Anthony Tyrrell, Kirk Stennett
Program #1
Spring 2016
4/13/16

Missionaries and Cannibals

- **Methodology:** These search algorithms were all more or less tested the same. We introduced a fourth test case to see how these algorithms expanded large values. Originally to ensure that the breadth-first search algorithm was functioning, we expanded the nodes by hand to check that everything was expanding properly. It wasn't until we encountered the Iterative Deepening DFS and A-Star that we had to start modifying the search parameters (past that of the queues for BFS and DFS).

  With the ID-DFS we decided that the algorithm would follow the standard and increase depth by one each time until it finds a solution. With the A-Star search, the heuristic was defined to prioritize moving the missionaries over to the left side of the bank without generating harmful states. The thinking behind this was that once the missionaries were all on the other side the cannibals could be moved over easily without the risk of eating all the missionaries.

- **Results:**

  n= Goal Depth
  n/m **M** = Nodes visited

	Breadth-First	**Depth-First**	**ID-DFS**	**A-Star**
Start 1	**11/16**	**11/11**	**11/11**	**11/15**
Start 2	**23/49**	**23/35**	**23/33**	**23/27**
Start 3	**333/4009**	**1279/1598**	**1279/1513**	**333/422**
Start 4	**1497/105709**	**34957/36419**	**34957/36022**	**1497/1898**

- **Discussion:** These results are somewhat as we expected. All of the search algorithms are able to get the first two tests perfectly as they should, with some difference in the expansion. However, there was some interesting results with the Depth-First and Iterative Deepening Depth-First Search, on test case 3. They both expanded down to 1279, which seemed to be odd as IDDFS is a hybrid of BFS and DFS, and therefore should have an optimal solution. We believe that this is part of how the algorithm operates in this particular case, with only unique states being generated, and therefore a lot of dead ends and nodes being discarded.

- **Conclusion:** From these results we can conclude that using an informed search is the most efficient way to generate the optimal solution in this scenario. The difference is most apparent in test case 4, which was one that we created to see how these handle larger data sets. This also shows that using a well thought out heuristic drastically reduces the amount of expansion that occurs in the graph and optimizes the search functionality. Other than that the results don't really start showing drastic differences until we get to the last test case. The results of the third test case show that the A-Star search is clearly the best out of all of the other search algorithms in that it only expands 422 nodes total instead of the next best of 1513, less than a third of that. And showing that Breadth-First Search is the most inefficient by a large margin.