**Họ và tên: Phạm Hàn Minh Chương**

**MSSV: 23110187**

**Câu 1: Cho 2 reward systems cho agent tìm đường thoát khỏi mê cung như bên dưới. Reward system nào tốt hơn? Tại sao?**

System 1: +10 khi thoát khỏi mê cung, 0 khi di chuyển 1 bước trong mê cung

System 2: 0 khi thoát khỏi mê cung, -10 khi di chuyển 1 bước trong mê cung

\* Giải:

System 1: +10 khi thoát khỏi mê cung, 0 khi di chuyển 1 bước trong mê cung

Phân tích:

* Đây là hệ thống thưởng "sparse reward" - chỉ có một phần thưởng duy nhất ở cuối quá trình
* Agent chỉ nhận được phần thưởng dương (+10) khi hoàn thành mục tiêu cuối cùng
* Không có hình phạt hay phần thưởng cho các hành động trung gian
* Không tạo động lực để tìm đường đi ngắn nhất, chỉ cần tìm được đường ra
* Agent có thể đi lòng vòng vô hạn mà không có bất kỳ hình phạt nào
* Đòi hỏi agent phải tự khám phá trong không gian trạng thái rộng lớn để tìm ra đường đi
* Hiệu quả trong việc dạy agent "tìm ra lối thoát bằng mọi giá" nhưng không hiệu quả trong việc dạy "tìm ra lối thoát nhanh nhất"

System 2: 0 khi thoát khỏi mê cung, -10 khi di chuyển 1 bước trong mê cung

Phân tích:

* Đây là hệ thống dựa trên hình phạt (penalty-based)
* Không có phần thưởng rõ ràng cho việc hoàn thành mục tiêu
* Mỗi bước di chuyển đều bị phạt (-10), tạo áp lực để agent giảm thiểu số bước di chuyển
* Tạo động lực mạnh mẽ để tìm đường đi ngắn nhất
* Thúc đẩy agent tránh các vòng lặp vô hạn hoặc đường đi dài không cần thiết
* Tạo cảm giác "khẩn cấp" cho agent để tìm lối ra nhanh chóng
* Có thể tạo ra các hành vi tiêu cực nếu agent bị mắc kẹt và không tìm được lối thoát (tích lũy hình phạt)

Kết luận:

System 2 tốt hơn cho bài toán tìm đường thoát khỏi mê cung vì nó thúc đẩy agent tìm đường đi ngắn nhất và hiệu quả nhất. System 2 tạo động lực để agent giảm thiểu số bước di chuyển, tránh vòng lặp vô hạn và tìm đường đi tối ưu. Trong khi đó, System 1 chỉ quan tâm đến việc đạt được mục tiêu cuối cùng mà không quan tâm đến hiệu quả của quá trình.

Mặc dù vậy, System 2 cũng có nhược điểm khi agent có thể gặp khó khăn trong môi trường phức tạp hoặc khi cần phải hy sinh lợi ích ngắn hạn để đạt được mục tiêu dài hạn. Tuy nhiên, với đặc thù của bài toán tìm đường trong mê cung, sự tối ưu về số bước đi chính là yếu tố quan trọng nhất, do đó System 2 vẫn là sự lựa chọn tốt hơn

**2. Thiết kế reward system cho agent tìm đường thoát khỏi mê cung mà bạn thấy tốt hơn 2 reward systems trong bài tập trên. Giải thích tại sao nó tốt hơn.**

\*Giải:

Reward System Đề Xuất

* +100 khi thoát khỏi mê cung
* -1 cho mỗi bước di chuyển
* -5 khi đi vào ô đã đi qua
* +5 khi khám phá ô mới
* +10 khi tiến gần lối ra hơn

Tại sao tốt hơn

* Kết hợp ưu điểm: Giữ phần thưởng lớn khi hoàn thành (+100) và hình phạt nhẹ cho mỗi bước đi (-1).
* Tránh lặp lại: Phạt nặng hơn (-5) khi đi vào ô đã đi qua, ngăn hành vi đi vòng tròn.
* Khuyến khích khám phá: Thưởng (+5) khi tìm thấy ô mới giúp mở rộng vùng khám phá.
* Reward shaping: Thưởng (+10) khi tiến gần đích tạo gradient dẫn hướng.
* Feedback liên tục: Khắc phục vấn đề "sparse reward" của System 1 và áp lực quá mức của System 2.
* Cân bằng: Tạo sự cân bằng tốt giữa khám phá (exploration) và khai thác (exploitation).

**Câu 3: Cho ít nhất 1 ví dụ cho mỗi loại environment sau (yêu cầu phải khác ví dụ trong bài giảng). Giải thích ngắn gọn cho mỗi ví dụ tại sao nó thuộc về loại environment đó.**

\* Giải:

\* Stochastic Environment

Ví dụ: Đầu tư chứng khoán

Giải thích: Mặc dù bạn có thể phân tích thị trường và đưa ra quyết định đầu tư, nhưng kết quả (giá cổ phiếu) chịu ảnh hưởng của nhiều yếu tố ngẫu nhiên và không thể dự đoán chính xác. Cùng một hành động (mua cổ phiếu) có thể dẫn đến nhiều kết quả khác nhau (lợi nhuận hoặc lỗ với mức độ khác nhau).

\* Episodic

Ví dụ:Game Trả lời câu hỏi (Quiz Show)

Giải thích:

* Mỗi câu hỏi là một tập riêng biệt, không ảnh hưởng đến các câu hỏi khác.
* Kết quả của một câu hỏi không thay đổi câu hỏi tiếp theo.

\* Fully Observable

ví dụ: Game Sudoku

Giải thích:

* Người chơi có thể nhìn thấy toàn bộ bảng Sudoku ngay từ đầu.
* Không có thông tin nào bị ẩn đi, tất cả số đã có đều hiển thị rõ ràng.

**Câu 4: Mô tả 5 thành phần của bài toán: tìm đường đi từ 1 tỉnh thành tới 1 tỉnh thành khác cho các tỉnh thành trong hình ở slide kế tiếp**

- Tác nhân (Agent): Đây là đối tượng thực hiện quá trình di chuyển từ một tỉnh này sang một tỉnh khác. Tác nhân có thể là một người lái xe đang tìm đường hoặc một thuật toán định tuyến tự động giúp xác định lộ trình tối ưu.

- Môi trường (Environment): Môi trường trong bài toán này chính là hệ thống đường sá kết nối giữa các tỉnh. Các tỉnh thành được xem như các điểm trên bản đồ (nodes), trong khi các tuyến đường nối giữa chúng được biểu diễn bằng các cạnh (edges) của đồ thị.

- Hành động (Actions): Tác nhân có thể lựa chọn đi theo các tuyến đường trực tiếp nối giữa các tỉnh. Chẳng hạn, nếu đang ở TP HCM, tác nhân có thể di chuyển đến Long An hoặc Vũng Tàu, miễn là có đường kết nối giữa các vị trí này.

- Trạng thái (States): Trạng thái của hệ thống được xác định bởi vị trí hiện tại của tác nhân trong hành trình. Khi tác nhân di chuyển từ tỉnh này sang tỉnh khác, trạng thái sẽ thay đổi tương ứng. Ví dụ, nếu tác nhân đang ở Tây Ninh, trạng thái hiện tại sẽ là "Tây Ninh".

- Phần thưởng (Reward): Đây là giá trị mà tác nhân nhận được dựa trên các tiêu chí tối ưu như quãng đường ngắn nhất, thời gian di chuyển ít nhất hoặc chi phí thấp nhất. Khi tác nhân hoàn thành hành trình thành công, có thể nhận được điểm thưởng cao, trong khi lựa chọn tuyến đường xa hơn hoặc gặp tắc nghẽn có thể làm giảm phần thưởng.