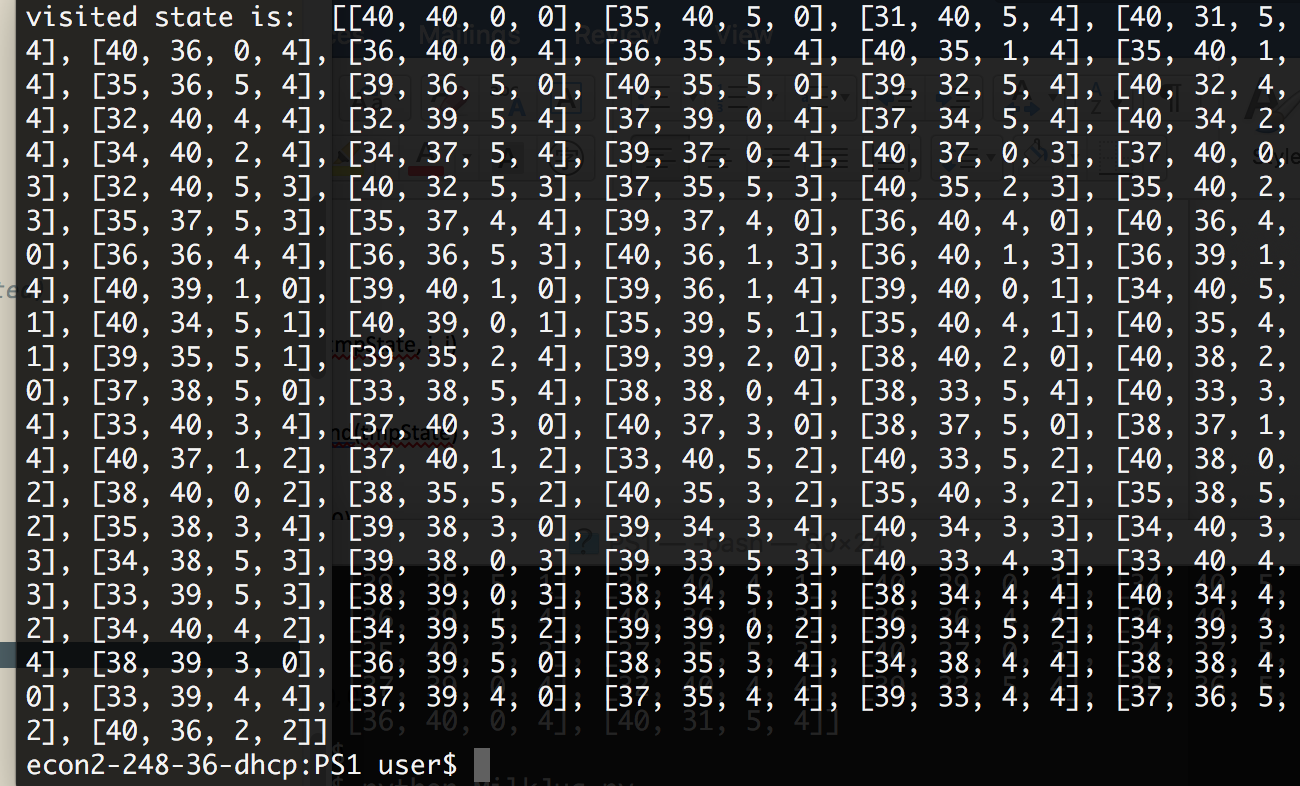
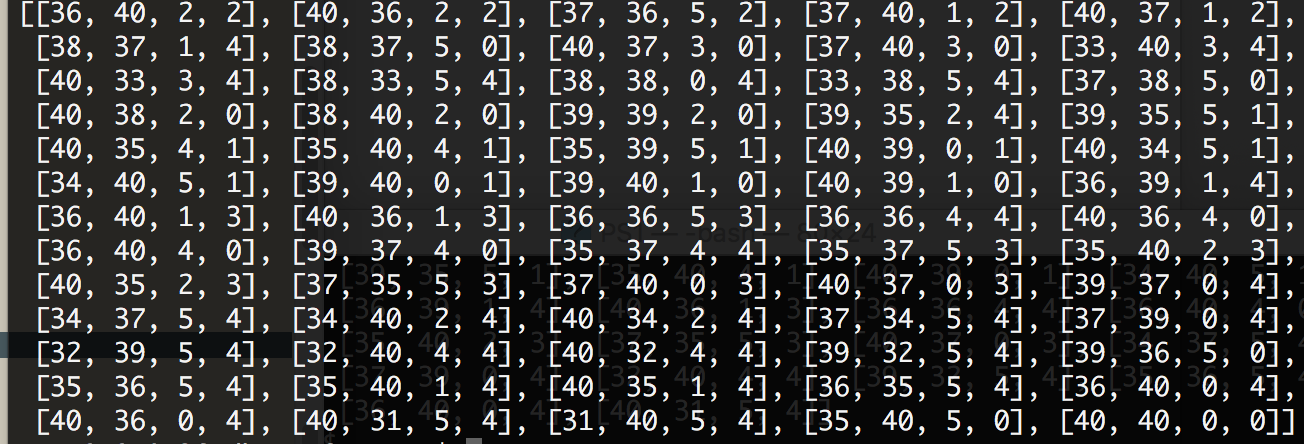
**The total size of the problem space is 105 states**



**The correct solution of total state is 65 states.**



#!/bin/bash python

# 4 milk cans capacity -> (x,y,z,w) where (x = y > z > w)

# initial\_state = (40,40,0,0)

# final state = (40,36,2,2) or (36,40,2,2)

# mark visited state

visited = []

# final solution

solution = []

def getState(state, From, toThere):

global count

a = state[0]

b = state[1]

c = state[2]

d = state[3]

tmpState = list(state)

#print tmpState

if toThere == 0:

Max = Capacity[0]

elif toThere == 1:

Max = Capacity[1]

elif toThere == 2:

Max = Capacity[2]

elif toThere == 3:

Max = Capacity[3]

# Check how much you can pour to

pour\_amount = Max - state[toThere]

if From != toThere:

if tmpState[From] <= pour\_amount:

tmpState[toThere] += tmpState[From]

tmpState[From] = 0

else:

tmpState[From] -= pour\_amount

tmpState[toThere] += pour\_amount

if tmpState in visited:

#print("State had been visited: " , visited)

return False

elif state[2] == 2 and state[3] == 2:

solution.append(tmpState)

return True

else:

visited.append(tmpState)

print visited

for i in range(0, 4):

for j in range(0, 4):

goal = getState(tmpState, i, j)

if goal == True:

solution.append(tmpState)

return True

initail\_state = (40,40,0,0)

Capacity = (40,40,5,4) #a, b, c ,d

numberOfState = 1

print ("Start...\n")

getState(initail\_state, 0, 0)

#print solution

#print solution.reverse()