

## COMPUTER SCIENCE AND DATA ANALYTICS

Course: **Artificial Intelligence**

Assignment 1

Informed Search

Student: **Shikhaliyev Anar**

Instructor: **Amrinder Arora**

## Baku 2023

## **1. Problem Description**

## In this project, water pitcher problem will be implemented to find the shortest path to obtain the target volume of water in the infinite capacity pitcher. Additionally, any type of water movement (from one pitcher to another) counts as one step.

## **2. Code analysis**

## This code is implementing the A\* algorithm to solve the water jug problem. The goal of the problem is to find the minimum number of steps required to obtain a target quantity of water using jugs of different sizes.

## The function takes two parameters: capacities, which is a list of integers representing the sizes of the jugs, and target, which is an integer representing the desired amount of water. All the inputs for the program are taken from input text file with 2 lines. First line holding a variable number of integers, comma separated and second line holding an integer representing the target value. The capacities list is modified by adding a large value at the end to represent a jug that can hold an infinite amount of water.

## The A\* algorithm is a search algorithm that uses heuristics to guide the search towards the goal. In this implementation, the heap data structure is used to store the possible states of the jugs, sorted by their estimated cost. The cost is calculated as the sum of the number of steps taken so far and the heuristic function h, which estimates the cost from the current state to the goal state. The steps variable represents the number of steps taken so far to reach the current state.

## The while loop continues until the heap is empty, indicating that all possible states have been explored, or until the goal state is reached. At each iteration of the loop, the state with the lowest estimated cost is removed from the heap, and its adjacent states are generated by pouring water from one jug to another or by filling or emptying a jug. The new states are added to the heap if they have not been visited before.

## If the goal state is found, the number of steps taken to reach it is returned. If the goal state is not reachable, -1 is returned.

## The heuristic function used in this implementation is called heuristic, which takes as input the current state of the jugs and the target quantity. It estimates the cost as the sum of the difference between the current amount of water in each jug and the target quantity, divided by the capacity of the corresponding jug. This heuristic is admissible, meaning it never overestimates the actual cost to reach the goal.

## **Heuristic**

## In this code, the heuristic function is used as a heuristic to estimate the cost from the current state to the goal state. The heuristic function is admissible, which means that it never overestimates the actual cost to reach the goal.

## The heuristic function calculates the minimum number of steps required to reach the goal state by pouring water from one jug to another. It takes the current state of the jugs, the goal state, and the capacities of the jugs as input.

## The function first checks if the goal state is achievable from the current state. If the goal state cannot be reached, the function returns the maximum integer value to indicate that the goal state is unreachable.

## The function then calculates the remaining volume of water that needs to be added to the jugs to reach the goal state. It iterates over the capacities of the jugs and calculates the minimum number of steps required to add the remaining volume to the jugs. The min\_steps variable is incremented by the integer division of the remaining volume by the current jug capacity, and the remaining volume is updated to the remainder of this division.

## The function returns the total minimum number of steps required to reach the goal state. Since the heuristic is admissible, it provides a lower bound on the cost to reach the goal state. This means that the actual cost to reach the goal state is guaranteed to be greater than or equal to the heuristic value. By using this lower bound estimate, the A\* algorithm can efficiently search for the optimal solution to the water jug problem.

## **3. Source Code**

## https://github.com/anar-sixeliyev/AI-assignment1