

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

# **Cơ sở trí tuệ nhân tạo**

Tìm kiếm có định hướng

**Nguyễn Ngọc Đức**

**2021**

# Nội dung



- 1 Best-first Search
- 2 Greedy Best-first Search
- 3 A\* Search
- 4 Giới hạn không gian tìm kiếm

# Tìm kiếm có định hướng

Tìm kiếm có định hướng hay tìm kiếm **heuristic**

- Bên cạnh định nghĩa còn sử dụng **tri thức** cụ thể về bài toán
- **Có khả năng** tìm lời giải hiệu quả hơn so với các chiến lược tìm kiếm mù

## Heuristic

- **Tri thức** về bài toán được truyền vào thuật toán tìm kiếm
- Heuristic **ước lượng** khoảng cách giữa trạng thái hiện tại so với trạng thái đích

# Best-first Search

# Best-first Search

- Thuật toán tìm kiếm tổng quát trên cây và đồ thị
- Một nút được chọn mở dựa trên **hàm đánh giá**,  $f(n)$ , được xây dựng dựa trên ước lượng chi phí.
  - Nút có ước lượng **thấp nhất** được mở trước
  - Cài đặt tương tự như UCS ngoại trừ việc sử dụng  $f$  thay vì  $g$  cho việc sắp xếp hàng đợi ưu tiên
  - Lựa chọn  $f$  định nghĩa **chiến lược tìm kiếm**
- Hầu hết hàm đánh giá  $f$  đều chứa **heuristic** -  $h$

# Greedy Best-first Search

# Greedy Best-first Search



- Cố gắng mở các nút được ước lượng gần với đích nhất

# Greedy Best-first Search

- Đánh giá trạng thái hiện tại chỉ dựa trên heuristic

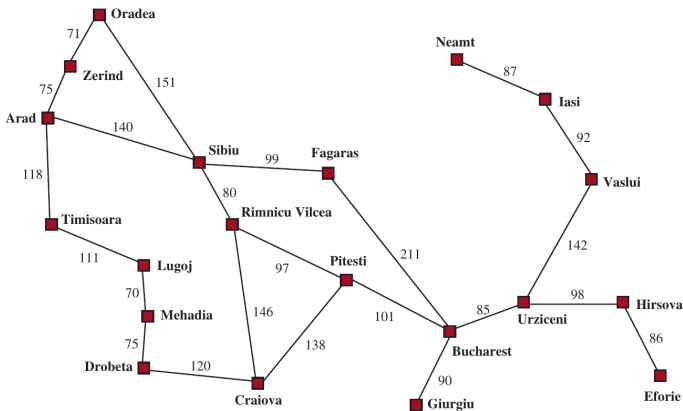
$$f(n) = h(n)$$

- Ví dụ: Khoảng cách Euclid giữa 2 điểm  $X$  và  $Y$  trong không gian  $d$  chiều

$$D_{XY} = \sqrt{\sum_{i=1}^d (X_i - Y_i)^2}$$

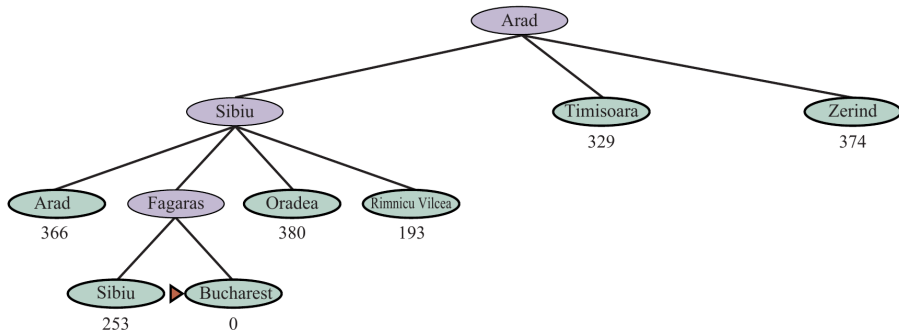


# Ví dụ



Bucharest	0
Giurgiu	77
Urziceni	80
Pitesti	100
Hirsova	151
Craiova	160
Eforie	161
Fagaras	176
Rimnicu Vilcea	193
Vaslui	199
Iasi	226
Neamt	234
Mehadia	241
Drobeta	242
Lugoj	244
Sibiu	253
Timisoara	329
Arad	366
Zerind	374
Oradea	380

# Cây tìm kiếm



- **Time:**  $O(b^m)$  thực thi nhanh với heuristic đủ tốt
- **Space:**  $O(b^m)$  giữ hết các nút trong bộ nhớ

# A\* Search

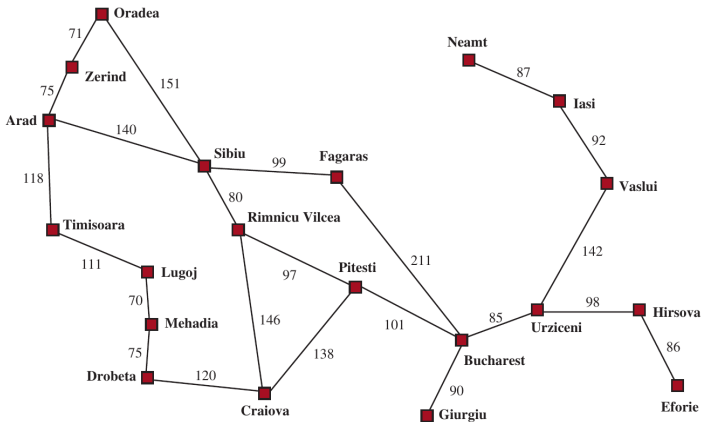
# A\* Search



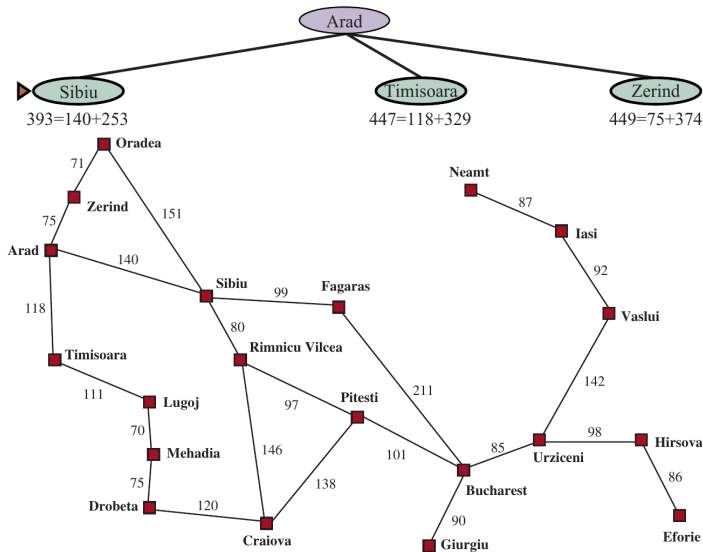
- A\* Search là dạng tìm kiếm Best-first Search phổ biến nhất
- Ý tưởng
  - Tích hợp heuristic vào quá trình tìm kiếm
  - Tránh các đường đi có chi phí lớn
- Hàm đánh giá:  $f(n) = g(n) + h(n)$ 
  - $g(n)$ : chi phí đường đi đến  $n$
  - $h(n)$ : ước tính khoảng cách đến đích
  - $f(n)$ : **ước tính** chi phí đến đích
- Gần như tương tự với UCS ngoại trừ  $f(n) = g(n) + h(n)$  thay vì  $f(n) = g(n)$



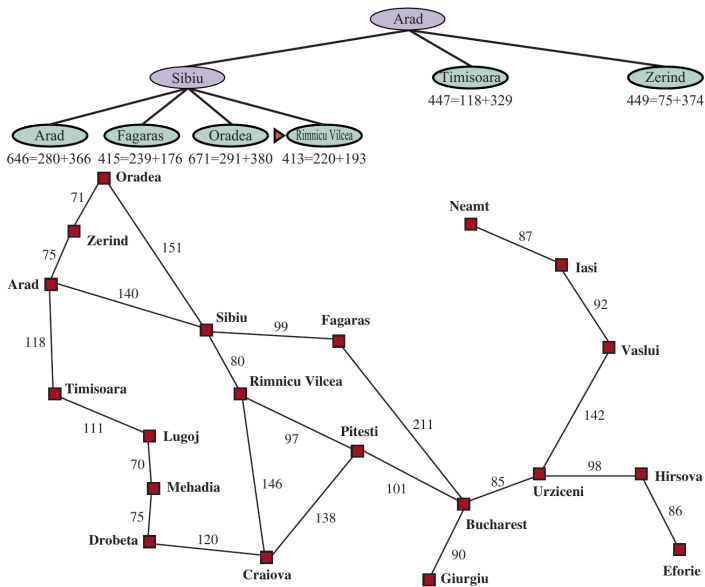
$$366 = 0 + 366$$



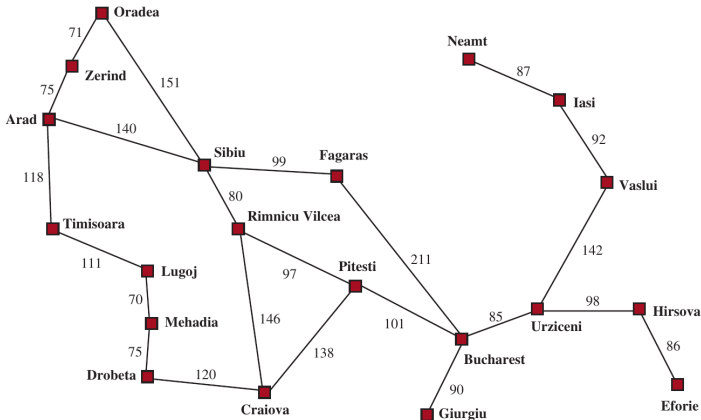
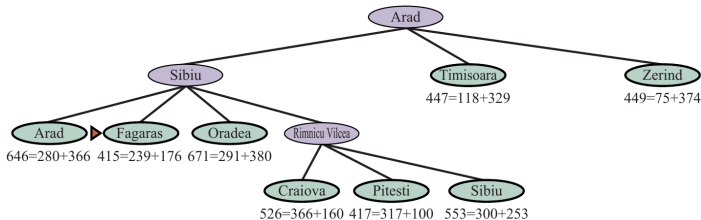
Bucharest	0
Giurgiu	77
Urziceni	80
Pitesti	100
Hirsova	151
Craiova	160
Eforie	161
Fagaras	176
Rimnicu Vilcea	193
Vaslui	199
Iasi	226
Neamt	234
Mehadia	241
Drobeta	242
Lugoj	244
Sibiu	253
Timisoara	329
Arad	366
Zerind	374
Oradea	380



Bucharest	0
Giurgiu	77
Urziceni	80
Pitesti	100
Hirsova	151
Craiova	160
Eforie	161
Fagaras	176
Rimnicu Vilcea	193
Vaslui	199
Iasi	226
Neamt	234
Mehadia	241
Drobeta	242
Lugoj	244
Sibiu	253
Timisoara	329
Arad	366
Zerind	374
Oradea	380

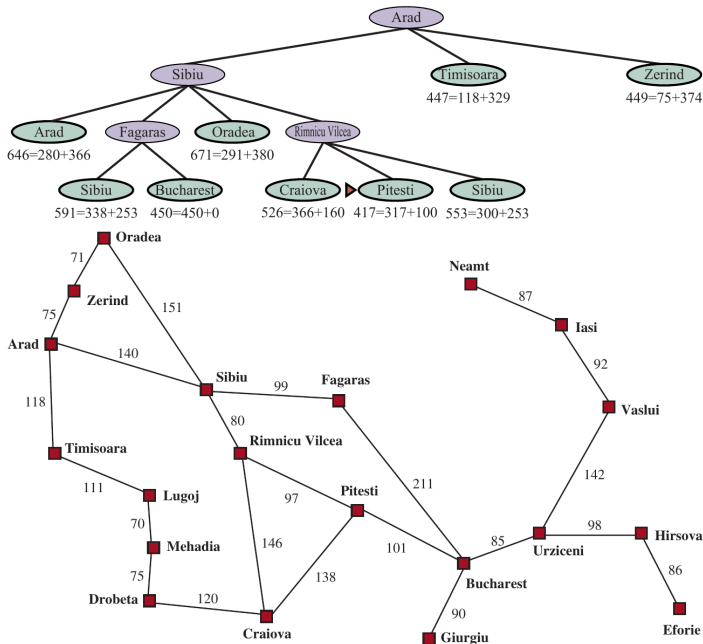


Bucharest	0
Giurgiu	77
Urziceni	80
Pitesti	100
Hirsova	151
Craiova	160
Eforie	161
Fagaras	176
Rimnicu Vilcea	193
Vaslui	199
Iasi	226
Neamt	234
Mehadia	241
Drobeta	242
Lugoj	244
Sibiu	253
Timisoara	329
Arad	366
Zerind	374
Oradea	380

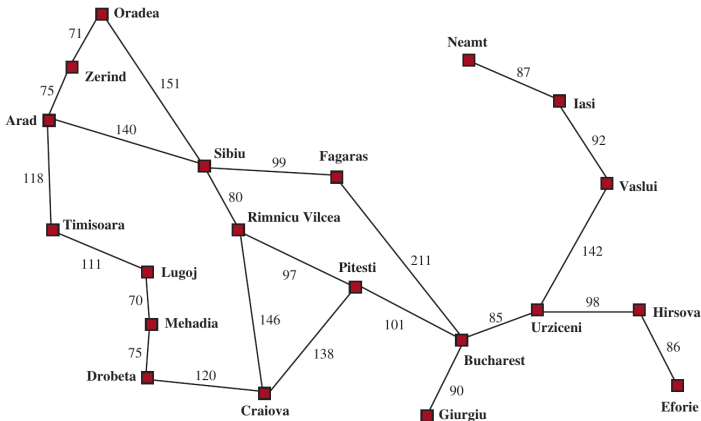
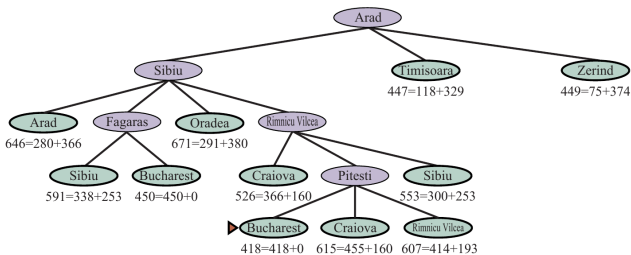


Bucharest	0
Giurgiu	77
Urziceni	80
Pitesti	100
Hirsova	151
Craiova	160
Eforie	161
Fagaras	176
Rimnicu Vilcea	193
Vaslui	199
Iasi	226
Neamt	234
Mehadia	241
Drobeta	242
Lugoj	244
Sibiu	253
Timisoara	329
Arad	366
Zerind	374
Oradea	380





Bucharest	0
Giurgiu	77
Urziceni	80
Pitesti	100
Hirsova	151
Craiova	160
Eforie	161
Fagaras	176
Rimnicu Vilcea	193
Vaslui	199
Iasi	226
Neamt	234
Mehadia	241
Drobeta	242
Lugoj	244
Sibiu	253
Timisoara	329
Arad	366
Zerind	374
Oradea	380



Bucharest	0
Giurgiu	77
Urziceni	80
Pitesti	100
Hirsova	151
Craiova	160
Eforie	161
Fagaras	176
Rimnicu Vilcea	193
Vaslui	199
Iasi	226
Neamt	234
Mehadia	241
Drobeta	242
Lugoj	244
Sibiu	253
Timisoara	329
Arad	366
Zerind	374
Oradea	380

# Đánh giá



- Với chi phí di chuyển thấp nhất là  $\epsilon$ ,  $C^*$  là chi phí lời giải tối ưu
- **Complete**: có nếu  $\epsilon > 0$  và không gian trạng thái hữu hạn
- **Optimal**: có nếu heuristic **hợp lý** và **nhất quán**
- **Time**: cấp số mũ
- **Space**: cấp số mũ

# Điều kiện tối ưu: hợp lý

## Hợp lý

$h(n)$  là một heuristic hợp lý khi:

- Không ước lượng quá cao chi phí đến đích
- $h(n) \leq h^*(n)$  chi phí thấp nhất từ  $n$  đến đích

VD: Khoảng cách Euclid trong việc tìm kiếm đường đi ở Romania

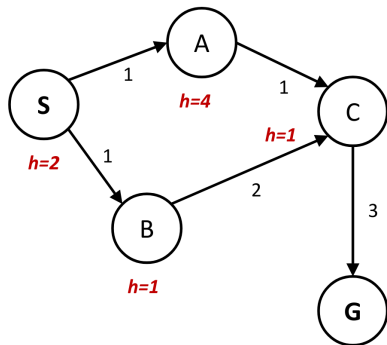
# Điều kiện tối ưu: hợp lý

Giả sử đường đi tối ưu có chi phí  $C^*$  nhưng thuật toán trả về đường đi với chi phí  $C > C^* \Rightarrow$  tồn tại nút  $n$  nằm trên đường đi tối ưu không được mở.

- $f(n) > C^*$  (do  $n$  không được mở)
- $f(n) = g(n) + h(n)$
- $f(n) = g^*(n) + h(n)$  (nguyên lý Bellman)
- $f(n) \leq g^*(n) + h^*(n)$  (vì heuristic **hợp lý**:  $h(n) < h^*(n)$ )
- $f(n) \leq C^*$  !!!

# Điều kiện tối ưu

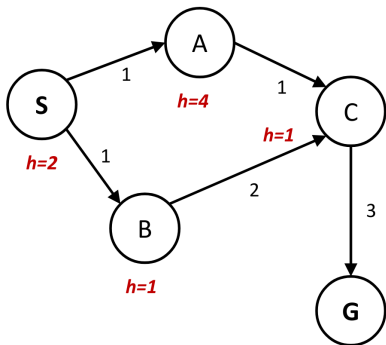
Liệu hợp lý là đã đủ?



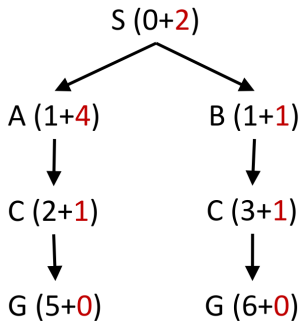
Hình 1: Không gian tìm kiếm

# Điều kiện tối ưu

Tại sao hợp lý vẫn chưa đủ?



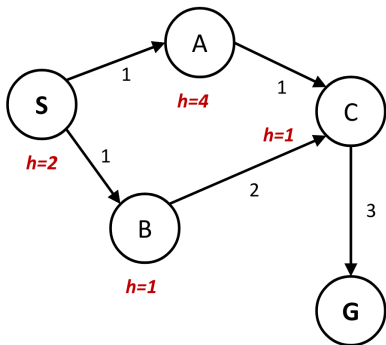
Hình 1: Không gian tìm kiếm



Hình 2: Cây tìm kiếm

# Điều kiện tối ưu

Tại sao hợp lý vẫn chưa đủ?



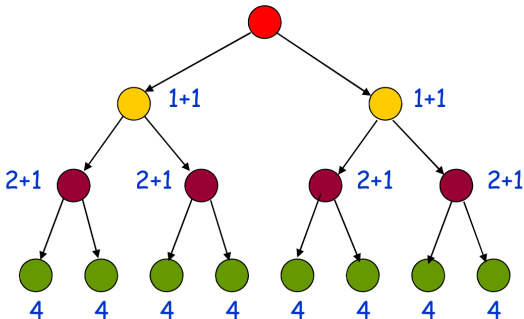
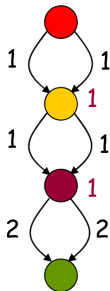
Nút C không được mở từ A!!!

Hình 1: Không gian tìm kiếm



# Nên làm gì với các trạng thái đã thăm?

- Loại bỏ nút mới, mở nút mục tiêu và trả về kết quả không tối ưu
- Không loại bỏ nút mới để đạt được kết quả tìm kiếm tối ưu

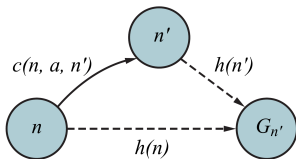


- **Space:**  $O(2^n)$  cấp số mũ!!!

# Điều kiện tối ưu: **nhất quán**

## Nhất quán

$h(n)$  là một heuristic nhất quán nếu với mỗi succesor  $n'$  của  $n$ , khoảng cách ước tính đến đích từ  $n$  không lớn hơn khoảng cách đến đích ước tính từ  $n'$  cộng với chi phí di chuyển từ  $n$  đến  $n'$ :  $h(n) \leq c(n, a, n') + h(n')$



Một heuristic **nhất quán** cũng là một heuristic **hợp lý**

# Điều kiện tối ưu: **nhất quán**

- Nếu  $h(n)$  **nhất quán**,  $f(n)$  không giảm dù đi bất kỳ đường nào.

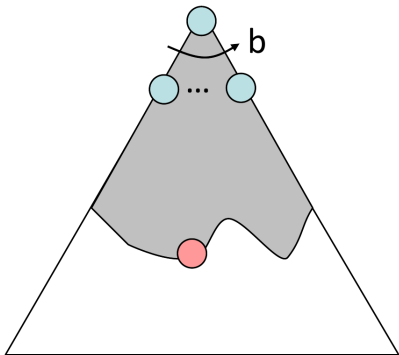
Chứng minh.

Nếu  $n'$  là successor của  $n$ :

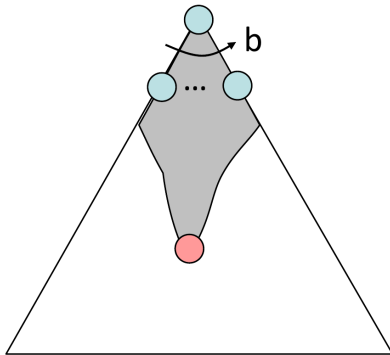
- $f(n') = g(n') + h(n') = g(n) + c(n, a, n') + h(n') \geq g(n) + h(n)$
- $f(n') \geq f(n)$

⇒ Nếu  $n'$  nằm trên đường đi tối ưu chứa  $n$  và  $g^*(n') > g^*(n)$ ,  $n'$  luôn luôn được mở sau  $n$ .

# Tính chất

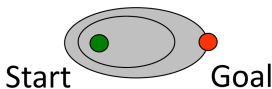
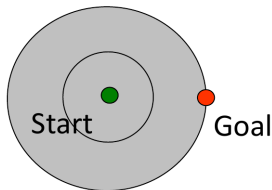


Hình 2: UCS



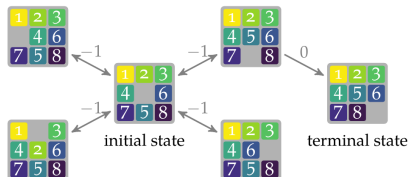
Hình 3:  $A^*$

- UCS mở các nút theo tất cả các hướng
- A\* mở theo hướng chính đến đích nhưng tính tối ưu dựa trên heuristic



# Bài toán 8-puzzle

- Chi phí lời giải trung bình cho một bài toán 8-puzzle ngẫu nhiên là 22 bước
- $b$  khoảng 3
  - $3^{22} \approx 3.1 \times 10^{10}$  trạng thái (cây tìm kiếm)
  - $9!/2 = 181440$  trạng thái có thể đi đến (không gian tìm kiếm)
- 15-puzzle: khoảng  $10^{13}$ !!!

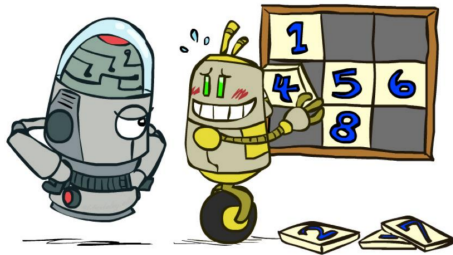


# Bài toán 8-puzzle I

- Nếu các ô có thể dịch chuyển tự do chồng lên các ô khác?
- Khoảng cách **Manhattan**

$$D_{XY} = \sum_{i=1}^d |X_i - Y_i|$$

- Tại sao heuristic này lại **hợp lý**?

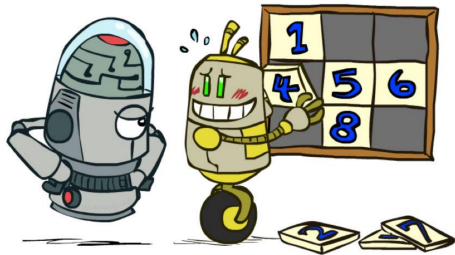


# Bài toán 8-puzzle I

- Nếu các ô có thể dịch chuyển tự do chồng lên các ô khác?
- Khoảng cách **Manhattan**

$$D_{XY} = \sum_{i=1}^d |X_i - Y_i|$$

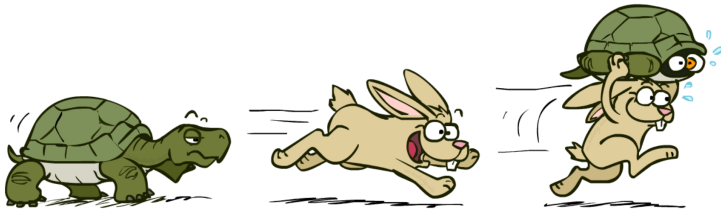
- Tại sao heuristic này lại **hợp lý**?
- Đây là heuristic cho bài toán nổi lổng





# A\* Search

- A\* dùng khoảng cách ước tính và chi phí di chuyển
- A\* tối ưu với heuristic nhất quán
- Thiết kế heuristic: sử dụng heuristic của bài toán nói lỏng



# Giới hạn không gian tìm kiếm

# Giới hạn không gian tìm kiếm

- Chi phí bộ nhớ cho thuật toán  $A^*$ :  $b^d$
- **Cấp số mũ!!!**
- Trên thực tế, máy tính thường bị tràn bộ nhớ khi thực thi thuật toán  $A^*$
- **Ý tưởng:** sử dụng hướng tiếp cận theo chiều sâu?

# Giới hạn không gian tìm kiếm

- Iterative-deepening A\* (IDA\*): Sử dụng giá trị cut-off cho **chi phí ước tính**.
- Recursive best-first search (RBFS): Giữ lại nút cha có **chi phí ước tính thay thế** tốt nhất của nút hiện tại.
  - **Space:**  $O(bd)$  !!!
  - **Time:** khó ước lượng nhưng vẫn có thể là cấp số mũ

# Tìm kiếm tốt hơn bằng phương pháp học?

"Those who do not learn from history are doomed to repeat it"

Tạm dịch: Sai lầm sẽ lặp lại với những người không chịu học hỏi

- Học từ kinh nghiệm di chuyển để tránh mở các cây con không mong muốn.
- Tài liệu thực hành tham khảo?

# Tổng kết



- **Best-first Search:** Tìm kiếm dựa trên chi phí ước tính  $f(n)$
- **Greedy Best-first Search:** Khoảng cách ước tính  $h(n)$
- **A\*:** Chi phí ước tính  $f(n) =$  Khoảng cách ước tính  $h(n) +$  Chi phí di chuyển  $g(n)$ 
  - Tối ưu: nhất quán
- **Thiết kế heuristic:** sử dụng bài toán nới lỏng (relaxed problem)
- **Giới hạn không gian tìm kiếm:** hướng tiếp cận theo chiều sâu
- **Tìm kiếm tốt hơn:** nhận xét, học hỏi trước khi hành động

# Tài liệu tham khảo

- [1] Bùi Tiến Lên, Bộ môn Khoa học máy tính  
?Bài giảng môn Cơ sở trí tuệ nhân tạo?
- [2] Michael Negnevitsky Russell, S. and Norvig, P. (2021).  
?Artificial intelligence: a modern approach.?
- [3] Berkelev University  
?CS188?
- [4] Prof. Eric Grimson, Prof. John Guttag  
?MIT 6.0002 Course?.