

# HỆ THỐNG TÌM KIẾM (PHẦN 3)

Trần Trung Kiên

[ttkien@fit.hcmus.edu.vn](mailto:ttkien@fit.hcmus.edu.vn)

# Tổng thể

---

Các thuật toán tìm kiếm sử dụng thông tin trạng thái đích để định hướng

- Greedy Search
- $A^*$

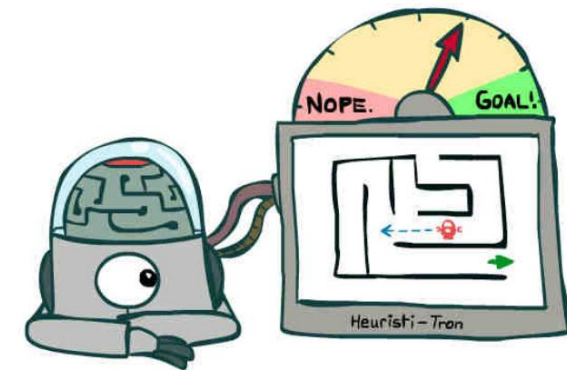
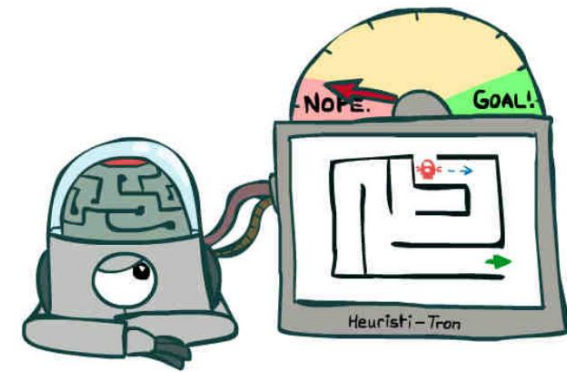
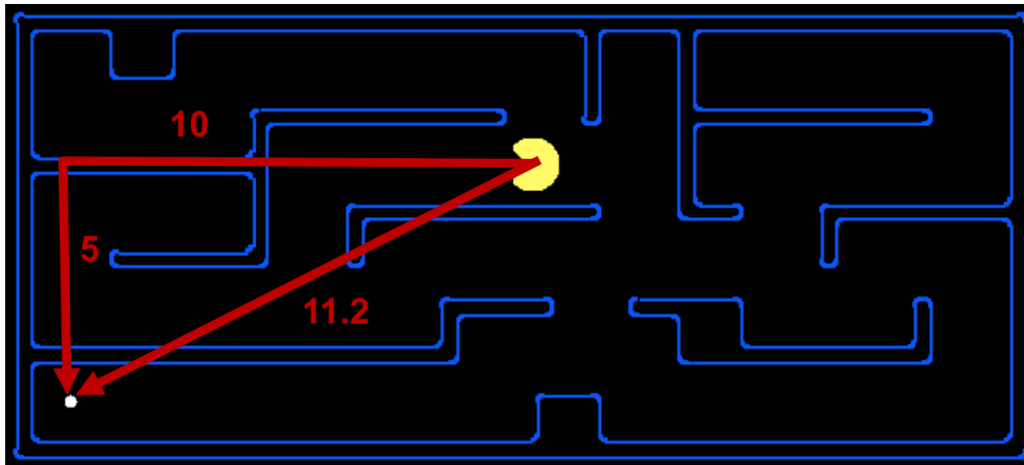
# Nhìn lại về UCS

---

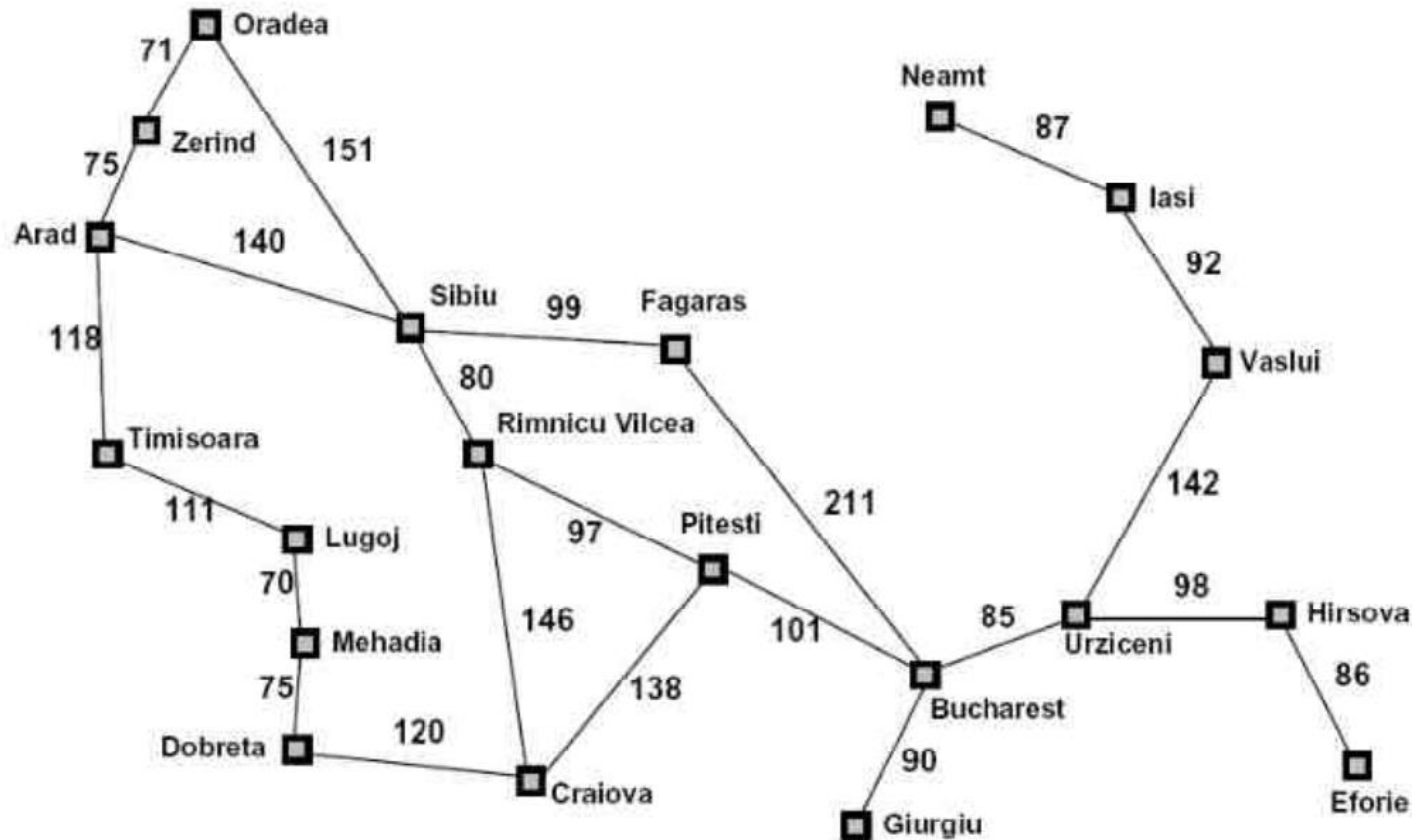
- UCS đảm bảo tìm được lời giải có chi phí nhỏ nhất
- Tuy nhiên, UCS mở rộng theo mọi hướng, không có định hướng về trạng thái đích → chạy chậm
  - Xem video demo ...
- Ta mong muốn tăng tốc quá trình tìm kiếm bằng cách sử dụng thông tin trạng thái đích để định hướng

# Heuristic

- Một **heuristic** là một hàm: nhận đầu vào là một trạng thái và trả về mức độ gần đích **ước lượng** của trạng thái này
- Heuristic được thiết kế riêng cho một bài toán cụ thể
- Vd, trong bài toán tìm đường đi cho Pacman đến một vị trí, heuristic có thể là khoảng cách Manhattan hoặc khoảng cách Euclidean



# Ví dụ về heuristic



| Straight-line distance to Bucharest |     |
|-------------------------------------|-----|
| Arad                                | 366 |
| Bucharest                           | 0   |
| Craiova                             | 160 |
| Dobreta                             | 242 |
| Eforie                              | 161 |
| Fagaras                             | 178 |
| Giurgiu                             | 77  |
| Hirsova                             | 151 |
| Iasi                                | 226 |
| Lugoj                               | 244 |
| Mehadia                             | 241 |
| Neamt                               | 234 |
| Oradea                              | 380 |
| Pitesti                             | 98  |
| Rimnicu Vilcea                      | 193 |
| Sibiu                               | 253 |
| Timisoara                           | 329 |
| Urziceni                            | 80  |
| Vaslui                              | 199 |
| Zerind                              | 374 |

$h(x)$

# Thuật toán Greedy Search

---



# Thuật toán Greedy Search

---

- Chiến lược chọn kế hoạch từ fringe để mở rộng: chọn kế hoạch gần với trạng thái đích nhất theo heuristic → chạy nhanh (nếu heuristic đủ tốt) nhưng không đảm bảo tìm được kế hoạch có chi phí thấp nhất
- Chạy Greedy Search với vd tìm đường đi từ Arad đến Bucharest: xem trên bảng ...
- Xem video demo ...

# Thuật toán A\*





# Thuật toán A\*

---



UCS



Greedy



A\*

# $A^* = \text{UCS} + \text{Greedy Search}$

**UCS**: chọn kế hoạch có có chi phí lùi g nhỏ nhất để mở rộng

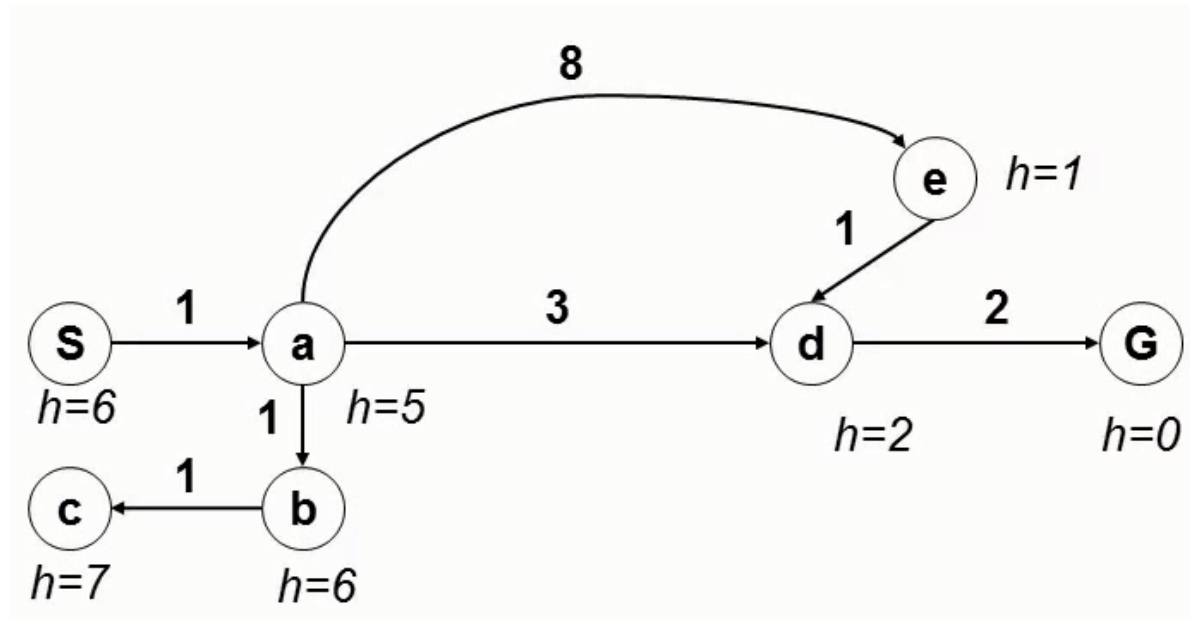
**Greedy Search**: chọn kế hoạch có chi phí tiến ước lượng h nhỏ nhất để mở rộng

Khi mở kế hoạch  $S \rightarrow a$ , ta được 3 kế hoạch:

- $S \rightarrow a \rightarrow b$  ( $g = 2$ ,  $h = 6$ )
- $S \rightarrow a \rightarrow d$  ( $g = 4$ ,  $h = 2$ )
- $S \rightarrow a \rightarrow e$  ( $g = 9$ ,  $h = 1$ )

**UCS** sẽ chọn  $S \rightarrow a \rightarrow b$  để mở tiếp dù kế hoạch này không hướng về đích ☹

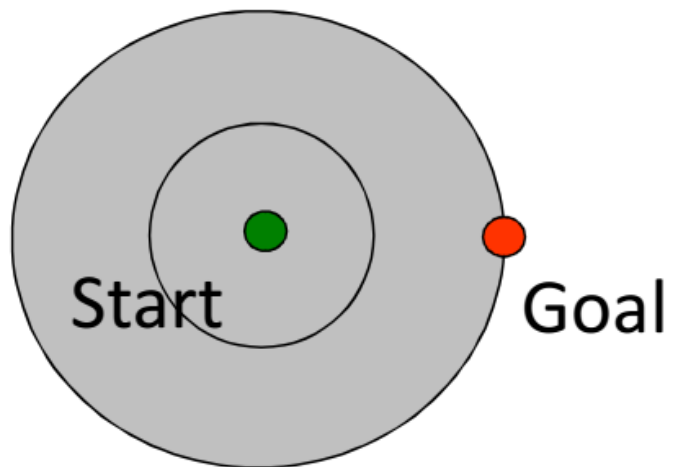
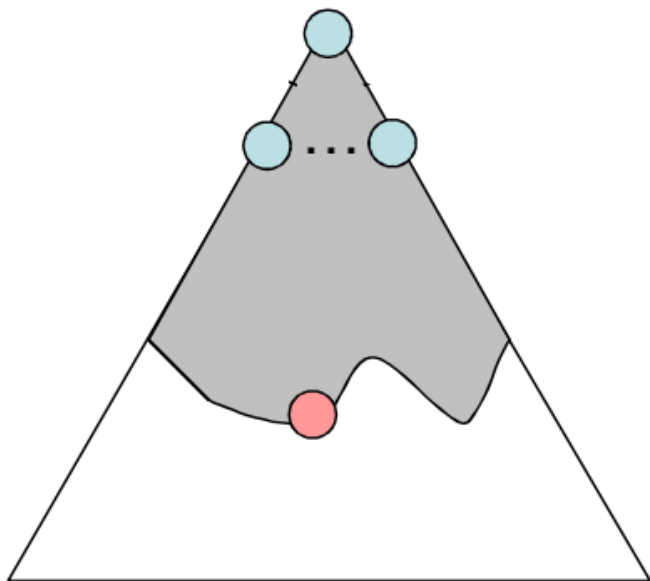
**Greedy** sẽ chọn  $S \rightarrow a \rightarrow e$  để mở tiếp dù kế hoạch này có chi phí (lùi) lớn ☹



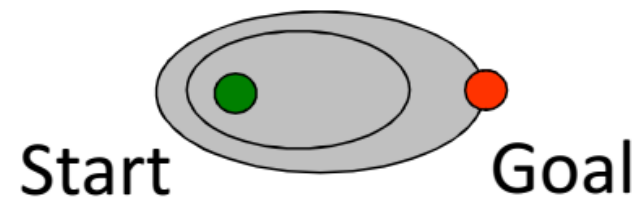
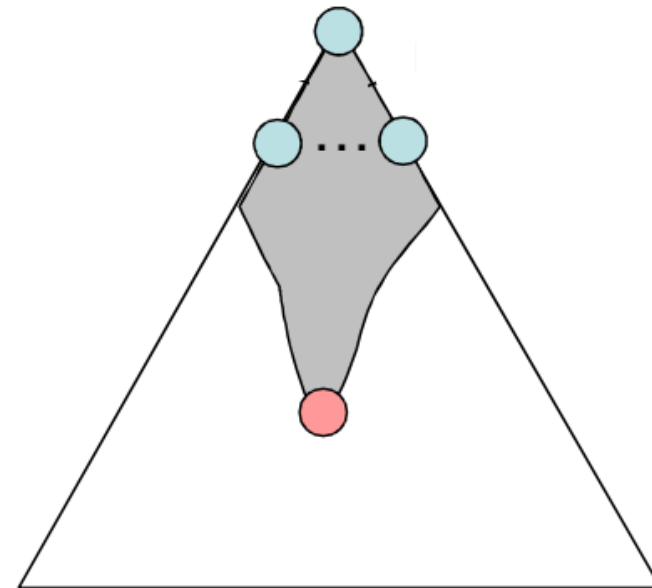
**$A^*$** : chọn kế hoạch có  $f = g + h$  nhỏ nhất để mở rộng  $\rightarrow$  không ưu tiên những hướng có chi phí lùi  $g$  quá lớn hoặc chi phí tiến  $h$  quá lớn

Trong vd trên,  **$A^*$**  sẽ chọn  $S \rightarrow a \rightarrow d$  (vừa không có  $g$  quá lớn, vừa không có  $h$  quá lớn) để mở tiếp ☺

# UCS



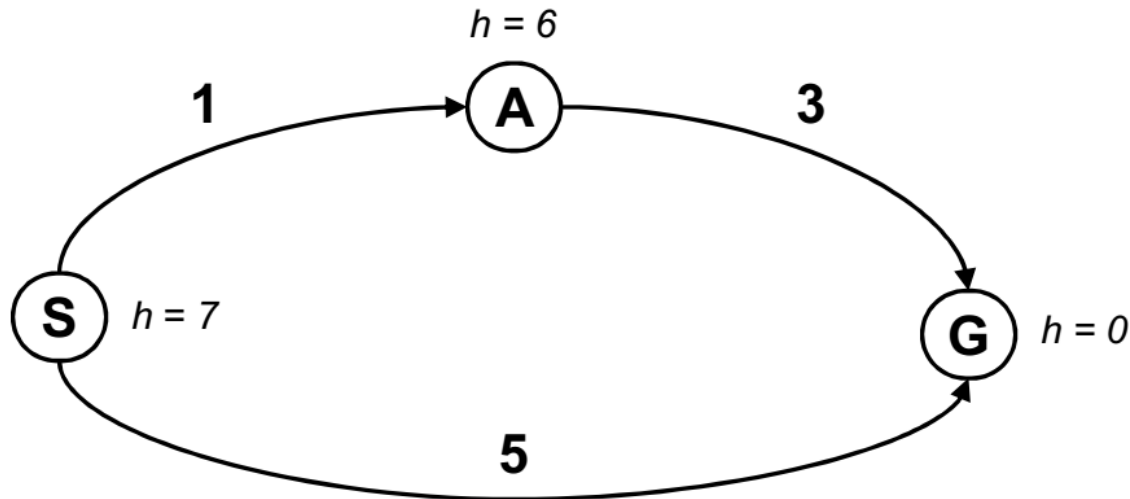
# A\*



Xem video demo ...

# A\* có tìm được kế hoạch tối ưu?

- Chạy A\* với đồ thị ở dưới ...
- Kế hoạch tìm được:  $S \rightarrow G$  😞. Tại sao không tối ưu?
  - Do: chi phí thật sự của kế hoạch dở  $<$  chi phí ước lượng (= chi phí lười thật sự + chi phí tiến ước lượng) của kế hoạch tốt
  - Chi phí tiến ước lượng của heuristic cần  $\leq$  chi phí tiến thật sự



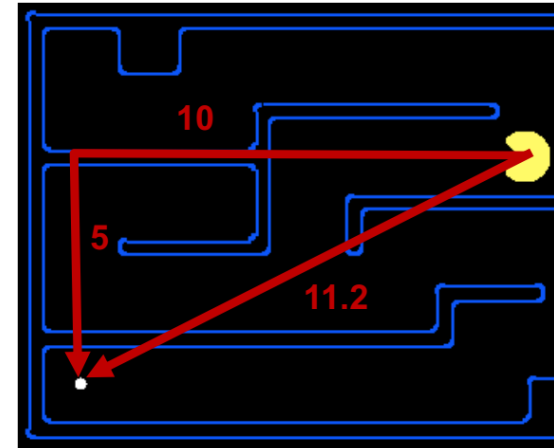
# Heuristic hợp lệ

- Để A\* tìm được kế hoạch tối ưu thì heuristic  $h$  phải **hợp lệ (admissible / optimistic)**:

$$0 \leq h(n) \leq h^*(n)$$

Chi phí tiến thật sự từ trạng thái  $n$  đến đích

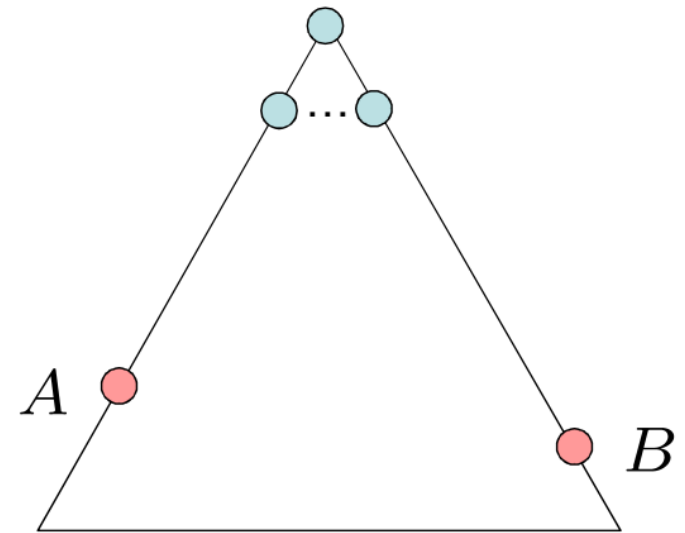
- Vd: trong bài toán tìm đường đi cho Pacman đến một vị trí thì khoảng cách Manhattan / Euclidean có phải là heuristic hợp lệ?
- Khi sử dụng A\*, việc quan trọng nhất là thiết kế ra một heuristic hợp lệ nhưng phải đủ chính xác để có thể định hướng



# C/m: nếu heuristic hợp lệ thì $A^*$ sẽ tìm được kế hoạch tối ưu

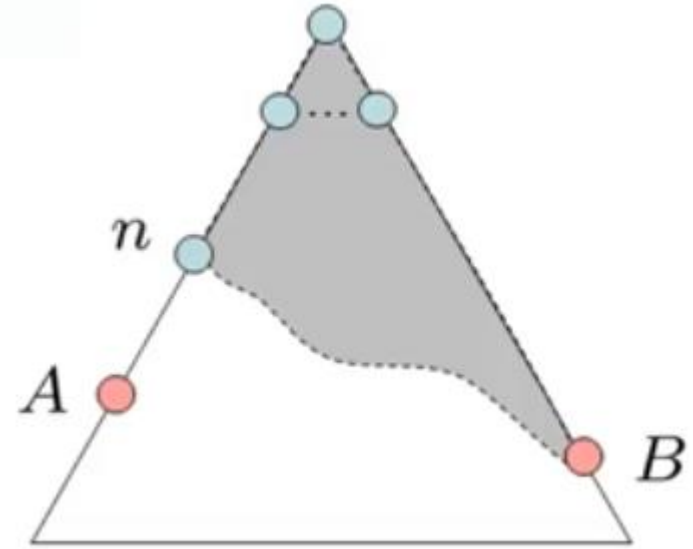
---

- Giả sử:
  - A là node ứng với kế hoạch đến đích tối ưu
  - B là node ứng với kế hoạch đến đích không tối ưu
  - Heuristic hợp lệ
- Ta sẽ c/m: A sẽ được lấy ra khỏi fringe trước B



# C/m: nếu heuristic hợp lệ thì A\* sẽ tìm được kế hoạch tối ưu

- Ta sẽ c/m: A sẽ được lấy ra khỏi fringe trước B
  - Giả sử B đang ở fringe, và tổ tiên n của A (hoặc A) cũng đang ở fringe
  - Dễ thấy:  $f(n) \leq f(A) \leq f(B)$ 
    - Do h hợp lệ
    - Do giả thiết A tối ưu hơn B
  - Do đó, từ tổ tiên n của A cho đến A sẽ được mở trước B



# Cách tạo heuristic hợp lệ

---

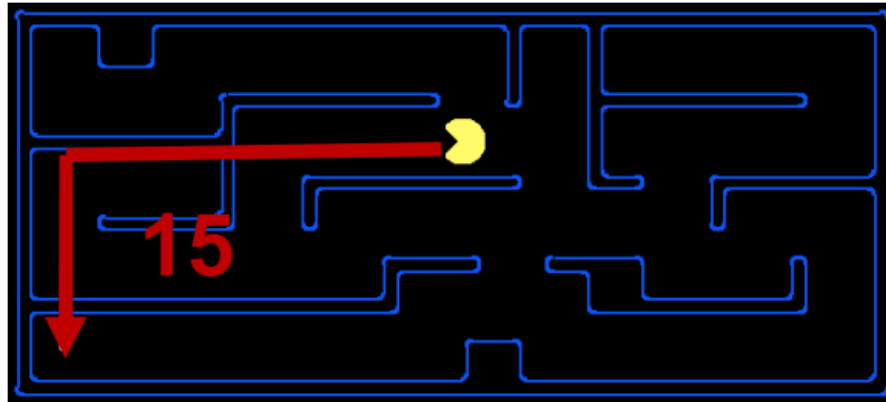
- Tính  $h(n)$  bằng cách chạy thuật toán tìm kiếm để tìm chi phí nhỏ nhất từ  $n$  đến đích?
- Cho  $h(n) = 0$ ?
- Heuristic hợp lệ thường được tạo ra bằng cách: nói lỏng bài toán (tức là làm cho bài toán dễ hơn), rồi tính  $h(n)$  bằng cách tính chi phí nhỏ nhất từ  $n$  đến đích trong bài toán nói lỏng này



## Ví dụ: bài toán tìm đường đi cho Pacman đến một vị trí

---

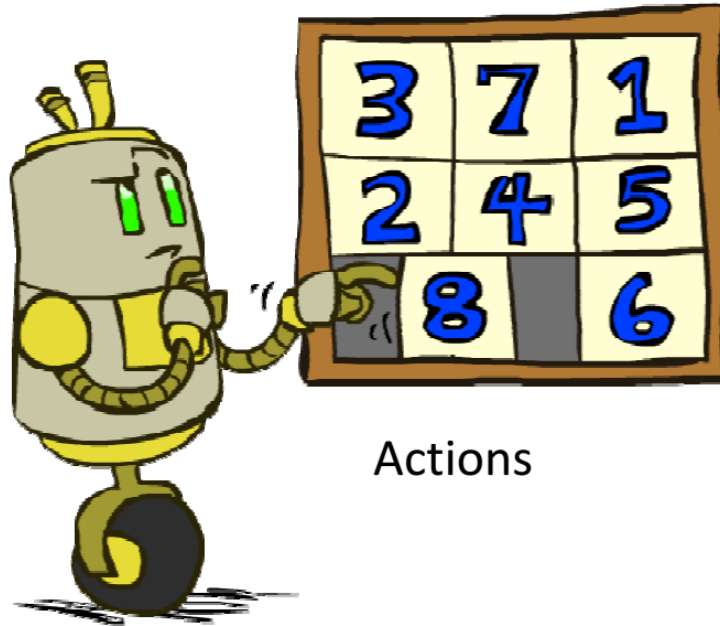
Dùng heuristic là khoảng cách Manhattan ứng với bài toán nói  
lỏng: Pacman có thể đi xuyên tường



# Ví dụ: bài toán 8 puzzle

|   |   |   |
|---|---|---|
| 7 | 2 | 4 |
| 5 |   | 6 |
| 8 | 3 | 1 |

Start State



|   |   |   |
|---|---|---|
| 3 | 7 | 1 |
| 2 | 4 | 5 |
|   | 8 | 6 |

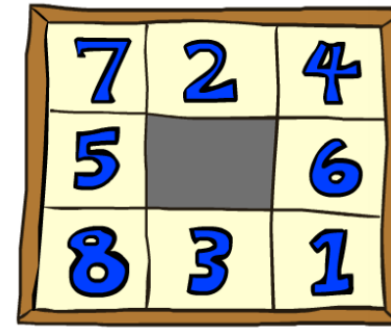
Actions

|   |   |   |
|---|---|---|
|   | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

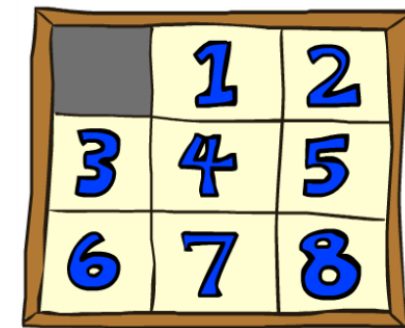
Goal State

# Ví dụ: bài toán 8 puzzle

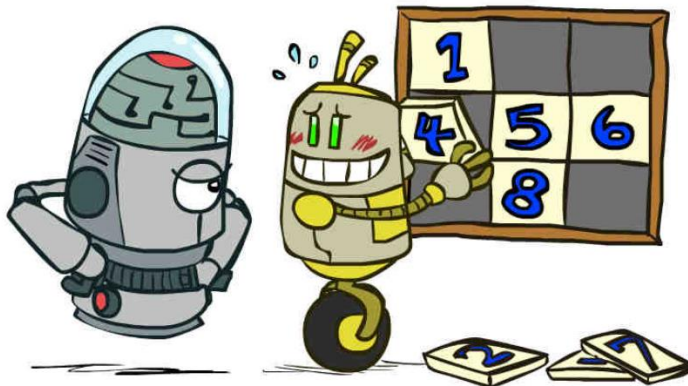
- Heuristic: số miếng ghép bị sai vị trí
  - Vd,  $h(\text{start}) = 8$
- Có hợp lệ không?
- Ứng với bài toán nói lỏng:



Start State



Goal State

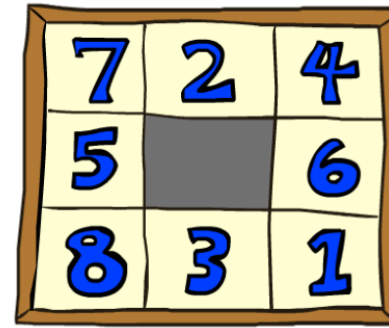


Average nodes expanded  
when the optimal path has...

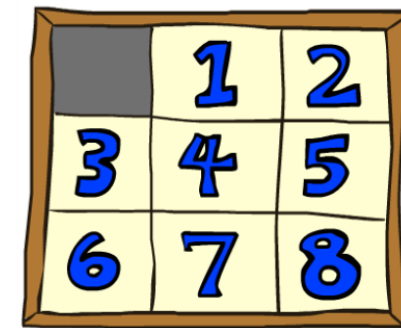
|       | ...4 steps | ...8 steps | ...12 steps       |
|-------|------------|------------|-------------------|
| UCS   | 112        | 6,300      | $3.6 \times 10^6$ |
| TILES | 13         | 39         | 227               |

# Ví dụ: bài toán 8 puzzle

- Bài toán ít nói lỏng hơn: ta có thể di chuyển một miếng ghép sang ô bên cạnh cho dù ô đó đã có miếng ghép khác
- Heuristic: tổng khoảng cách Manhattan từ mỗi miếng ghép đến vị trí đúng
  - Vd,  $h(\text{start}) = 3 + 1 + 2 + \dots = 18$
- Có hợp lệ không?



Start State



Goal State

| Average nodes expanded when the optimal path has... |            |            |             |
|-----------------------------------------------------|------------|------------|-------------|
|                                                     | ...4 steps | ...8 steps | ...12 steps |
| TILES                                               | 13         | 39         | 227         |
| MANHATTAN                                           | 12         | 25         | 73          |

# Ý rút ra từ ví dụ 8 puzzle

---

- Heuristic càng chính xác thì càng giúp định hướng tốt (giúp giảm số trạng thái phải mở)
- Tuy nhiên, cũng không nên cố gắng tạo ra heuristic quá chính xác vì thời gian tính heuristic sẽ lâu → dù mở rất ít trạng thái, nhưng với mỗi thái lại tốn nhiều thời gian để tính heuristic → không đạt được mục tiêu tăng tốc ban đầu của  $A^*$
- Heuristic tốt là heuristic chính xác vừa phải; vừa để có thể định hướng được, vừa để có thể tính nhanh được

# Quiz: ôn lại các thuật toán tìm kiếm

---

Xem video và trả lời ...