13/11/20

AI lab test-1

Revanth.NM
1BM18CS081

Problem:

You are given two jugs, 4-litre one and a 3-litre one. Neither has any measuring marks on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 litre of water into 4-litre jug? implement this using Depth first search.

→ Let us consider X as water in 4-litre jug; i.e $0 \leq X \leq 4$

Let Y be water in 3-litre jug; i.e $0 \leq Y \leq 3$

The state space can be described as (X, Y)

∴ Start state : (0, 0) Final state : (2, 0)

Lets start with.

	State	
→	Fill 4-litre jug	$(X, Y \mid X \leq 4) \Rightarrow (4, 0)$
→	Fill 3 litre jug with 4-litre jug	$(X, Y \mid Y < 3) \Rightarrow (1, 3)$ $* X+Y \leq 4$
→	Empty 3 litre jug	$(X, Y \mid Y > 0) \Rightarrow (1, 0)$
→	Pour left 1 litre to 3 litre jug	$(X, Y \mid X > 0) = (0, 1)$
→	Fill 4 litre jug	$(4, 1)$ ∴ $Y \leq 3$
→	Fill 3 litre jug with 4 litre jug	$(2, 3)$ ∴ $Y \leq 3$
→	Empty 3 litre jug	$(2, 0)$

Final state is reached

Revanth.NM
1BM18CS081

Code:

```
def gcd(a,b):          // return gcd of
    if b==0                two values.
        return a
    return gcd(b,a%b)

def Pour(tojug, fromjug, d):
    from = fromjug
    to = 0
    step = 1
    while(from is not d) and (to is not d):
        temp = min(from, tojug - to) //min of fromjug & tojug - to
        to = to + temp
        from = from - temp
        step = step+1
        if ((from == d) or (to == d)):
            break;
        if (from == 0):
            from = fromjug
            step = step + 1
        if (to == tojug):
            to = 0
            step = step + 1;
    return step
def minsteps(n, m, d):
    if m>n:
        temp = m
        m = n
        n = temp
    if (d%(gcd(n,m)) is not 0)
        return -1
    return (min(Pour(n,m,d), Pour(m,n,d))
```