Artificial Intelligence CSCI 6511 Project 1

Team 2

Ulvi Aslanli, Asad Dashdamirli,

Emin Alizada, Turgay Umudzada

Repository Link

Writeup

After we read input from text files, we call the A_star function which takes pitcher capacities as a list and a single target value which is an integer. A_star function is the implementation of the A* search algorithm.

Geeks for Geeks portal defines the A* algorithm as follows:

What A* Search Algorithm does is that at each step it picks the node according to a value-‘f’ which is a parameter equal to the sum of two other parameters – ‘g’ and ‘h’. At each step it picks the node/cell having the lowest ‘f’, and processes that node/cell.

We define ‘g’ and ‘h’ as simply as possible below

g = cost to get to the current state

h = the estimated cost to the target. This is often referred to as the heuristic, which is nothing but a kind of smart guess.

The first thing we do is to check if the case provided is impossible to solve. If the target is not divisible by the greatest common divisor of pitcher capacities, then it is impossible to solve the problem, in this case -1 is returned.

Then we proceed to initialize the states. The state is a tuple that contains tuples of current water capacity pairs. It also contains two infinite pitchers out of which one always contains an infinite amount of water, and one initially contains 0, which is our target pitcher.

After that, we initialize the scores(f,g,h) and two sets (open and closed) and start iterating over the open set till we have reached the goal or there are no unvisited states. First, we pop the state with the lowest f score, which is the sum of g score and h score, from the open set and print the path and the number of steps if we have reached the target. If we have not reached the target yet, we add the state we selected to the closed set and start iterating over the states that can be generated from the current state.

We also check if we have reached the next step of our current state before, and at more cost, then we replace it and its path with the next step and next step’s path.

The heuristic we selected is a simple one:

- It calculates the absolute value of the distance between the target value and amount of water contained in our target pitcher
- divide the result by the capacity of the pitcher with a maximum capacity
- multiplies it by 2 to account for emptying the pitcher to infinite pitcher

This way we calculate the lower bound by which target can be reached.

We have tried several different heuristics before making the final decision. The reason why we decided to choose this one is that we couldn't find a test case when this heuristic gave a wrong answer and this heuristic is admissible. We have found at least one test case that other our heuristics have failed.

In case if hyperlink above doesn't work, link to the repo:
https://github.com/emin-alizada/GW-AI-Project1