

# Problem 365: Water and Jug Problem

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 44.25%

**Paid Only:** No

**Tags:** Math, Depth-First Search, Breadth-First Search

## Problem Description

You are given two jugs with capacities `x` liters and `y` liters. You have an infinite water supply. Return whether the total amount of water in both jugs may reach `target` using the following operations:

\* Fill either jug completely with water. \* Completely empty either jug. \* Pour water from one jug into another until the receiving jug is full, or the transferring jug is empty.

**Example 1:**

**Input:**  $x = 3$ ,  $y = 5$ ,  $target = 4$

**Output:** true

**Explanation:**

Follow these steps to reach a total of 4 liters:

1. Fill the 5-liter jug (0, 5).
2. Pour from the 5-liter jug into the 3-liter jug, leaving 2 liters (3, 2).
3. Empty the 3-liter jug (0, 2).
4. Transfer the 2 liters from the 5-liter jug to the 3-liter jug (2, 0).
5. Fill the 5-liter jug again (2, 5).
6. Pour from the 5-liter jug into the 3-liter jug until the 3-liter jug is full. This leaves 4 liters in the 5-liter jug (3, 4).
7. Empty the 3-liter jug. Now, you have exactly 4 liters in the 5-liter jug (0, 4).

Reference: The [Die

Hard]([https://www.youtube.com/watch?v=BVtQNK\\_ZUJg&ab\\_channel=notnek01](https://www.youtube.com/watch?v=BVtQNK_ZUJg&ab_channel=notnek01)) example.

**\*\*Example 2:\*\***

**\*\*Input:\*\***  $x = 2, y = 6, \text{target} = 5$

**\*\*Output:\*\*** false

**\*\*Example 3:\*\***

**\*\*Input:\*\***  $x = 1, y = 2, \text{target} = 3$

**\*\*Output:\*\*** true

**\*\*Explanation:\*\*** Fill both jugs. The total amount of water in both jugs is equal to 3 now.

**\*\*Constraints:\*\***

$1 \leq x, y, \text{target} \leq 10^3$

## Code Snippets

**C++:**

```
class Solution {  
public:  
    bool canMeasureWater(int x, int y, int target) {  
  
    }  
};
```

**Java:**

```
class Solution {  
public boolean canMeasureWater(int x, int y, int target) {  
  
}  
}
```

**Python3:**

```
class Solution:  
    def canMeasureWater(self, x: int, y: int, target: int) -> bool:
```