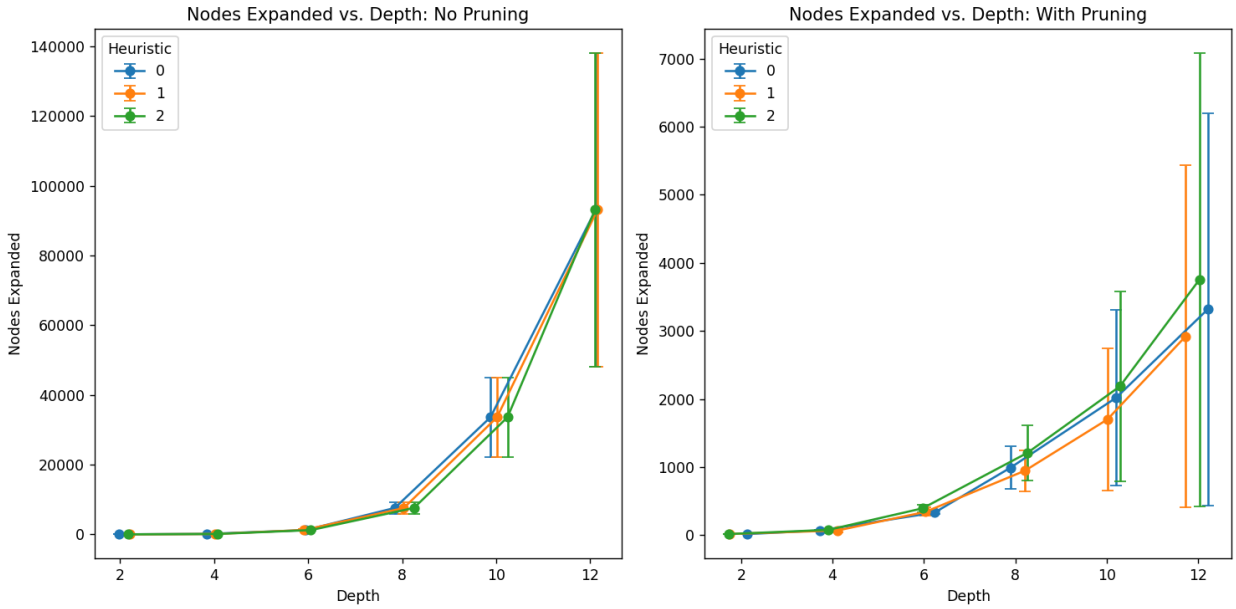
CS331 - 4x4 Othello Assignment
5/17/2023
Nathan Rumsey
Edson Fuentes

Contribution Statement:
Edson Fuentes: Report write-up
Nathan Rumsey: Code Implementation and Generation of Report Results

4x4 Othello Report

Plot of Nodes Expanded vs. Search Depth



Win Rates for Each Heuristic Against Every Other Heuristic & Player Win Rates By Depth

```
Heuristic 0 vs. Heuristic 1
Wins: 1, Losses: 3, Ties: 0

Heuristic 0 vs. Heuristic 2
Wins: 0, Losses: 4, Ties: 0

Heuristic 1 vs. Heuristic 0
Wins: 1, Losses: 3, Ties: 0

Heuristic 1 vs. Heuristic 2
Wins: 1, Losses: 3, Ties: 0

Heuristic 2 vs. Heuristic 0
Wins: 1, Losses: 3, Ties: 0

Heuristic 2 vs. Heuristic 1
Wins: 2, Losses: 2, Ties: 0
```

```
Depth: 2
        P1 Wins: 4
        P1 Win Rate: 0.6666666666666666
        P2 Wins: 2
        P2 Win Rate: 0.3333333333333333
Depth: 4
        P1 Wins: 2
        P1 Win Rate: 0.3333333333333333
        P2 Wins: 4
        P2 Win Rate: 0.6666666666666666
Depth: 6
        P1 Wins: 0
        P1 Win Rate: 0.0
        P2 Wins: 6
        P2 Win Rate: 1.0
Depth: 8
        P1 Wins: 0
        P1 Win Rate: 0.0
        P2 Wins: 6
        P2 Win Rate: 1.0
```

```
Heuristic 0 total victories: 7
Heuristic 1 total victories: 7
Heuristic 2 total victories: 10
```

Result Commentary

Overall, we see very similar results for the three heuristics with no pruning. Taking a closer look at each of them though, we can observe that heuristic 0 expanded the most nodes, followed by heuristic 1 then 2. The amount of nodes expanded appears to grow exponentially as depth increases. With pruning however, the results vary slightly. It's almost the opposite. Heuristic 2 expanded the most nodes followed by Heuristic 2 then 1. Additionally, with pruning it's important to note that the average number of nodes expanded with pruning for each Heuristic is nearly half that of without pruning. We can say that H0 overall has a larger number of node expansions due to the fact that the evaluation of the difference in number of pieces is simpler and does not require quality or strategic value leading to a larger search space compared to the other two. H1 takes into account the number of legal moves each player will have in a state which gives a better idea of who would win since the more possibilities means the more opportunities to flip pieces.The custom evaluation function calculates the difference in the sum of the number of symbols that would be flipped by each player in each successor of the given state, which gives a better idea in the “potential” of a state since the number of pieces a player can flip represents how much further they can flip the flow of the game in their way. Since H1 and H2 progressively take into account more nuanced information, their search space is reduced because they can better determine which actions are poor, compared to H0’s naive approach.

The line plot above shows the number of nodes expanded by each heuristic with pruning enabled and disabled at each search depth. Please note that it contains 95% confidence intervals for the number of nodes expanded at each depth by each heuristic to show the variability of the number of nodes expanded at each depth. Additionally, we can see from the plots above that as depth increases, then the variability of nodes also increases.

In terms of winning rates there is an important thing to note from our results. As depth increases, player 2 is performing better. We suspect this to be the case because initially both players are not playing optimally but with more trial and error player 2 begins to play more optimally. Since player 2 is playing optimally, this means that player 2 should be always winning. This is the reasoning we have deduced that Player 2 is not winning all the time as stipulated in the program requirements.