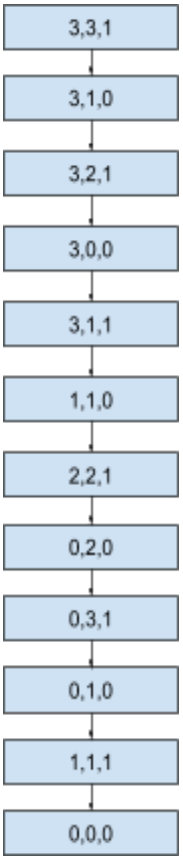
Assignment #0

Aniruddha Prithul
aprithul@nevada.unr.edu

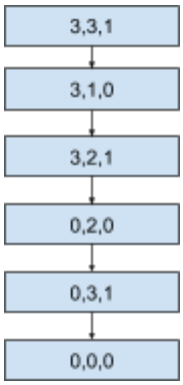
1. The state can be represented as a vector of 3 numbers, <M,C,B>. The three components of the vector represent the number of Missionaries, Cannibals and Boat on the wrong side of the river respectively. Initially, we have the state <3,3,1> with all 6 people and the boat on the wrong side. The goal is to reach the state <0,0,0>. At any state, we have a finite number of actions, that when acted upon, transitions the system into a new state. The actions are also 3 component vectors which are added (to move people to the wrong side) or subtracted (to move people to the correct side) in order to transition into new states.

2. To solve the problem we first generate a tree of state transitions. The tree is generated by an algorithm similar to bfs. Each time we generate a new state, we save a reference to the previous state in it. We stop the generation process when we first reach a goal state. This ensures that the goal state we find is optimal (since it's the first one to be found using a bfs like approach). This also means that once we generate the tree, we already have the solution path. We just have to follow the parent references from the goal to reach the root state and this will give us the solution.
   If we don't keep track of repeated states, it's possible to spend a very large portion of the cpu time being stuck in a loop and generating repeated state transitions. While a goal state will still be found eventually, this isn't optimal use of resources. That's why 'visited' flags were maintained for each state to avoid checking repeated states.

3. The diagram below shows the state transitions of the solution path.



4. The initial algorithm was extended so that the boat can hold a maximum of 3 people. Unlike the case with 2 people, it's now possible to have a invalid state on the boat when there are 2 cannibals and 1 missionary. This adds 3 more actions to the existing 5. The `generate_actions(state, max_occupancy)` function takes this into consideration. The below diagram shows the state transition with 3 people allowed on the boat.



5. Although the state space is relatively small with only 32 states, it can still be hard to solve the problem. I think most people start by randomly moving around the entities in the problem to see if they can reach a goal state. This way it's also easy to get stuck in a loop where they keep visiting the same state multiple times.