

Group7

Sihan Adi

Ziyan Guo

Zihan Li

Siton Wang

Lower Bound

Choose the node with the smallest f value.

$$f = g + h$$

g : is the operand from the starting point (0, 0, 0, ... 0) to reach a node. Since it is stipulated in our algorithm that each node expansion can only be performed once, filling or emptying a cup or pouring water between two cups. So every time, the g of the node is the $g+1$ of the node's parent node.

h : The expected number of operations required for the current node to reach the target node. In the water cup problem, we formulate the problem as the target water volume minus the manipulated water volume divided by the appropriate cup to get its expected number of operations to reach the target water volume. Suppose the target water volume is smaller than the operated water volume.

In that case, h is still the water volume required by the current state from the target state.

Writeup

Design Guide:

The project requires changing the water volume of the cup through specific rules to obtain the target water volume. So, we transform this problem into a shortest path problem. The water volume status of each cup and the difference from the target water volume is used as a sign of a node. According to the project requirements, we use the A* algorithm to solve the problem of the shortest path from the starting point to the target node.

In the A* algorithm, we first add the initial node to the openlist, and the water volume of each cup in the initial node root is 0. Then, transfer the node from openlist to closelist, and the newly expanded node will be added to openlist. According to the A* algorithm, we select the node with the smallest $f(n)$ from the openlist for expansion. When expanding the node, we will change the amount of water in each cup in a node. And take into account the case of operating one cup and two cups. New extension node, we will update the content of this node. Its $g(n)$, $h(n)$, and $f(n)$ will

be updated, and its parent will point to the node that generated it.

When a node is added to the openlist, if the node is in the oepnlist, it will be added to the openlist only if it represents a better path; if the node is not in the closelist or the node is not in the openlist, it can be directly added to the openlist.

The A* algorithm needs to continuously select the node with the smallest $f(n)$ from the openlist.

When the selected node is the target node, the loop ends, and the output path and number of steps are output.