PROGRAM

import operator

from search import Problem

from .state import State

from .state\_constants import GOAL\_STATE

from .state\_constants import INITIAL\_STATE

class MissionariesAndCannibals(Problem):

def \_\_init\_\_(self):

initial\_state = State.value\_of(INITIAL\_STATE)

goal\_state = State.value\_of(GOAL\_STATE)

super().\_\_init\_\_(initial\_state, goal\_state)

def actions(self, state):

all\_actions = self.get\_all\_actions()

return self.get\_valid\_actions(state, all\_actions)

@staticmethod

def get\_all\_actions():

return {

(1, 0, 1),

(2, 0, 1),

(0, 1, 1),

(0, 2, 1),

(1, 1, 1)

}

def get\_valid\_actions(self, state, all\_actions):

is\_action\_valid\_lambda = self.get\_is\_action\_valid\_lambda(state)

return set(filter(is\_action\_valid\_lambda, all\_actions))

def get\_is\_action\_valid\_lambda(self, state):

return lambda action: self.is\_action\_valid(state, action)

def is\_action\_valid(self, state, action):

operate = self.get\_operation(state.boat)

result = operate(state, action)

return result.is\_valid()

def result(self, state, action):

operate = self.get\_operation(state.boat)

return operate(state, action)

@staticmethod

def get\_operation(boat):

"""Subtract action from state if boat is on initial side of river."""

return operator.sub if boat == 1 else operator.add

OUTPUT

