**CONNECT4**

**Task 1: Defining the Search Problem [10 points]**

Mục tiêu

Cần **mô hình hóa trò chơi** (ví dụ: Connect 4) **dưới dạng một bài toán tìm kiếm (search problem)** để các thuật toán như *Minimax* hoặc *Alpha-Beta Pruning* có thể hoạt động trên đó.

1. **Initial state** – Bàn cờ trống lúc bắt đầu.

* Là **bàn cờ trống** khi trò chơi bắt đầu.
* Trong Connect 4: đó là một ma trận 6x7 (6 hàng, 7 cột) chứa toàn giá trị rỗng (0 hoặc "."), chưa có quân của người chơi nào.

1. **Actions** – Các nước đi hợp lệ (những cột có thể thả quân).

* Là **tất cả nước đi hợp lệ** mà người chơi có thể thực hiện trong trạng thái hiện tại.
* Ví dụ trong Connect 4: người chơi có thể **thả quân vào bất kỳ cột nào chưa đầy**

1. **Transition model (result function)** – Cách bàn cờ thay đổi khi thực hiện nước đi.

* Mô tả **kết quả khi thực hiện một hành động trong một trạng thái**.
* Tức là: nếu ta thả quân của người chơi A vào cột col, trạng thái bàn cờ mới sẽ là gì?

1. **Goal state & Utility** – Khi nào trò chơi kết thúc và điểm số của người chơi.

* Trạng thái kết thúc là **khi có người thắng hoặc bàn cờ đầy**.
* Utility (điểm thưởng):
  + +1 nếu người chơi A thắng
  + −1 nếu người chơi B thắng
  + 0 nếu hòa

Code:

A computer screen shot of a program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

| **Thành phần** | **Hàm** | **Mô tả** |
| --- | --- | --- |
| **Initial state** | initial\_state() | Khởi tạo bàn cờ trống 6x7 |
| **Actions** | actions(state) | Trả về danh sách cột có thể thả quân |
| **Transition model** | result(state, action, player) | Sinh ra bàn cờ mới sau khi thực hiện nước đi |
| **Goal & Utility** | terminal\_test(state), utility(state, player) | Xác định trạng thái kết thúc và điểm số cho người chơi |

Giải thích kết quả:

Tất cả 7 cột đều trống nên **mọi hành động hợp lệ** → Đúng với logic actions(state).

Quân của người chơi 1 (giá trị 1) được thả đúng **ở hàng cuối cùng (row 5)** của **cột 3**.  
→ Đúng với logic “rơi xuống dưới cùng”.

Trò chơi chưa kết thúc vì:

* Không ai có 4 quân liên tiếp.
* Bàn cờ chưa đầy → terminal\_test() chạy đúng.

Không ai thắng, không ai thua → điểm số trung lập (0) → utility() hoạt động đúng.

**How big is the state space? Give an estimate and explain it.**

Your answer goes here.

Ước lượng kích thước không gian trạng thái (State space) của trò chơi Connect 4

**1. Giới hạn trên (ước lượng thô ban đầu)**

Bàn cờ Connect 4 có 6 hàng × 7 cột = 42 ô.

Mỗi ô có thể ở 3 trạng thái:

Trống (0),

Quân của người chơi 1 (1),

Quân của người chơi 2 (-1).

Nếu không xét quy tắc nào, tổng số trạng thái có thể là:

3^42 = 109,418,989,131,512,359,209 ≈ 1.09 x 10^20

Đây là giới hạn trên lý thuyết (naïve upper bound), vì nó tính cả những bàn cờ không hợp lệ (vi phạm luật trọng lực hoặc thứ tự lượt chơi).

**2. Lý do không gian thật nhỏ hơn rất nhiều**

Trong thực tế, phần lớn các bàn cờ trong không thể xảy ra trong trò chơi Connect 4 thật, vì:

Luật trọng lực (gravity):

Quân cờ luôn “rơi” xuống ô trống thấp nhất của cột, không thể có quân lơ lửng.

Luân phiên lượt đi:

Số quân của 2 người chơi chỉ có thể chênh lệch nhiều nhất là 1.

Luật kết thúc:

Nếu một bên đã thắng (4 quân liên tiếp), trò chơi sẽ dừng lại ngay — không thể có trạng thái nào có nhiều hơn 4 quân liên tiếp được đặt thêm sau đó.

Những ràng buộc này loại bỏ hầu hết các cấu hình không hợp lệ, khiến không gian trạng thái thực tế nhỏ hơn rất nhiều.

**3.Kết quả chính xác (đã được tính toán)**

Theo các nghiên cứu đã công bố (và được dùng trong các dự án giải Connect 4 chuyên dụng),

tổng số trạng thái hợp lệ (legal positions) trong Connect 4 là: 4.53 x 10^12

Tức là khoảng 4.5 nghìn tỷ trạng thái hợp lệ.

So sánh: 3^42 / (4.53 x 10^12) ≈ 2.4 x 10^7

Giới hạn lý thuyết lớn gấp khoảng 24 triệu lần so với không gian thật.

**4. Ý nghĩa đối với tìm kiếm (search)**

Mặc dù nhỏ hơn nhiều so với 10^20 nhưng 4.5 nghìn tỷ trạng thái vẫn là một con số khổng lồ —việc duyệt toàn bộ không gian này là bất khả thi với Minimax thuần túy

Vì vậy, các thuật toán chơi Connect 4 thực tế phải dùng:

    Alpha–Beta pruning để cắt nhánh không cần thiết,

    Heuristic evaluation để ước lượng giá trị trạng thái trung gian,

    Bảng nhớ (transposition table) và chiến lược chọn nước đi thông minh

**How big is the game tree that minimax search will go through? Give an estimate and explain it.**

**Ý tưởng chính**

Kích thước game tree mà Minimax phải duyệt phụ thuộc vào:

  b = branching factor (số nước đi trung bình trên mỗi bước), và

  d = độ sâu (số plies/moves tối đa mà ta xem xét).

Một ước lượng thô thường là nodes ≈ b^d. Với Connect-4:

tối đa mỗi bước có 7 lựa chọn (7 cột), nhưng về trung bình số cột khả dụng nhỏ hơn (một số cột dần đầy). Ta có thể dùng b ≈ 5–6 làm ước lượng hợp lý.

một ván đầy có tối đa 42 nước (6×7), tức d ≤ 42

Tính nhanh

**1) Nếu Minimax duyệt toàn bộ đến cuối ván (d = 42)**

Giả sử b=5.5 (một giá trị trung bình giữa 5 và 6),

Số nút ≈ 5.5^42 ≈ 1.25 x 10^31

Kết luận: không thể thực hiện (con số khổng lồ, vượt xa khả năng máy tính thực tế).

**2) Nếu Minimax chỉ duyệt tới các độ sâu thực tế thường dùng (ví dụ d = 8, 10, 12, 16)**

**(giả sử vẫnb=5.5):**

    d = 6 → ≈ 2.8 x 10^4 nut

    d = 8 → ≈ 8.4 x 10^5 nut

    d = 10 → ≈ 2.5 x 10^7 nut

    d=12 → ≈ 7.7 x 10^8 nut

    d= 16 → ≈ 7 x 10^11 nut

Những con số này cho thấy: duyệt đến 8–12 plies là khả thi trên máy thường, còn sâu hơn (14–16+) đã bắt đầu rất nặng.

**3) Tác dụng của Alpha–Beta pruning (với ordering tốt)**

Với alpha–beta và thứ tự đánh giá nước đi tốt, hiệu quả thường xấp xỉ giảm branching về b^1/2

Nếu hiệu ứng này đạt được, thì “hiệu dụng” b ≈ sqrt(5.5) ≈ 2.345

Với d = 42, số nút xấp xỉ 2.345^42 ≈ 3.5 x 10^15 — vẫn rất lớn, nhưng nhỏ hơn b^d ban đầu hàng chục mũ.

Tức là alpha–beta có thể giảm đáng kể kích thước tree nhưng vẫn không biến việc duyệt toàn bộ đến độ sâu 42 thành khả thi; cần thêm transposition tables, symmetry reductions, heuristics, v.v.

So sánh với không gian trạng thái

Số vị trí hợp lệ (legal positions) của Connect-4 4.531985219092 × 10^12 (khoảng 4.53 nghìn tỷ)

Tuy nhiên số cây trò chơi (số chuỗi nước đi khác nhau, tức số lá nếu duyệt tới các kết thúc khác nhau) thường lớn hơn số trạng thái hợp lệ vì nhiều chuỗi khác nhau có thể dẫn tới cùng một trạng thái (những trạng thái lặp lại được lưu trong bảng băm sẽ giúp cắt bớt công việc).

**Kết luận ngắn gọn**

Game tree của Minimax đầy đủ (duyệt tới tất cả các plies hợp lệ) là khổng lồ — nếu tính theo b^d với b ≈ 5.5 và d <= 42 ta có 1.25 x 10^31 nút (ước lượng thô).

Thực tế người ta không duyệt toàn bộ: thường giới hạn độ sâu (8–12 plies), dùng alpha–beta pruning, transposition tables, move ordering, và heuristics. Những tối ưu này làm Minimax khả thi trong thực tế và thậm chí cho phép giải (strong-solve) Connect-4 bằng các phương pháp chuyên biệt. (Số trạng thái hợp lệ của trò chơi≈ 4.53 x 10^12.

**Task 2: Game Environment and Random Agent [25 point]**

Mục tiêu:

Tạo **môi trường trò chơi** (game environment) cho Connect 4.

Dùng **numpy character array** làm bàn cờ (' ' cho trống, 'R' cho Red, 'Y' cho Yellow).

Cài đặt các hàm trợ giúp (helper functions):

* actions(state) – các nước đi hợp lệ
* result(state, player, action) – kết quả sau khi đi
* terminal(state) – kiểm tra trò chơi kết thúc chưa
* utility(state, player) – điểm cho người chơi

Viết **random agent** (người chơi chọn nước đi ngẫu nhiên).

Code:

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program code

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

A computer screen shot of a program code

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Giải thích

| **Hàm** | **Chức năng** |
| --- | --- |
| initial\_state(rows, cols) | Tạo bàn cờ trống (kích thước tùy ý) |
| actions(state) | Liệt kê các cột có thể đi hợp lệ |
| result(state, player, action) | Sinh trạng thái mới sau khi người chơi đi |
| terminal(state) | Kiểm tra bàn cờ kết thúc chưa (thắng hoặc hòa) |
| check\_winner(state) | Kiểm tra xem ai thắng |
| utility(state, player) | Tính điểm cho người chơi |
| random\_agent\_action(state) | Chọn nước đi ngẫu nhiên (random agent) |