Assignment-2

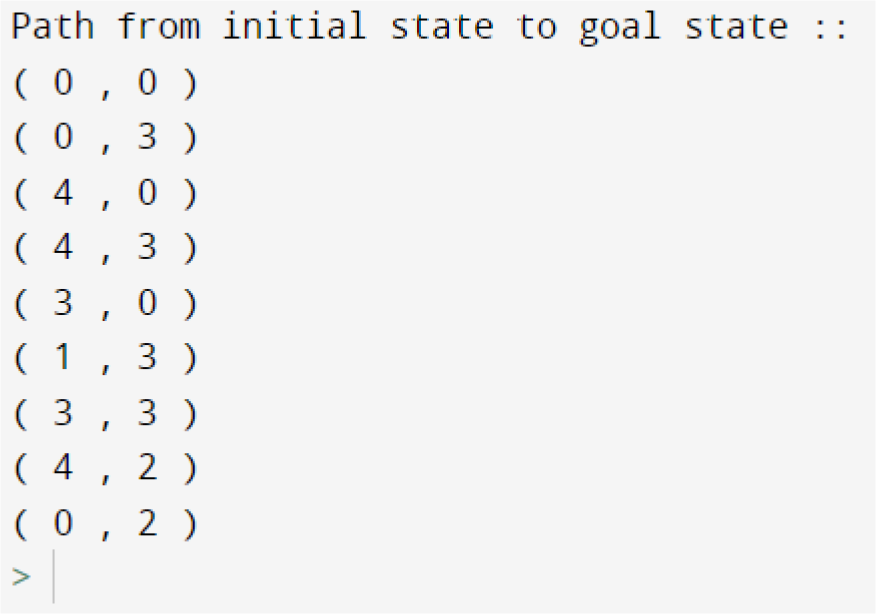
1) Water jug problem:

Code:-

```
from collections import deque

def BFS(r, s, goal):

    m = {}
    isSolvable = False
    path = []



    a = deque()

    a.append((0, 0))

    while (len(a) > 0):


        u = a.popleft()



        if ((u[0], u[1]) in m):
            continue
```

```
if ((u[0] > r or u[1] > s or
     u[0] < 0 or u[1] < 0)):
    continue



path.append([u[0], u[1]])



m[(u[0], u[1])] = 1



if (u[0] == goal or u[1] == goal):
    isSolvable = True


    if (u[0] == goal):
        if (u[1] != 0):


            path.append([u[0], 0])
    else:
        if (u[0] != 0):


            path.append([0, u[1]])


    sz = len(path)
    for i in range(sz):
        print("(", path[i][0], ",",
                    path[i][1], ")")
    break



a.append([u[0], s])
a.append([r, u[1]])

for ap in range(max(r, s) + 1):

    c = u[0] + ap
    d = u[1] - ap



    if (c == r or (d == 0 and d >= 0)):
        a.append([c, d])

    c = u[0] - ap
    d = u[1] + ap



    if ((c == 0 and c >= 0) or d == s):
        a.append([c, d])
        a.append([r, 0])



        a.append([0, s])



    if (not isSolvable):
        print ("No solution")



if __name__ == '__main__':

    Jug1, Jug2, goal = 4, 3, 2
    print("Path from initial state "
          "to goal state ::")

    BFS(Jug1, Jug2, goal)
```

Output:



```
Path from initial state to goal state ::
( 0 , 0 )
( 0 , 3 )
( 4 , 0 )
( 4 , 3 )
( 3 , 0 )
( 1 , 3 )
( 3 , 3 )
( 4 , 2 )
( 0 , 2 )
>
```