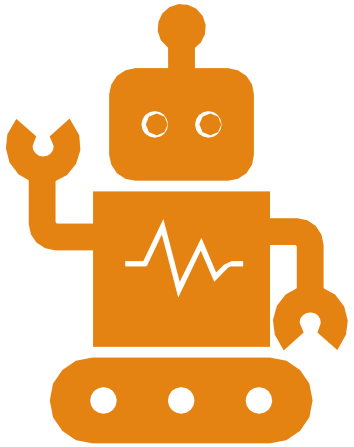


CS341
Artificial Intelligence

Lecture 6

DR. HEBA MOHSEN

Heuristic Search

- On the average they improve the quality of the paths that are explored.
- Using Heuristics, we can hope to get good (though possibly non-optimal) solutions
- There are good general purpose heuristics that are useful in a wide variety of problem domains.
- Special purpose heuristics exploit domain specific knowledge
- Heuristic search uses **Heuristic Function:** This is a function that maps from problem state descriptions to measures of desirability, usually represented as numbers.

Best-first search

Idea: use an evaluation function *f(n)* for each node

- f(n) provides an estimate for the total cost.

→Expand the node n with smallest f(n).

Implementation:

Order the nodes in fringe increasing order of cost.

Special cases:

- greedy best-first search
- A* search

Heuristics for 8-puzzle

These heuristics were obtained by relaxing constraints …

h1: The number of misplaced tiles (squares with number).

h2: The sum of the distances of the tiles from their goal positions.

Heuristics for 8-puzzle I

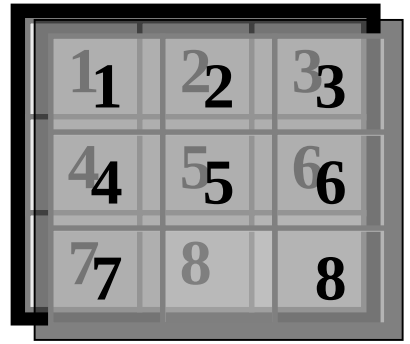
•The number of **misplaced tiles** (not including the blank)

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	

In this case, only "**8**" is misplaced, so the heuristic function evaluates to 1.

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

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

Heuristics for 8-puzzle II

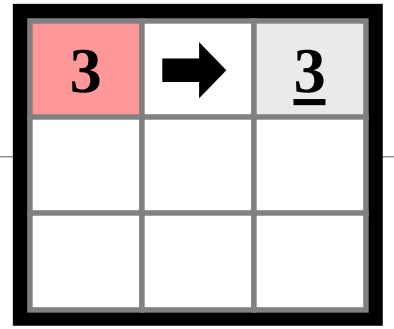
Current State

3	2	8
4	5	6
7	1	

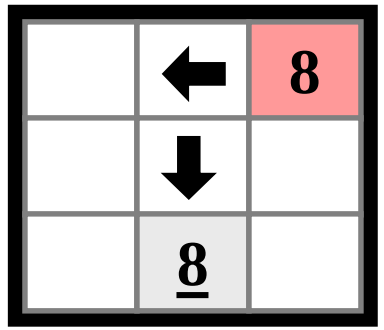
Goal State

1	2	3
4	5	6
7	8	

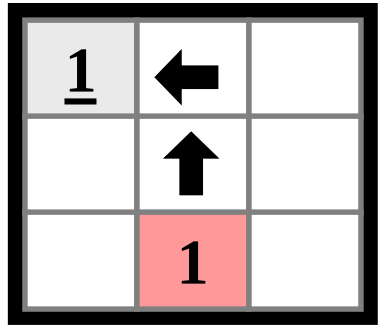
•The **Manhattan Distance** (not including the blank)



2 spaces



3 spaces



3 spaces

Total 8

In this case, only the "**3**", "**8**" and "**1**" tiles are misplaced, by 2, 3, and 3 squares respectively, so the heuristic function evaluates to 8.

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

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

A* Search

(with systematic checking of repeated states)

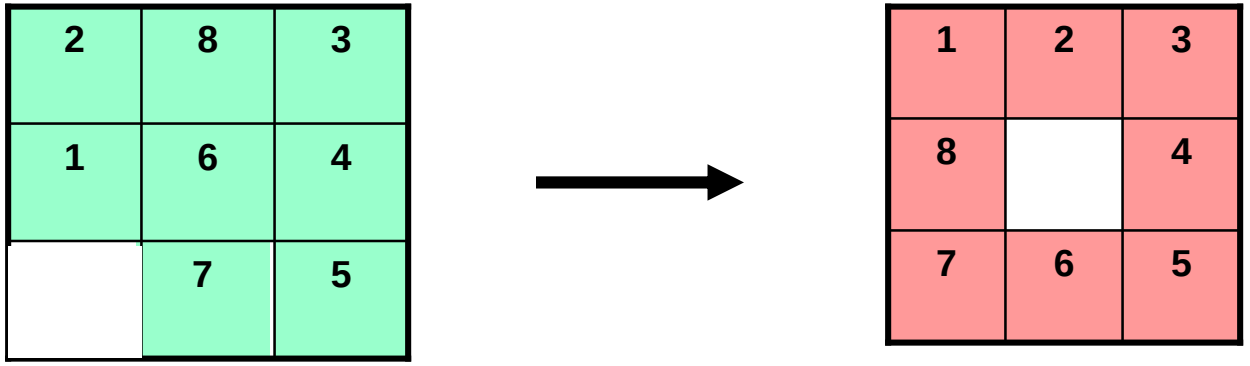
8-Puzzle with h1()

h1(): the number of misplaced tiles

8-Puzzle Problem

Solve the following 8-puzzle problem using A* search algorithm as search strategy and the following function f(n) as heuristic: f(n)=g(n)+h(n)

- h(n):the number of misplaced tiles
- g(n):the number of steps from the initial state



A* Search

(with systematic checking of repeated states)

8-Puzzle with h2()

h2(): Manhattan Distance

8-Puzzle Problem

Solve the following 8-puzzle problem using A* search algorithm as search strategy and the following function f(n) as heuristic: f(n)=g(n)+h(n)

- h(n):the Manhattan Distance
- g(n):the number of steps from the initial state