

EXP NO:2 STATE SPACE SEARCH ALGORITHMS

2.1 VACCUUM CLEANER WORLD

Algorithm

1. Start
2. initializing goal_state
3. 0 indicates Clean and 1 indicates Dirty
4. Goal state is defined as $\text{goal_state} = \{ 'A': '0', 'B': '0' \}$ and $\text{cost} = 0$
5. user_input of location vacuum is placed
6. user_input if location is dirty or clean
7. if location_input is 'A', Location ,A is Dirty.
8. suck the dirt and mark it as clean
9. calculate cost.
10. $\text{goal_state}['A'] = '0'$
11. if location_input is 'A', Location ,A is Clean
12. Do Nothing.
13. Moving right to the Location B.
14. cost for moving right
15. If dirty suck the dirt and mark it as clean
16. $\text{goal_state}['B'] = '0'$
17. cost for suck
18. If B is clean Do Nothing
19. Else if location_input is 'B', Location ,B is Dirty.
20. suck the dirt and mark it as clean
21. calculate cost.
22. $\text{goal_state}['B'] = '0'$
23. if location_input is 'B', Location ,B is Clean
24. Do Nothing.
25. Moving Left to the Location A.
26. cost for moving left
27. If dirty suck the dirt and mark it as clean
28. $\text{goal_state}['A'] = '0'$
29. cost for suck
30. If A is clean Do Nothing
31. Print Goal State
32. Print Final Cost as Performance Measurement
33. Stop

PROGRAM

```
def vacuum_world():  
    # 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  
    status_input_complement = input("Enter status of other room")  
    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()

```

RESULT

The above program has been successfully executed and output obtained is verified.

OUTPUT

```

Enter Location of VacuumA
Enter status of A1
Enter status of other room1
Initial Location Condition{'A': '0', 'B': '0'}
Vacuum is placed in Location A
Location A is Dirty.
Cost for CLEANING A 1
Location A has been Cleaned.
Location B is Dirty.
Moving right to the Location B.
COST for moving RIGHT2
COST for SUCK 3

```

Location B has been Cleaned.

GOAL STATE:

{'A': '0', 'B': '0'}

Performance Measurement: 3