**Course-Project**

**ARTIFICIAL INTELLIGENCE**

****

**Title: -** **The Missionaries and Cannibals Problem**

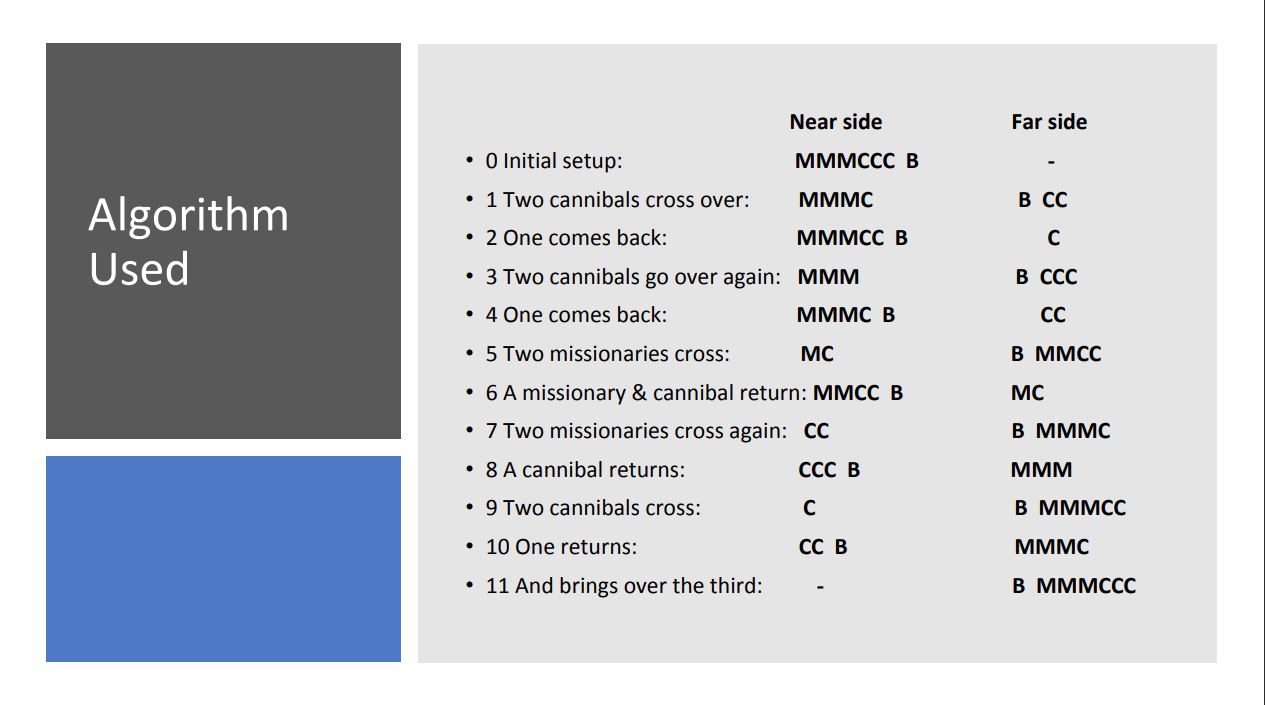
**Language used: Prolog**

**Group Details:**

**Group No: TY-12 Div: B Batch: B-3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Name** | **Roll.No** | **Gr.No** |
| 1 | Shubham Kakde | 49 | 11811244 |
| 2 | Sidharth Kakde | 50 | 11811170 |
| 3 | Vedant Katti | 55 | 11810142 |
| 4 | Vinayak Kokane | 58 | 11810499 |
| 5 | Dhanraj Kore | 60 | 11810970 |

**Algorithm:**

****

**Code:**

% Represent a state as [CL,ML,B,CR,MR]

%

start([3,3,left,0,0]).

goal([0,0,right,3,3]).

legal(CL,ML,CR,MR) :-

% is this state a legal one?

ML>=0, CL>=0, MR>=0, CR>=0,

(ML>=CL ; ML=0),

(MR>=CR ; MR=0).

% Possible moves:

move([CL,ML,left,CR,MR],[CL,ML2,right,CR,MR2]):-

% Two missionaries cross left to right.

MR2 is MR+2,

ML2 is ML-2,

legal(CL,ML2,CR,MR2).

move([CL,ML,left,CR,MR],[CL2,ML,right,CR2,MR]):-

% Two cannibals cross left to right.

CR2 is CR+2,

CL2 is CL-2,

legal(CL2,ML,CR2,MR).

move([CL,ML,left,CR,MR],[CL2,ML2,right,CR2,MR2]):-

% One missionary and one cannibal cross left to right.

CR2 is CR+1,

CL2 is CL-1,

MR2 is MR+1,

ML2 is ML-1,

legal(CL2,ML2,CR2,MR2).

move([CL,ML,left,CR,MR],[CL,ML2,right,CR,MR2]):-

% One missionary crosses left to right.

MR2 is MR+1,

ML2 is ML-1,

legal(CL,ML2,CR,MR2).

move([CL,ML,left,CR,MR],[CL2,ML,right,CR2,MR]):-

% One cannibal crosses left to right.

CR2 is CR+1,

CL2 is CL-1,

legal(CL2,ML,CR2,MR).

move([CL,ML,right,CR,MR],[CL,ML2,left,CR,MR2]):-

% Two missionaries cross right to left.

MR2 is MR-2,

ML2 is ML+2,

legal(CL,ML2,CR,MR2).

move([CL,ML,right,CR,MR],[CL2,ML,left,CR2,MR]):-

% Two cannibals cross right to left.

CR2 is CR-2,

CL2 is CL+2,

legal(CL2,ML,CR2,MR).

move([CL,ML,right,CR,MR],[CL2,ML2,left,CR2,MR2]):-

% One missionary and one cannibal cross right to left.

CR2 is CR-1,

CL2 is CL+1,

MR2 is MR-1,

ML2 is ML+1,

legal(CL2,ML2,CR2,MR2).

move([CL,ML,right,CR,MR],[CL,ML2,left,CR,MR2]):-

% One missionary crosses right to left.

MR2 is MR-1,

ML2 is ML+1,

legal(CL,ML2,CR,MR2).

move([CL,ML,right,CR,MR],[CL2,ML,left,CR2,MR]):-

% One cannibal crosses right to left.

CR2 is CR-1,

CL2 is CL+1,

legal(CL2,ML,CR2,MR).

% Recursive call to solve the problem

path([CL1,ML1,B1,CR1,MR1],[CL2,ML2,B2,CR2,MR2],Explored,MovesList) :-

move([CL1,ML1,B1,CR1,MR1],[CL3,ML3,B3,CR3,MR3]),

not(member([CL3,ML3,B3,CR3,MR3],Explored)),

path([CL3,ML3,B3,CR3,MR3],[CL2,ML2,B2,CR2,MR2],[[CL3,ML3,B3,CR3,MR3]|Explored],[ [[CL3,ML3,B3,CR3,MR3],[CL1,ML1,B1,CR1,MR1]] | MovesList ]).

% Solution found

path([CL,ML,B,CR,MR],[CL,ML,B,CR,MR],\_,MovesList):-

output(MovesList).

% Printing

output([]) :- nl.

output([[A,B]|MovesList]) :-

output(MovesList),

write(B), write(' -> '), write(A), nl.

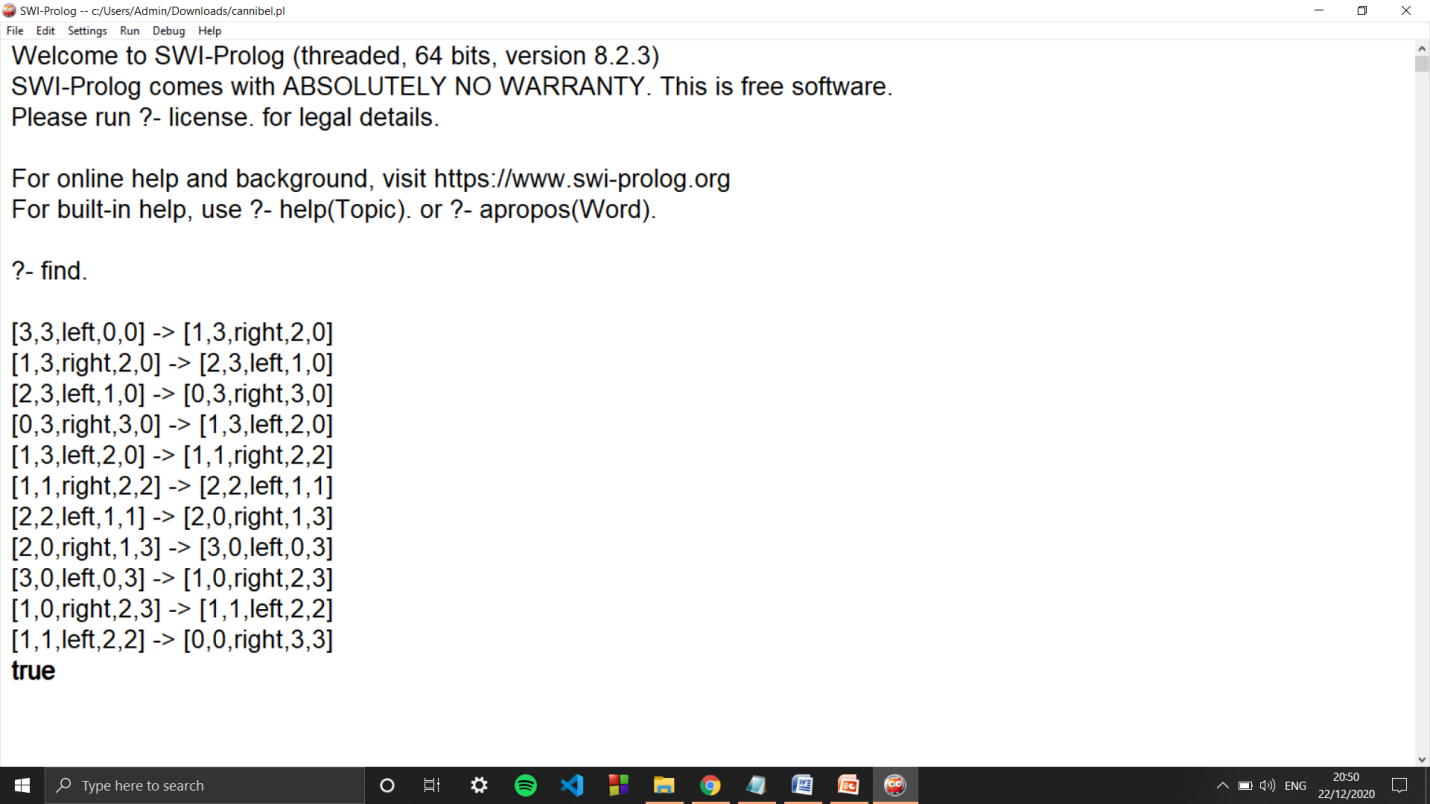
% Find the solution for the missionaries and cannibals problem

find :-

path([3,3,left,0,0],[0,0,right,3,3],[[3,3,left,0,0]],\_).

%arguments are left side,right side and goal.

**Output Screen:**

****