**1. Thuật toán tìm kiếm trên đồ thị**

(a) Tìm kiếm theo chiều rộng (Breadth-first - BFS)

Thứ tự các node được thêm vào Explored Set: S, A, C, D, B, F, E, G

Đường đi trả về: S→ D →G

Tổng chi phí: 2 + 8 = 10

(b) Tìm kiếm theo chiều sâu (Depth-first - DFS)

Thứ tự các node được thêm vào Explored Set: S, A, C, F, G

Đường đi trả về: S→C→F →G

Tổng chi phí: 2 + 1 + 4 = 7

(c) Lặp lại tìm kiếm theo chiều sâu (Iterative deepening - IDDFS)

Thứ tự các node được thêm vào Explored Set: S, A, C, F, D, B, G

Đường đi trả về: S→D →G

Tổng chi phí: 2 + 8 = 10

**2. Tháp Hà Nội (Tower of Hanoi)**

(a) Biểu diễn trạng thái:

Một bộ n giá trị,wps, trong đó wps là cọc mà đĩa i (đĩa nhỏ nhất là i=1) đang nằm.

(b) Kích thước không gian trạng thái: wps

(c) Trạng thái bắt đầu:

wps (tất cả n đĩa trên cọc 1).

(d) Các hành động hợp pháp:

Di chuyển đĩa trên cùng từ cọc nguồn i sang cọc đích j wps, miễn là đĩa đang di chuyển nhỏ hơn đĩa trên cùng của cọc đích (hoặc cọc đích trống).

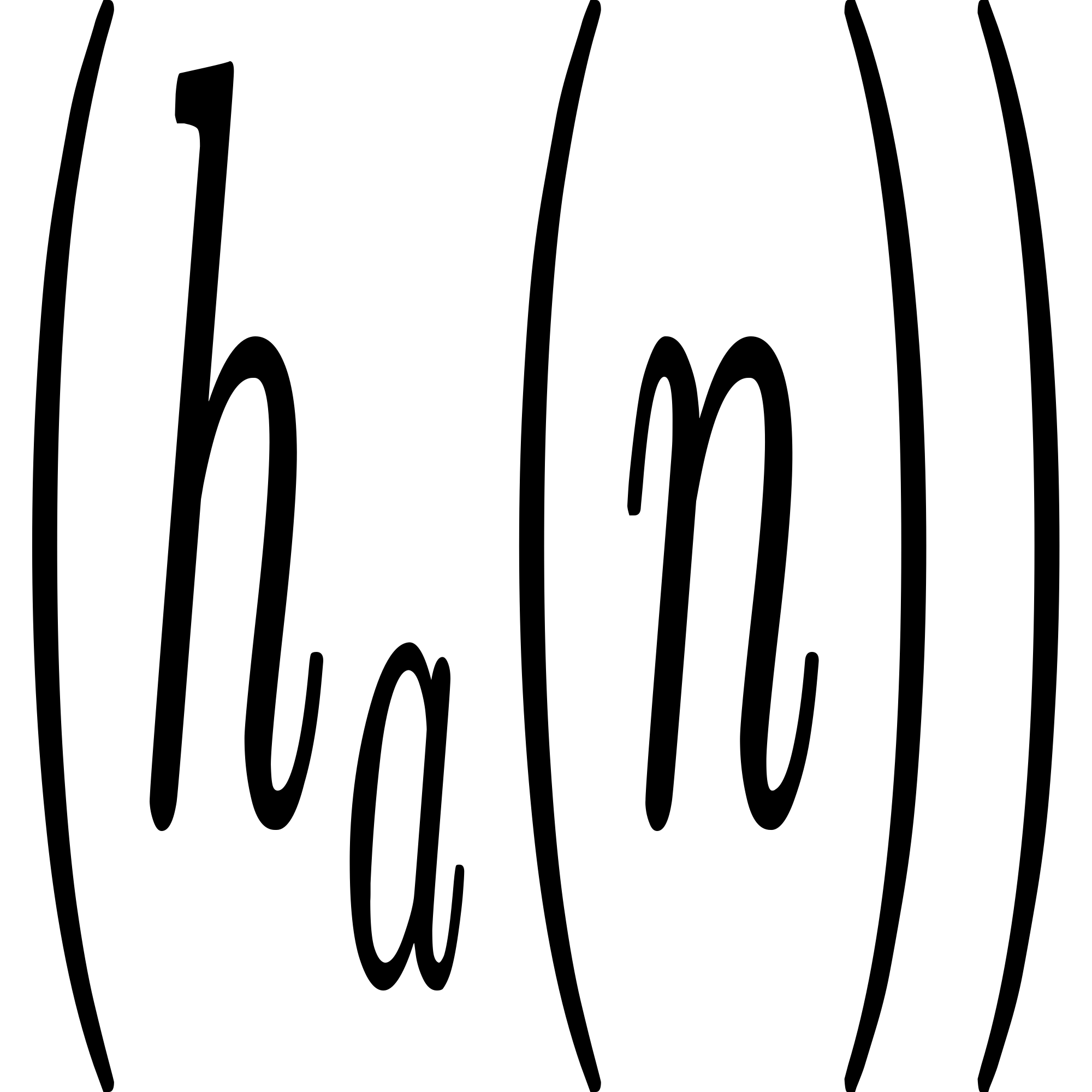
(e) Kiểm tra trạng thái đích:

wps cho tất cả wps (tất cả đĩa trên cọc 3).

1. **Thiết kế và Hiểu Heuristic (Designing & Understanding Heuristics)**

(Sử dụng đồ thị trên trang 3. Chi phí tối ưu thực tế h\*(n): S=8, A=10, B=5, C=6, D=4, G=0)

Heuristic mẫu:

* Heuristic Admissible & Consistent wps: S=8, A=10, B=5, C=6, D=4, G=0 (bằng h\*).
* Heuristic Admissible nhưng KHÔNG Consistent : S=7, A=8, B=3, C=3, D=0, G=0. (Vi phạm tính nhất quán tại wps nhưngwps.

(a) Đường đi tìm được bởi wps sử dụng heuristic nhất quán C:/Users/Windows 11/AppData/Local/Temp/wps.iRUVXHwps

Đường đi trả về: wps

Tổng chi phí: 8

(b) Heuristic admissible nhưng không consistent:

Sử dụng $h\_a(n)$ đã thiết kế ở trên: S=7, A=8, B=3, C=3, D=0, G=0.

(c) Giải thích tại sao một consistent heuristic phải là admissible:

Giải thích: Tính nhất quán (consistency) là wps. Áp dụng điều kiện này dọc theo đường đi tối ưu từ n đến node đích G và biết wps, ta có:wps.

Vì wps là định nghĩa của tính chấp nhận được (admissible), nên mọi heuristic nhất quán đều phải chấp nhận được.

**4. Săn lùng Kho báu (Treasure Hunting)**

(Start A, Goals G và H. Phá vỡ hòa theo thứ tự bảng chữ cái.)

(a) Depth-First Search (DFS)

Explored set: A, B, D, F, H

Path returned: A → B → D → F → H

Cost: 11 (2 + 3 + 6 + 0)

(b) Breadth-First Search (BFS)

Explored set: A, B, C, D, E, F, G

Path returned: A → C → E → G

Cost: 3 (1 + 1 + 1)

(c) Uniform-Cost Search (UCS)

Explored set: A, C, B, E, G

Path returned: A → C → E → G

Cost: 3 (1 + 1 + 1)

(d) Greedy Search

Explored set: A, B, D, F, H

Path returned: A → B → D → F → H

Cost: 11 (2 + 3 + 6 + 0)

(e) A\* Search

Explored set: A, B, C, E, G

Path returned: A → C → E → G

Cost: 3 (1 + 1 + 1)

**5. Tìm kiếm đối kháng (Adversarial Search)**

(Minimax với Alpha-Beta Pruning, Root là Maximizer, từ trái sang phải.)

Giá trị cuối cùng được trả về tại Root (MAX): 3

Giá trị tại các Min Node (tầng 2): Min 1 = 3, Min 2 = -2, Min 3 = -7

Giá trị tại các Max Node (tầng 3): Max 1 = 3, Max 2 = 6, Max 3 = 10, Max 4 = 12, Max 5 = -2, Max 7 = 10, Max 8 = -7

Các cạnh bị cắt tỉa (Pruned Edges):

* Cạnh từ Min 2 đến Max 6 (vì giá trị của Max 5 là -2, nhỏ hơnwps của Root, nên Max 6 bị cắt).
* Cạnh từ Min 3 đến Max 9 (vì giá trị của Max 8 là -7, nhỏ hơn wps của Root, nên Max 9 bị cắt).

**6. Phần Đúng/Sai (True/False Section)**

(a) DFS luôn mở rộng ít nhất bằng số node so với A\* search với admissible heuristic.

SAI. DFS có thể đi sâu vào một nhánh rất dài, vô dụng và mở rộng nhiều node hơn A\*, vốn được hướng tới mục tiêu.

(b) Manhattan distance là admissible heuristic cho xe pháo (rook).

SAI. Chi phí thực tế (số lần di chuyển tối thiểu) của xe pháo là 1 hoặc 2. Manhattan distance thường lớn hơn 2, nên nó ước tính quá cao chi phí thực tế (h(n) > h\*(n)).

(c) Euclidean distance là admissible heuristic cho Pacman path-planning problems.

ĐÚNG. Khoảng cách Euclidean (đường chim bay) luôn nhỏ hơn hoặc bằng chi phí đường đi thực tế trên lưới (chỉ đi ngang/dọc), do đó nó không bao giờ ước tính quá cao (h(n) ≤ h\*(n)).

(d) The sum of several admissible heuristics is still an admissible heuristic.

SAI. Tổng của các heuristic chấp nhận được có thể lớn hơn chi phí tối ưu (hsum > h\*), do đó nó không chấp nhận được.

(e) Admissibility of a heuristic for A\* search implies consistency as well.

SAI. Tính nhất quán (consistency) là một điều kiện mạnh hơn. Một heuristic chấp nhận được có thể không nhất quán.

(f) A\* with graph search is always optimal with an admissible heuristic.

ĐÚNG. Nếu heuristic không bao giờ ước tính quá cao chi phí thực tế, A\* với graph search được đảm bảo tìm thấy đường đi có chi phí tối ưu.

(g) VM (Minimax) is guaranteed to be less than or equal to VE (Expectimax).

SAI. VM (kịch bản xấu nhất) không được đảm bảo nhỏ hơn VE (trung bình). Nếu các kết quả ngẫu nhiên trong Expectimax có xác suất cao rơi vào trạng thái cực kỳ tệ, VE có thể thấp hơn VM.

(h) Using the optimal minimax policy in the game corresponding to the modified (chance) game tree is guaranteed to result in a payoff of at least VE.

SAI. Chính sách Minimax tối ưu hóa cho kịch bản xấu nhất. Chính sách tối ưu Expectimax được thiết kế để tối đa hóa lợi ích kỳ vọng. Do đó, chính sách Minimax không được đảm bảo mang lại lợi ích kỳ vọng bằng hoặc lớn hơn VE.