

⇒ 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]

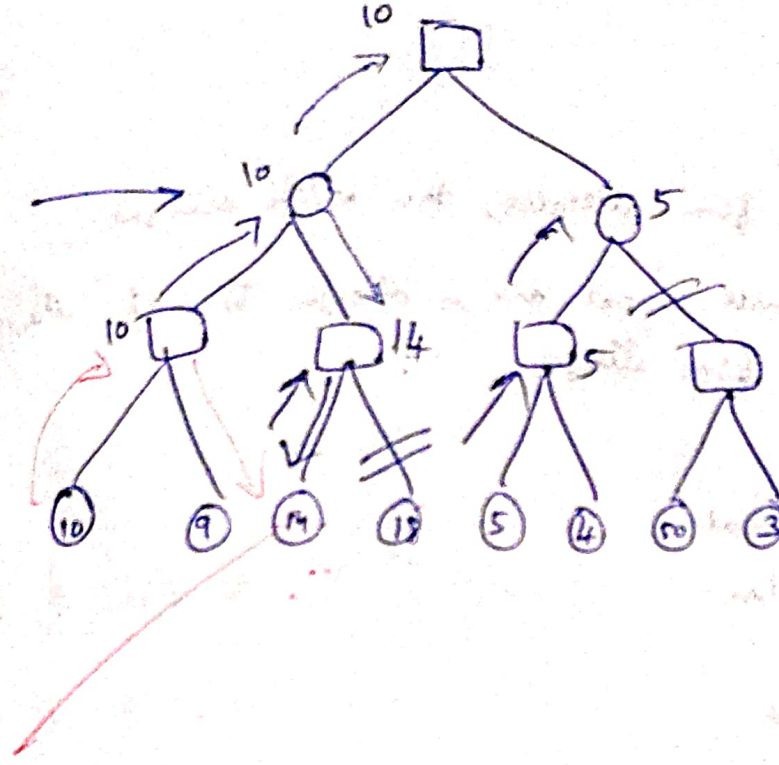
    if maximizing_player:
        // maximizer
        max_eval = float(-∞)
        best_board = None
        for row in range(self.size):
            if self.is_size(board, row, col):
                board[row][col] = 1
                eval_score, pot_board = self.alpha_beta_search(board, col+1, alpha, 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 . . . . .
. . . . a . .
. . . . . a .
a . . . . . .
. . a . . . .
. . . . . a .
. . . a . . .
. . . . . . Q
```



$L(\max)$

$\beta(\min)$

$L(\max)$

Ans
3/12/24

Given array: $\{10, 10, 10, 10, 10, 10, 10, 10, 10, 10\}$

Let $f(i)$ be the maximum value of $L(\max)$ for the array $\{a_1, a_2, \dots, a_i\}$.

Then, $f(i) = \max\{f(i-1), L(\max)\}$.

where $L(\max)$ is the value of $L(\max)$ for the array $\{a_1, a_2, \dots, a_i\}$.

For the given array, the value of $f(i)$ is 10.

Therefore, the answer is 10.

Time complexity: $O(n)$.

Space complexity: $O(1)$.

Hope this helps!