

## AI Lab test-1

Pramod D.Y  
IBM19CS405  
13/11/2020

AIM:- You are given two jugs. a 4-litre One and a 3-litre One. Neither has any measuring marker on it. How can you get exactly 2 litres of water into 4 litre Jug? Implement this Using Depth First Search.

Prepost collection

```
def main():
```

```
    start-node = [(0, 0)]
```

```
    jug = get-jug()
```

```
    goal-amount = get-goal(jug)
```

```
    check-dict = {}
```

```
    is-depth = True
```

```
    Search( start-node, jug, goal-amount,  
            check-dict, is depth )
```

```
def next-transition (jug, path, check-dict):
```

```
    result = []
```

```
    next node = []
```

```
    node = []
```

(1)

Pramod D.Y



~~a = max~~

a\_max = jugs[0]

b\_max = jugs[1]

a = path[-1][0]

b = path[-1][1]

node.append(a\_max)

node.append(b)

if not been\_there(node, check\_dist):

next\_nodes.append(node)

node = []

node.append(a)

node.append(b\_max)

if not been\_there(node, check\_dist):

next\_nodes.append(node)

node = []

node.append(min(a+b))

node.append(b - (node[0] - a))

if not been\_there(node, check\_dist):

next\_nodes.append(node)

node = []

node.append(0)

node.append(b)

(2)

Ramendra

If not been there (node, check-dict):

next-nodes.append(node)

node = []

node.append(a)

node.append(o)

If not been there (node, check-dict):

next-nodes.append(node)

def Search (starting-node, jug1, goal-amount, check-dict, is-depth):

If is-depth:

print("Implementing DFS")

else

print("Implementing BFS")

goal = []

accomplished = False

q = collections.deque()

q.appendleft(starting-node)

while len(q) != 0:

path = q.popleft()

check-dict[get\_index(path[-1])] = True

If length(path) >= 2:

print(transaction(path[-2], path[-1]))

③ Recursion



Page). path(-1)

if ! goal (path, goal-amount):

    accomplished = True

    goal = path

    break

next move = next-transition, (page, path, check-date)

for i in next-move:

    if ! depth:

        q.appendleft(i)

    else:

        q.append(i)

if accomplished:

    print("the goal is achieved")

    print.path("Problem cannot be Solved")