

Q-2 - Water Jug problem with DFS

Pranjal Sinha  
IBM18CS013  
AI LAITEST-1

```
def main()
```

```
    Starting node = [[0,0]]
```

```
    Jugs = get-jugs()
```

```
    goal amount = 2
```

```
    check dict = {}
```

```
    Search(Starting-node, jugs, goal amount, check-dict.)
```

```
def get-index(node)
```

returns a key value for a given node

node : a list of three integer representing current state of the jugs

```
return pow(7, node[0]) * pow(5, node[1])
```

```
def get-jugs()
```

Return a list of three integer representing volume of the jugs - Take volume of the jugs as an input from the user.

```
return jugs
```



def get\_god(jugs)

Returns desired amount of water

Takes desired amount as an input from the user

jug: a list of two integers representing volume of the jug

return goal\_amount

def is\_goal(path, goal\_amount)

return True, if the given path terminates at the goal node

path: a list of nodes representing the path to be checked

goal amount = an integer representing the desired amount

return path[-1][0] == goal\_amount or path[-1][1] == goal\_amount

def next\_transitions(jugs, path, check\_det)

Returns a list of all possible transitions which do not cause loops

return result



def transition

returns a string explaining the transition from old state/  
node to new state/node

def print-path(path, jugs)

prints the goal path

def search(starting-node, jugs, goal state, check dist)

Searches for a path between starting node & goal node