6. Compare the effect of increasing search depth (come up with a method to demonstrate your point).

To compare the effect of increasing search depth, I copied the original alpha-beta search and modified its depth to 8. Then I made 2 players (player1, player2). Therefore, player 1 had alpha-beta with a depth of 4, whereas player 2 has a depth of 8.

```
#Main function to run
def main():
    game = Othello()


    '''
    Comment out to choose different style of players
    '''

    '''
    Player 1
    '''
    #Choose player 1 method
    #player1 = human_player              #Played by the uswer
    player1 = ai_player                  #Played by the alpha-beta original



    '''
    Player 2
    '''
    #Choose player 2 method
    #player2 = ai_player                  #Played by original alpha-beta by cpu
    #player2 = human_player               #Played by the user
    player2 = better_ai_player           #Played by the better alpha-beta by cpu. I
    #player2 = weight_ai_player             #Played by the 2nd eval function (weight
    #player2 = movement_ai_player          #Played by the 3rd eval function (number
```

Player1 alpha-beta

```
# Alpha-beta cut off from AIMA
def alpha_beta_cutoff_search(state, game, d=4, cutoff_test=None, eval_fn=None):
```

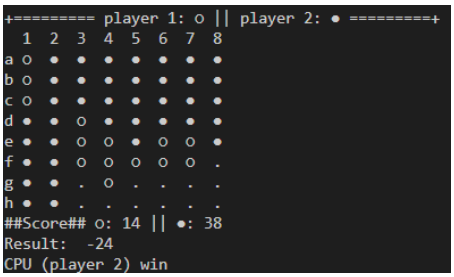
Player2 alpha-beta

```
#Modified Aima alpha-beta cut off with deeper depth.
def alpha_beta_cutoff_search_ver(state, game, d=8, cutoff_test=None, eval_fn=None):
```

After running 10 times, each time player2 (with higher depth) always won the game. However, it took longer to take action.

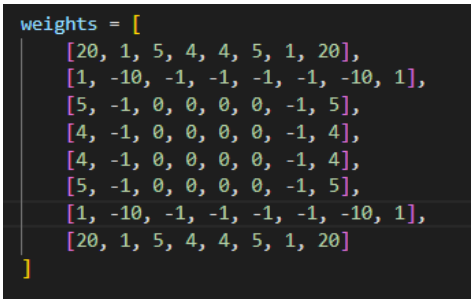
Additionally, I have also increased the depth for the player 2, and as depth increased, with showed stronger gameplayer, however running time increase a lot.

Result for agent vs agent (1 vs 4 / 4 vs 8 / 1 vs 6) agent with the higher depth always won the game.



7. Implement at least two evaluation functions with varying quality. Compare the effect of improving the evaluation function.

One of the evaluation functions was to set the weight on the board. By looking at the tactics of Othello, taking the corner increases the win rate a lot. Therefore, I have created a board with a weight on it.



```
weights = [
    [20, 1, 5, 4, 4, 5, 1, 20],
    [1, -10, -1, -1, -1, -1, -10, 1],
    [5, -1, 0, 0, 0, 0, -1, 5],
    [4, -1, 0, 0, 0, 0, -1, 4],
    [4, -1, 0, 0, 0, 0, -1, 4],
    [5, -1, 0, 0, 0, 0, -1, 5],
    [1, -10, -1, -1, -1, -1, -10, 1],
    [20, 1, 5, 4, 4, 5, 1, 20]
]
```

Taking the corner is the best, however, taking the edge is also effective. It is also important to not make a movement that takes spots that are beside the edge. Because it leads opponents to take the edge or the corner.

Order to use this evaluation function, it was important to increase the depth as well. When this evaluation function had a low depth, it performed low win rate than the original function. That is because, with the low depth, it cannot include the cornder or the edges as the part of the calculation.

The second evaluation function is by taking leftover movement. In Othello, it is possible to make the opponent skip its turn if opponent cannot make any more movement. If opponent cannot make turn, the player can make consecutive movement, and that increases the win rate significantly.