

PRINCIPLES OF ARTIFICIAL INTELLIGENCE

LABORATORY PROGRAMS

INTRODUCTION TO ALPHA-BETA PRUNING PROGRAM:

Note:

Modified Version of MINIMAX Algorithm

SOURCE CODE:

```
MAX, MIN = 1000, -1000
def minimax(depth, nodeIndex, maximizingPlayer, values, alpha, beta):
    if depth == 3:
        return values[nodeIndex]

    if maximizingPlayer:
        best = MIN
        for i in range(0, 2):
            val = minimax(depth + 1, nodeIndex * 2 + i, False, values, alpha, beta)
            best = max(best, val)
            alpha = max(alpha, best)
            if beta <= alpha:
                break
        return best
    else:
        best = MAX
        for i in range(0, 2):
            val = minimax(depth + 1, nodeIndex * 2 + i, True, values, alpha, beta)
            best = min(best, val)
            beta = min(beta, best)
            if beta <= alpha:
                break
        return best

if __name__ == "__main__":
    values = [3, 5, 6, 9, 1, 2, 0, -1]
    print("The optimal value is:", minimax(0, 0, True, values, MIN, MAX))
```

OUTPUT:

```
24-04-2024 23:15:01 23:15:01 ALPHA BETA PRUNING.py - C:/Users/ur mom/Documents/PRINCIPLES OF AI/SANTHOSHKUMAR S... IDLE Shell 3.9.10
File Edit Format Run Options Window Help File Edit Shell Debug Options Window Help
MAX, MIN = 1000, -1000
def minimax(depth, nodeIndex, maximizingPlayer, values, alpha, beta):
    if depth == 3:
        return values[nodeIndex]

    if maximizingPlayer:
        best = MIN
        for i in range(0, 2):
            val = minimax(depth + 1, nodeIndex * 2 + i, False, values, alpha, beta)
            best = max(best, val)
            alpha = max(alpha, best)
            if beta <= alpha:
                break
        return best
    else:
        best = MAX
        for i in range(0, 2):
            val = minimax(depth + 1, nodeIndex * 2 + i, True, values, alpha, beta)
            best = min(best, val)
            beta = min(beta, best)
            if beta <= alpha:
                break
        return best

if __name__ == "__main__":
    values = [3, 5, 6, 9, 1, 2, 0, -1]
    print("The optimal value is:", minimax(0, 0, True, values, MIN, MAX))

Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/ur mom/Documents/PRINCIPLES OF AI/SANTHOSHKUMAR S...
024 231501147 ALPHA BETA PRUNING.py
The optimal value is: 5
>>>
```