

# Ai Application & Ethics(TC-7)

## Lab 2: - Program to solve water jug problem.

---

**Name:** - Utkarsh Bhangale

**PRN:** - 20200802124

**DS1**

---

### Algorithm

An approach using BFS has been discussed in the previous post. In this post an approach using memorization and recursion has been discussed. At any point, there can be a total of six possibilities:

1. Empty the first jug completely
  2. Empty the second jug completely
  3. Fill the first jug
  4. Fill the second jug
  5. Fill the water from the second jug into the first jug until the first jug is full or the second jug has no water left
  6. Fill the water from the first jug into the second jug until the second jug is full or the first jug has no water left
- 

```
In [6]: # WaterJugProblem

from collections import defaultdict

#User Input
m = int(input("Enter Jug 1 capacity:- "))    # m = Jug 1
n = int(input("Enter Jug 2 capacity:- "))    # n = Jug 2
t = int(input("Enter aim Value:- "))
j1 = int(input("Enter water availabel in j1 at starting point:- "))
j2 = int(input("Enter water availabel in j2 at starting point:- "))

visited = defaultdict(lambda: False)

def waterJugSolver(j1, j2):

    if (j1 == t and j2 == 0) or (j1 == 0 and j2 == t):
        print(j1, j2)

        return True

    if visited[(j1, j2)] == False:
        print(j1, j2)
```

```

        visited[(j1, j2)] = True

        return(waterJugSolver(0, j2) or
               waterJugSolver(j1, 0) or
               waterJugSolver(m, j2) or
               waterJugSolver(j1, n) or
               waterJugSolver(j1 + min(j2, (m-j1)), j2 - min(j2, (m - j1))) or
               waterJugSolver(j1 + min(j1, (n-j2)), j2 + min(j2, (n - j2)))
               )

    else:
        return False
print("Steps:- ")
waterJugSolver(j1, j2)

```

```

Enter Jug 1 capacity:- 4
Enter Jug 2 capacity:- 5
Enter aim Value:- 3
Enter water availabel in j1 at starting point:- 0
Enter water availabel in j2 at starting point:- 0
Steps:-
0 0
4 0
4 5
0 5
4 1
0 1
1 0
1 5
4 2
0 2
2 0
2 5
4 3
0 3
Out[6]: True

```

## Conclusion

We have successfully implemented code to solve water jug problem.