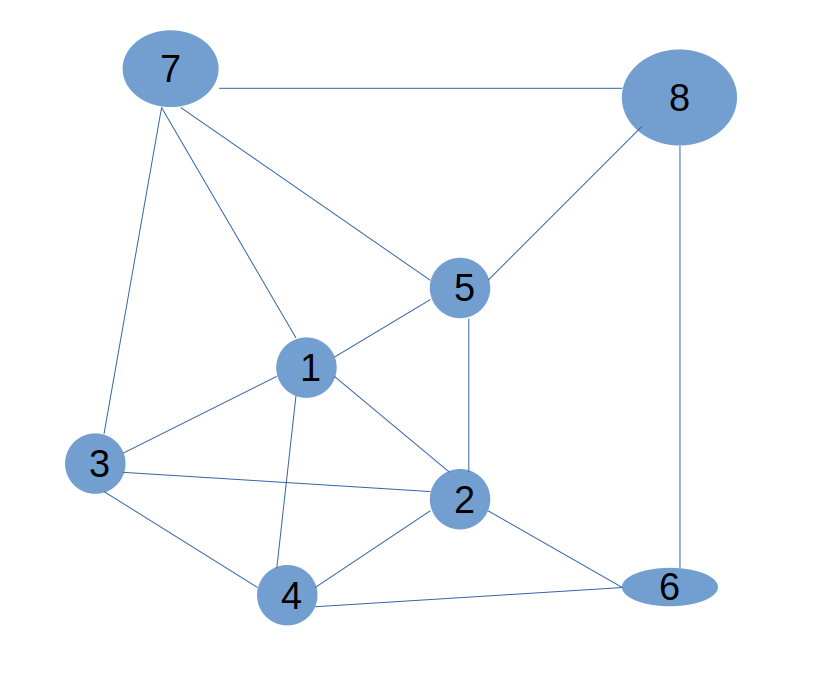
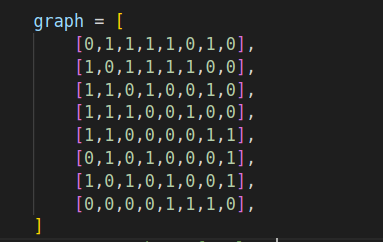
# Tô Màu Bản Đồ ( các điểm liền kề nhau không được trùng màu ) bằng thuật toán Quay lui

Quy ước:

* chúng ta sẽ sử dụng bản đồ sau để tiến hành tô màu
* mỗi số trên bản đồ là một vùng khác nhau



* Chúng ta tiến hành chuyển đổi bản đồ trên sang ma trận tương ứng



* chúng ta sẽ cho 8 màu và sẽ quy ước như sau

{

1 - đỏ

2 - xanh dương

3- vàng

4 - tím

5 - xanh lá

6 - đen

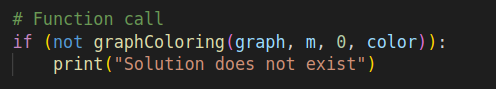
7 - hồng

8 - cam

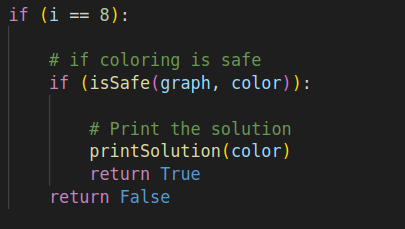
}



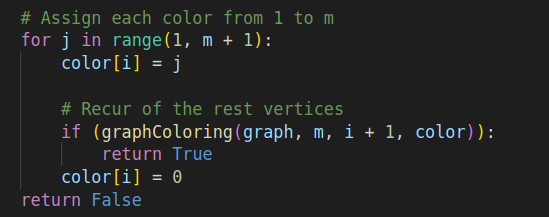
* tiến hành gọi hàm graphColoring đây là một hàm trả về true hoặc false và đây cũng chính là hàm gọi lại chính nó ( đệ quy ) để thực hiện bài toán quay lui. các thông số được truyền vào bao gồm bản đồ,số màu,vị trí hiện tại,mảng màu



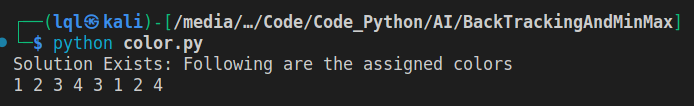
* nếu vị trí hiện tại của i = 8 ( đã duyệt hết 8 vị trí ) thì sẽ tiến hành kiểm tra xem các màu đánh số có thỏa mãn hay không bằng cách gọi ra hàm isSafe, nếu isSafe thỏa mãn thì sẽ tiến hành in ra giải pháp các màu, nếu không sẽ trả về false và ghi ra dòng chữ “solution does not exits"



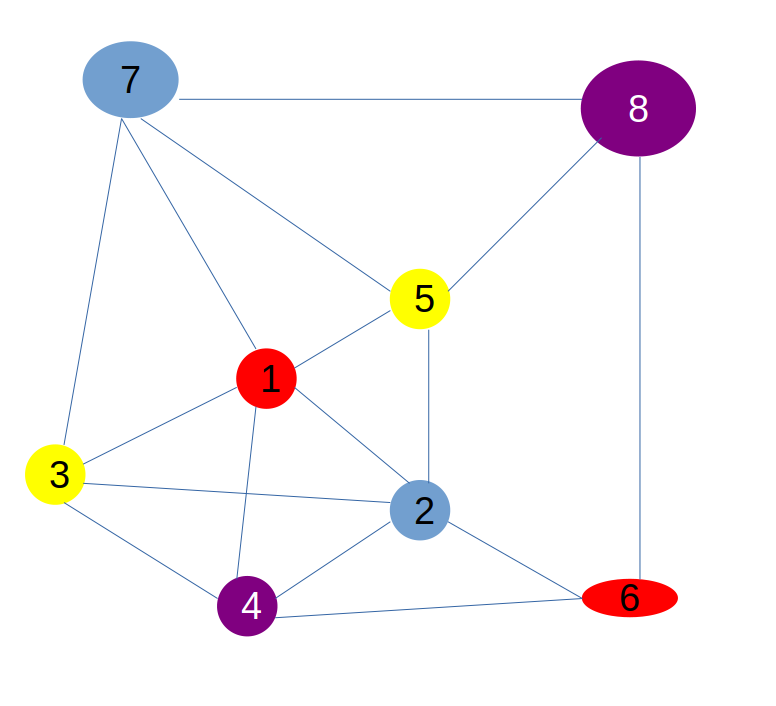
* tiến hành duyệt các màu trong mảng bắt đầu từ 1 và tiến hành gán màu cho vị trí hiện tại và tiến hành gọi lại chính nó với vị trí tiếp theo (i+1) cho đến khi i = 8 sẽ kết thúc



* kết quả



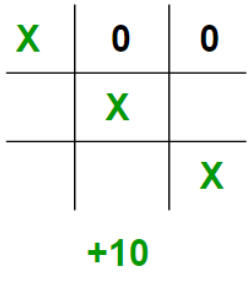
* và theo quy ước
* 1 - đỏ
* 2 - xanh dương
* 3- vàng
* 4 - tím
* ta tiến hành tô màu vào bản đồ lúc đầu và thỏa mãn với bài toán tô màu bản đồ



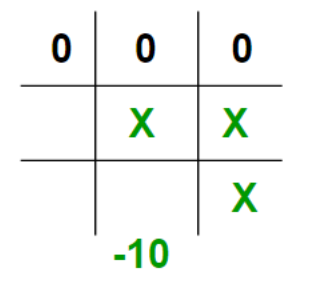
# *Tic tac toe* bằng min max

* Quy ước x là player

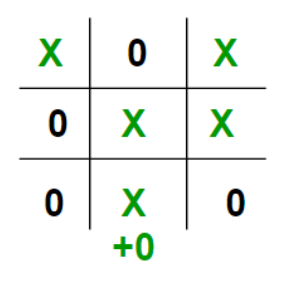
nếu x thắng thì evaluate là +10



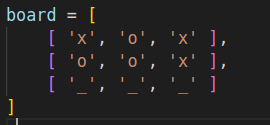
nếu o thắng thì evaluate là -10



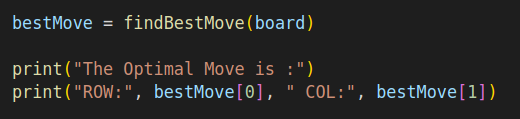
nếu không có bên nào là +0



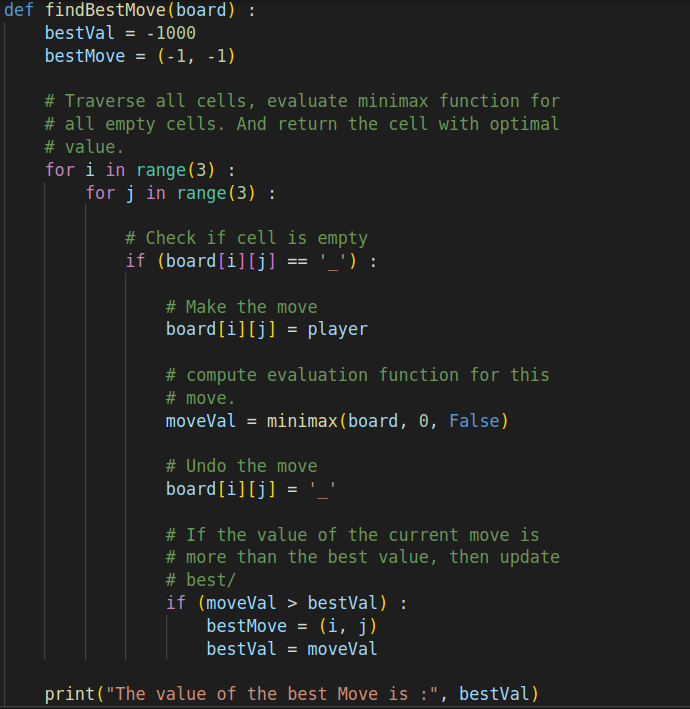
* Ta sẽ dùng bảng tic tac toe sau để tính ra nước cờ tối ưu nhất



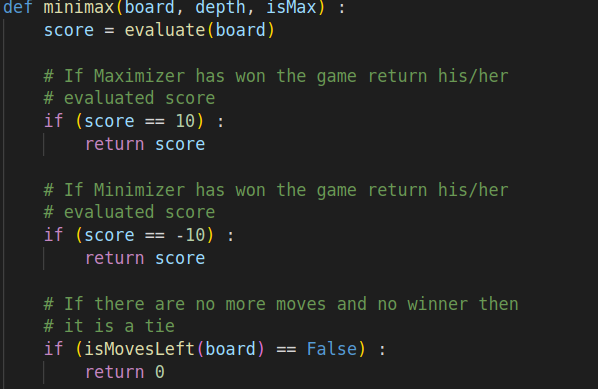
* tiến hành gọi hàm “findBestMove” để tìm ra bước tốt nhất



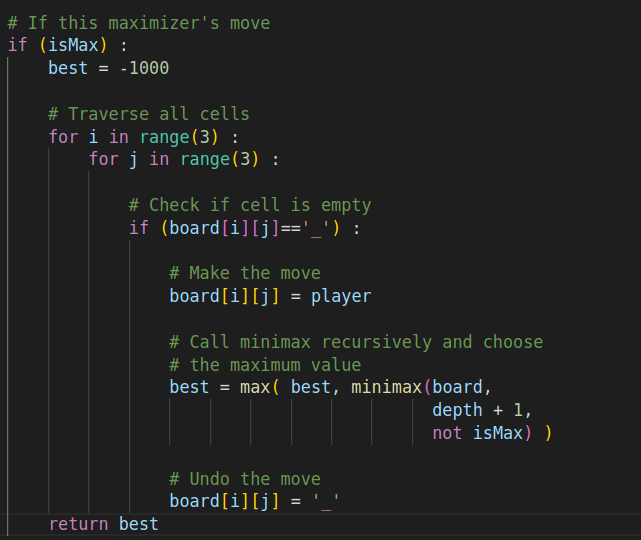
* trong hàm “findBestMove” sẽ tiếm hành dùng hai vòng lặp lòng nhau để tìm ra vị trí đang trống và gán nó cho player sau đó tiến hành gọi tới hàm minmax nhằm để trả về giá trị tốt nhất và nếu giá trị tốt nhất > giá trị tốt nhất được đặt ở đầu là -1000 thì sẽ trả về vị trí i,j ( dòng và cột tương ứng )



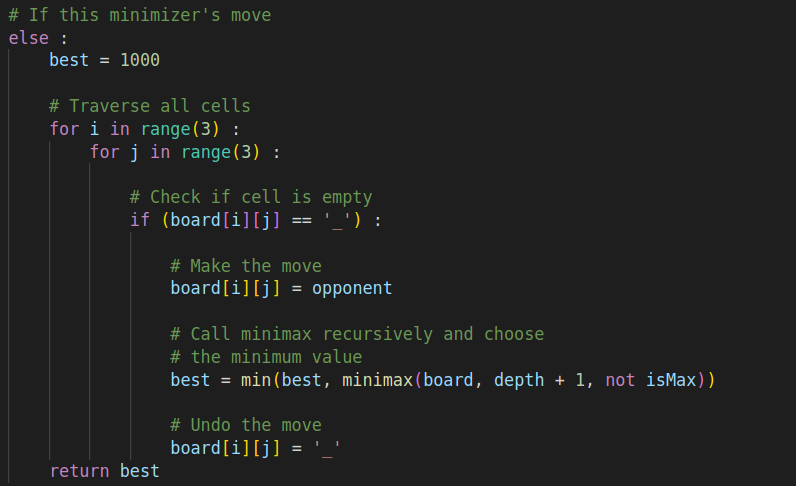
* trong hàm minmax sẽ tiến hành gọi hàm evaluate để tính ra số điểm tương ứng của bước đi nếu bước đi là -10,10 thì sẽ trả về số điểm đó, và nếu như không còn các ô trống thì sẽ trả về false (hàm isMovesLeft) và return với kết quả là 0



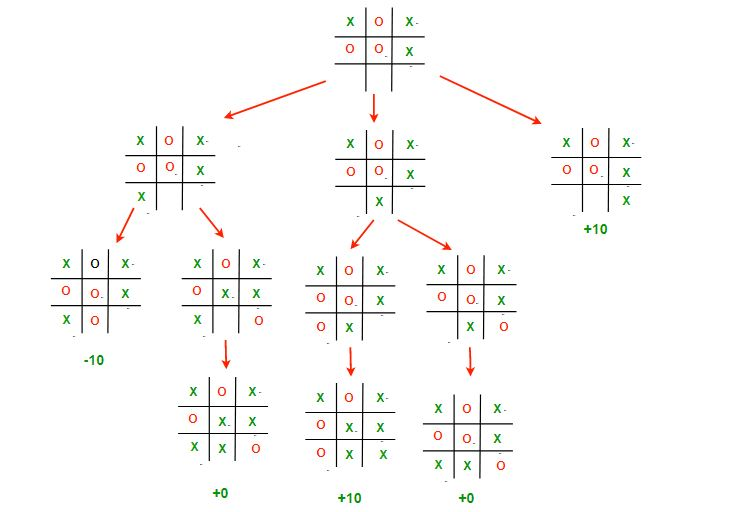
* nếu như đây là bước chọn tối ưu nhất thì sẽ tiến hành kiểm trả xem có còn ô nào trống nữa hay không nếu có thì sẽ tiến hành lấy max của kết quả tốt nhất và minmax ( đệ quy chính nó ) nhằm kiểm tra phải là tốt nhất không và trả về best



* nếu khôn phải max thì cũng tương tự như là max nhưng ở đây chúng ta sẽ tính min và kết quả trả về là min



* kết quả muốn có, chúng ta thấy rằng nếu ta đánh x tại vị trí cột 2 dòng 2 ( bắt đầ từ 0) thì x sẽ chắc chắn thắng ngoài ra lưu ý rằng nếu đánh ở cột 1 dòng 2 thì vẫn có thể thắng nhưng do có thể tính toán được là có thể ra +0 nên sẽ chọn vị trí x (2,2) là tốt nhất



* kết quả thực tế như vậy thuật toán đã tính đúng

