

Sanjana Shetty

5-E

## LAB 2 - VACUUM WORLD

```
#INSTRUCTIONS
#Enter LOCATION A/B in captial letters
#Enter Status 0/1 accordingly where 0 means CLEAN and 1 means DIRTY

def vacuum_world_2q():

    # initializing goal_state
    # 0 indicates Clean and 1 indicates Dirty
    goal_state = {'A': '0', 'B': '0'}
    cost = 0

    location_input = input("Enter Location of Vacuum ") #user_input of
location vacuum is placed
    status_input = input("Enter status of " + location_input) #user_input
if location is dirty or clean
    other_location = 'B' if location_input == 'A' else 'A'
    status_input_complement = input("Enter status of " + other_location +
" (0 for CLEAN, 1 for DIRTY): ")

    # Initialize status dictionary
    initial_status = {location_input: status_input, other_location:
status_input_complement}
    print("Initial Status: " + str(initial_status))
    print("Initial Location Condition: " + str(goal_state))

    if location_input == 'A':
        # Location A is Dirty.
        print("Vacuum is placed in Location A")
        if status_input == '1':
            print("Location A is Dirty.")
            # suck the dirt and mark it as clean
            goal_state['A'] = '0'
            cost += 1                    #cost for suck
```

```

print("Cost for CLEANING A " + str(cost))
print("Location A has been Cleaned.")

if status_input_complement == '1':
    # if B is Dirty
    print("Location B is Dirty.")
    print("Moving right to the Location B. ")
    cost += 1                                #cost for moving right
    print("COST for moving RIGHT" + str(cost))
    # suck the dirt and mark it as clean
    goal_state['B'] = '0'
    cost += 1                                #cost for suck
    print("COST for SUCK " + str(cost))
    print("Location B has been Cleaned. ")
else:
    print("No action" + str(cost))
    # suck and mark clean
    print("Location B is already clean.")

if status_input == '0':
    print("Location A is already clean ")
    if status_input_complement == '1':# if B is Dirty
        print("Location B is Dirty.")
        print("Moving RIGHT to the Location B. ")
        cost += 1                            #cost for moving right
        print("COST for moving RIGHT " + str(cost))
        # suck the dirt and mark it as clean
        goal_state['B'] = '0'
        cost += 1                            #cost for suck
        print("Cost for SUCK" + str(cost))
        print("Location B has been Cleaned. ")
    else:
        print("No action " + str(cost))
        print(cost)
        # suck and mark clean
        print("Location B is already clean.")

else:
    print("Vacuum is placed in location B")
    # Location B is Dirty.

```

```

if status_input == '1':
    print("Location B is Dirty.")
    # suck the dirt and mark it as clean
    goal_state['B'] = '0'
    cost += 1 # cost for suck
    print("COST for CLEANING " + str(cost))
    print("Location B has been Cleaned.")

if status_input_complement == '1':
    # if A is Dirty
    print("Location A is Dirty.")
    print("Moving LEFT to the Location A. ")
    cost += 1 # cost for moving right
    print("COST for moving LEFT" + str(cost))
    # suck the dirt and mark it as clean
    goal_state['A'] = '0'
    cost += 1 # cost for suck
    print("COST for SUCK " + str(cost))
    print("Location A has been Cleaned.")

else:
    print(cost)
    # suck and mark clean
    print("Location B is already clean.")

if status_input_complement == '1': # if A is Dirty
    print("Location A is Dirty.")
    print("Moving LEFT to the Location A. ")
    cost += 1 # cost for moving right
    print("COST for moving LEFT " + str(cost))
    # suck the dirt and mark it as clean
    goal_state['A'] = '0'
    cost += 1 # cost for suck
    print("Cost for SUCK " + str(cost))
    print("Location A has been Cleaned. ")
else:
    print("No action " + str(cost))
    # suck and mark clean
    print("Location A is already clean.")

```

```
# done cleaning
print("GOAL STATE: ")
print(goal_state)
print("Performance Measurement: " + str(cost))

vacuum_world_2q()
```

## OUTPUT:

1.

```
Enter Location of Vacuum a
Enter status of a1
Enter status of A (0 for CLEAN, 1 for DIRTY): 1
Initial Status: {'a': '1', 'A': '1'}
Initial Location Condition: {'A': '0', 'B': '0'}
Vacuum is placed in location B
Location B is Dirty.
COST for CLEANING 1
Location B has been Cleaned.
Location A is Dirty.
Moving LEFT to the Location A.
COST for moving LEFT2
COST for SUCK 3
Location A has been Cleaned.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 3
```

2.

```
Enter Location of Vacuum B
Enter status of B0
Enter status of A (0 for CLEAN, 1 for DIRTY): 1
Initial Status: {'B': '0', 'A': '1'}
Initial Location Condition: {'A': '0', 'B': '0'}
Vacuum is placed in location B
0
Location B is already clean.
Location A is Dirty.
Moving LEFT to the Location A.
COST for moving LEFT 1
Cost for SUCK 2
Location A has been Cleaned.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 2
```

3.

```
Enter Location of Vacuum B
Enter status of B0
Enter status of A (0 for CLEAN, 1 for DIRTY): 1
Initial Status: {'B': '0', 'A': '1'}
Initial Location Condition: {'A': '0', 'B': '0'}
Vacuum is placed in location B
0
Location B is already clean.
Location A is Dirty.
Moving LEFT to the Location A.
COST for moving LEFT 1
Cost for SUCK 2
Location A has been Cleaned.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 2
```

4.

```
Enter Location of Vacuum B
Enter status of B1
Enter status of A (0 for CLEAN, 1 for DIRTY): 0
Initial Status: {'B': '1', 'A': '0'}
Initial Location Condition: {'A': '0', 'B': '0'}
Vacuum is placed in location B
Location B is Dirty.
COST for CLEANING 1
Location B has been Cleaned.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 1
```