

# **BÁO CÁO ĐỒ ÁN 2**

Lecture: Pham Trong Nghia

TA: Nguyen Thai Vu

**CƠ SỞ TRÍ TUỆ NHÂN TẠO - ADVERSARIAL**

# MỤC LỤC

<b>1. Tổng quan</b>	2
<b>1.1. Thông tin sinh viên</b>	2
<b>1.2. Vấn đề</b>	2
<b>2. Ý tưởng thuật toán Minimax</b>	2
<b>2.1. Khái niệm</b>	2
<b>2.2. Cách hoạt động</b>	2
<b>3. Tính hoàn chỉnh</b>	3
<b>4. Độ phức tạp thuật toán</b>	4
<b>5. Nguồn tham khảo</b>	4

# 1. Tổng quan

## 1.1. Thông tin sinh viên

✚ Họ và tên: Phạm Hiền Đoàn Trang  
✚ MSSV: 20127362

## 1.2. Vấn đề

\_ Để giải quyết trò chơi sử dụng AI, chúng ta sẽ dùng Adversarial Search mà rõ ràng hơn là sử dụng thuật toán Minimax. Đầu tiên là khái niệm về cây trò chơi, các trạng thái khác nhau của trò chơi được biểu diễn bằng các nút trong cây trò chơi. Trong cây trò chơi, các nút được sắp xếp theo các cấp độ tương ứng với lượt của mỗi người chơi trong trò chơi sao cho nút “gốc” của cây (thường ở trên cùng của sơ đồ) là vị trí bắt đầu trong trò chơi - là 1 bảng rỗng, không có đánh dấu X hoặc O. Dưới gốc, ở cấp thứ hai, có những trạng thái có thể xảy ra từ các nước đi của người chơi đầu tiên, có thể là X hoặc O. Chúng tôi gọi các nút này là con của nút gốc. Mỗi nút ở cấp độ thứ hai sẽ có thêm các nút con là các nước có thể đi của 1 lượt chơi. Điều này cứ tiếp tục đến khi người thắng hoặc máy thắng hoặc hòa (cả 2 không thể di chuyển thêm bước nào nữa).

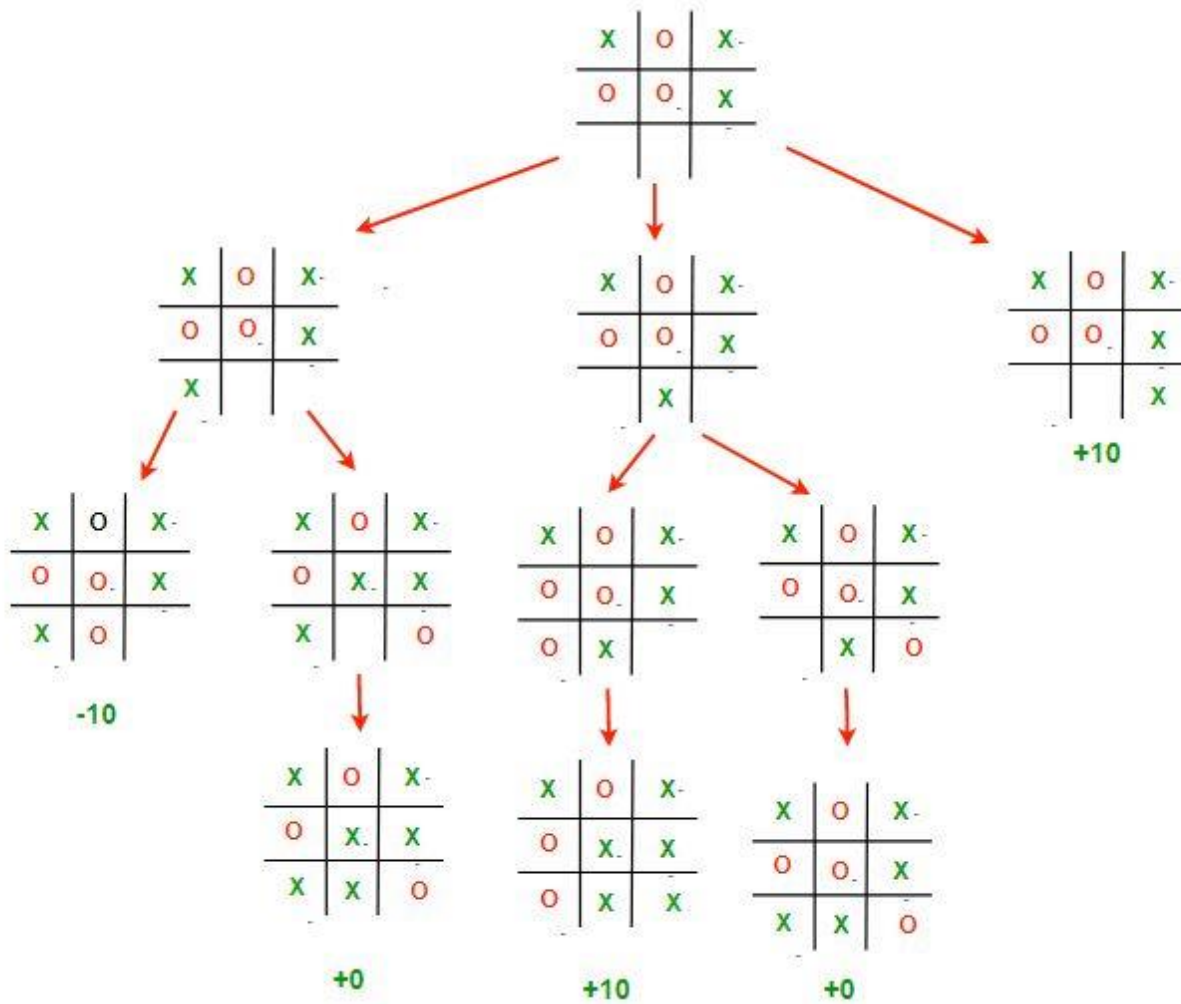
# 2. Ý tưởng thuật toán Minimax

## 2.1. Khái niệm

\_ Minimax là một trí tuệ nhân tạo được áp dụng trong các trò chơi hai người chơi, chẳng hạn như tic-tac-toe, cờ vua và cờ vây. Trò chơi này được gọi là trò chơi có tổng bằng 0, bởi vì trong một biểu diễn toán học: lượt chơi máy thắng (+1) hoặc lượt chơi máy thua (-1) hoặc hòa (0).

## 2.2. Cách hoạt động

\_ Thuật toán tìm kiếm các nước đi một cách đệ quy, nước đi tốt nhất dẫn đến việc máy thắng hoặc hòa. Nó xem xét trạng thái hiện tại của trò chơi và các nước đi có sẵn ở trạng thái đó, sau đó đối với mỗi nước đi hợp lệ, nó sẽ chơi (tối thiểu và tối đa xen kẽ) cho đến khi tìm thấy trạng thái cuối cùng (thắng, hòa hoặc thua).



### 3. Tính hoàn chỉnh

\_ Minimax như đã xem xét là thuật toán buộc phải có toàn bộ không gian trạng thái để triển khai, có thể gán giá trị cho các nút lá và tính ngược lại → Việc này không khả thi vì không gian bài toán là rất lớn. Giả sử với những map có không gian rộng hơn như 5x5, 10x10.... thì thuật toán sẽ tốn rất nhiều thời gian chạy. Theo em, thuật toán Minimax chưa có tính tối ưu, chưa phải là 1 thuật toán hoàn thiện nhất.

## 4. Độ phức tạp thuật toán

- \_ Thời gian:  $O(m^d)$
- \_ Không gian:  $O(md)$
- \_ Với  $m$ : số bước đi hợp lệ tại mỗi nút  
     $d$ : độ sâu lớn nhất của cây

## 5. Nguồn tham khảo

- \_ <https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-3-tic-tac-toe-ai-finding-optimal-move/>
- \_ <https://www.youtube.com/watch?v=trKjYdBASyQ&t=1396s>
- \_ <https://mindovermetal.org/ung-dung-cua-thuat-toan-minimax-1639417544/>

## 6. Link video demo

<https://drive.google.com/file/d/18LyxJADAXXc5cin-kouElCcCsaNZSw1g/view?usp=sharing>