Lab - 11

→ Solve 8-queen's problem using Alpha-beta pruning algorithm

Algorithm:

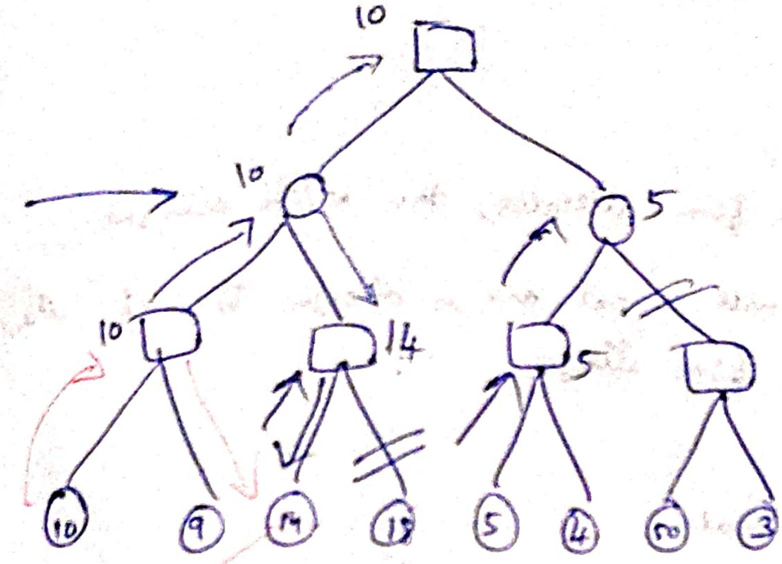
```
def alpha_beta(self, board, col, alpha, beta, maximizing_player):
    if col >= self.size:
        return 0, [row[:] for row in board]

    if maximizing_player:                    //maximizer
        max_eval = float(-∞)
        best_board = None
        for row in range(self.size):
            if self.is_safe(board, row, col):
                board[row][col] = 1
                eval_score, pot_board = self.alpha_beta_search(board,
                    col + 1, alpha, beta, False)
                board[row][col] = 0
                if eval_score > max_eval:
                    max_eval = eval_score
                    best_board = pot_board
                alpha = max(alpha, eval_score)
                if beta <= alpha:
                    break
        return max_eval, best_board

    else:                                    //minimizer
        beta = min(beta, eval_score)
        if beta <= alpha:
            break
        return min_eval, best_board
```

Output:

```
. Q . . . . . .
. . . . Q . . .
. . . . . . Q .
Q . . . . . . .
. . Q . . . . .
. . . . . . . Q
. . . . . Q . .
. . . Q . . . .
```



α (max)

β (min)

α (max)

3/12/24