CS-331 Program Assignment 1

Zhaolong Wu, Rutger Farry

April 21, 2017

- **Methodology (5 points)**

In this assignment, we implemented the cannibals and missionaries puzzle solution finding algorithm by using the uniformed and informed search algorithms. We implemented all four algorithms by using the pseudocode provided by the textbook in Python. We set up the game environment by defining a *state* class and a *node* class. We also made *child_node* function which takes a state and returns the set of possible successor states

For the implementation of breath-first search algorithm, we simply using a FIFO queue for the frontier, then a new node will be added to the back of the queue, the old last node will be expanded first.

For the depth first search algorithm, everything is the same as the breath-first algorithm except we used a LIFO queue as a frontier instead of using FIFO queue.

The iterative deepening search is implemented by repeatedly applies depth-limited search algorithm with increasing the limits, the algorithm gets stopped when solution is found or the depth limited search algorithm returns a failure. The depth-limited search function has a recursive depth-limited search function inside. We set the depth limit to the max-system size*10 in here but I'm sure we could have used any reasonable small numbers like 1000.

For the A-star search algorithm, we first made a heuristic search function, where we made a *score* function that constantly comparing how many cannibals and missionaries are currently on the left side of river to the solution state, by subtracting current number to the goal number, where the lower number mains better. Then what we did is just like what we did in the breath-first and depth first search, but we used priority queue in here.

- **Results (10 points)**

	Test Case 1	Test Case 2	Test Case 3
BFS	12 steps 45 nodes expanded	34 steps 1658917 nodes expanded	
DFS	12 steps 26 nodes expanded	40 steps 133 nodes expanded	1122 steps 8881 nodes expanded
IDDFS	12 steps Nodes expanded		
a-star	12 steps 28 nodes	34 steps 132 nodes expanded	378 steps 3461 nodes expanded

- **Discussion (10 points)**

For the breath-first search, it does always give an optimal solution, despites that it has $O(b^d)$ for both time and space complexity but throughout the experiment we found that it runs relatively slow compare with the other 3.

For the depth-first search algorithm, it did not give optimal solution for most of cases.

The iterative deepening search takes most time and space among the 4 algorithms, though the searching completeness is probably the best among the 4, the time complexity is exponential $O(b^d)$ where b is the branching factor and d is the max depth, the space complexity is O(d) which is not too much better neither. If we don't care about searching completeness we should definitely avoid using the IDDFS algorithm.

A star search theoretically gives optimal solution with the fastest running time, by adding the heuristic feature. Which is what we observed during the test runs.

- **Conclusion (5 points)**

Among the 4 algorithms, we conclude that the a-star search definitely performed the best, but it is not the easiest algorithm to implement. The BFS is also surprisingly always gave the optimal solution, most importantly it is the most straight forward and easiest algorithm to implement among the 4 algorithms, the DFS is a bit fuzzy sometimes that doesn't give the optimal solution where IDDFS algorithm does provide the optimal solution but takes forever to run, we choose the A-star for the cannibals and missionaries puzzle.