Experiment 1

Aim: To implement Water Jug Problem.

Problem Definition: Given 3 jugs of capacities 12, 8 and 5 litres. The 12 litre jug is completely filled initially. Using these 3 jugs split the water to exactly obtain 6 litres

Language used: Java

Theory: Water pouring puzzles (also called water jug problems or measuring puzzles) are a class of puzzle involving a finite collection of water jugs of known integer capacities (in terms of a liquid measure such as litres or gallons). Initially each jug contains a known integer volume of liquid, not necessarily equal to its capacity. Puzzles of this type ask how many steps of pouring water from one jug to another (until either one jug becomes empty or the other becomes full) are needed to reach a goal state, specified in terms of the volume of liquid that must be present in some jug or jugs.
It is a common assumption, stated as part of these puzzles, that the jugs in the puzzle are irregularly shaped and unmarked, so that it is impossible to accurately measure any quantity of water that does not completely fill a jug. Other assumptions of these problems may include that no water can be spilled, and that each step pouring water from a source jug to a destination jug stops when either the source jug is empty or the destination jug is full, whichever happens first.

The rules are sometimes formulated by adding a source (tap) and a drain (sink) which provide an infinite amount of additional water and an opportunity to pour all liquid from any jug into the sink. Filling a jug to the rim from the tap or pouring the entire contents of jug into the drain each count as one step while solving the problem.

This variant is identical to the original, as a third container capable of holding the contents of the first two is mathematically equivalent to a tap or drain capable of filling or emptying both containers.
If the number of jugs is three, the filling status after each step can be described in a diagram of barycentric coordinates, because the sum of all three integers stays the same throughout all steps. In consequence the steps can be visualized as some kind of billiard moves in the (clipped) coordinate system on a triangular lattice.

For example, if we have a jug J1 of 5 litre (n = 5) and another jug J2 of 3 litre (m = 3) and we have to measure 1 litre of water using them. The associated equation will be 5n + 3m = 1. First of all this problem can be solved since GCD(3,5) = 1 which divides 1. Using the Extended Euclid algorithm, we get values of n and m for which the equation is satisfied which are n = 2 and m = -3. These values of n, m also have some meaning like here n = 2 and m = -3 means that we have to fill J1 twice and empty J2 thrice.

Now to find the minimum no of operations to be performed we have to decide which jug should be filled first. Depending upon which jug is chosen to be filled and which to be emptied we have two different solutions and the minimum among them would be our answer.

Solution 1 (Always pour from m litre jug into n litre jug)

1. Fill the m litre jug and empty it into n litre jug.
2. Whenever the m litre jug becomes empty fill it.
3. Whenever the n litre jug becomes full empty it.
4. Repeat steps 1,2,3 till either n litre jug or the m litre jug contains d litres of water.

Each of steps 1, 2 and 3 are counted as one operation that we perform. Let us say algorithm 1 achieves the task in C1 no of operations.

Solution 2 (Always pour from n litre jug into m litre jug)

1. Fill the n litre jug and empty it into m litre jug.
2. Whenever the n litre jug becomes empty fill it.
3. Whenever the m litre jug becomes full empty it.
4. Repeat steps 1, 2 and 3 till either n litre jug or the m litre jug contains d litres of water

Conclusion: Thus, we have implemented the water jug problem, which can be solved by both DFS and BFS. The most efficient solution is achieved when the water supply is unlimited.