ARTIFICIAL INTELLIGENCE SOKOBAN

Thành viên nhóm

NGUYỄN NGỌC MINH - 21110784
NGUYỄN NGỌC MẠNH - 21110781
VÕ HOÀNG TÙNG - 21110811
ĐẶNG VĂN THÔNG - 19110056

NỘI DUNG

01 GIỚI THIỆU SOKOBAN

02 GIẢI THUẬT DFS TRONG SOKOBAN

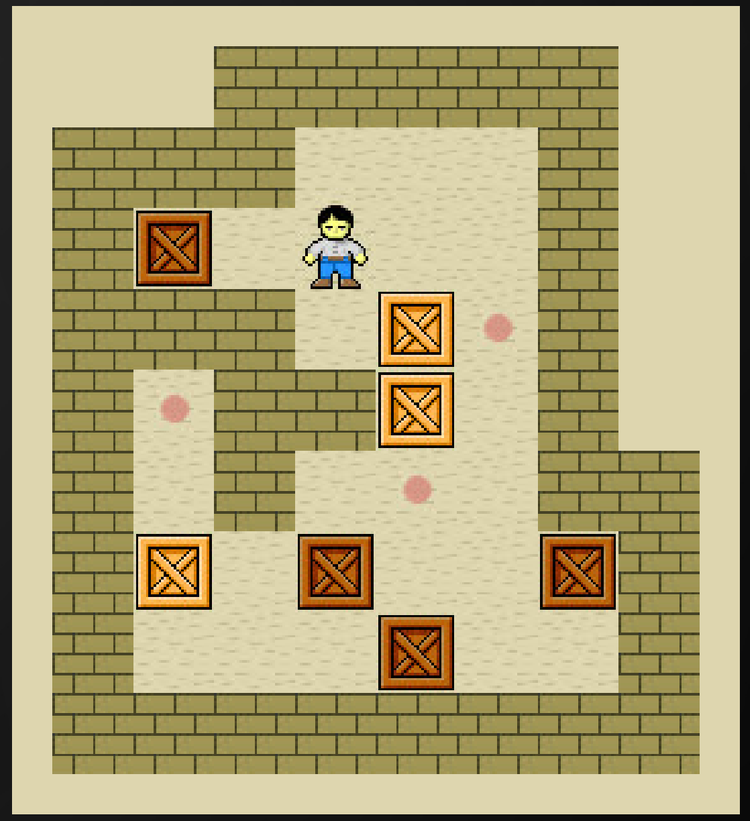
03 GIẢI THUẬT A STAR TRONG SOKOBAN

04 SO SÁNH VÀ ĐỐI CHIẾU

1. TỔNG QUAN VỀ SOKUBAN AI

Giới thiệu Sokoban

- Sokoban là một trò chơi thuộc thể loại giải đố.
- Người chơi điều khiển một nhân vật, có thể di chuyển theo các hướng lên, xuống, trái, phải. Trên bảng có các khối vuông, được đặt ở các vị trí khác nhau.
- Mục tiêu của trò chơi là đẩy tất cả các khối vuông đến các vị trí đích được đánh dấu trên bảng.



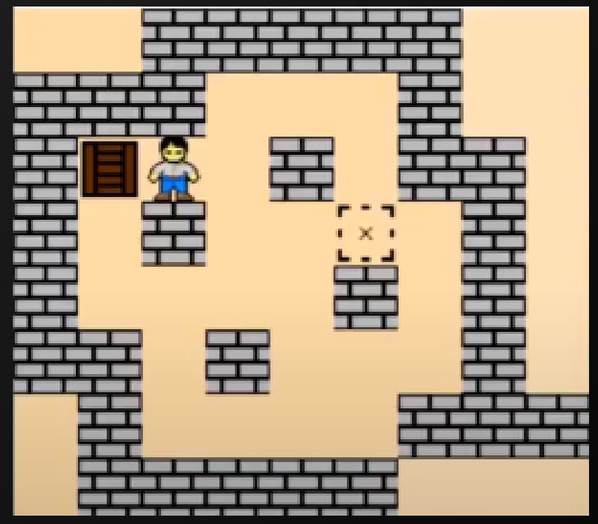
Hình 1: Sokuban AI

Định nghĩa các trạng thái

-Trạng thái của bài toán là 1 ma trận 2 chiều lưu vị trí của các thực thể.
- Mỗi bước di chuyển của nhân vật sẽ thay đổi trạng thái của bài toán.

Hình 2: Trạng thái (state)

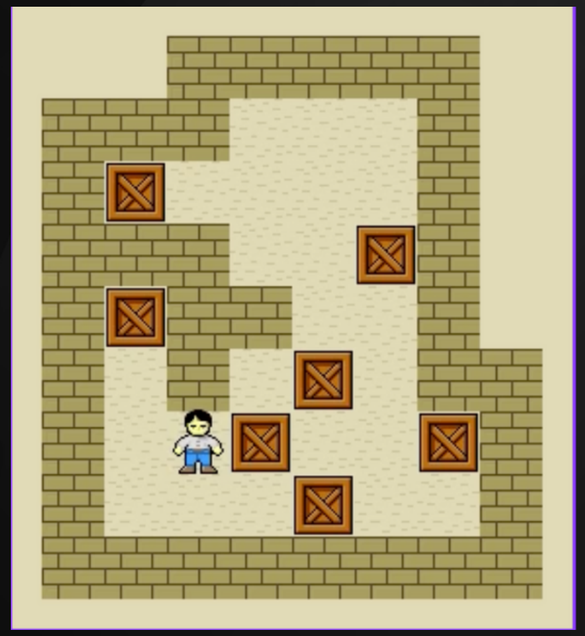
- Trạng thái mà khóa chết (deadlock): khi hộp nằm vào góc chết và nhân vật không thể di chuyển hộp được nữa



Hình 3: khóa chêt (deadlock)

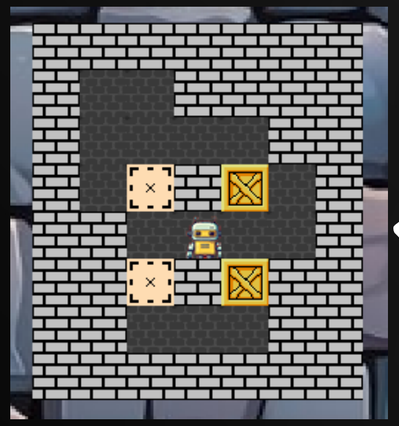
Định nghĩa các trạng thái

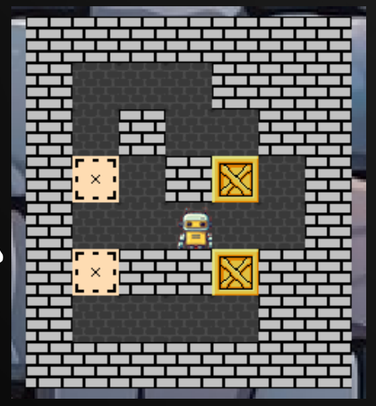
Goal state : Là trạng thái mà tất cả các hộp vào đúng vị trị dích



Hình 4: Goal state

Cách thức thực thi Sokuban





Chọn map

Depth First Search

SPACE

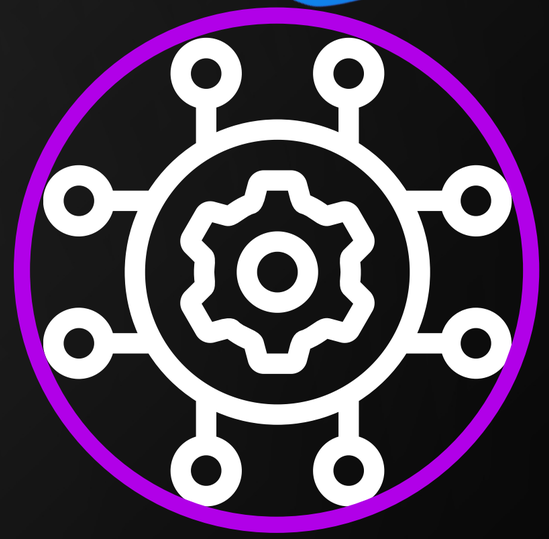
Search A*

Chọn giải thuật

ENTER

Giải

2. CÁC GIẢI THUẬT TRONG SOKUBAN

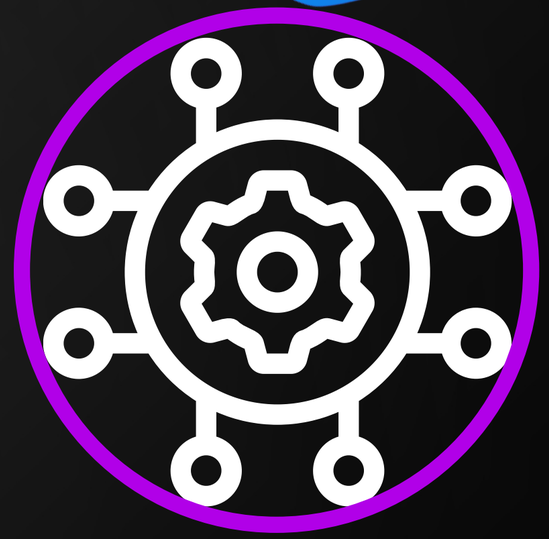


2. Giải thuật cho bài toán

- Input: Trạng thái khởi đầu của bài toán
- Ouput: Solution dưới dạng chuỗi các bước di chuyển 'U', D', 'L', 'R'
    + Nếu không giải được (deadlock hoặc map sai) => Solution là "NoSol"
    + Nếu hết thời gian (Time out) => Solution là "Timeout"

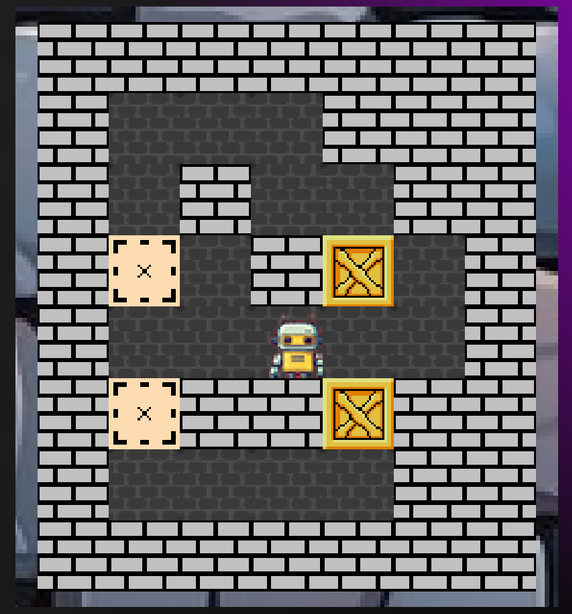
2.1. GIẢI THUẬT DFS TRONG SOKOBAN AI

- Là một giải thuật vét cạn, dùng để duyệt qua tất cả các trạng thái di chuyển có thể xảy ra khi áp dụng vào sokoban để tìm tới trạng thái mục tiêu
- Từ trạng thái ban đầu, sinh ra tối đa 4 trạng thái với các nước đi của người chơi tiếp theo có thể đi. Từ đó tạo thành một cầu trúc cây. Cuối cùng dùng DFS để vét cạn cây tìm trạng thái mục tiêu



Bắt đầu

* Kiểm tra nếu đã thắng trong trạng thái ban đầu
  * Nếu có, trả về đường đi
  * Nếu không, tạo trạng thái ban đầu danh sach đã thăm và sẽ thăm



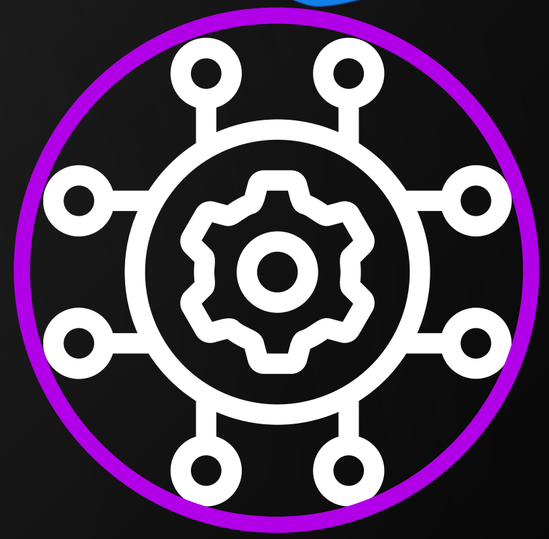
Lặp

* Lấy trạng thái hiện tại
  * Tìm vị trí hiện tại của người chơi trên bảng game
  * Lấy danh sách các vị trí mà người chơi có thể di chuyển đến
* Đối với mỗi vị trí mà người chơi có thể di chuyển đến:
  * Tạo một trạng thái mới
  *Kiểm tra xem trong trạng thái mới các hộp có bị kẹt góc hay không
  * Kiểm tra nếu trạng thái mới đã được duyệt trước đó
    * Nếu có, bỏ qua trạng thái mới
  * Kiểm tra nếu trạng thái mới có khả năng thắng
    * Nếu có, trả về đường đi
    * Nếu không, thêm trạng thái mới vào ngăn xếp và danh sách đã thăm

Kết thúc

* Nếu không tìm thấy đường đi, in thông báo
  * Trả về kết quả là không tìm thấy đường đi

2.2. GIẢI THUẬT A*
TRONG SOKOBAN AI



2.2 Giải thuật A* trong Sokoban

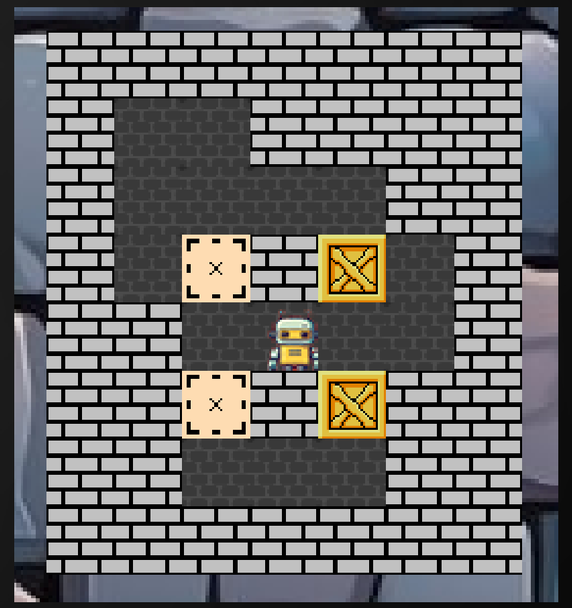
A*, là một giải thuật heuristic sử dụng một hàm lượng giá và một hàng đợi ưu tiên để tính toán các nước đi ngắn nhất dẫn đến trạng thái mục tiêu

Tương tự với DFS, tuy nhiên các trạng thái sinh ra sẽ có giá trị lượng giá riêng và được sắp xếp vào một hàng đợi ưu tiên để từ đó giải thuật có thể lấy từ hàng đợi ra duyệt để đi chế độ rạp chiếu trạng thái mục tiêu nhanh

2.2 Giải thuật A* trong Sokoban

Bắt đầu

* Kiểm tra nếu đã thắng trong trạng thái ban đầu
* Nếu có, trả về đường đi
* Nếu không, tạo trạng thái ban đầu
* Đưa trạng thái ban đầu vào hàng đợi ưu tiên .



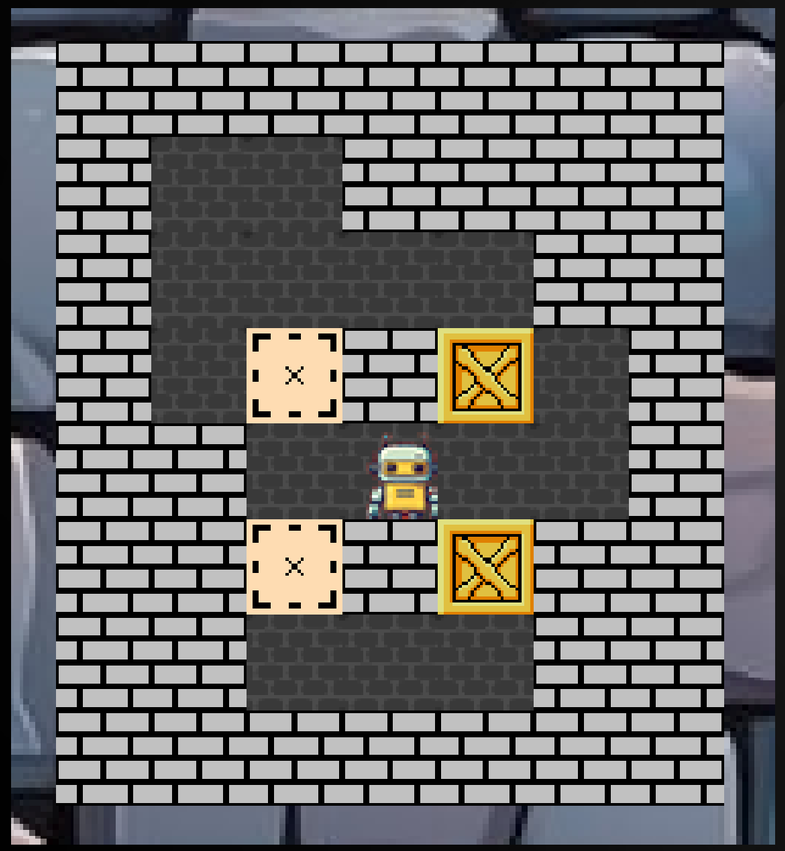
Lặp

*. Lấy trạng thái có độ ưu tiên cao nhất ra.
* Lấy danh sách các bước di chuyển có thể
* Đối với mỗi bước di chuyển có thể:
    * Tạo một trạng thái mới
    * Kiểm tra nếu trạng thái mới đã được duyệt trước đó
    * Nếu có, bỏ qua trạng thái mới
    *Kiểm tra xem trong trạng thái mới các hộp có bị kẹt góc hay không
    * Nếu không, kiểm tra nếu trạng thái mới có khả năng thắng
    * Nếu có, trả về đường đi
    * Nếu không, thêm trạng thái mới vào danh sách và hàng đợi ưu tiên

Kết thúc

* Nếu không tìm thấy đường đi, in thông báo
* Trả về kết quả là không tìm thấy đường đi

2.2 Giải thuật A* trong Sokoban



Tính Toán Heuristic Cho Mỗi Trạng Thái Mới

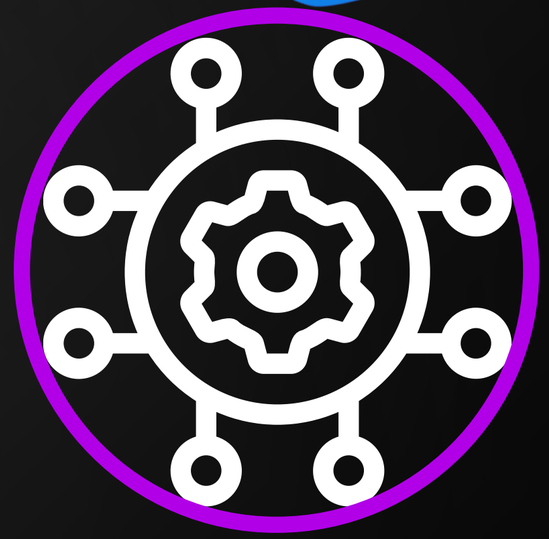
Tìm vị trí các hộp trên bảng.

Tính heuristic dựa trên tổng khoảng cách từ hộp đến điểm kiểm tra tương ứng.

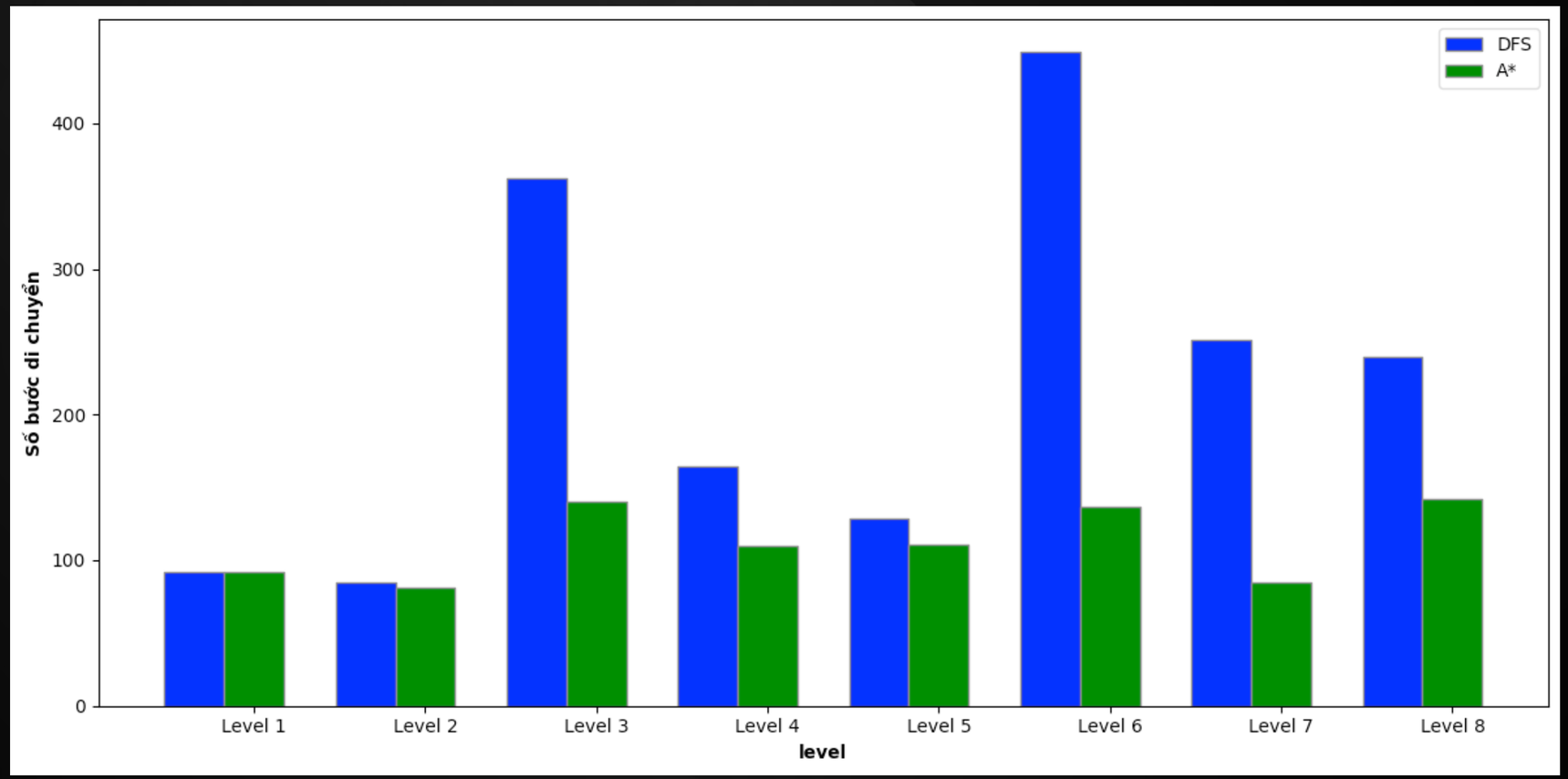
Cập Nhật Heuristic Trong Trạng Thái

Heuristic = Chi phí đến trạng thái hiện tại hộp + Tổng khoảng cách từ hộp đến điểm kiểm tra.

3. SO SÁNH DFS VÀ A*

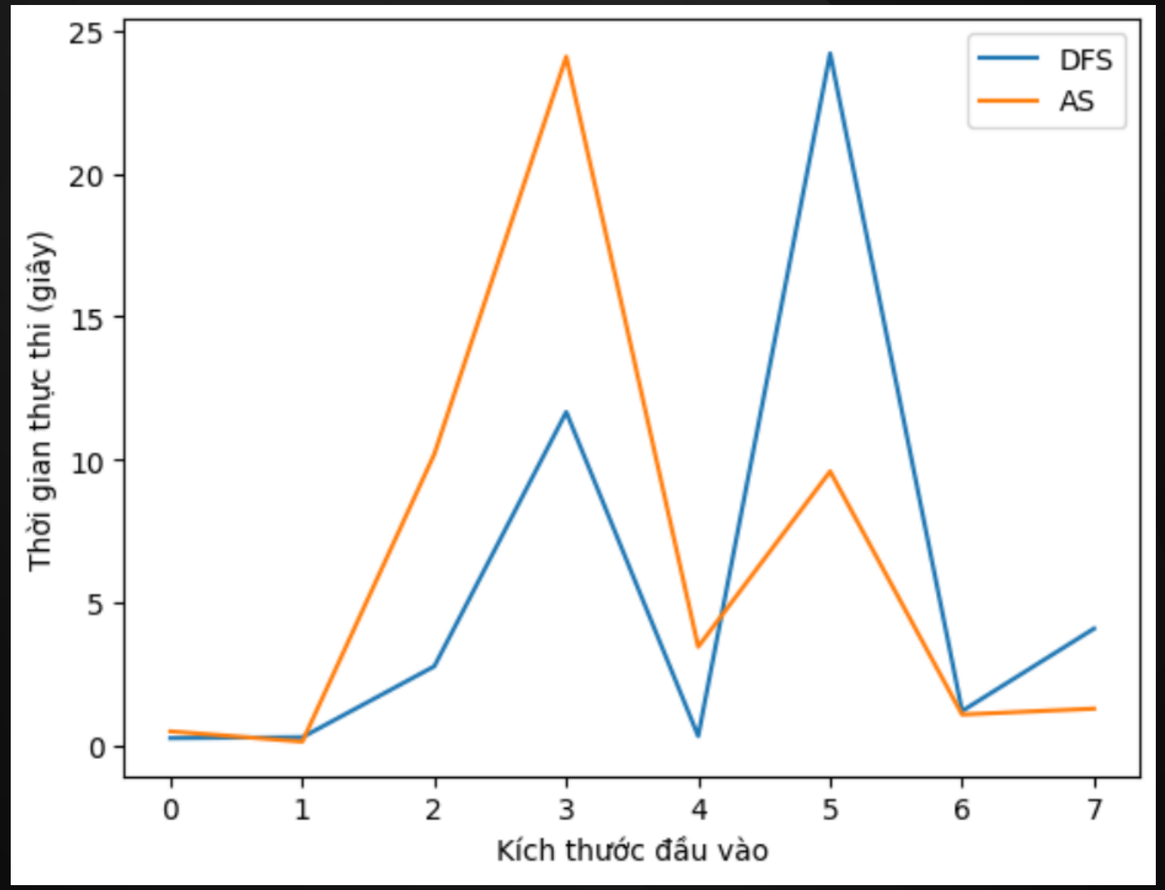


3.1 DFS vs A* ALGORITHM (Số bước di chuyển)



Theo thống kê, A* luôn cho ra số state duyệt ít hơn DFS

3.2 DFS vs A Star ALGORITHM (Thời Gian)



My team dedicate

THANKS EVERYONE

For watching this presentation