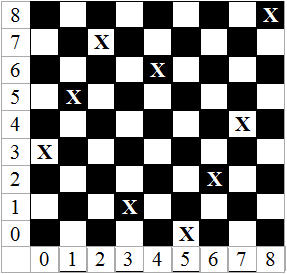
**Ứng dụng Hill Climbing – Bài toán 8 Quân hậu**

# Giới thiệu

Bài toán tám quân hậu là bài toán đặt tám quân hậu trên bàn cờ vua kích thước 8×8 sao cho không có quân hậu nào có thể “ăn” được quân hậu khác. Như vậy, lời giải của bài toán là một cách xếp tám quân hậu trên bàn cờ sao cho không có hai quân nào đứng trên cùng hàng, hoặc cùng cột hoặc cùng đường chéo. Bài toán tám quân hậu có thể tổng quát hóa thành bài toán đặt n quân hậu trên bàn cờ n×n(n ≥ 4).

# Bài toán

Cho một bàn cờ kích thước n x n. Chúng ta cần đặt n (n>=4) quân hậu vào bàn cờ sao cho chúng không tấn công được nhau, tức là không có cặp con hậu nào nằm cùng hàng, cùng cột, cùng đường chéo.



# Phân tích và thiết kế thuật toán

Để giải quyết bài toán, ta cần lưu trữ, sắp xếp các quân hậu sao cho:

* Không có 2 quân hậu nào nằm trên một hàng
* Không có 2 quân hậu nằm trên một cột
* Không có 2 quân hậu nào nằm trên một đường chéo

Thông thường, chúng ta sẽ lưu bàn cờ dưới dạng mảng 2 chiều. Nhưng trong bài toán N-Queen, cách lưu trữ này tỏ ra thiếu hiệu quả và tốn dữ liệu. Một các lưu trữ khác như sau: Tạo một mảng một chiều gồm N phần tử. Mỗi phần tử sẽ đại diện cho một quân hậu. Trong đó, chỉ số của mỗi phần tử được coi là chỉ số cột của quân hậu, giá trị của phần tử được coi là chỉ số hàng của quân hậu.

Ví dụ ta có mảng

Queen = [1,2,6,5,4,5,3,0]

Queen[0] = 1 tức là trên bàn cờ sẽ có 1 quân hậu ở vị trí hàng 1 cột 0

Với cách lưu trữ này, ta có thể giảm ô số cần lưu trữ và loại đi rất nhiều trường hợp xếp hậu sai vị trí (chỉ số không trùng nên chỉ số cột sẽ không trùng).

Việc kiểm tra va chạm của các quân hậu như sau

* Nếu hai quân hậu X1 X2 cùng hàng :

X1.getRow() = X2.getRow()

* Nếu hai quân hậu X1 X2 cùng cột :

X1.getColumn() = X2.getColumn()

* Nếu hai quân hậu X1 X2 cùng đường chéo:

abs(X1.getRow() – X2.getRow()) = abs(X1.getColumn() – X2.getColumn())

Ý tưởng của hill-climbing là từ trạng thái ban đầu về trạng thái đích bằng thực hiện các nước đi hợp hệ rồi chọn trạng thái tiếp theo tốt nhất nhờ một hàm chi phí.

Với trạng thái ban đầu là trạng thái ngẫu nhiên. Trạng thái đích là trạng thái mà các quân hậu không thể tấn công nhau. Từ trạng thái ban đầu, ta thực hiện thay đổi vị trí quân hậu để tạo ra trạng thái tiếp theo. Hàm chi phí tính bằng số quân hậu có thể bị tấn công. Ta sẽ chọn trạng thái nào có giá trị hàm nhỏ nhất. Và để cải thiện hàm chi phí, chúng ta sẽ chọn quân hậu nào có khả năng bị tấn công nhiều nhất dịch chuyển tạo ra trạng thái mới.

B1: Khởi tạo bàn cờ ngẫu nhiên các vị trí của quân hậu. Ta gọi đây là trạng thái ban đầu.

B2: Chọn quân cờ có khả năng bị tấn công nhiều nhất. Thực hiện dịch chuyển các nước đi phù hợp để tạo ra các trạng thái con.

B3: Từ các trạng thái con này chọn ra trạng thái có hàm chi phí nhỏ nhất. Chọn nó là thái đầu tiên và lưu chúng vào danh sách trạng thái được chọn.

B4 Lặp lại B2 cho đến khi không còn quân cờ nào trong tình trạng bị tấn công.

Do giải thuật hilllclimbing chưa chắc đã có thể tìm ra trạng thái đích, nên nếu không thể tìm thấy trạng thái đích ta khởi tạo 1 giá trị random mới cho vị trí các quân hậu và tiếp tục tìm kiếm.