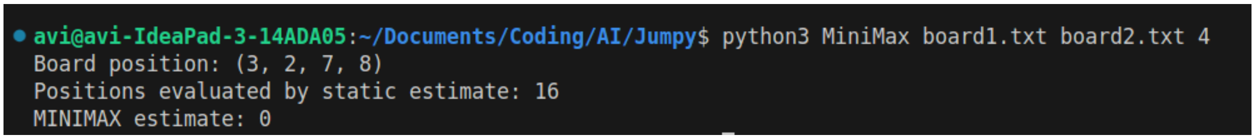
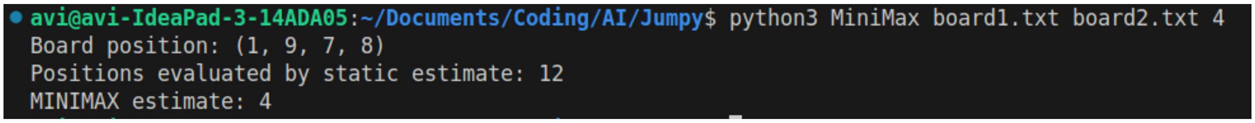
MINIMAX:

1. Running MiniMax with board1.txt = 1278



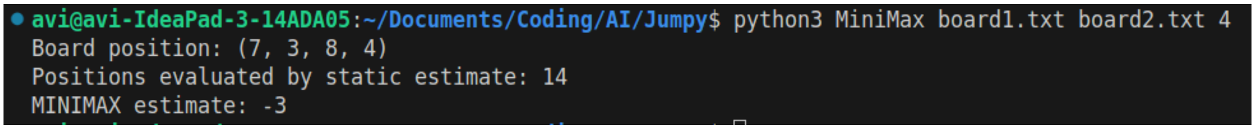
```
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 MiniMax board1.txt board2.txt 4
Board position: (3, 2, 7, 8)
Positions evaluated by static estimate: 16
MINIMAX estimate: 0
```

2. Running MiniMax with board1.txt = 1678



```
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 MiniMax board1.txt board2.txt 4
Board position: (1, 9, 7, 8)
Positions evaluated by static estimate: 12
MINIMAX estimate: 4
```

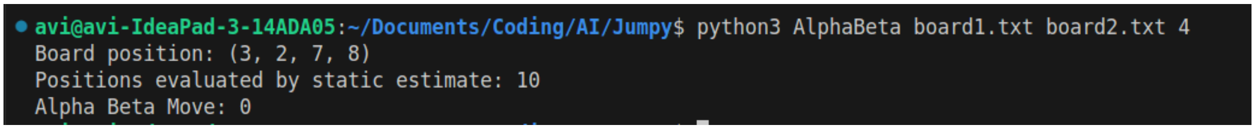
3. Running MiniMax with board1.txt = 7124



```
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 MiniMax board1.txt board2.txt 4
Board position: (7, 3, 8, 4)
Positions evaluated by static estimate: 14
MINIMAX estimate: -3
```

ALPHA-BETA PRUNING:

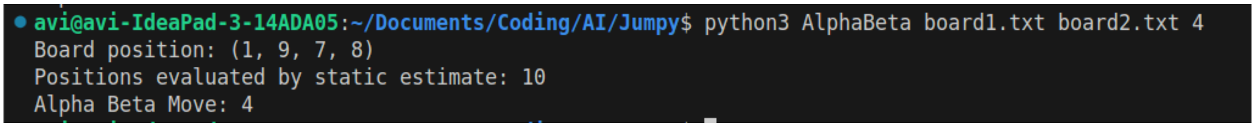
1. Running AlphaBeta with board1.txt = 1278



```
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 AlphaBeta board1.txt board2.txt 4
Board position: (3, 2, 7, 8)
Positions evaluated by static estimate: 10
Alpha Beta Move: 0
```

6 less nodes evaluated than MiniMax with board1.txt = 1278

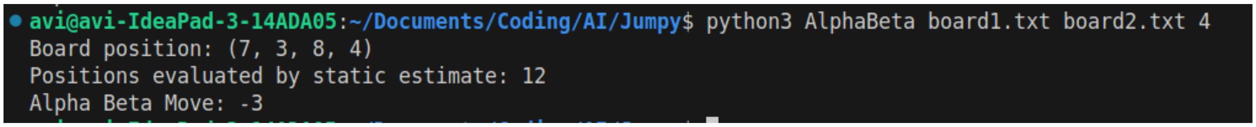
2. Running AlphaBeta with board1.txt = 1678



```
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 AlphaBeta board1.txt board2.txt 4
Board position: (1, 9, 7, 8)
Positions evaluated by static estimate: 10
Alpha Beta Move: 4
```

2 less nodes evaluated than MiniMax with board1.txt = 1678

3. Running AlphaBeta with board1.txt = 7124

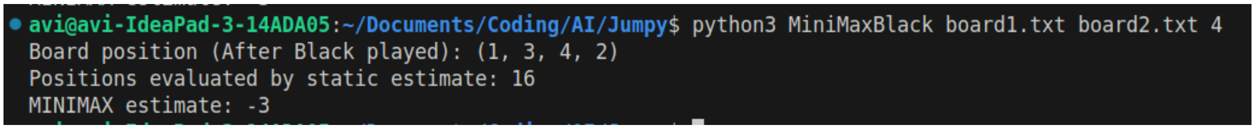


```
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 AlphaBeta board1.txt board2.txt 4
Board position: (7, 3, 8, 4)
Positions evaluated by static estimate: 12
Alpha Beta Move: -3
```

2 less nodes evaluated than MiniMax with board1.txt = 7124

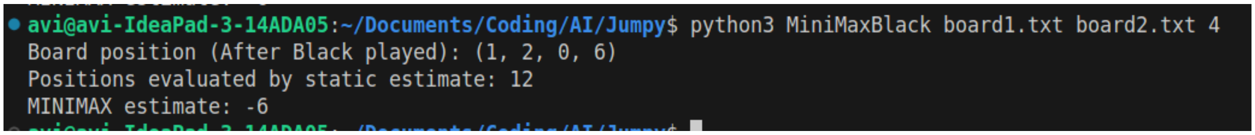
MINIMAXBLACK:

1. Running MiniMaxBlack with board1.txt = 1345



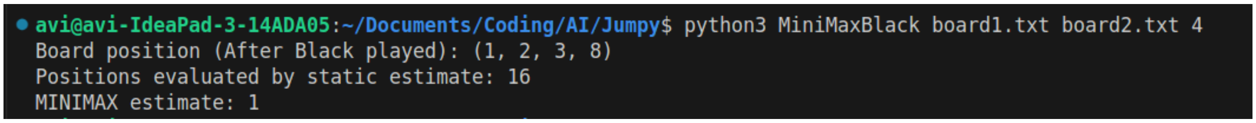
```
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 MiniMaxBlack board1.txt board2.txt 4
Board position (After Black played): (1, 3, 4, 2)
Positions evaluated by static estimate: 16
MINIMAX estimate: -3
```

2. Running MiniMaxBlack with board1.txt = 1236



```
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 MiniMaxBlack board1.txt board2.txt 4
Board position (After Black played): (1, 2, 0, 6)
Positions evaluated by static estimate: 12
MINIMAX estimate: -6
```

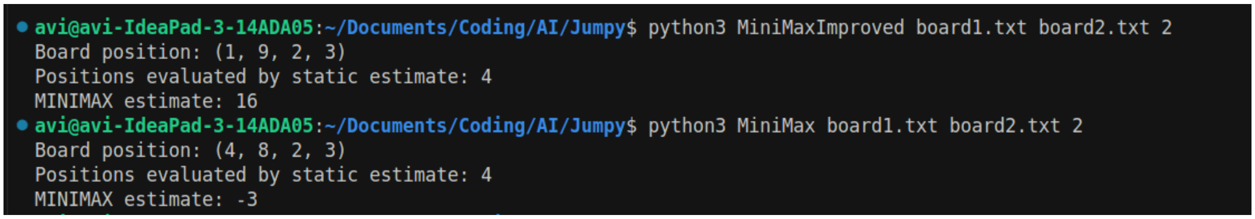
3. Running MiniMaxBlack with board1.txt = 1458



```
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 MiniMaxBlack board1.txt board2.txt 4
Board position (After Black played): (1, 2, 3, 8)
Positions evaluated by static estimate: 16
MINIMAX estimate: 1
```

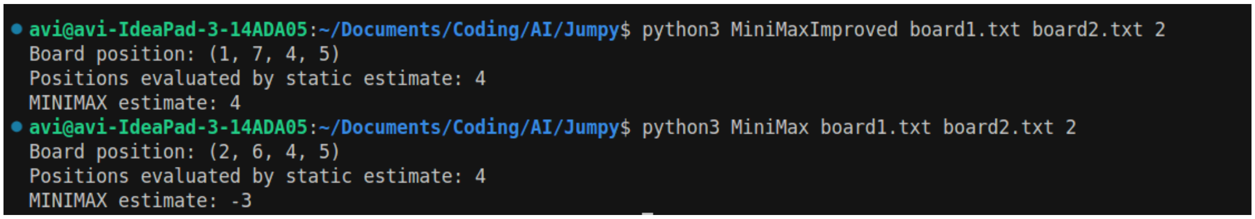
MINIMAXIMPROVED:

1. Running both MiniMax and MiniMaxImproved with board1.txt = 1823



```
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 MiniMaxImproved board1.txt board2.txt 2
Board position: (1, 9, 2, 3)
Positions evaluated by static estimate: 4
MINIMAX estimate: 16
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 MiniMax board1.txt board2.txt 2
Board position: (4, 8, 2, 3)
Positions evaluated by static estimate: 4
MINIMAX estimate: -3
```

2. Running both MiniMax and MiniMaxImproved with board1.txt = 1645



```
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 MiniMaxImproved board1.txt board2.txt 2
Board position: (1, 7, 4, 5)
Positions evaluated by static estimate: 4
MINIMAX estimate: 4
avi@avi-IdeaPad-3-14ADA05:~/Documents/Coding/AI/Jumpy$ python3 MiniMax board1.txt board2.txt 2
Board position: (2, 6, 4, 5)
Positions evaluated by static estimate: 4
MINIMAX estimate: -3
```

I believe that my MiniMaxImproved is a better static estimate because it heavily awards winning (if w1 == 9 and w2 == 9 or b1 == 0 and b2 == 0, then assign 1000 or -1000 respectively), similar to the instructor's estimate, as well as encouraging partial rewards for being closer to winning. It doubles the estimate for the white piece that is closer to winning and the same for the black piece. The function does this by calculating the bigger value for white using the max function and doubling it and finding the smaller value for black using the min function and doubling it. It finally returns the sum of the findings, similar to the instructor's function.