Methodology

For this assignment, the wolves and chickens puzzle was implemented and solved using both uninformed and informed search algorithms. For each algorithm implementation, I included: the fringe, the given initial state, and the final goal state.

For the implementation of breadth first search, a first in first out queue for the frontier is used. Then a new node will be added to the end of the queue. Then the old last node will be expanded first.

For the depth first search algorithm, everything is the same as the breadth first search algorithm except a last in first out queue is used.

The iterative deepening search was implemented

Results

	Test 1	Test 2	Test 3
Breadth First Search	11,14	33, 96	391, 980
Depth First Search	11, 11	59, 72	401, 35936
Iterative Deepening Depth First Search	11, 106	33, 4978	391, 5 066 459
A*	11, 13	33, 92	391, 976

*(Solution Path Length, Total Nodes Expanded)

Discussion

With BFS I was surprised with how well it performed even in the larger tests. Considering that it has a space and time complexity of $O(b^d)$, I was going into the test cases under the impression that it would take noticeably longer to run on the final test case. However, it would be possible that the test case was not large enough to noticeably affect runtime. BFS is a satisfactory search algorithm considering it found an optimal solution in a reasonable runtime.

DFS did not find optimal solutions for test cases 2 and 3.

For IDDFS, it found the optimal solution however it took much longer than the other algorithms. Where others took a brief moment to run the final test case, IDDFS took what felt like minutes to run. Since IDDFS is DFS being performed iteratively, the runtime being longer is to be expected. Even though IDDFS and BFS both have a runtime complexity of $O(b^d)$, I was surprised at how much faster BFS found a solution that was still optimal.

In theory, A* should find the optimal solution with the fastest run time with the addition of the heuristic function. This was observed in the test runs. The results were: optimal, quick, and complete.

Conclusion

Of the 4 algorithms, A* performed the best which was closely followed by breadth first search. However, I found the implementation of A* to be noticeably more difficult than BFS. Additionally, what surprised me the most was that BFS found solution paths that were just as good as A* with a similar number of nodes expanded in the given test cases. DFS performed well on the smaller test cases and somewhere along the line, it became very inefficient. DFS was also the only algorithm that did not find an optimal solution, which indicates a lack of completeness. While the iterative deepening search algorithm found an optimal solution it was by far the slowest.