FAST National University of
Computer and Emerging Sciences

Artificial Intelligence Assignment 3:

Name: Muneel Haider

Roll no: 21i-0640

Section: G

QUESTION 1 CODE OUTPUT:

```
180
181   board = [
182       ['x', 'o', 'x'],
183       ['o', 'o', 'x'],
184       ['_', '_', '_']
185   ]
186
187   bestMove = findBestMove(board)
188   print("\nThe Optimal Move is :")
```

PROBLEMS DEBUG CONSOLE TERMINAL POSTMAN CONSOLE

```
PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3> python -u "d:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3\i210640_A3_Q1.py"

Best possible move found.

Board:
    0   1   2
0   x  | o  | x
   -----------
1   o  | o  | x
   -----------
2   _  | _  | $

The Optimal Move is :
ROW: 2  COL: 2
```

QUESTION 2 CODE OUTPUT:

- ## PART 1 CODE OUTPUT:
  - ### CASE 1:

```
89    board = [
90        ['x', '_', 'x'],
91        ['_', 'o', '_'],
92        ['_', '_', '_']
93    ]
94
95    result = compare_minimax(board)
96    print("\nNode counts comparison: \n", result, "\n")
97
```

PROBLEMS DEBUG CONSOLE TERMINAL POSTMAN CONSOLE

```
PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3> python -u "d:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3\i210640_A3_Q2_P1.py"

Node counts comparison:
 {'Without Pruning': [1152, 1152, 1152], 'With Pruning': [404, 404, 404]}

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3>
```

- ### Case 2:

```
88
89  board = [
90      ['x', '_', '_'],
91      ['_', 'o', '_'],
92      ['_', '_', '_']
93  ]
94
95  result = compare_minimax(board)
96  print("\nNode counts comparison: \n", result, "\n")
97
```

```
PROBLEMS  DEBUG CONSOLE  TERMINAL  POSTMAN CONSOLE

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3> python -u "d:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3\i210640_A3_Q2_P1.py"

Node counts comparison:
 {'Without Pruning': [11056, 11056, 11056], 'With Pruning': [2170, 2170, 2170]}

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3>
```

- ### Case 3:

```
88
89  board = [
90      ['x', '_', 'x'],
91      ['_', 'o', 'x'],
92      ['_', '_', '_']
93  ]
94
95  result = compare_minimax(board)
96  print("\nNode counts comparison: \n", result, "\n")
97
```

```
PROBLEMS  DEBUG CONSOLE  TERMINAL  POSTMAN CONSOLE

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3> python -u "d:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3\i210640_A3_Q2_P1.py"

Node counts comparison:
 {'Without Pruning': [214, 214, 214], 'With Pruning': [122, 122, 122]}

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3>
```

Part 2 Code Output

- ### Case 1:

```
134  # Sequential Minimax
135  startTime = time.perf_counter()
136  bestSolution = minimax(board, 0, True, -100000, 100000)
137  endTime = time.perf_counter()
138  print(f"\nSequential Minimax Best Value: {bestSolution}, Time Taken: {endTime - startTime}")
139
140  # Parallel Minimax
141  startTime = time.perf_counter()
142  best_move, bestSolution = parallel_minimax(board, True)
143  endTime = time.perf_counter()
144  print(f"\nParallel Minimax Best Value: {bestSolution}, Time Taken: {endTime - startTime}")
145  print()
```

```
PROBLEMS  DEBUG CONSOLE  TERMINAL  POSTMAN CONSOLE

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3> python -u "d:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3\i210640_A3_Q2_P2.py"

Sequential Minimax Best Value: 10, Time Taken: 0.0003240999940317124

Parallel Minimax Best Value: 10, Time Taken: 0.006477500006440096

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3>
```

- ## CASE 2:

```
134   # Sequential Minimax
135   startTime = time.perf_counter()
136   bestSolution = minimax(board, 0, True, -100000, 100000)
137   endTime = time.perf_counter()
138   print(f"\nSequential Minimax Best Value: {bestSolution}, Time Taken: {endTime - startTime}")
139
140   # Parallel Minimax
141   startTime = time.perf_counter()
142   best_move, bestSolution = parallel_minimax(board, True)
143   endTime = time.perf_counter()
144   print(f"\nParallel Minimax Best Value: {bestSolution}, Time Taken: {endTime - startTime}")
145   print()
```

```
PROBLEMS  DEBUG CONSOLE  TERMINAL  POSTMAN CONSOLE

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3> python -u "d:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3\i210640_A3_Q2_P2.py"

Sequential Minimax Best Value: 10, Time Taken: 0.0003772999916691333

Parallel Minimax Best Value: 10, Time Taken: 0.002837000007275492

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3>
```

- ## CASE 3:

```
134   # Sequential Minimax
135   startTime = time.perf_counter()
136   bestSolution = minimax(board, 0, True, -100000, 100000)
137   endTime = time.perf_counter()
138   print(f"\nSequential Minimax Best Value: {bestSolution}, Time Taken: {endTime - startTime}")
139
140   # Parallel Minimax
141   startTime = time.perf_counter()
142   best_move, bestSolution = parallel_minimax(board, True)
143   endTime = time.perf_counter()
144   print(f"\nParallel Minimax Best Value: {bestSolution}, Time Taken: {endTime - startTime}")
145   print()
```

```
PROBLEMS  DEBUG CONSOLE  TERMINAL  POSTMAN CONSOLE

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3> python -u "d:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3\i210640_A3_Q2_P2.py"

Sequential Minimax Best Value: 10, Time Taken: 0.0003245999978389591

Parallel Minimax Best Value: 10, Time Taken: 0.005047599988756701

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3>
```

- ## PART 3 CODE OUTPUT:
  - ### CASE 1:

```
print("The Optimal Move is:", bestMove)
print("Heuristic minimax result:", nodeCount)
print("Heuristic minimax time taken:", endTime - startTime)
```

```
PROBLEMS  DEBUG CONSOLE  TERMINAL  POSTMAN CONSOLE

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3> python -u "d:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3\i210640_A3_Q2_P3.py"
The Optimal Move is: (1, 0)
Heuristic minimax result: 6
Heuristic minimax time taken: 3.9099992136470973e-05
PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3>
```

  - ### CASE 2:

```
board = [
    ['x', 'o', 'x'],
    ['_', 'x', '_'],
    ['_', '_', '_']
]

nodeCount = 0
startTime = time.perf_counter()
bestMove = findBestMove(board)
endTime = time.perf_counter()

print("The Optimal Move is:", bestMove)
```

```
PROBLEMS  DEBUG CONSOLE  TERMINAL  POSTMAN CONSOLE

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3> python -u "d:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3\i210640_A3_Q2_P3.py"
The Optimal Move is: (1, 0)
Heuristic minimax result: 5
Heuristic minimax time taken: 3.549999382812536-05
PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3>
```

  - ### CASE 3:

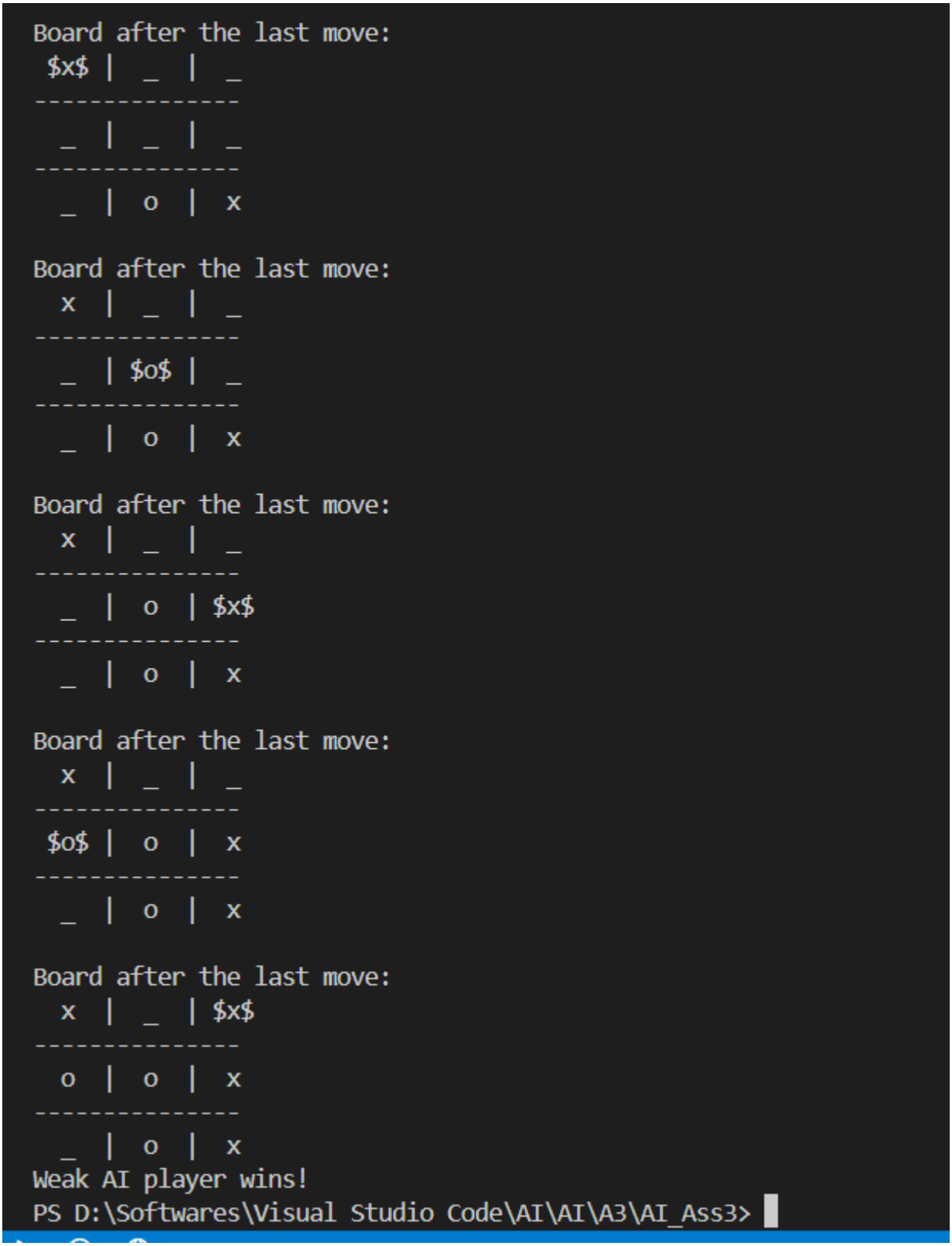
```
board = [
    ['x', 'o', 'x'],
    ['o', '_', 'x'],
    ['o', '_', '_']
]

nodeCount = 0
startTime = time.perf_counter()
bestMove = findBestMove(board)
endTime = time.perf_counter()

print("The Optimal Move is:", bestMove)
```

```
PROBLEMS  DEBUG CONSOLE  TERMINAL  POSTMAN CONSOLE

PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3> python -u "d:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3\i210640_A3_Q2_P3.py"
The Optimal Move is: (1, 1)
Heuristic minimax result: 3
Heuristic minimax time taken: 3.2199997804127634e-05
PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3>
```

- ## Part 4 Code Output:
  - ### Case 1:

```
Board after the last move:
  _  |  _  |  _
--------------
  _  |  x  |  x
--------------
 $o$ |  _  |  o

Board after the last move:
  _  |  _  | $x$
--------------
  _  |  x  |  x
--------------
  o  |  _  |  o

Board after the last move:
  _  |  _  |  x
--------------
 $o$ |  x  |  x
--------------
  o  |  _  |  o

Board after the last move:
  _  |  _  |  x
--------------
  o  |  x  |  x
--------------
  o  | $x$ |  o

Board after the last move:
 $o$ |  _  |  x
--------------
  o  |  x  |  x
--------------
  o  |  x  |  o
It's a tie!
PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3>
```

- **CASE 2:**


Board after the last move:
x | _ | _

_ | _ | _

_ | o | x
Board after the last move:
x | _ | _

_ | o | _

_ | o | x
Board after the last move:
x | _ | _

_ | o | x

_ | o | x
Board after the last move:
x | _ | _
---
o | o | x

_ | o | x
Board after the last move:
x | _ | x
---
o | o | x

_ | o | x
Weak AI player wins!
PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3>

- ## CASE 3:

```
Board after the last move:
  x  | $x$ |  _
--------------
  _  |  o  |  _
--------------
  _  |  _  |  _

Board after the last move:
  x  |  x  | $o$
--------------
  _  |  o  |  _
--------------
  _  |  _  |  _

Board after the last move:
  x  |  x  |  o
--------------
  _  |  o  | $x$
--------------
  _  |  _  |  _

Board after the last move:
  x  |  x  |  o
--------------
  _  |  o  |  x
--------------
 $o$ |  _  |  _

Board after the last move:
  x  |  x  |  o
--------------
  _  |  o  |  x
--------------
  o  | $x$ |  _
Weak AI player loses!
PS D:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3> python -u "d:\Softwares\Visual Studio Code\AI\AI\A3\AI_Ass3\i210640_A3_Q2_P4.py"
```