# Introduction to Artificial Intelligence

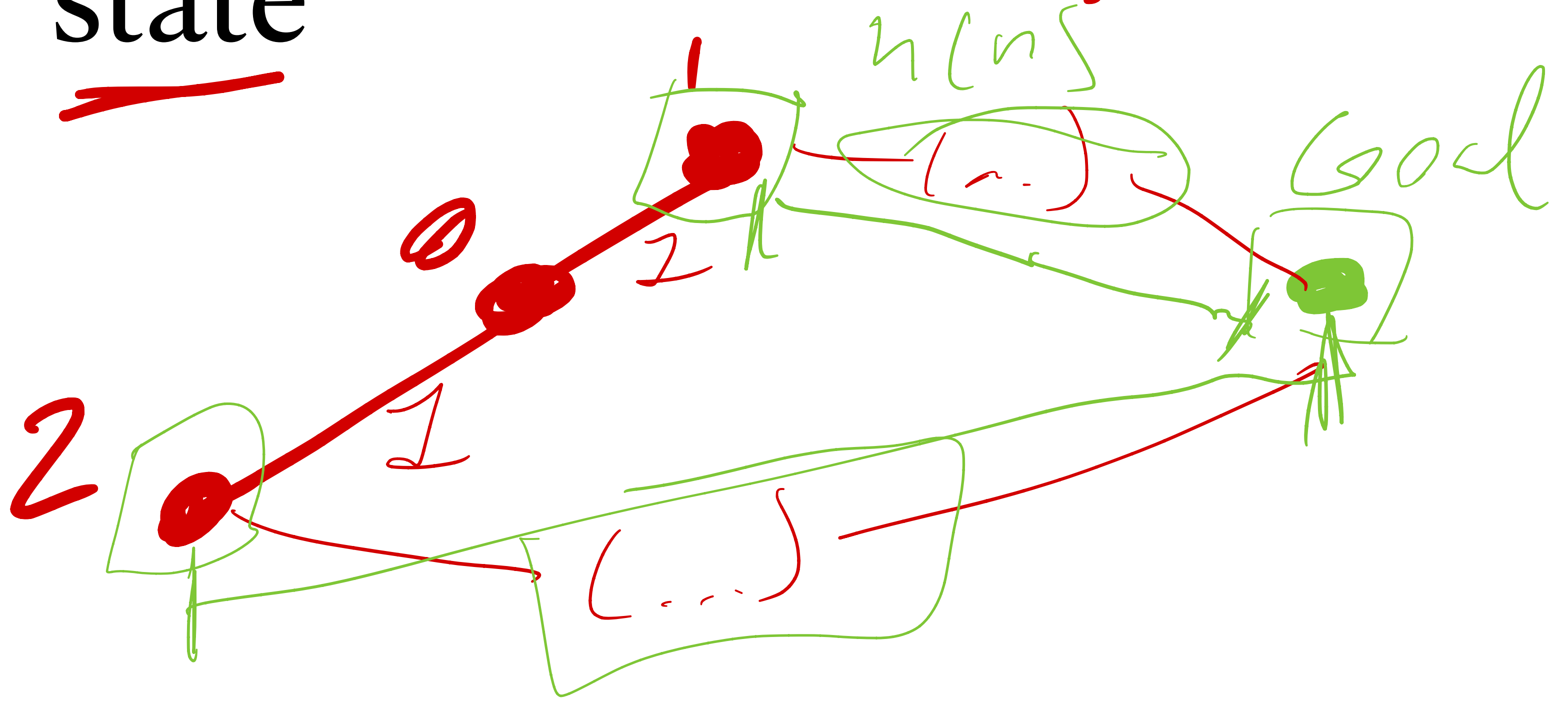
## Informed Strategies

Edgar Granados

# Informed Search

BFS - UCS - DFS

- Previous approaches only use g(n) (cost of current path) to evaluate
  - $f(n) = g(n)$
- **Informed approaches**: Use problem-specific knowledge
- **Heuristic** Function:
  - h(n) - Estimated cost of the cheapest path from the state at node *n* to the goal state

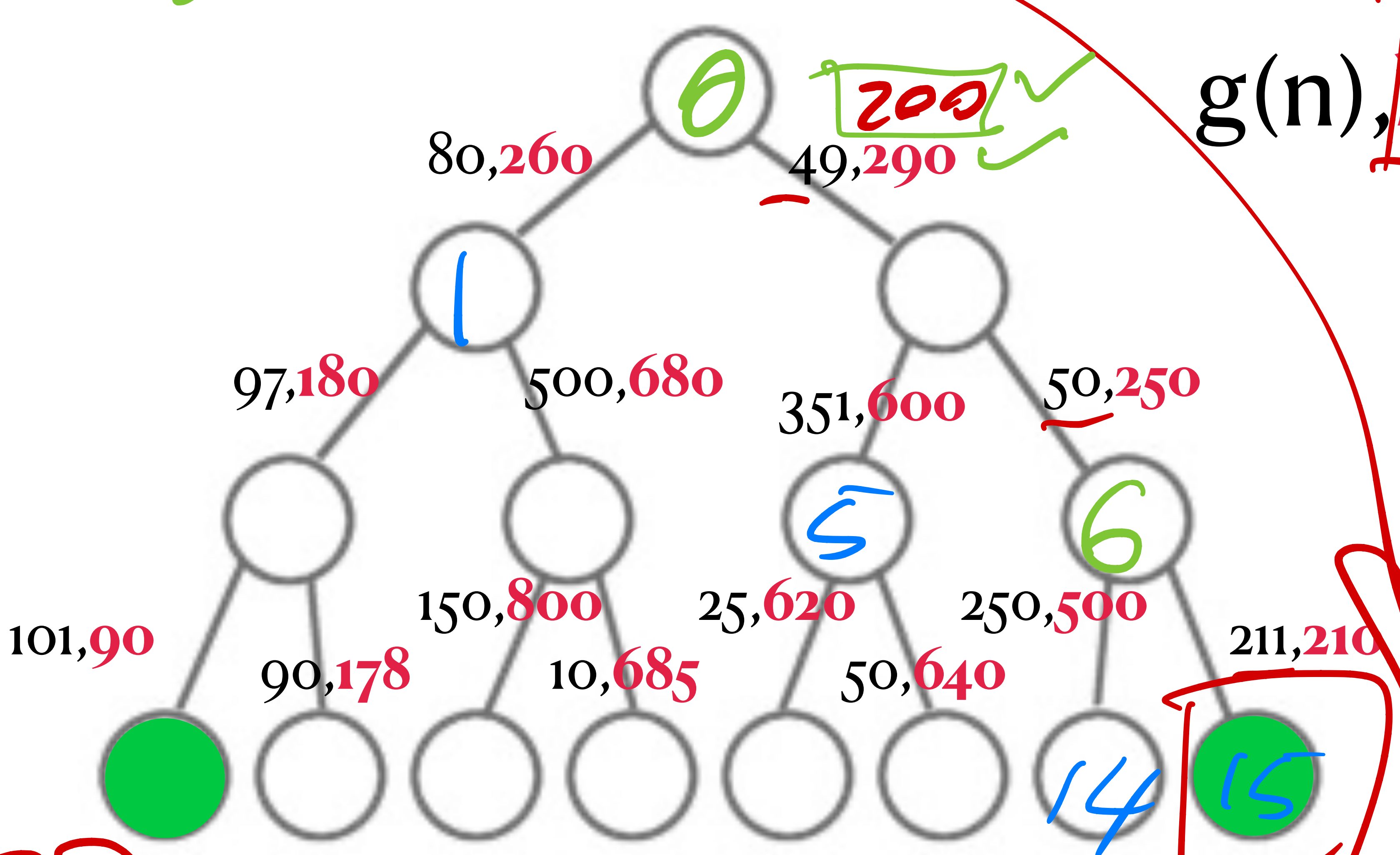


# Greedy Best-First Search

~~f(n) = g(n)~~

Frontier = [1 260, ~~2 200~~, 5 600, ~~6 250~~, 14 500, ~~15 210~~]

- f(n) = h(n)




g(n), **h(n)**

278

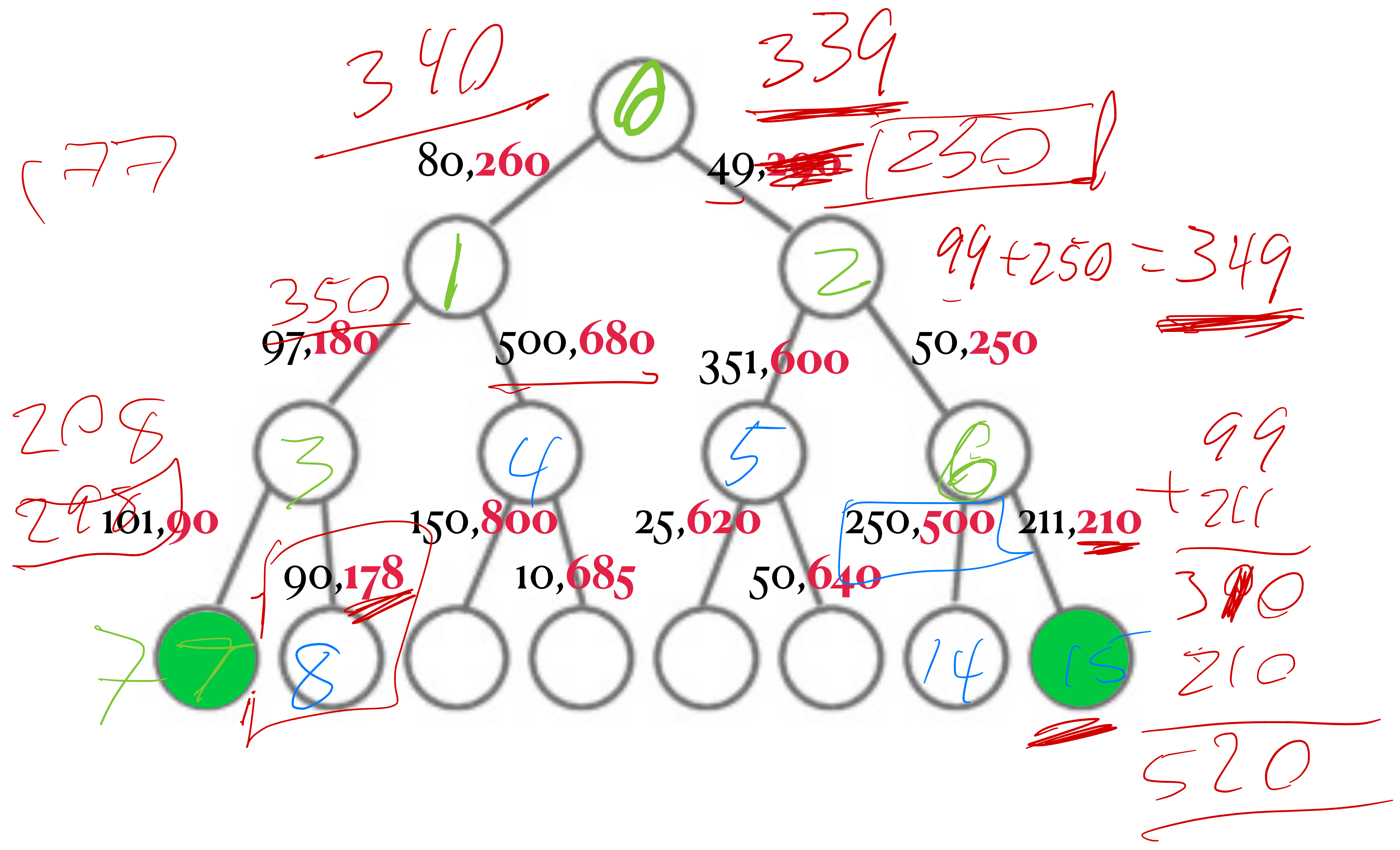
310

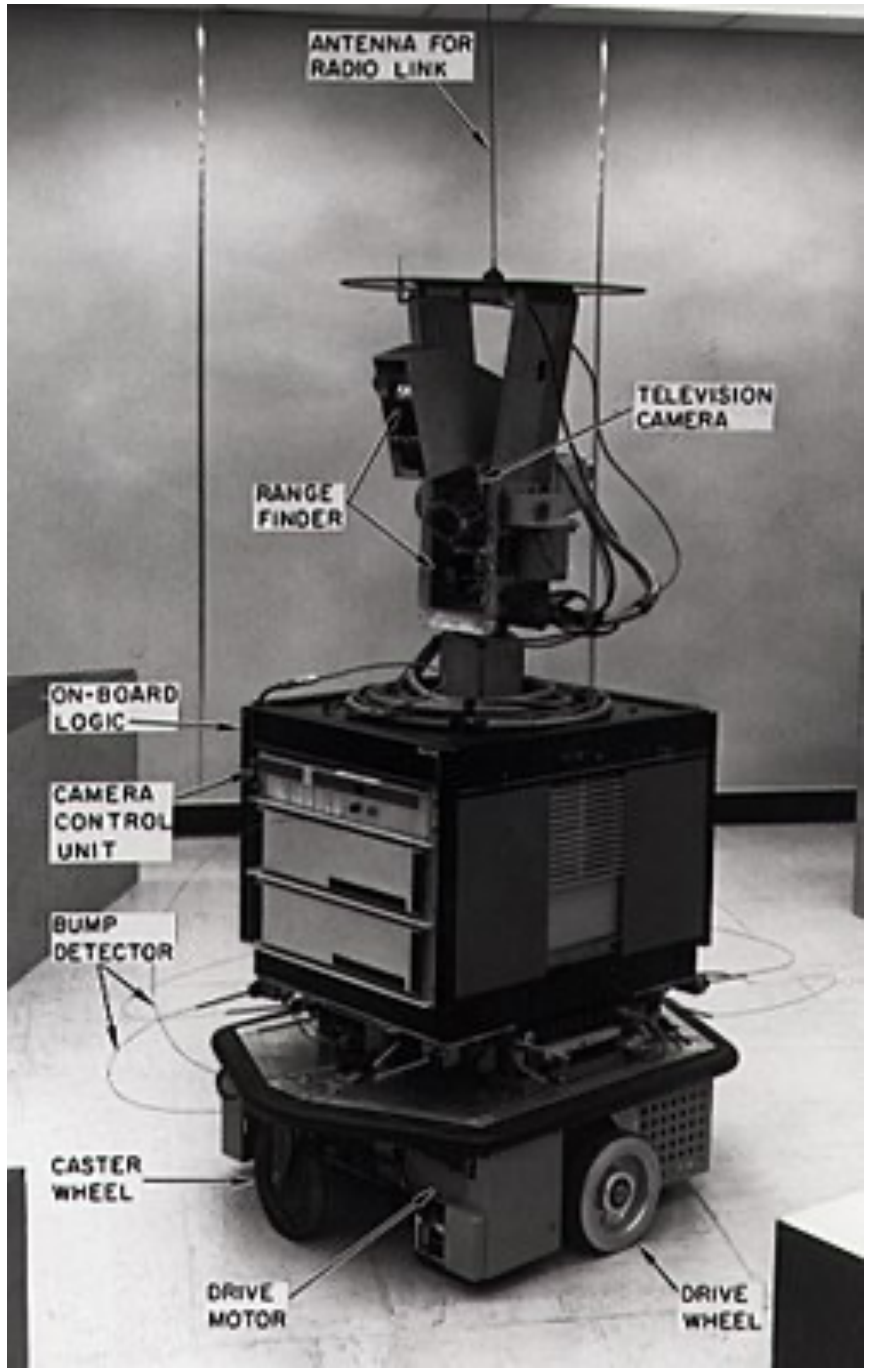
# Greedy Best-First Search

- **Completeness:** No
- **Optimality:** No
- **Time complexity:** $O(b^m)$
- **Space complexity:** $O(b^m)$

# A* Algorithm

- $f(n) = g(n) + h(n)$



$\sqrt{\Delta x^2 + \Delta y^2}$

# A* Algorithm

## Example


Manhattan distance

$\Delta x + \Delta y$

$2 + 3 = 5$

RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

# Introduction to Artificial Intelligence

## A* Analysis & Heuristics

$F(n) = g(n) + h(n)$

Edgar Granados

# A*: Algorithm

## Analysis

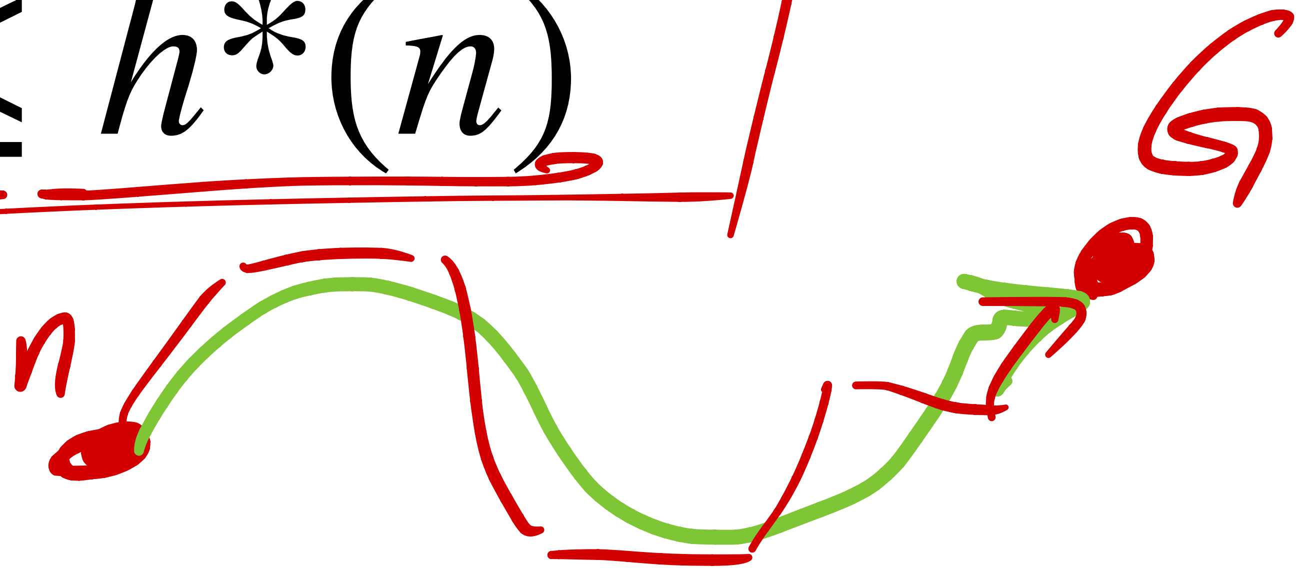
cost of the path

$= g(n) + \boxed{h^*(n)} = f$

cost to come

- Behavior depends on the heuristic
- **Admissible** Heuristic:
  - Never overestimate:

$$\forall n : \boxed{h(n)} \leq h^*(n)$$




**Theorem**

lemma

If h is admissible then the A* is optimal

# A*: Algorithm

## Lemma

- Assume *h* admissible
- Let *n* be a node such that goal-function(n) is true but is part of a non-optimal path
- Let n’ be a node on the optimal path
- Then, A* expands *n′* before n

# A*: Algorithm

## Proving the Lemma

- Assume *h* admissible
- Let *n* be a node such that goal-function(n) is true but is part of a non-optimal path
- Let n' be a node on the optimal path
- Then, A* expands *n'* before n

$C^* \sim$ cost of the opt path

$$f(n') = g(n') + h(n') \boxed{\leq C^* <} f(n) = g(n) + h(n)$$

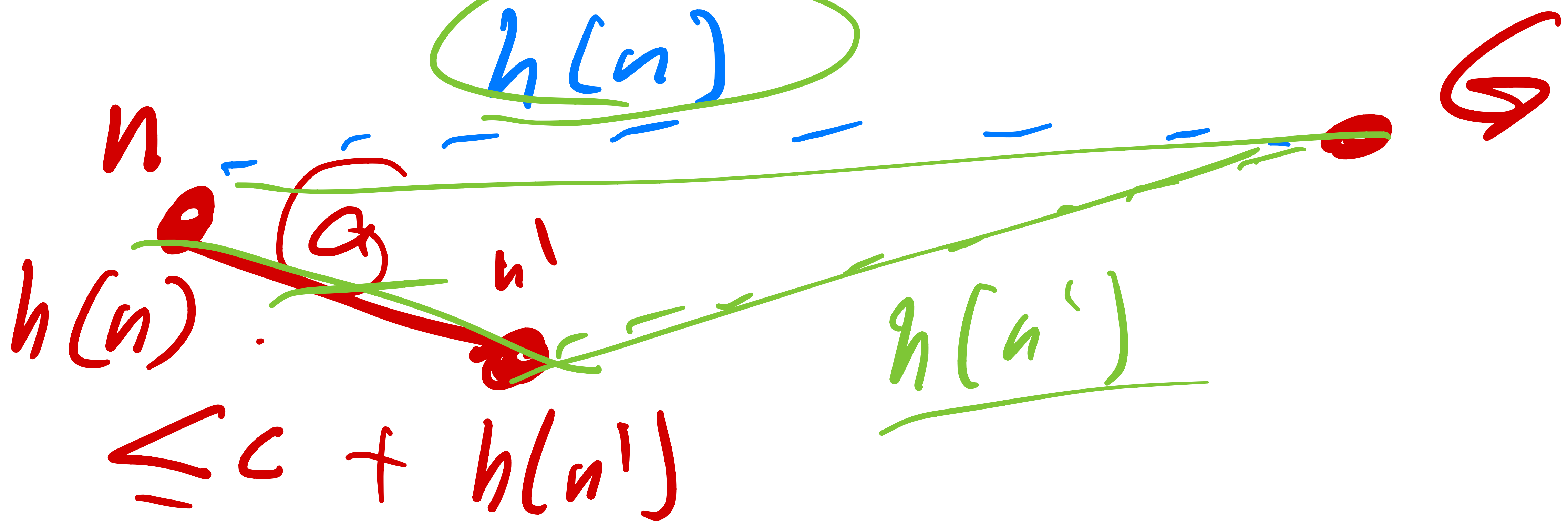
n' expands before n

# A*: Algorithm

## Analysis

- **Consistent** Heuristic:

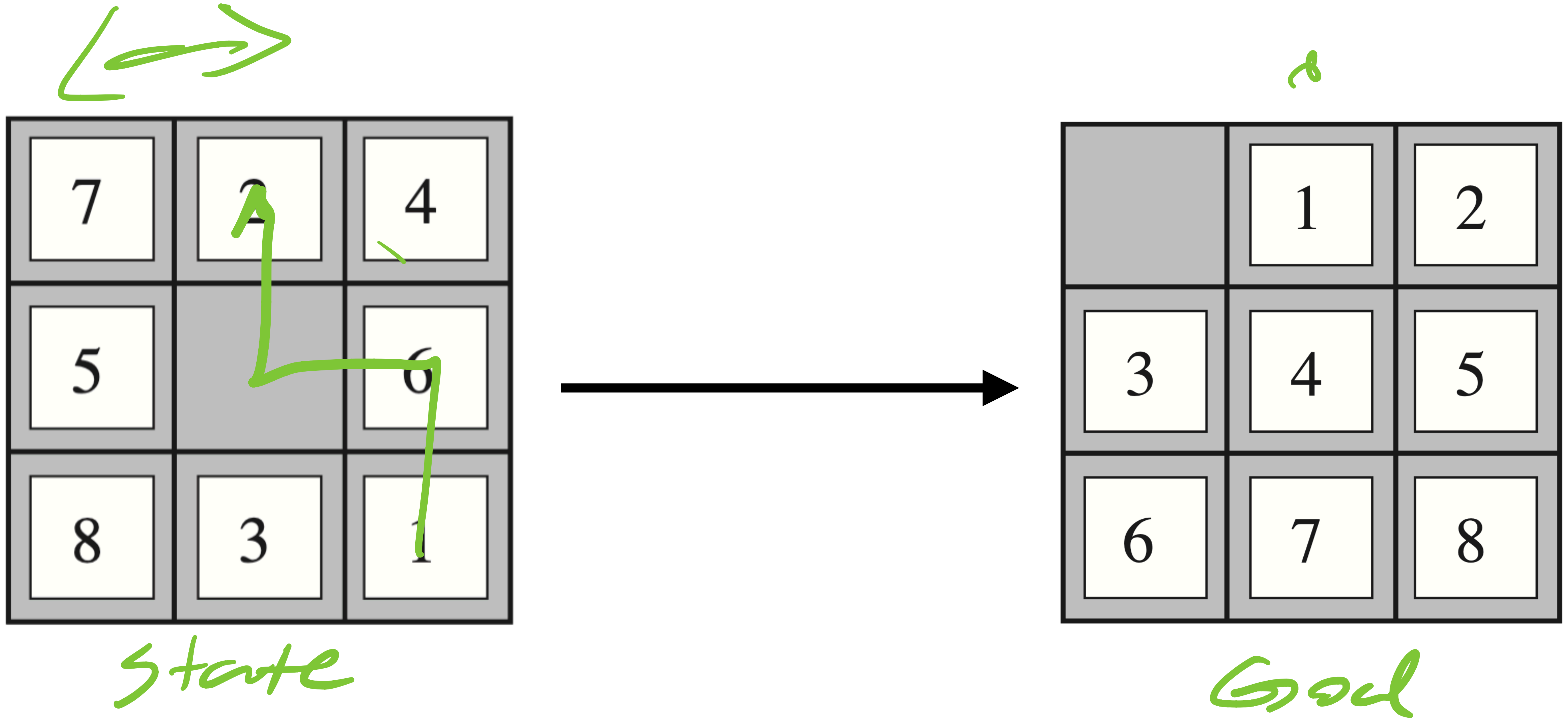
$$\forall (n, a, n') : h(n) \leq c(n, a, n') + h(n')$$




## Theorem

If h is consistent then graph search A* is optimal

# Designing Heuristics

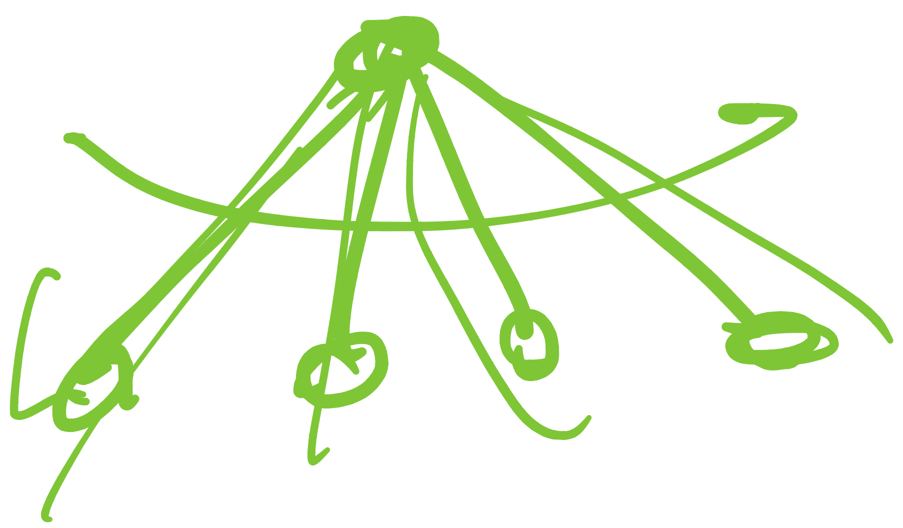



- Not trivial!
- h1 - # tiles to move
- h2 - Manhattan dist. $3+1+2+2+2 \ldots \le h^*$

# Designing Heuristics

$\boxed{h(n) = 0}$

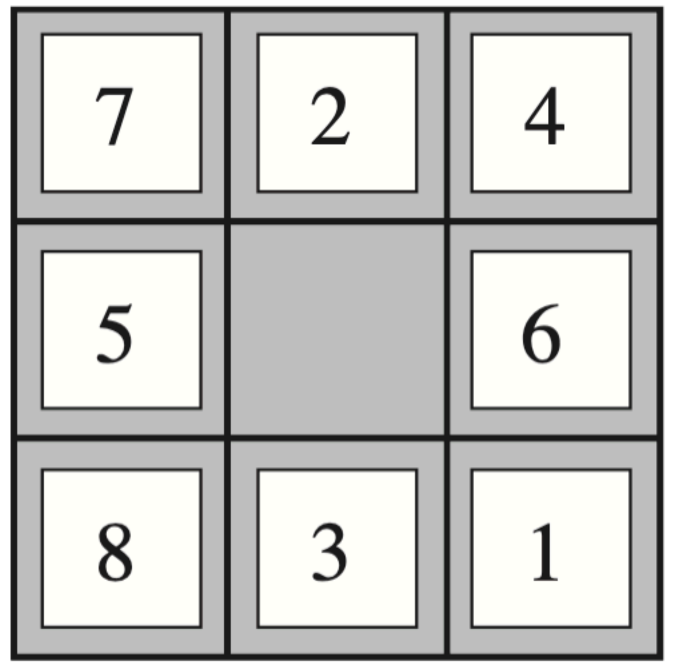
$A^*$



- Minimize the number of expanded nodes
- Automated heuristics?    $A^*$
- Evaluate the number of nodes expanded: the effective branching factor b* $\rightarrow 1$

$$N = 1 + b^* + (b^*)^2 + \cdots + (b^*)^d$$

$h_1 \leq h_2$

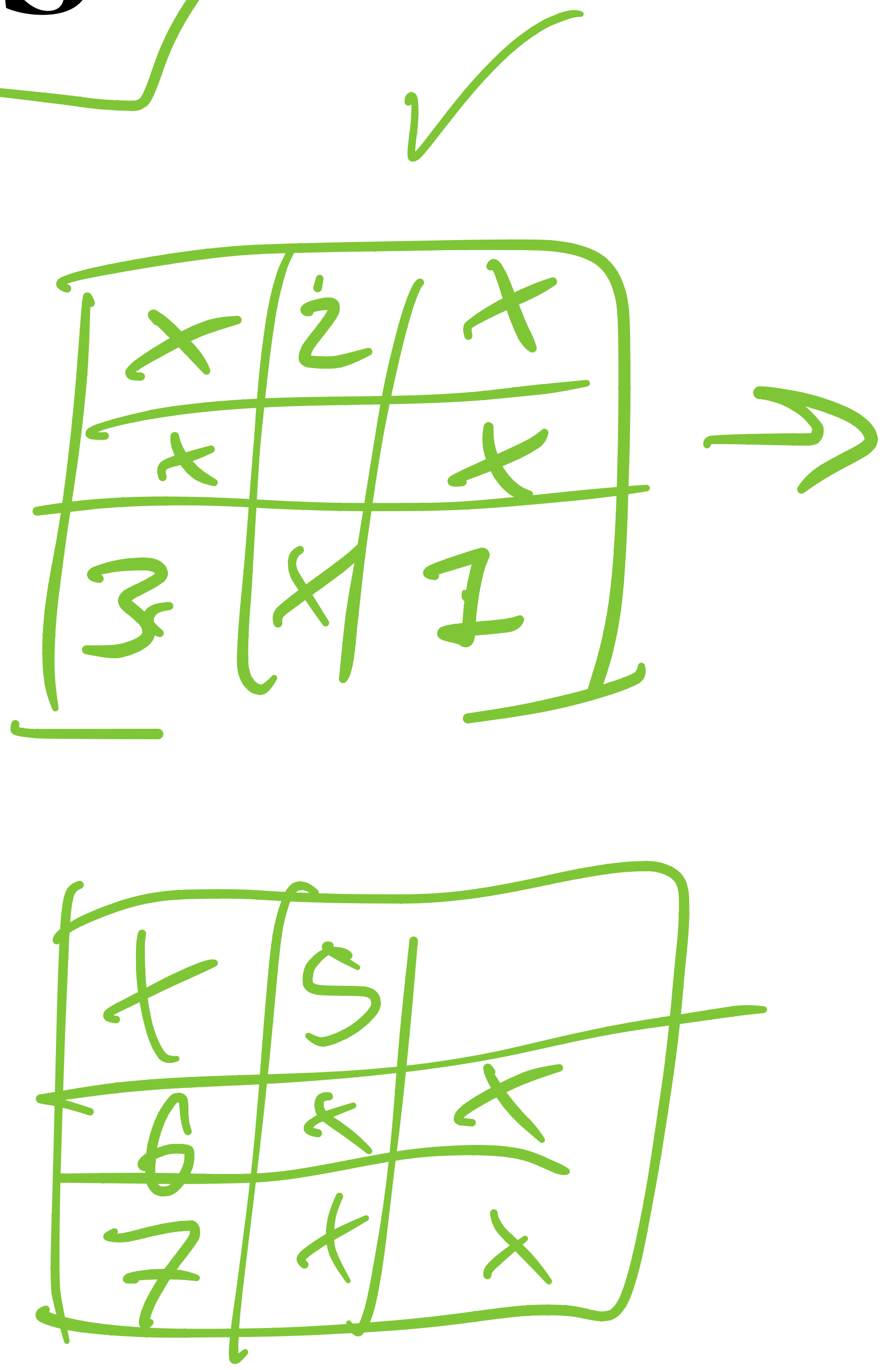


$g(n) + \boxed{h(n)}$

# Designing Heuristics

## Coming up with heuristics

- Use relaxed (quickly solved) versions of the problem
- Database of precomputed solutions
- Combine heuristics
- New approaches: ML



# Combining Heuristics

$w h_1 + (w-1) h_2$

- Given multiple heuristics, how to select one?

$$h_{max}(n) = \max\{h_1(n), h_2(n), \ldots, h_n(n)\}.$$

- Admissibility of $h_{max}$?

$h(n) \le h^*(n)$

- Consistency of $h_{max}$?