AI Presentation

Presented by kim lopes

Problem Statenment

Three missionaries and three cannibals are on one side of a river. They have a boat that can carry at most two people. If at any time the cannibals outnumber the missionaries on either side of the river, the missionaries will be eaten. The goal is to find a way to get all three missionaries and three cannibals safely to the other side of the river without violating these rules.
The problem can be formulated as a search problem, where the states represent the configurations of people on each side of the river and the actions represent moving people from one side to the other. The task is then to find a sequence of actions (moves) that lead from the initial state to the goal state without violating the constraints.

Process To Solution

01

LStart with all three missionaries and three cannibals on the left side of the river.
Take two cannibals across the river to the right side.

02

Send one cannibal back to the left side.
Take two missionaries across to the right side

03

Send one cannibal back to the left side.
Take two cannibals across to the right side.

04

Send one cannibal back to the left side.
Take two missionaries across to the right side.

State-space search

This is a general problem-solving technique where you explore a set of possible states to find a solution. It involves defining the initial state, possible actions, transitions between states, and a goal state.

Depth-first search

DFS is an algorithm for traversing or searching tree or graph data structures. It starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking

A search*: A*

A* is a best-first search algorithm that finds the least-cost path from a given initial node to one goal node (out of one or more possible goals). It evaluates nodes by combining the cost to reach the node and an estimate of the cost to reach the goal from the node

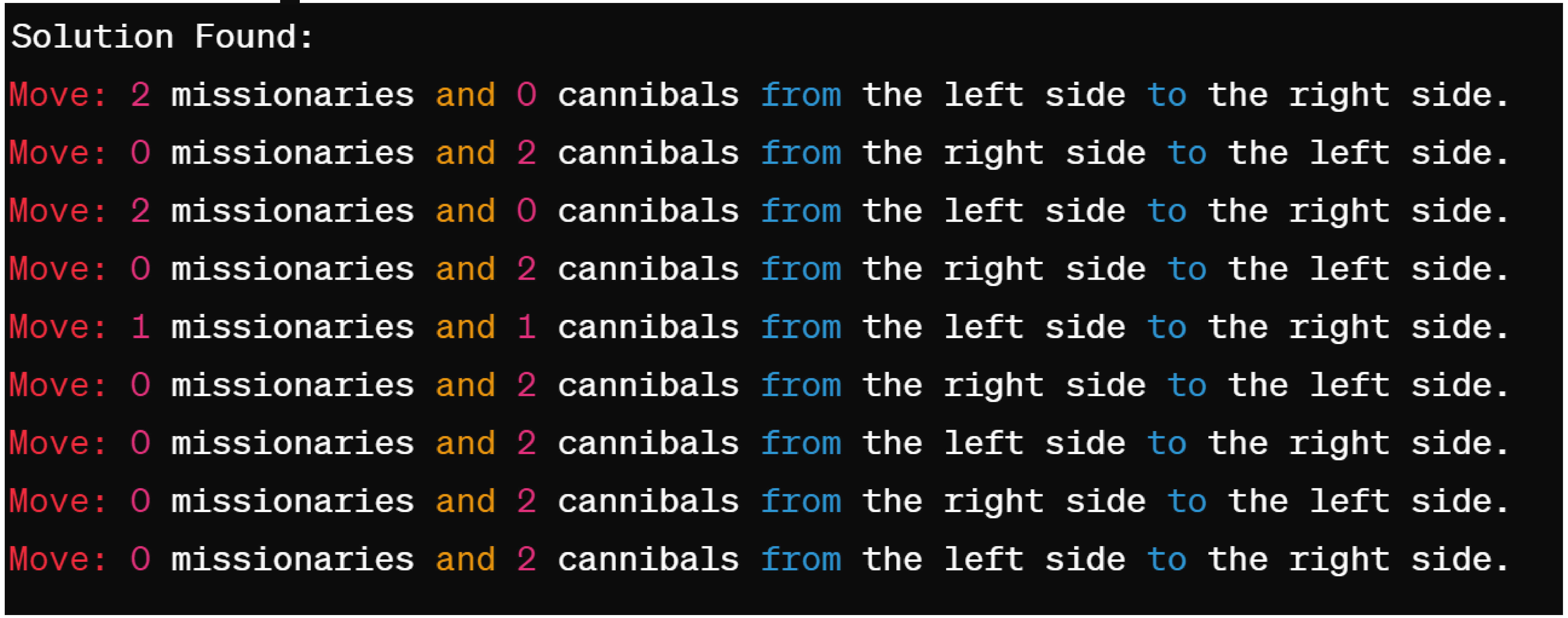
Breadth-first search

BFS is an algorithm for traversing or searching tree or graph data structures. It starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level.

Solution using BFS

1. Data Structures Definition:
    - State struct: Represents the state of the problem, including the number of missionaries and cannibals on the left side of the river, the boat's position, and is_valid() and is_goal() functions to validate the state and check if it's a goal state.
    - Node struct: Represents a node in the search tree, containing a state and the index of its parent node.
    - Queue struct: Implements a basic queue for BFS using an array.
2. Queue Operations:
    - initialize_queue(): Initializes the queue.
    - is_empty(): Checks if the queue is empty.
    - enqueue(): Adds a node to the rear of the queue.
    - dequeue(): Removes a node from the front of the queue.
3. BFS Function (bfs()):
    - Initializes a queue and enqueues the initial state node with a parent index of -1.
    - Enters a while loop that continues until the queue is empty.
    - Dequeues a node from the front of the queue.
    - Checks if the dequeued node's state is the goal state. If so, it prints the solution path and returns.
    - Generates child nodes by iterating over possible moves (adding or subtracting 1 or 2 missionaries and cannibals).
    - Checks if the generated child state is valid and enqueues it if it is.
    - Repeats the process until a solution is found or the queue becomes empty.
4. Printing Solution Path (print_path()):
    - Recursively prints the solution path by traversing the parent nodes starting from the last node in the queue.
5. Main Function:
    - Initializes the initial state.
    - Calls the bfs() function with the initial state to start the breadth-first search

Output



```
Solution Found:
Move: 2 missionaries and 0 cannibals from the left side to the right side.
Move: 0 missionaries and 2 cannibals from the right side to the left side.
Move: 2 missionaries and 0 cannibals from the left side to the right side.
Move: 0 missionaries and 2 cannibals from the right side to the left side.
Move: 1 missionaries and 1 cannibals from the left side to the right side.
Move: 0 missionaries and 2 cannibals from the right side to the left side.
Move: 0 missionaries and 2 cannibals from the left side to the right side.
Move: 0 missionaries and 2 cannibals from the right side to the left side.
Move: 0 missionaries and 2 cannibals from the left side to the right side.
```