

README

Name: Pratik Rajendra Deshpande

NetId: PRD190001

Email : prd190001@utdallas.edu

Homework 1 AI CS6364

Problems Solved:

Q1 – a. describe the state space, b. state update function, utility function

Q2 – Manual – a. formulate the search problem, b. sketch manual solution, motivate.

Q2 – Programming – a) Use AIMA code, to implement b) Describe search space c) Give Solution Path

Q2 – Extra Credit – Listed first 5 steps for – Greedy Best First Search, A* Search, Uniform Cost Search

Q3 – Manual Solution for A*

File and Contents

HW1-prd190001.pdf : This file contains the manual solutions and the solutions of the above problems.

Code For Q2:

The code for question 2 can be found in the folders missionaries-and-cannibals.

Running “*python main.py*” would run the code and demonstrate the solution for uniform cost search

- **Folder: missionaries_and_cannibals:**
 - **Folder: missionaries_and_cannibals:**
 - **__init__.py**
 - **missionaries_and_cannibals.py:**
 - A subclass of Problem from aima code.
 - It is the definition of this problem
 - This has all the actions of the problem.
 - Imports state_constants.py that has INITIAL_STATE and GOAL_STATE
 - Uses a function to get all the valid actions by operating on the state instance
 - Operation depends upon the boat, if boat ==1 then the operation is subtract as boat is on the left side else its add as boat is on the right.
 - **state.py**
 - Every node has an instantiation of this state class.
 - The data of this state is a tuple (missionaries, cannibals, boat)
 - It has a function to check whether the state is valid? This is used after to model the constraints of the problem.

- The state also has a function operate that, takes other state, and this state and operates on the two to result in a new state.
 - this file also uses operate_on_tuples from util files and other utility functions
- **state_constants.py**
 - Holds the state constants INITIAL_STATE and GOAL_STATE
- **Folder search:**
 - **__init__.py**
 - **search.py** (Contains search implementations of Aima Code)
 - **node.py** (Contains node class from Aima code)
 - **problem.py** (Contains the abstract problem class from Aima code, and a heuristic function that is used by A*, RBFS, Greedy Best First Search. The heuristic function solves the relaxed problem, that is at minimum one person can be transferred to the other side of the river. ***So the heuristic function is: no of missionaries on initial side + no of cannibals on the initial side - 1***)
- **Folder: util**
 - **__init__.py**
 - **Tuple_util.py:** (imports the operator library from python and defines operations on tuples which is used by the problem , and the state to add or subtract to the states.
- **File: main.py** (this is the file that has the main function, which instantiates the problem, calls the search method, and returns the solution, this file also prints the path. Change the “uniform_cost_search” to “astar_search” or “iterative deepening search” or “greedy_best_first_graph_search” and the algorithm would be run)
- **File utils.py** (This file is the copy of the utils file on Aima code, as PriorityQueue and memoization implementation is in this file)