- Operators: *Up, Down, Left, Right*
- Example: **(3 2 8 4 5 6 7 1 _)** $\Rightarrow$ *Down* **(3 2 8 4 5 _ 7 1 6)**
- M&C problem has < 30 possible states
- 8-puzzle has 9! possible states = 362,880
- 15-puzzle has 16! possible states = 20,922,789,888,000
- Demo



Right
Down
Right
Down
Down
Right
Down
Right
Down
Left
Right
Right
Down
Up
?
?
?

Heuristic Function 1

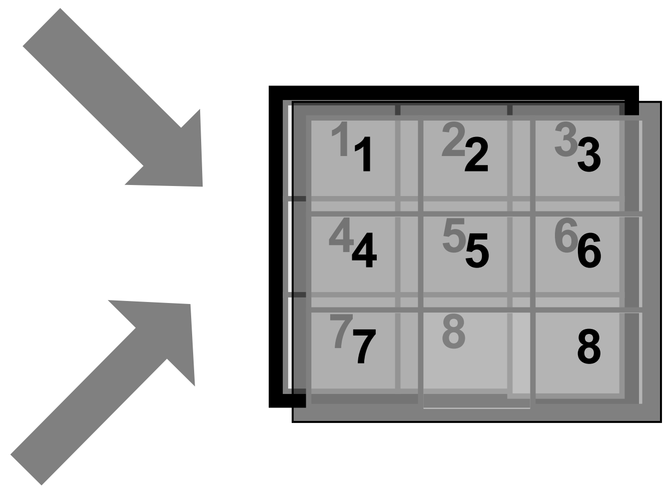
Number of misplaced tiles
(not including the blank)

Current State

1	2	3
4	5	6
7		8

Goal State

1	2	3
4	5	6
7	8	

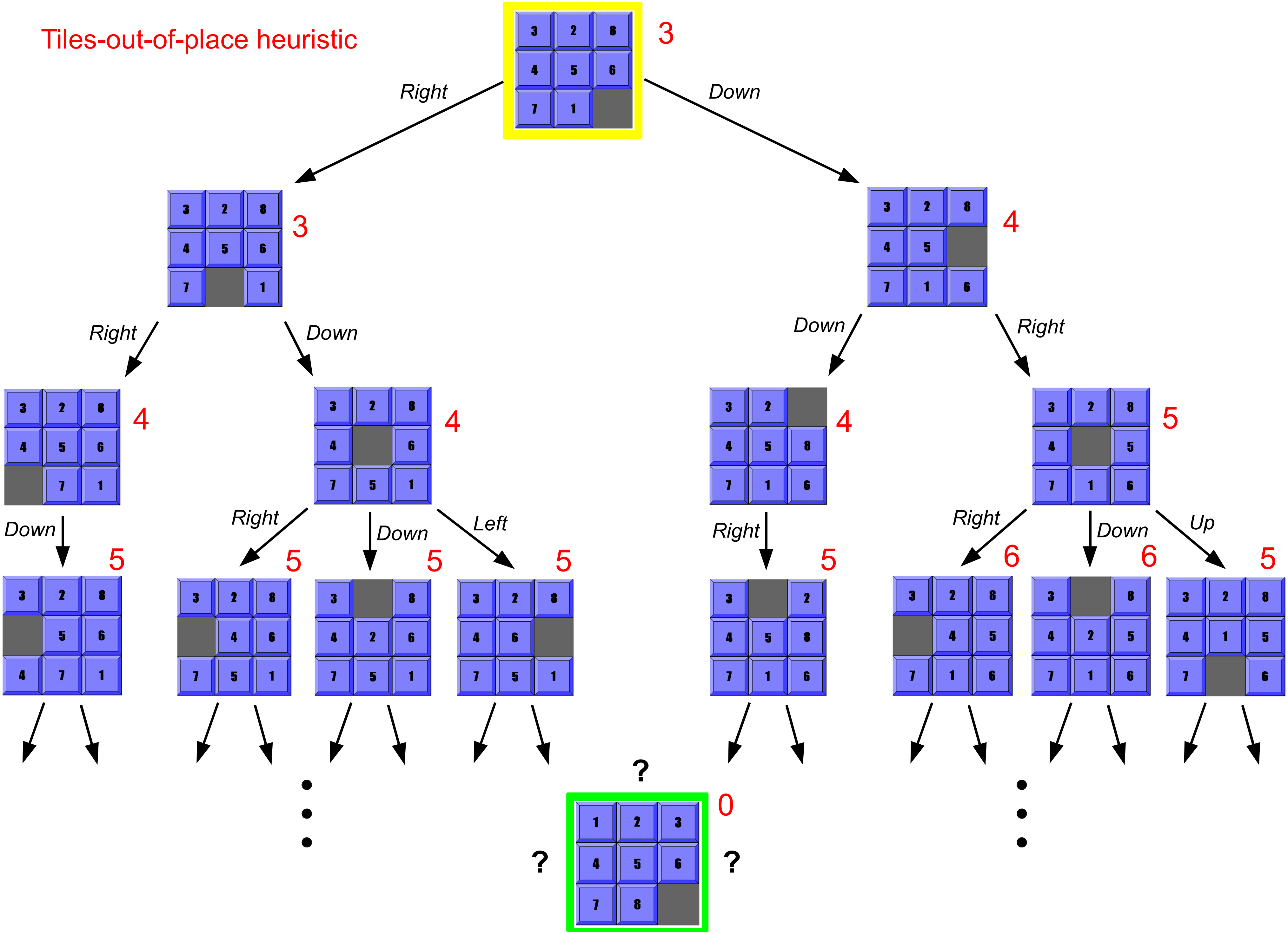


N	N	N
N	N	N
N	Y	

Only tile **8** is misplaced, so heuristic estimate = 1

In other words, the heuristic is telling us that it thinks a solution might be reached in 1 more move.

Notation: *h*(n) *h*(current state) = 1



Tiles-out-of-place heuristic
3
Right
Down
3
4
Right
Down
Down
Right
4
4
4
5
Down
Right
Down
Left
Right
Right
Down
Up
5
5
5
5
5
6
6
5
?
?
?
0

Heuristic Function 2

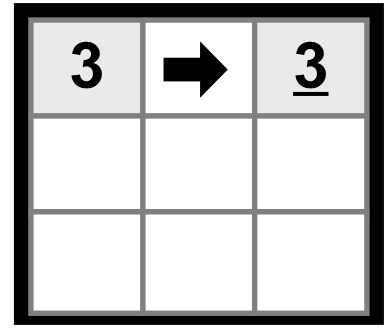
Manhattan Distance
(not including the blank)

Current State

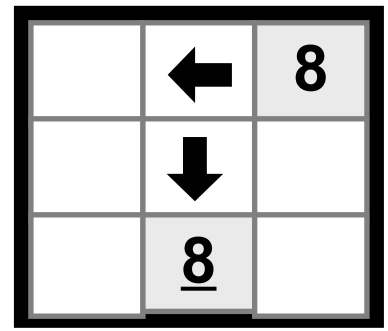
3	2	8
4	5	6
7	1	

Goal State

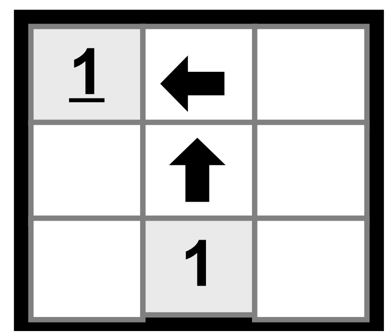
1	2	3
4	5	6
7	8	



2 spaces



3 spaces



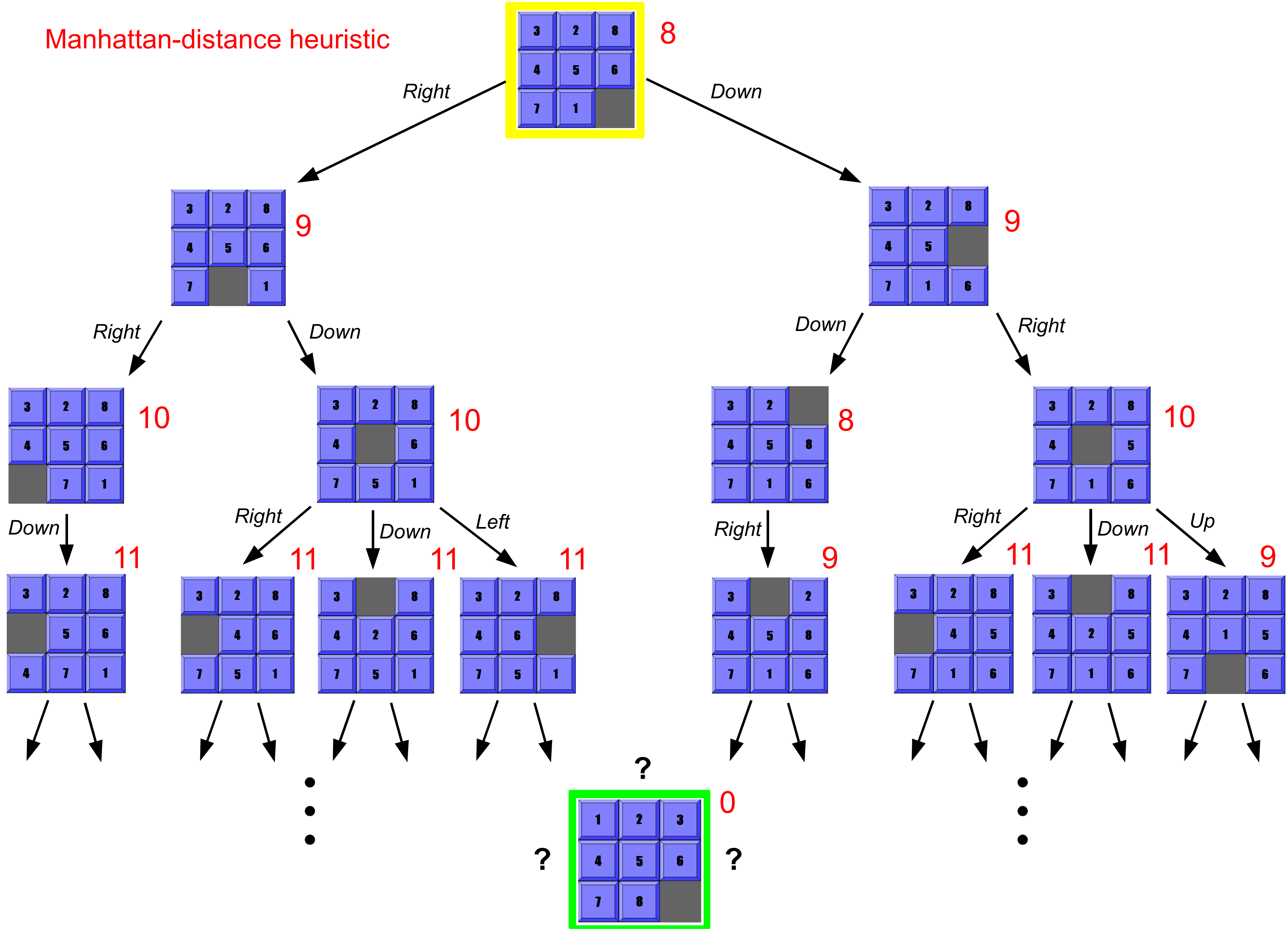
3 spaces

Total = 8

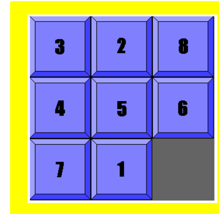
Only tiles **3**, **8** and **1** are misplaced (by 2, 3, and 3 spaces, respectively) so the heuristic function evaluates to 8

In other words, the heuristic is telling us that it thinks a solution is reachable in 8 more moves.

Notation: *h*(n) *h*(current state) = 8

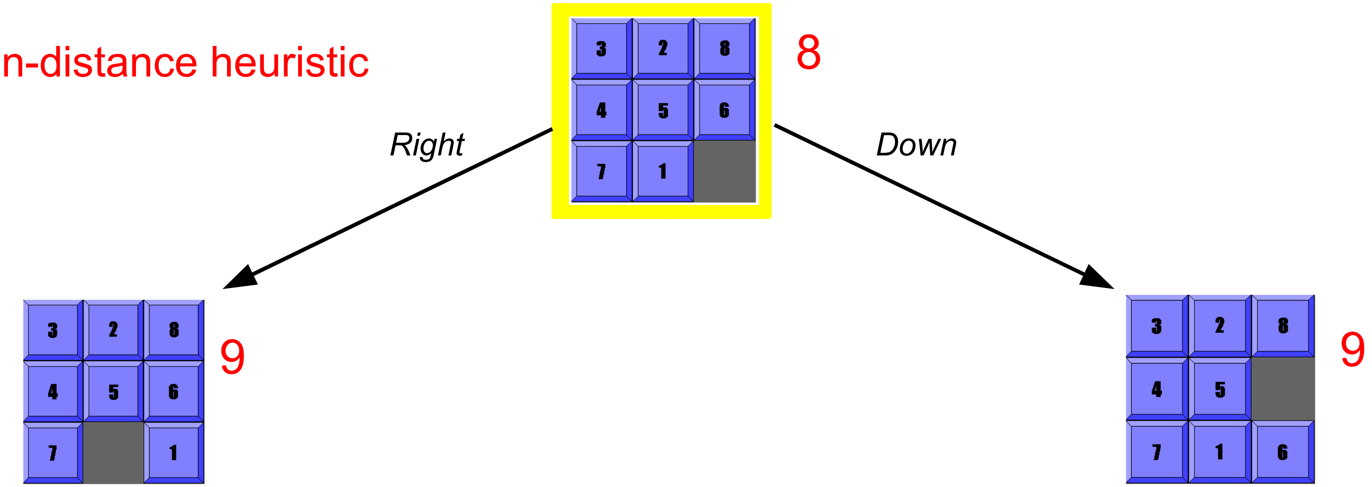

Manhattan-distance heuristic
8
Right
Down
9
9
Right
Down
Down
Right
10
10
8
10
Down
Right
Down
Left
Right
Right
Down
Up
11
11
11
11
9
11
11
9
?
0
?
?

Manhattan-distance heuristic

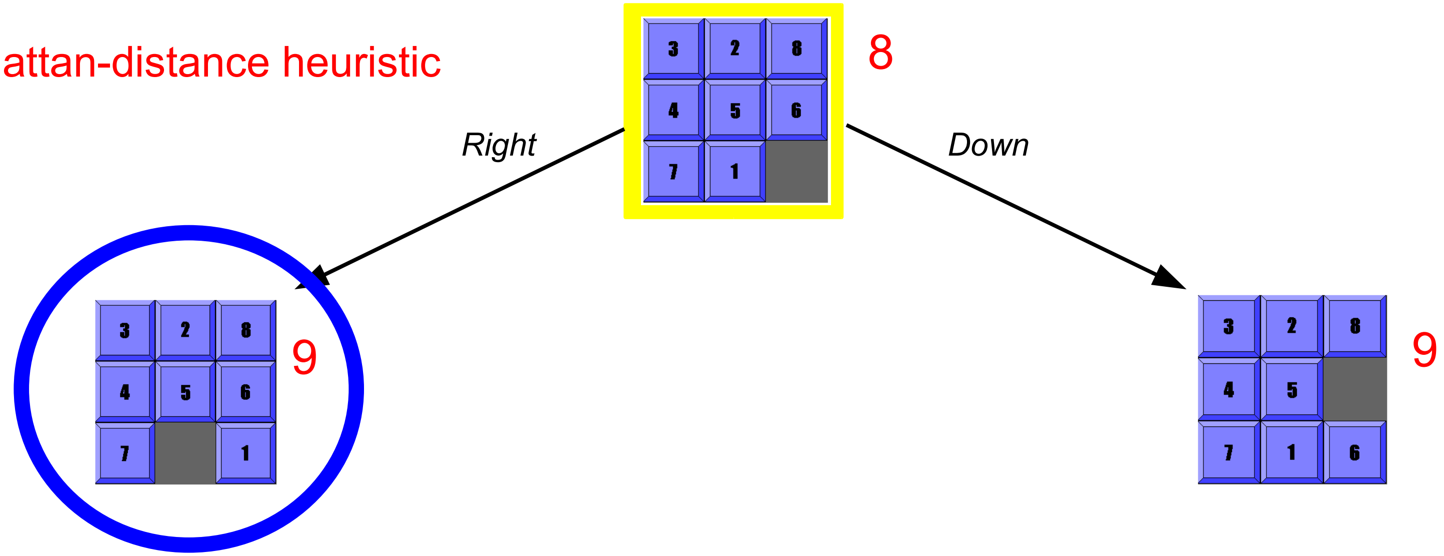


8

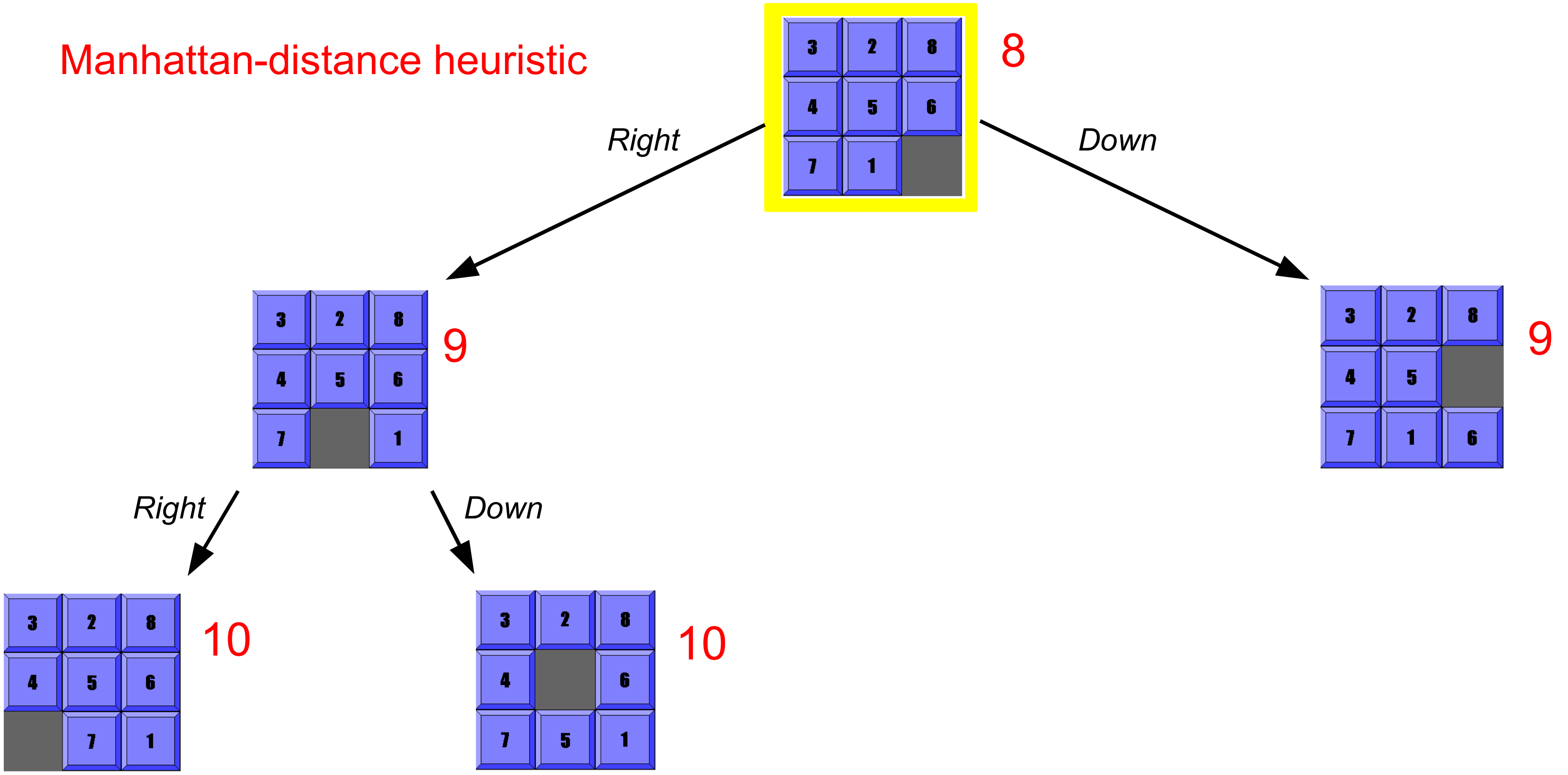
Manhattan-distance heuristic



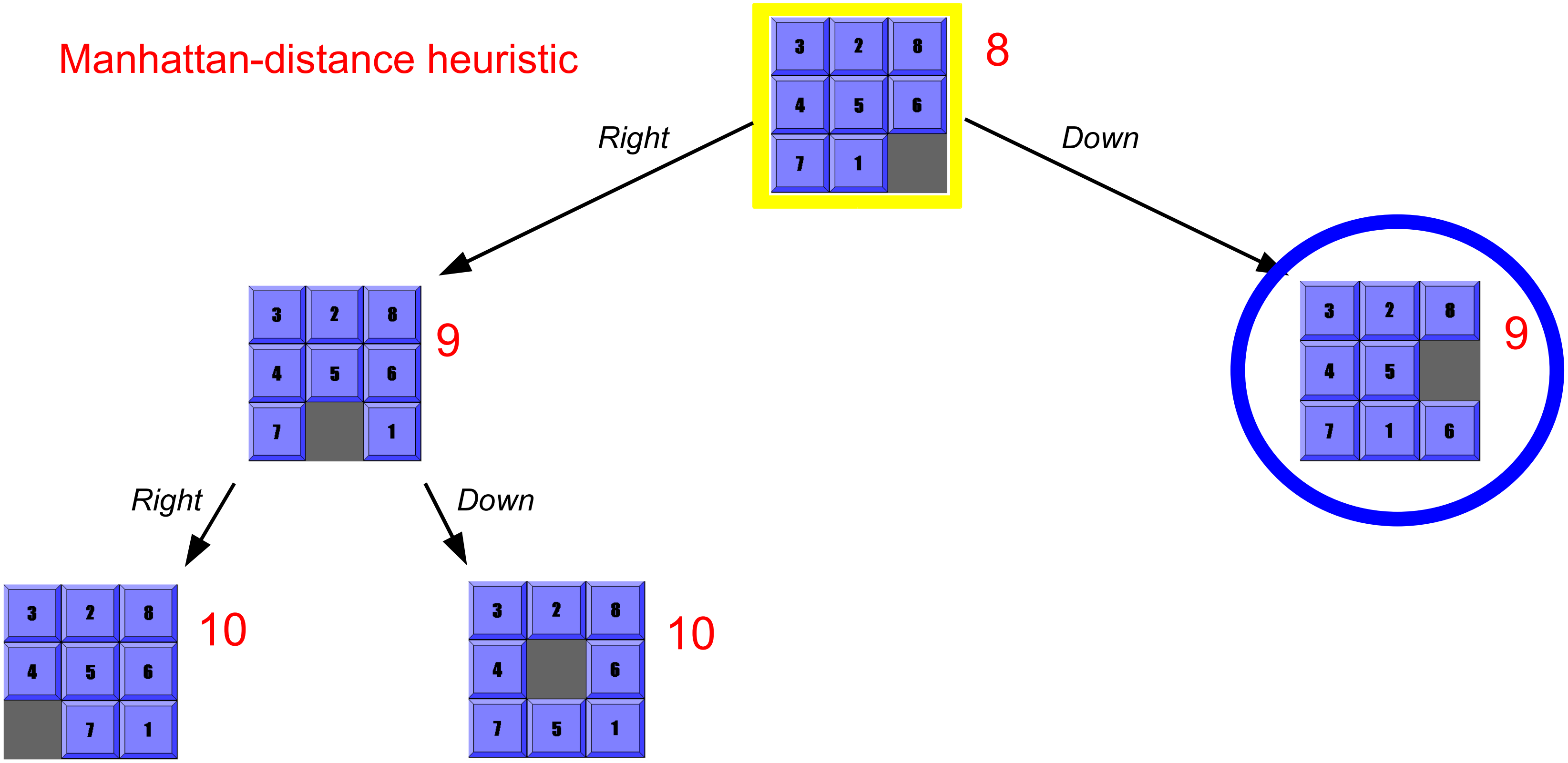
Manhattan-distance heuristic



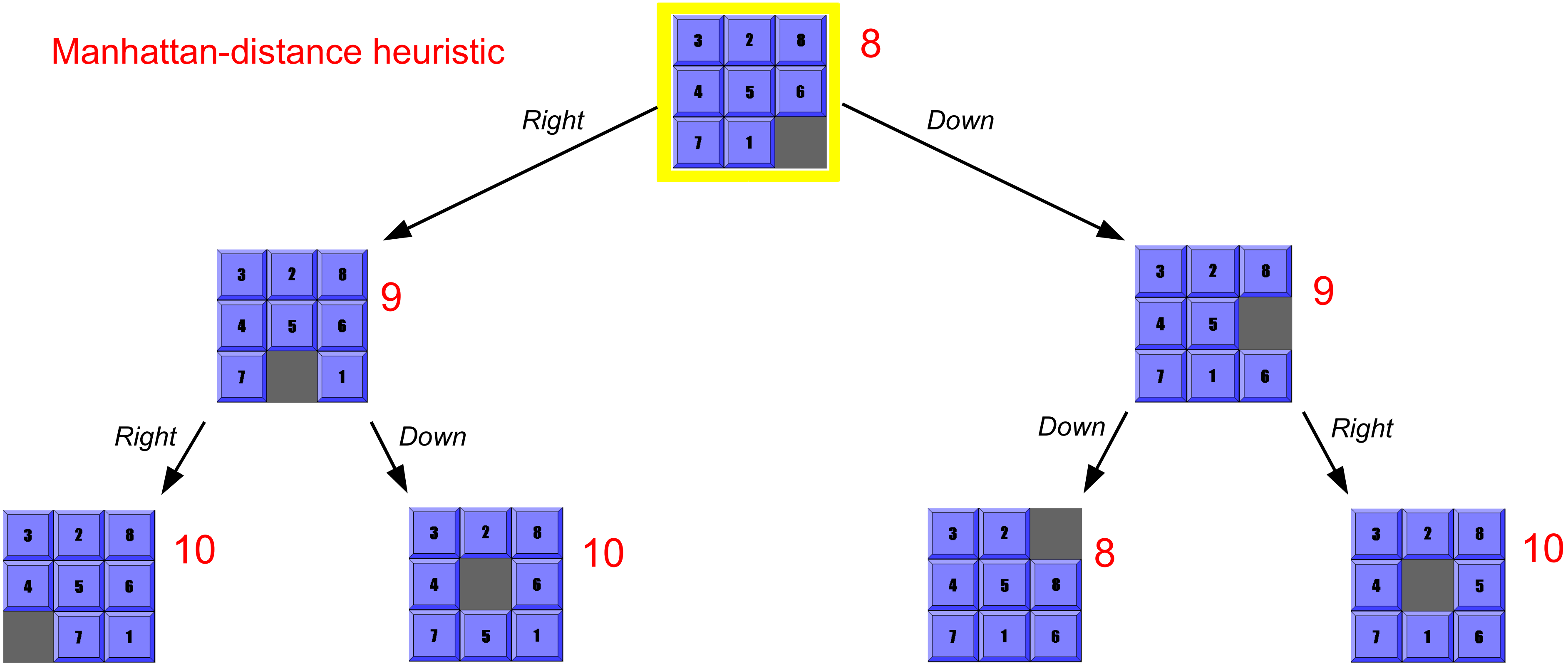
Manhattan-distance heuristic



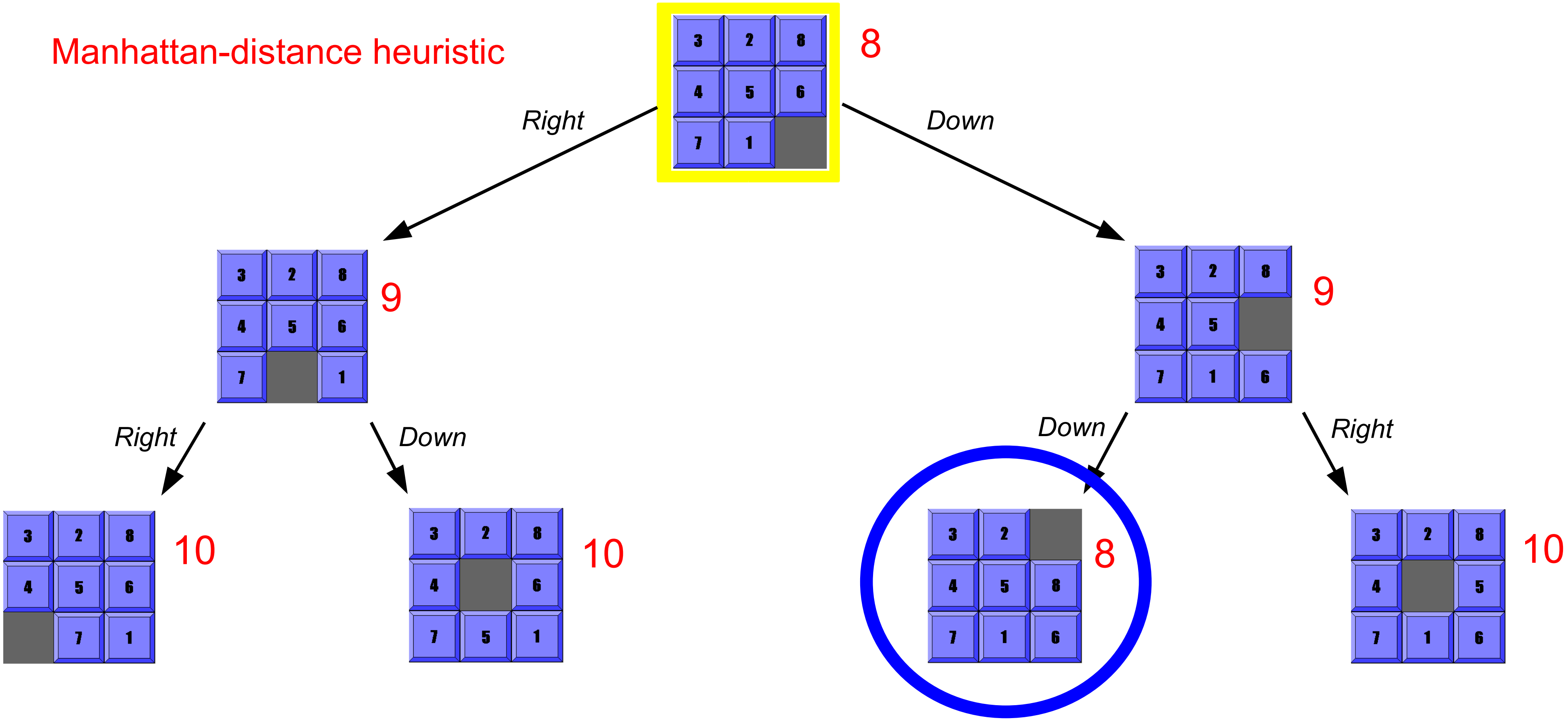
Manhattan-distance heuristic



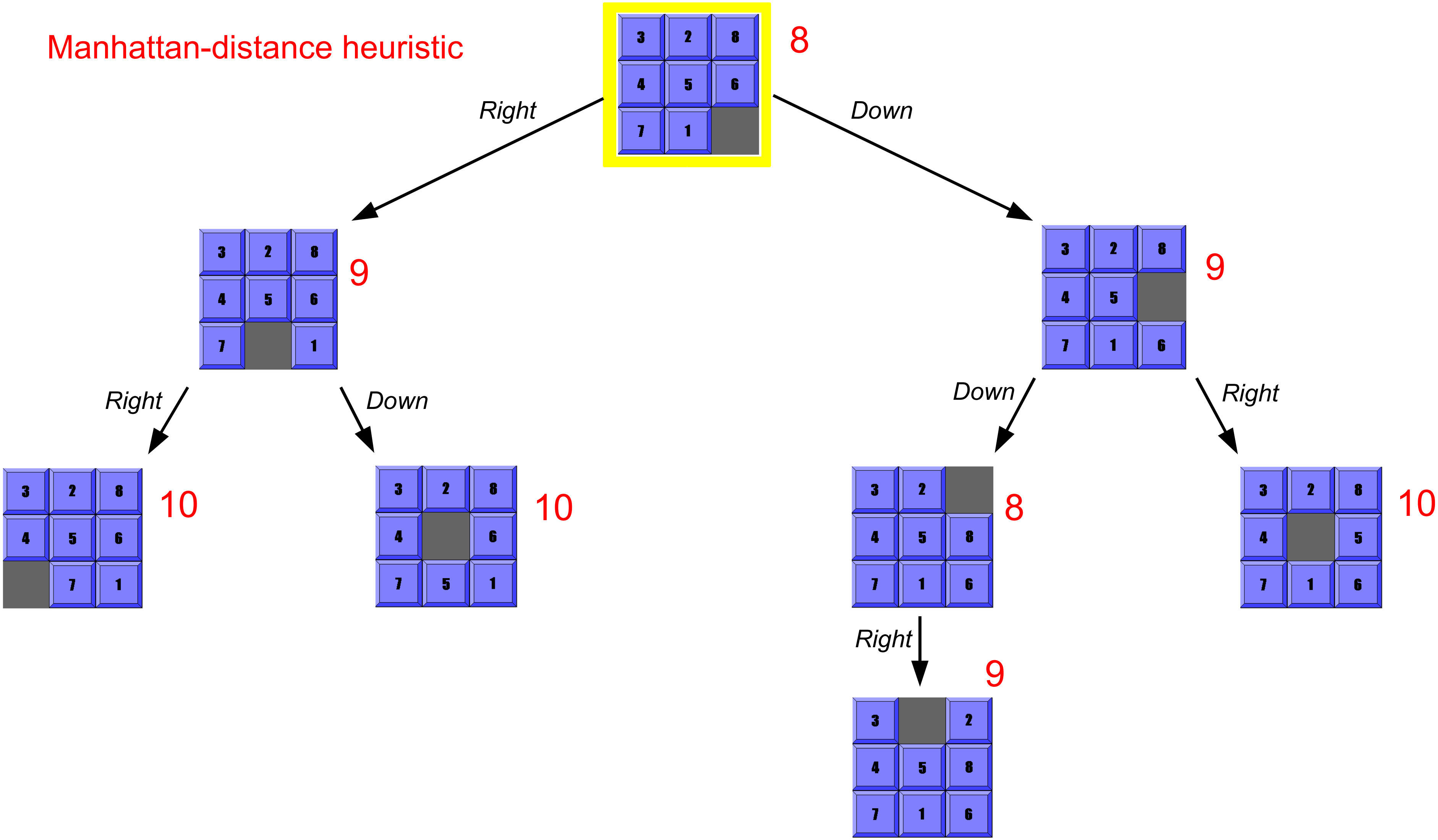
Manhattan-distance heuristic



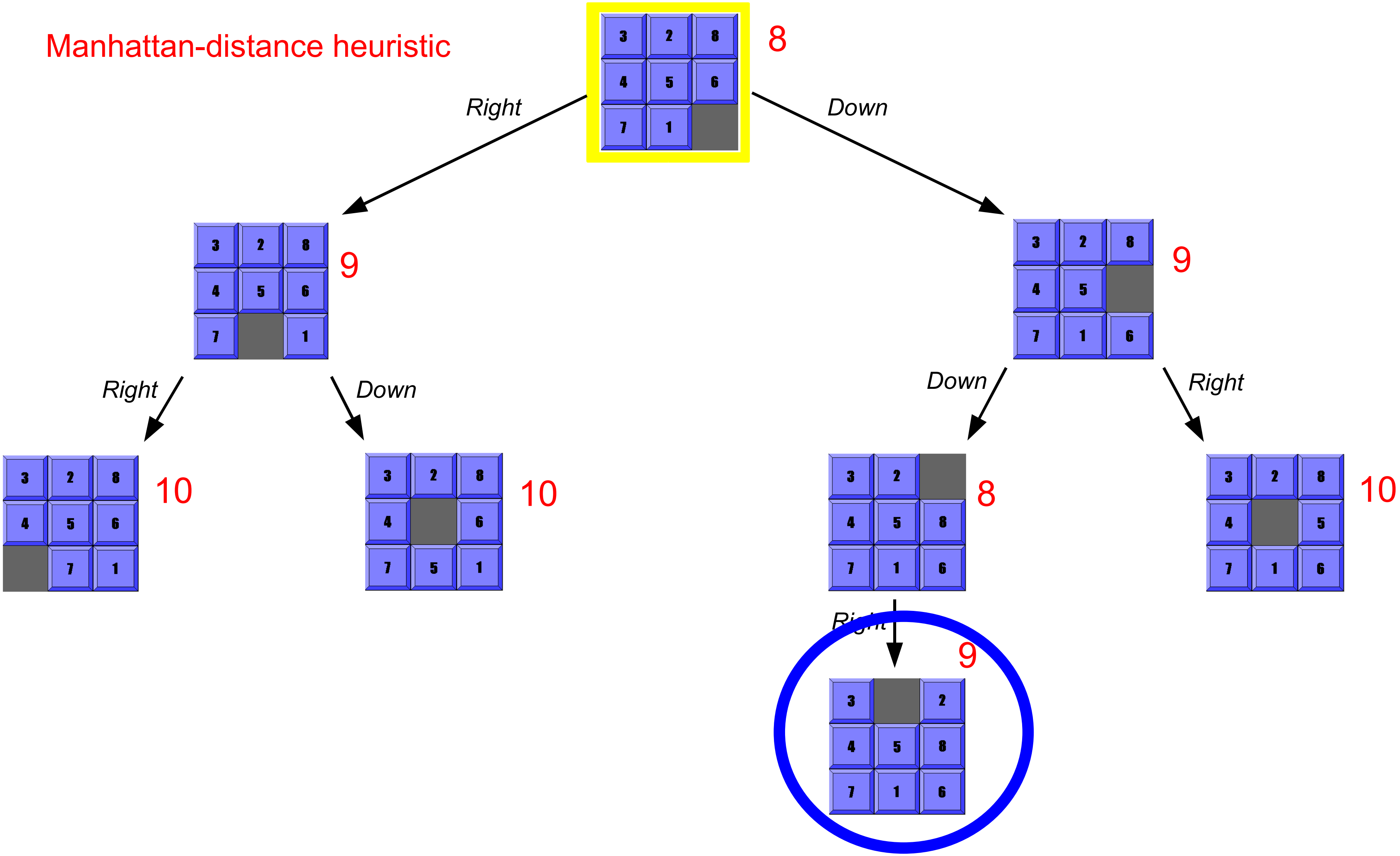
Manhattan-distance heuristic

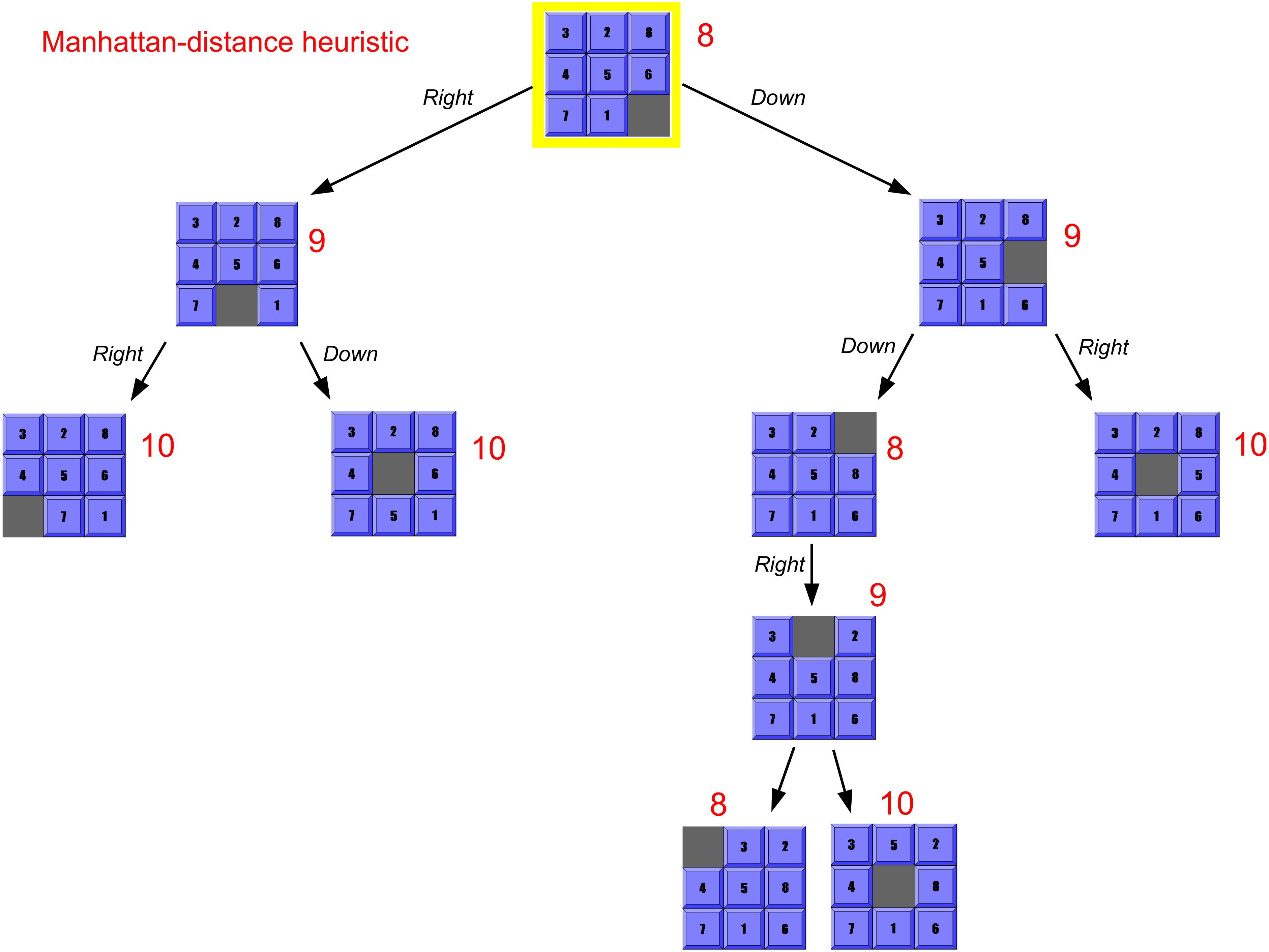


Manhattan-distance heuristic

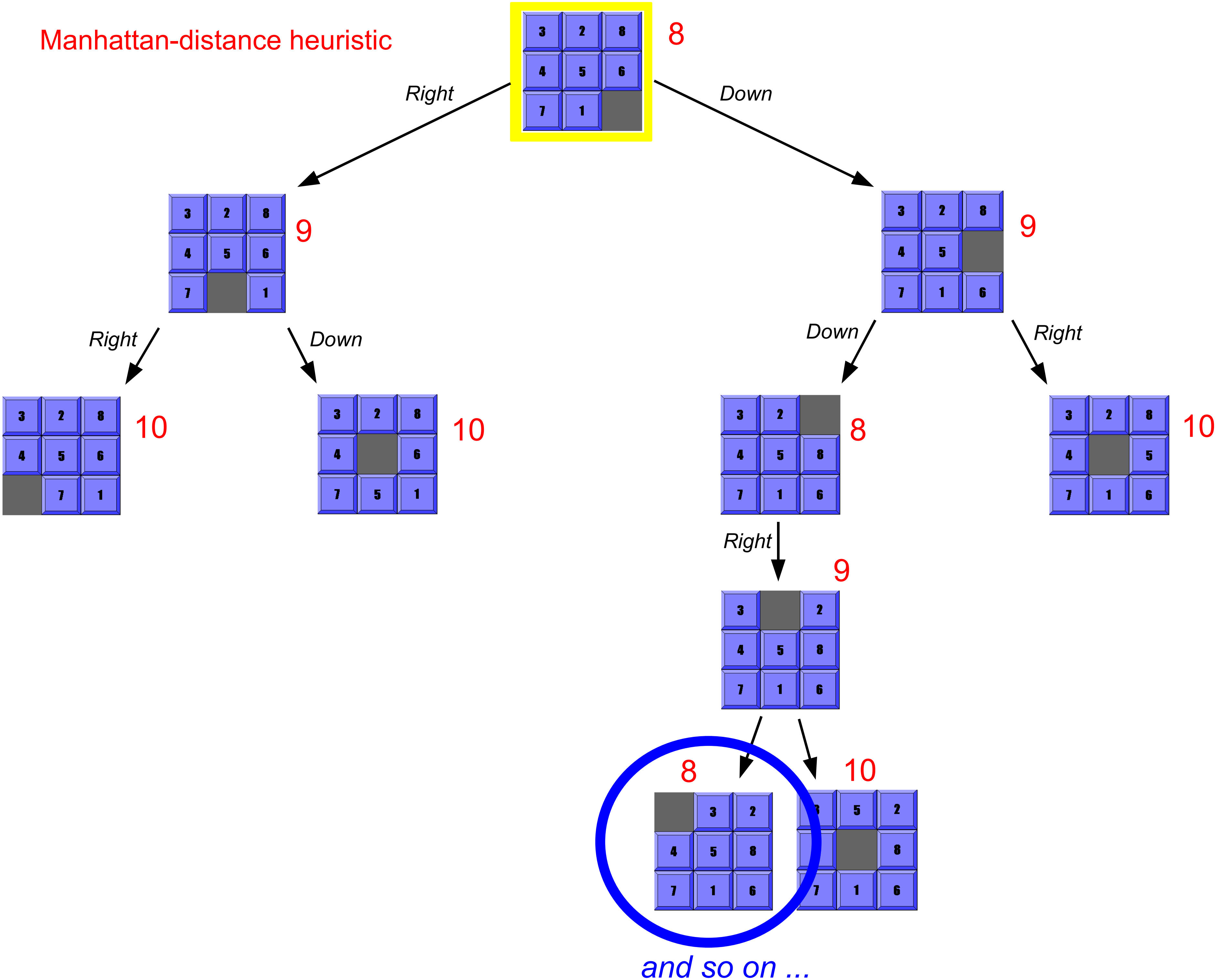


Manhattan-distance heuristic



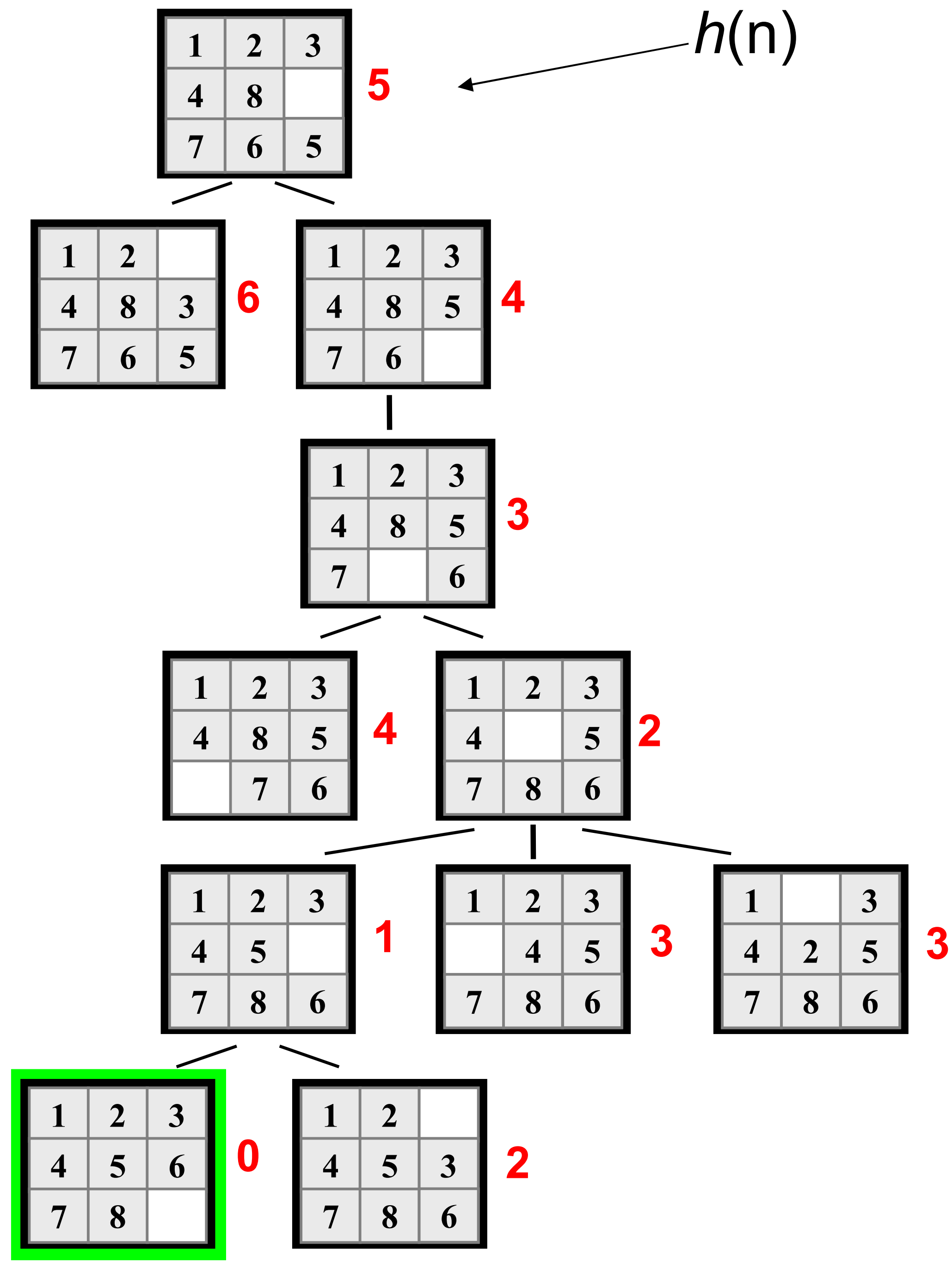


Manhattan-distance heuristic
3 2 8
4 5 6
7 1
8
Right
Down
3 2 8
4 5 6
7 1
9
3 2 8
4 5
7 1 6
9
Right
Down
Down
Right
3 2 8
4 5 6
7 1
10
3 2 8
4 6
7 5 1
10
3 2
4 5 8
7 1 6
8
3 2 8
4 5
7 1 6
10
Right
9
3 2
4 5 8
7 1 6
8
10
3 2
4 5 8
7 1 6
3 5 2
4 8
7 1 6



Manhattan-distance heuristic
8
Right
Down
9
9
Right
Down
Down
Right
10
10
8
10
Right
9
8
10
and so on ...

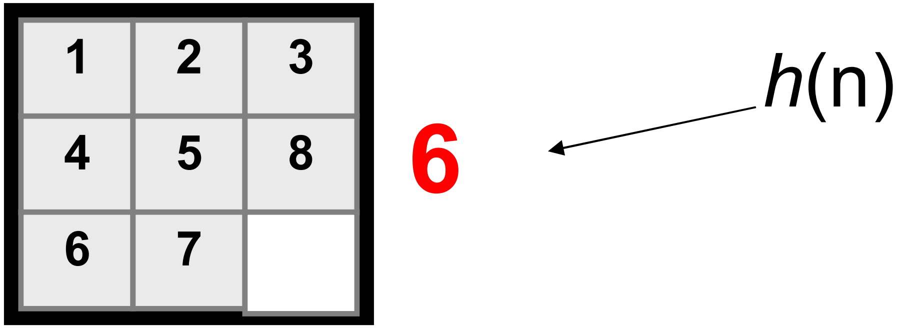
Always choosing the next state based on the lowest heuristic value is called **hill climbing**.



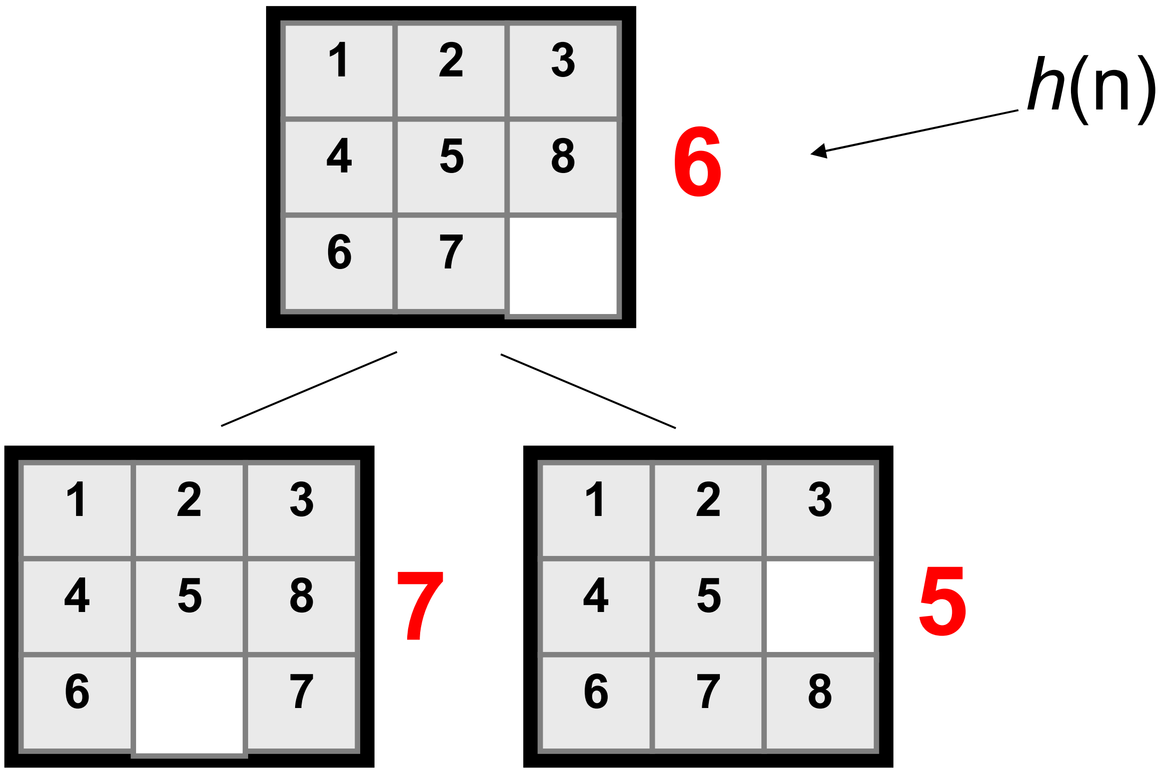
In this example, the Manhattan Distance heuristic guides us quickly to a solution.

But hill climbing has a problem...

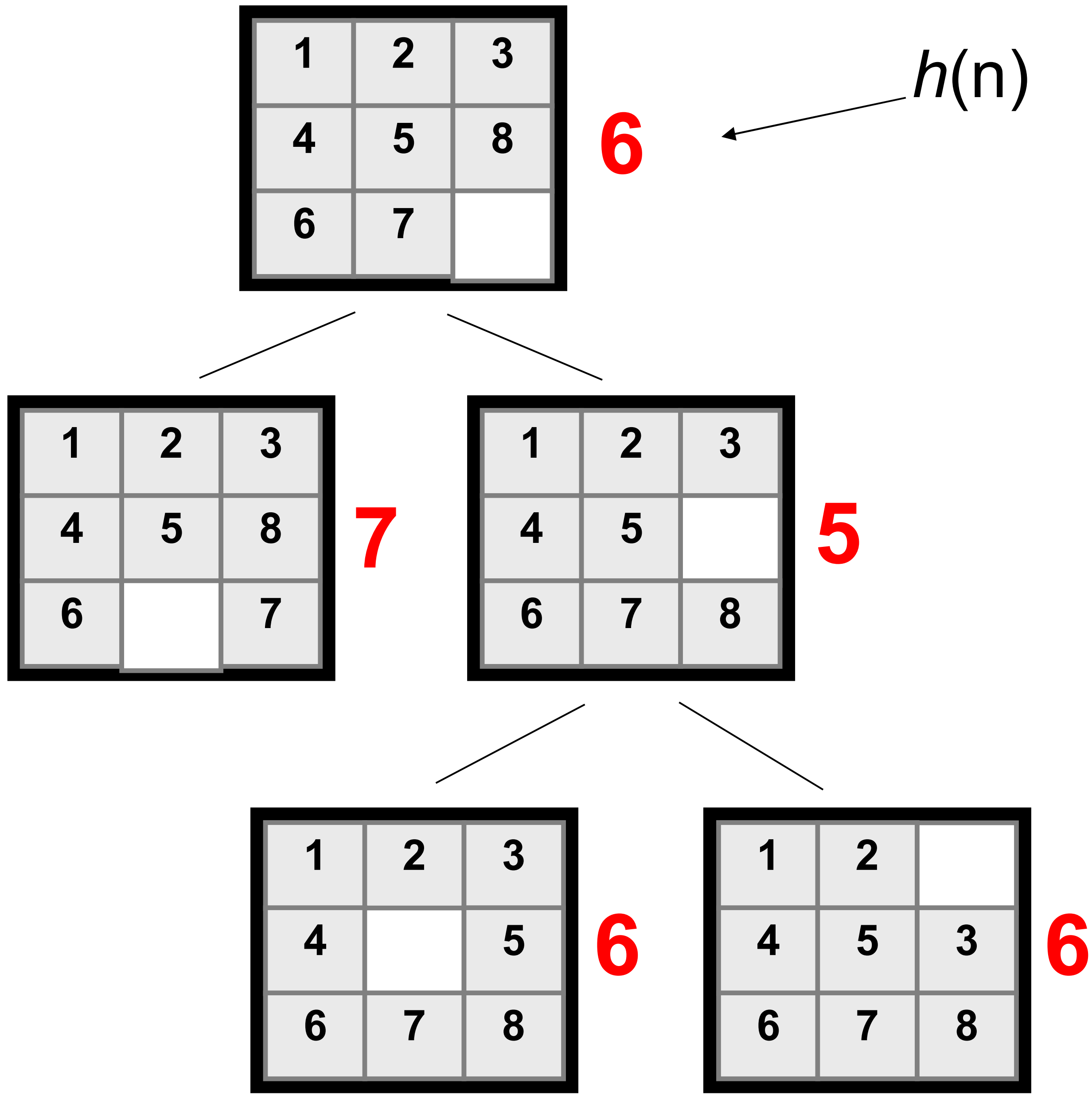
In this example, hill climbing does not work!



In this example, hill climbing does not work!



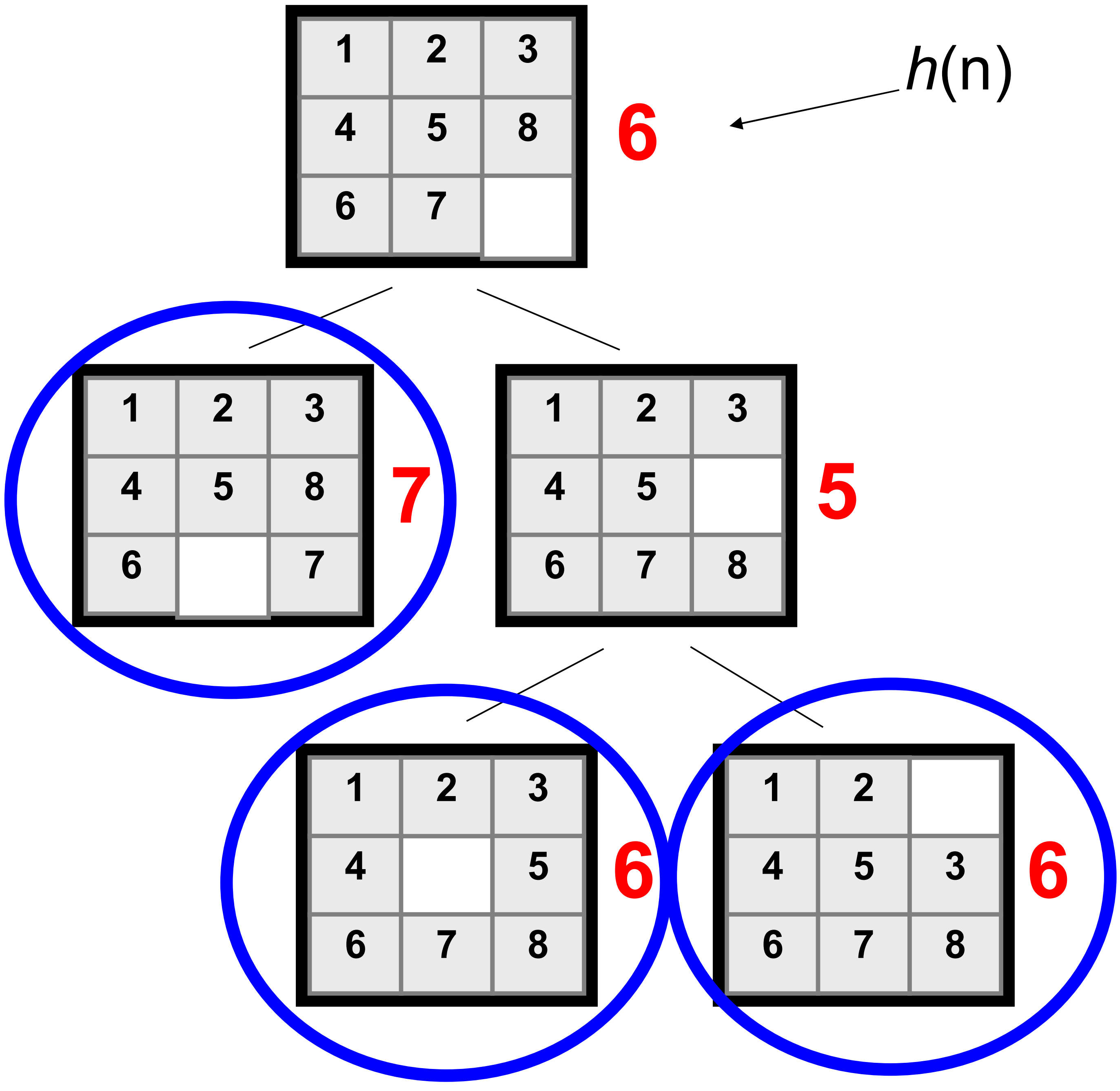
In this example, hill climbing does not work!



In this example, hill climbing does not work!

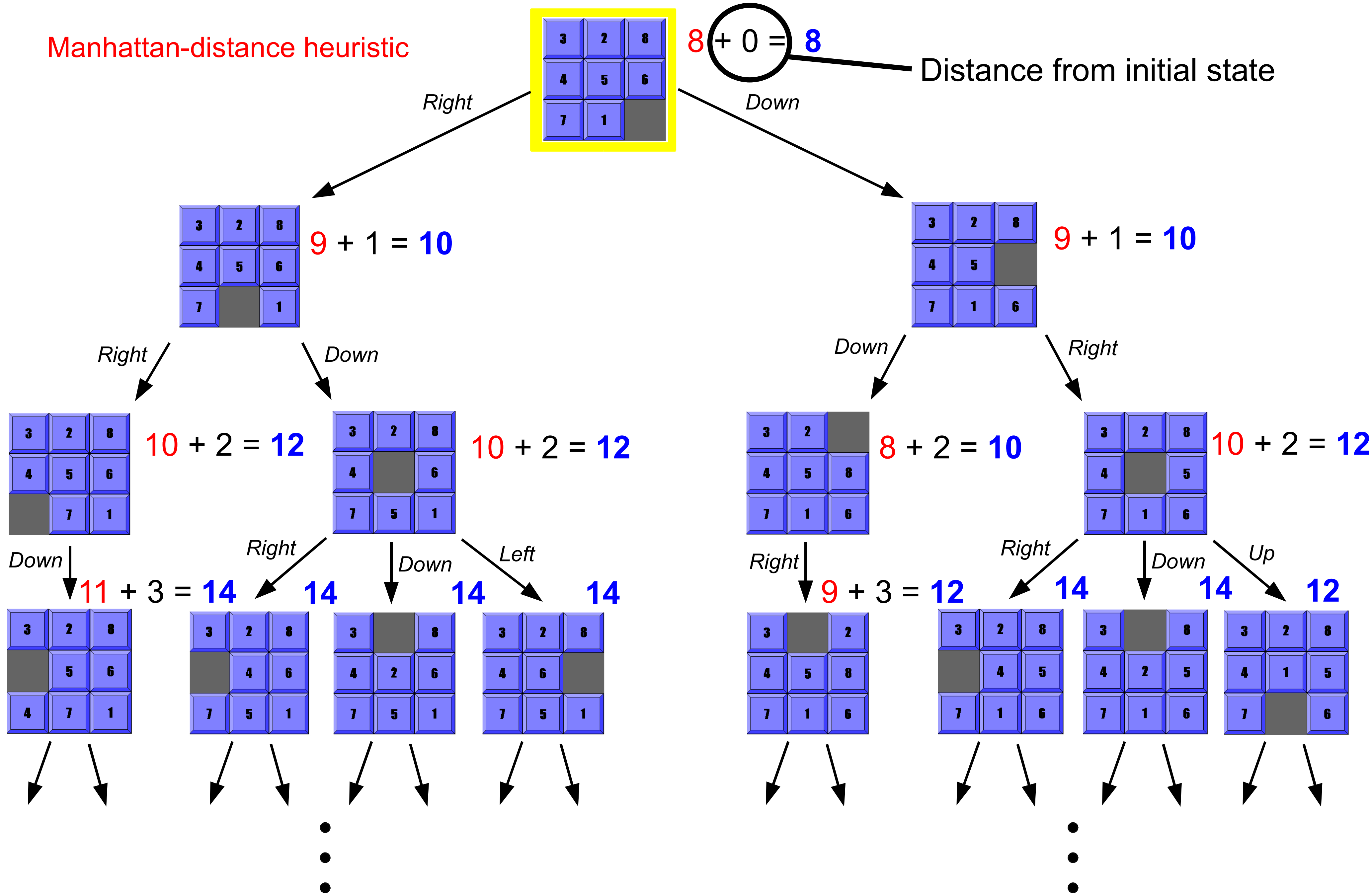
All the nodes on the leaves are taking a step backwards (a “local minimum”)

Note that this puzzle *is* solvable in just 12 more steps.



The A* Algorithm

- Just like before, but we also keep track of the **distance from the initial state** to each intermediate state
- Choose the next node based on the **sum** of two values:
  - How far we think we have left to go (the heuristic estimate)
  - How long it took to get here
  - $f(n) = h(n) + g(n)$
- If our heuristic function **never overestimates**, this algorithm is guaranteed to find the shortest path to the goal!
- Underestimating is fine, but the closer the estimate is to the true value, the faster the search will find the goal.
- If it *does* occasionally overestimate the number of steps left, the search process may get lost in blind alleys, or may find a suboptimal path to the goal



Manhattan-distance heuristic
8 + 0 = 8
Distance from initial state
Right
Down
9 + 1 = 10
9 + 1 = 10
Right
Down
Down
Right
10 + 2 = 12
10 + 2 = 12
8 + 2 = 10
10 + 2 = 12
Down
Right
Down
Left
Right
Right
Down
Up
11 + 3 = 14
14
14
14
9 + 3 = 12
14
14
12