

Submission: Please submit all files through MyGateway. Please do not email final solutions to me, but feel free to ask me questions and code by email.

In this project you will be attempting to solve the Cannibals and Missionaries problem:

Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place.

Your task is to implement and solve this problem by generating an appropriate tree structure and then using an appropriate search algorithm to search this tree.

First you will want to formulate the problem precisely and ensure that you understand what a valid solution is. In order to simplify this for you, I suggest you use the following tips, though you can use your own formulation if you want:

Represent the current state using a simple vector $\langle a, b, c \rangle$. This would represent the number of missionaries on the wrong side, cannibals on the wrong side and number of boats on the wrong side. So starting out, all the cannibals, boats (we only have one) and missionaries are on the wrong side, for $\langle 3, 3, 1 \rangle$

Represent actions by using vector subtraction/addition to change the state. For example, suppose a boat carrying one cannibal crosses the river, you would subtract $\langle 0, 1, 1 \rangle$ from our $\langle 3, 3, 1 \rangle$ to get $\langle 3, 2, 0 \rangle$. Meaning that we now have 3 missionaries on the wrong side, 2 cannibals on the wrong side, and no boats on the right side.

For each state, there are 5 possible actions, $\langle 1, 0, 1 \rangle$, $\langle 2, 0, 1 \rangle$, $\langle 0, 1, 1 \rangle$, $\langle 0, 2, 1 \rangle$, $\langle 1, 1, 1 \rangle$. You start at $\langle 3, 3, 1 \rangle$ and start generating a search tree, getting the successor nodes by subtracting these actions from it. This generates the child states. From these children, you get to the grandchildren by adding these actions to it. So at each depth you reverse the addition or subtraction. Continue this until you reach a node of $\langle 0, 0, 0 \rangle$, which is the solution.

Make sure to check for invalid states, that is, if any node has more cannibals than missionaries on either bank. For example, from the initial state of $\langle 3, 3, 1 \rangle$, only 3 valid children will be generated, $\langle 3, 2, 0 \rangle$, $\langle 3, 1, 0 \rangle$ and $\langle 2, 2, 0 \rangle$.

You must use iterative deepening for this search. I suggest you write a function that, given a maximum depth, searches for a solution and either returns it or says it did not find a solution. Then repeat this until a solution is found. I expect your program to present the entire solution (ie: Which actions were taken) after it finds a solution.