robot_vacuum

February 18, 2022

1 Intelligent Agents: Reflex-Based Agents for the Vacuum-cleaner World

1.1 Instructions

Total Points: Undergrads 100 / Graduate students 110

Complete this notebook. Use the provided notebook cells and insert additional code and markdown cells as needed. Submit the completely rendered notebook as a PDF file.

1.2 Introduction

In this assignment you will implement a simulator environment for an automatic vacuum cleaner robot, a set of different reflex-based agent programs, and perform a comparison study for cleaning a single room. Focus on the **cleaning phase** which starts when the robot is activated and ends when the last dirty square in the room has been cleaned. Someone else will take care of the agent program needed to navigate back to the charging station after the room is clean.

1.3 PEAS description of the cleaning phase

Performance Measure: Each action costs 1 energy unit. The performance is measured as the sum of the energy units used to clean the whole room.

Environment: A room with $n \times n$ squares where $n = 5$. Dirt is randomly placed on each square with probability $p = 0.2$. For simplicity, you can assume that the agent knows the size and the layout of the room (i.e., it knows n). To start, the agent is placed on a random square.

Actuators: The agent can clean the current square (action `suck`) or move to an adjacent square by going `north`, `east`, `south`, or `west`.

Sensors: Four bumper sensors, one for north, east, south, and west; a dirt sensor reporting dirt in the current square.

1.4 The agent program for a simple randomized agent

The agent program is a function that gets sensor information (the current percepts) as the arguments. The arguments are:

- A dictionary with boolean entries for the for bumper sensors `north`, `east`, `west`, `south`. E.g., if the agent is on the north-west corner, `bumpers` will be `{"north" : True, "east" : False, "south" : False, "west" : True}`.
- The dirt sensor produces a boolean.

The agent returns the chosen action as a string.

Here is an example implementation for the agent program of a simple randomized agent:

[40]:
```
import numpy as np

actions = ["north", "east", "west", "south", "suck"]

def simple_randomized_agent(bumpers, dirty):
    return np.random.choice(actions)
```

[41]:
```
# define percepts (current location is NW corner and it is dirty)
bumpers = {"north" : True, "east" : False, "south" : False, "west" : True}
dirty = True

# call agent program function with percepts and it returns an action
simple_randomized_agent(bumpers, dirty)
```

[41]:
```
'east'
```

Note: This is not a rational intelligent agent. It ignores its sensors and may bump into a wall repeatedly or not clean a dirty square. You will be asked to implement rational agents below.

1.5 Simple environment example

We implement a simple simulation environment that supplies the agent with its percepts. The simple environment is infinite in size (bumpers are always `False`) and every square is always dirty, even if the agent cleans it. The environment function returns a performance measure which is here the number of cleaned squares (since the room is infinite and all squares are constantly dirty, the agent can never clean the whole room as required in the PEAS description above). The energy budget of the agent is specified as `max_steps`.

[42]:
```
def simple_environment(agent, max_steps, verbose = True):
    num_cleaned = 0

    for i in range(max_steps):
        dirty = True
        bumpers = {"north" : False, "south" : False, "west" : False, "east" :␣
↪False}

        action = agent(bumpers, dirty)
        if (verbose): print("step", i , "- action:", action)
```

```
        if (action == "suck"):
            num_cleaned = num_cleaned + 1

    return num_cleaned
```

Do one simulation run with a simple randomized agent that has enough energy for 20 steps.

simple_environment(simple_randomized_agent, max_steps = 20)

2 Tasks

2.1 General [10 Points]

1. Make sure that you use the latest version of this notebook. Sync your forked repository and pull the latest revision.
2. Your implementation can use libraries like math, numpy, scipy, but not libraries that implement inteligent agents or complete search algorithms. Try to keep the code simple! In this course, we want to learn about the algorithms and we often do not need to use object-oriented design.
3. You notebook needs to be formated professionally.
   - Add additional markdown blocks for your description, comments in the code, add tables and use mathplotlib to produce charts where appropriate
   - Do not show debugging output or include an excessive amount of output.
   - Check that your PDF file is readable. For example, long lines are cut off in the PDF file. You don't have control over page breaks, so do not worry about these.
4. Document your code. Add a short discussion of how your implementation works and your design choices.

2.2 Task 1: Implement a simulation environment [20 Points]

The simple environment above is not very realistic. Your environment simulator needs to follow the PEAS description from above. It needs to:

- Initialize the environment by storing the state of each square (clean/dirty) and making some dirty. (Help with random numbers and arrays in Python)
- Keep track of the agent's position.
- Call the agent function repeatedly and provide the agent function with the sensor inputs.

- React to the agent's actions. E.g, by removing dirt from a square or moving the agent around unless there is a wall in the way.
- Keep track of the performance measure. That is, track the agent's actions until all dirty squares are clean and count the number of actions it takes the agent to complete the task.

The easiest implementation for the environment is to hold an 2-dimensional array to represent if squares are clean or dirty and to call the agent function in a loop until all squares are clean or a predefined number of steps have been reached (i.e., the robot runs out of energy).

The simulation environment should be a function like the `simple_environment()` and needs to work with the simple randomized agent program from above. Use the same environmnt for all your agent implementations in the tasks below.

Note on debugging: Debugging is difficult. Make sure your environment prints enough information when you use `verbose = True`. Also, implementing a function that the environment can use to displays the room with dirt and the current position of the robot at every step is very useful.

[77]:
```
# Your code and description goes here
import random;
def my_environment(agent, size=5, verbose = True):
    # sets the size of the environment
    rows, cols = (size, size)
    num_clean = 0 # keeps track of the number of clean spaces
    start = [random.randint(0, size-1), random.randint(0, size-1)] # randomizes␣
 ↪the starting position of the agent
    curr = start
    num_dirty = 0 # keeps track of the number of dirty spaces
    dirty = False # variable that is passed to the agent
    # creating a 2D array with python lists used this https://www.geeksforgeeks.
 ↪org/python-using-2d-arrays-lists-the-right-way/ for inspiration
    arr=[]
    for i in range(rows):
        col = []
        for j in range(cols):
            if random.random() < 0.2: # randomly chooses squares to make dirty␣
 ↪with probability of 0.2
                col.append('dirty')
                num_dirty += 1
            else:
                col.append('clean')
        arr.append(col)
    if (verbose):
        for row in arr: # prints the environment before being cleaned
            print(row)
    step_count = 0 # used to keep track of total number of actions agent took

    if(verbose):
        print('number of dirty:', num_dirty)
    while num_clean < num_dirty: # while there are still dirty squares in the␣
 ↪environment
        bumpers = {"north" : False, "south" : False, "west" : False, "east" :␣
 ↪False} # defaults bumpers to false every time
        if(arr[curr[0]][curr[1]] == 'dirty'): # check if current square is dirty
            dirty = True
        else:
            dirty = False
        # checks to see if agent is touching a wall and updates bumpers
```

```
        if(curr[0]==0):
            bumpers["north"] = True
        if(curr[0]==size-1):
            bumpers["south"] = True
        if(curr[1]==0):
            bumpers['west'] = True
        if(curr[1]==size-1):
            bumpers['east'] = True

        action = agent(bumpers, dirty)

        if(verbose): # prints the current step, position, if it's dirty, bumpers, and what action it will perform
            print('step', step_count, 'position', curr, 'dirty:', dirty, '\n', 'bumpers:', bumpers, '\n', 'action:', action, '\n')
        step_count += 1
        # takes the action from the agent and checks with current bumpers then either moves the agent or cleans a square and updates the current position
        if(action == 'suck' and arr[curr[0]][curr[1]] == 'dirty'):
            num_clean += 1
            arr[curr[0]][curr[1]] = 'clean'
        if(action == 'north' and bumpers['north'] != True):
            curr[0] = curr[0] - 1
        if(action == 'east' and bumpers['east'] != True):
            curr[1] = curr[1] + 1
        if(action == 'south' and bumpers['south'] != True):
            curr[0] = curr[0] + 1
        if(action == 'west' and bumpers['west'] != True):
            curr[1] = curr[1] - 1

    if(verbose): # prints the 2D array after everything is cleaned to make sure everything is correct
        for row in arr:
            print(row)
    return step_count

# my_environment()
```

2.3 Task 2: Implement a simple reflex agent [10 Points]

The simple reflex agent randomly walks around but reacts to the bumper sensor by not bumping into the wall and to dirt with sucking. Implement the agent program as a function.

Note: Agents cannot directly use variable in the environment. They only gets the percepts as the arguments to the agent function.

[48]:
```
# Your code and description goes here

def simple_reflex_agent(bumpers, dirty):
    # first checks if the square is dirty so it can clean it
    if(dirty):
        return 'suck'
    # lists all the possible actions depending on where the agent is
    actions = ["north", "east", "west", "south"]
    north = ['east', 'west', 'south']
    northeast_corner = ["west", "south"]
    northwest_corner = ['east', 'south']
    east = ['north', 'south', 'west']
    south = ['north', 'east', 'west']
    southeast_corner = ['north', 'west']
    southwest_corner = ['north', 'east']
    west = ['north', 'south', 'east']

    # determines which walls the agent is touching then gives a randomized␣
 ↪direction to go so it doens't bump into a wall
    if(bumpers['north'] == True):
        if(bumpers['south'] == False and bumpers['east'] == False and␣
 ↪bumpers['west'] == False): return random.choice(north)
        if(bumpers['east'] == True and bumpers['west'] == False and␣
 ↪bumpers['south'] == False): return random.choice(northeast_corner)
        if(bumpers['east'] == False and bumpers['west'] == True and␣
 ↪bumpers['south'] == False): return random.choice(northwest_corner)
    elif(bumpers['south'] == True):
        if(bumpers['north'] == False and bumpers['east'] == False and␣
 ↪bumpers['west'] == False): return random.choice(south)
        if(bumpers['north'] == False and bumpers['east'] == True and␣
 ↪bumpers['west'] == False): return random.choice(southeast_corner)
        if(bumpers['north'] == False and bumpers['east'] == False and␣
 ↪bumpers['west'] == True): return random.choice(southwest_corner)
    elif(bumpers['east'] == True):
        if(bumpers['north'] == False and bumpers['west'] == False and␣
 ↪bumpers['south'] == False): return random.choice(east)
    elif(bumpers['west'] == True): return random.choice(west)
    else: return random.choice(actions)

print(my_environment(simple_reflex_agent))
```

```
['dirty', 'clean', 'clean', 'clean', 'clean']
['dirty', 'clean', 'clean', 'clean', 'clean']
['dirty', 'clean', 'clean', 'clean', 'clean']
['clean', 'clean', 'dirty', 'clean', 'clean']
['clean', 'dirty', 'clean', 'clean', 'clean']
number of dirty: 5
```

```
step 0 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 1 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west

step 2 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 3 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 4 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west

step 5 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 6 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 7 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 8 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 9 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 10 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 11 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north
```

```
step 12 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 13 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 14 position [3, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 15 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 16 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 17 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 18 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 19 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 20 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 21 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 22 position [4, 3] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: east

step 23 position [4, 4] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': True}
 action: north
```

```
step 24 position [3, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 25 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 26 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 27 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 28 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 29 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 30 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 31 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 32 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 33 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 34 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 35 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south
```

```
step 36 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 37 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 38 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 39 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 40 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 41 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 42 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 43 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 44 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 45 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 46 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 47 position [2, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east
```

```
step 48 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 49 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 50 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 51 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 52 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 53 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 54 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 55 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 56 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 57 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 58 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 59 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north
```

```
step 60 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 61 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 62 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 63 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 64 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 65 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 66 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 67 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 68 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west

step 69 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 70 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 71 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west
```

```
step 72 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 73 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west

step 74 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 75 position [1, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 76 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 77 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west

step 78 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 79 position [1, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 80 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 81 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 82 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 83 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west
```

```
step 84 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 85 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 86 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 87 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 88 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 89 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 90 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 91 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 92 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 93 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 94 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 95 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north
```

```
step 96 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 97 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 98 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 99 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 100 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 101 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 102 position [3, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 103 position [4, 4] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': True}
 action: north

step 104 position [3, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 105 position [4, 4] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': True}
 action: west

step 106 position [4, 3] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: east

step 107 position [4, 4] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': True}
 action: north
```

```
step 108 position [3, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 109 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 110 position [3, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 111 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 112 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 113 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 114 position [3, 2] dirty: True
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: suck

step 115 position [3, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 116 position [3, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 117 position [3, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 118 position [4, 2] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: east

step 119 position [4, 3] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: east
```

```
step 120 position [4, 4] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': True}
 action: north

step 121 position [3, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 122 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 123 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 124 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 125 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 126 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 127 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 128 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 129 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 130 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 131 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west
```

```
step 132 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 133 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 134 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 135 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 136 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 137 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 138 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 139 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 140 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 141 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 142 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 143 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east
```

```
step 144 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 145 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west

step 146 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 147 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 148 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 149 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 150 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 151 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 152 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 153 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 154 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 155 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east
```

```
step 156 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 157 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 158 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 159 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 160 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 161 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 162 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 163 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 164 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 165 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 166 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 167 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east
```

```
step 168 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 169 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 170 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 171 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 172 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 173 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 174 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 175 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 176 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 177 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 178 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 179 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east
```

```
step 180 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 181 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 182 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 183 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 184 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 185 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 186 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 187 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 188 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 189 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west

step 190 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west

step 191 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west
```

```
step 192 position [0, 0] dirty: True
 bumpers: {'north': True, 'south': False, 'west': True, 'east': False}
 action: suck

step 193 position [0, 0] dirty: False
 bumpers: {'north': True, 'south': False, 'west': True, 'east': False}
 action: south

step 194 position [1, 0] dirty: True
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: suck

step 195 position [1, 0] dirty: False
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: north

step 196 position [0, 0] dirty: False
 bumpers: {'north': True, 'south': False, 'west': True, 'east': False}
 action: south

step 197 position [1, 0] dirty: False
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: east

step 198 position [1, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 199 position [2, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 200 position [1, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 201 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 202 position [1, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 203 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west
```

```
step 204 position [0, 0] dirty: False
 bumpers: {'north': True, 'south': False, 'west': True, 'east': False}
 action: south

step 205 position [1, 0] dirty: False
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: north

step 206 position [0, 0] dirty: False
 bumpers: {'north': True, 'south': False, 'west': True, 'east': False}
 action: east

step 207 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west

step 208 position [0, 0] dirty: False
 bumpers: {'north': True, 'south': False, 'west': True, 'east': False}
 action: east

step 209 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 210 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 211 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 212 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 213 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 214 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 215 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south
```

```
step 216 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 217 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 218 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: south

step 219 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 220 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 221 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 222 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 223 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 224 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 225 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 226 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 227 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south
```

```
step 228 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 229 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 230 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 231 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 232 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 233 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 234 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 235 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 236 position [1, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 237 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west

step 238 position [0, 0] dirty: False
 bumpers: {'north': True, 'south': False, 'west': True, 'east': False}
 action: east

step 239 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south
```

```
step 240 position [1, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 241 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 242 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 243 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 244 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 245 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 246 position [2, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 247 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 248 position [2, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 249 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 250 position [2, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 251 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west
```

```
step 252 position [2, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 253 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 254 position [2, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 255 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 256 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: west

step 257 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 258 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 259 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 260 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 261 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 262 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 263 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east
```

```
step 264 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 265 position [3, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: south

step 266 position [4, 4] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': True}
 action: west

step 267 position [4, 3] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: west

step 268 position [4, 2] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: west

step 269 position [4, 1] dirty: True
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: suck

step 270 position [4, 1] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: north

step 271 position [3, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 272 position [4, 1] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: north

step 273 position [3, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 274 position [3, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 275 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south
```

```
step 276 position [4, 3] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: north

step 277 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 278 position [3, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: west

step 279 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 280 position [3, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 281 position [2, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 282 position [1, 4] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': True}
 action: north

step 283 position [0, 4] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': True}
 action: west

step 284 position [0, 3] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 285 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 286 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 287 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west
```

```
step 288 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 289 position [2, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 290 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 291 position [1, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 292 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 293 position [1, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 294 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 295 position [2, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: east

step 296 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 297 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 298 position [3, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 299 position [4, 2] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: north
```

```
step 300 position [3, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 301 position [2, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 302 position [2, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 303 position [3, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: west

step 304 position [3, 0] dirty: False
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: north

step 305 position [2, 0] dirty: True
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: suck

['clean', 'clean', 'clean', 'clean', 'clean']
['clean', 'clean', 'clean', 'clean', 'clean']
['clean', 'clean', 'clean', 'clean', 'clean']
['clean', 'clean', 'clean', 'clean', 'clean']
['clean', 'clean', 'clean', 'clean', 'clean']
306
```

2.4 Task 3: Implement a model-based reflex agent [20 Points]

Model-based agents use a state to keep track of what they have done and perceived so far. Your agent needs to find out where it is located and then keep track of its current location. You also need a set of rules based on the state and the percepts to make sure that the agent will clean the whole room. For example, the agent can move to a corner to determine its location and then it can navigate through the whole room and clean dirty squares.

Describe how you define the **agent state** and how your agent works before implementing it. (Help with implementing state information on Python)

3 Your short description of the state and your implementation goes here

I designed my agent to have three states. One keeps track if the agent is in the starting position or not. The other two states keep track if the robot is currently cleaning down or up the room. The way I designed the agent is it starts in the north-west corner of the room then completes a snake to sweep the whole room by going all the way south then moving east one square then all the north and repeating until all the squares are clean.

[70]:
```
# Your code goes here
class Agent:
    def __init__(self, room_size=0, name = "An Agent", bumpers = {"north" :␣
↪False, "south" : False, "west" : False, "east" : False}):
        self.room_size = room_size # keeps track of the size of the room but␣
↪NOT NEEDED
        self.name = name # name of the agent also NOT NEEDED
        self.bumpers = bumpers # keeps track of the current bumpers
        self.start_pos = False # state to determine if the agent is in the␣
↪starting position, which is the top left corner of the room
        self.go_down = True    # state to determine if the agent needs to go up␣
↪and clean
        self.go_up   = False   # state to determine if the agent needs to go␣
↪down and clean

    # make start find corner state and cleaning state
    # keep states as class variables
    def act(self, bumpers, dirty):
        self.bumpers = bumpers
        if(self.start_pos == False): # while agent is not in starting position,␣
↪it goes all the way to the top then all the way left
            if(self.bumpers['north'] == True and self.bumpers['west'] == True):␣
↪# checks if in top left corner then sets the start_pos state to true
                self.start_pos = True
                if(dirty == True): return 'suck'
                else: return 'south'
            if(self.bumpers['north'] == False): return 'north'
            if(self.bumpers['west'] == False): return 'west'

        else:
            if(self.go_down): # while go_down state is active it goes all the␣
↪way to the bottom of room while cleaning any dirty squares
                if(dirty): return 'suck'
                if(self.bumpers['south'] == False): return 'south'
                if(self.bumpers['south']): # when it hits the bottom it turns␣
↪on the go_up state and turns off the go_down state
```

```
                    self.go_up = True      # then it moves one square to the␣
 ↪right
                    self.go_down = False
                    return 'east'
            elif(self.go_up): # while the go_up state is active it goes all the␣
 ↪wayto the top of teh room while cleaning any dirty squares
                if(dirty): return 'suck'
                if(self.bumpers['north'] == False): return 'north'
                if(self.bumpers['north']): # when it hits the bottom it turns␣
 ↪on the go_down state and turns off the go_up state then moves one square to␣
 ↪the right
                    self.go_down = True
                    self.go_up   = False
                    return 'east'

agent1 = Agent(room_size = 5, name = 'agent1')
my_environment(agent1.act)
```

```
['clean', 'dirty', 'clean', 'dirty', 'clean']
['clean', 'clean', 'clean', 'clean', 'clean']
['dirty', 'clean', 'clean', 'clean', 'clean']
['clean', 'clean', 'clean', 'clean', 'clean']
['clean', 'clean', 'clean', 'clean', 'clean']
number of dirty: 3
step 0 position [3, 0] dirty: False
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: north

step 1 position [2, 0] dirty: True
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: north

step 2 position [1, 0] dirty: False
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: north

step 3 position [0, 0] dirty: False
 bumpers: {'north': True, 'south': False, 'west': True, 'east': False}
 action: south

step 4 position [1, 0] dirty: False
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: south

step 5 position [2, 0] dirty: True
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: suck
```

```
step 6 position [2, 0] dirty: False
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: south

step 7 position [3, 0] dirty: False
 bumpers: {'north': False, 'south': False, 'west': True, 'east': False}
 action: south

step 8 position [4, 0] dirty: False
 bumpers: {'north': False, 'south': True, 'west': True, 'east': False}
 action: east

step 9 position [4, 1] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: north

step 10 position [3, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 11 position [2, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 12 position [1, 1] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 13 position [0, 1] dirty: True
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: suck

step 14 position [0, 1] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: east

step 15 position [0, 2] dirty: False
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: south

step 16 position [1, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 17 position [2, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south
```

```
step 18 position [3, 2] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: south

step 19 position [4, 2] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: east

step 20 position [4, 3] dirty: False
 bumpers: {'north': False, 'south': True, 'west': False, 'east': False}
 action: north

step 21 position [3, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 22 position [2, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 23 position [1, 3] dirty: False
 bumpers: {'north': False, 'south': False, 'west': False, 'east': False}
 action: north

step 24 position [0, 3] dirty: True
 bumpers: {'north': True, 'south': False, 'west': False, 'east': False}
 action: suck

['clean', 'clean', 'clean', 'clean', 'clean']
['clean', 'clean', 'clean', 'clean', 'clean']
['clean', 'clean', 'clean', 'clean', 'clean']
['clean', 'clean', 'clean', 'clean', 'clean']
['clean', 'clean', 'clean', 'clean', 'clean']
```

[70]: 25

3.1 Task 4: Simulation study [30 Points]

Compare the performance (the performance measure is defined in the PEAS description above) of the agents using environments of different size. E.g., 5×5, 10×10 and 100×100. Use 100 random runs for each. Present the results using tables and graphs. Discuss the differences between the agents. (Help with charts and tables in Python)

[102]:
```python
# Your code goes here
import pandas as pd
sizes = [5, 10, 100]
```

```
# dataframe to store all the data
df = pd.DataFrame(columns=['Size','Model based reflex','simple␣
 ↪reflex','random'])
for i in sizes:
    for x in range(100):
        agent1 = Agent(room_size = i, name = 'agent1')

        model_steps = my_environment(agent1.act, i, False)

        simple_steps = my_environment(simple_reflex_agent, i, False)
        random_steps = my_environment(simple_randomized_agent, i, False)
        #adds new row to dataframe
        df.loc[len(df.index)] = [i, model_steps, simple_steps, random_steps]

df
```

[102]:

	Size	Model based reflex	simple reflex	random
0	5	25	104	279
1	5	33	276	248
2	5	33	66	393
3	5	35	54	212
4	5	36	90	196
..	…	…	…	…
295	100	12103	328914	788475
296	100	12074	409461	862742
297	100	12015	331214	732434
298	100	12118	270034	761444
299	100	12140	388735	802286

[300 rows x 4 columns]

[258]:
```
df.groupby('Size').mean()
```

[258]:

Size	Model based reflex	simple reflex	random
5	28.45	95.13	395.90
10	122.47	930.27	3001.36
100	12096.09	354561.17	880405.42

Fill out the following table with the average performance measure for 100 random runs (you may also create this table with code):

Size	Randomized Agent	Simple Reflex Agent	Model-based Reflex Agent
5x5			
10x10			
100x100			

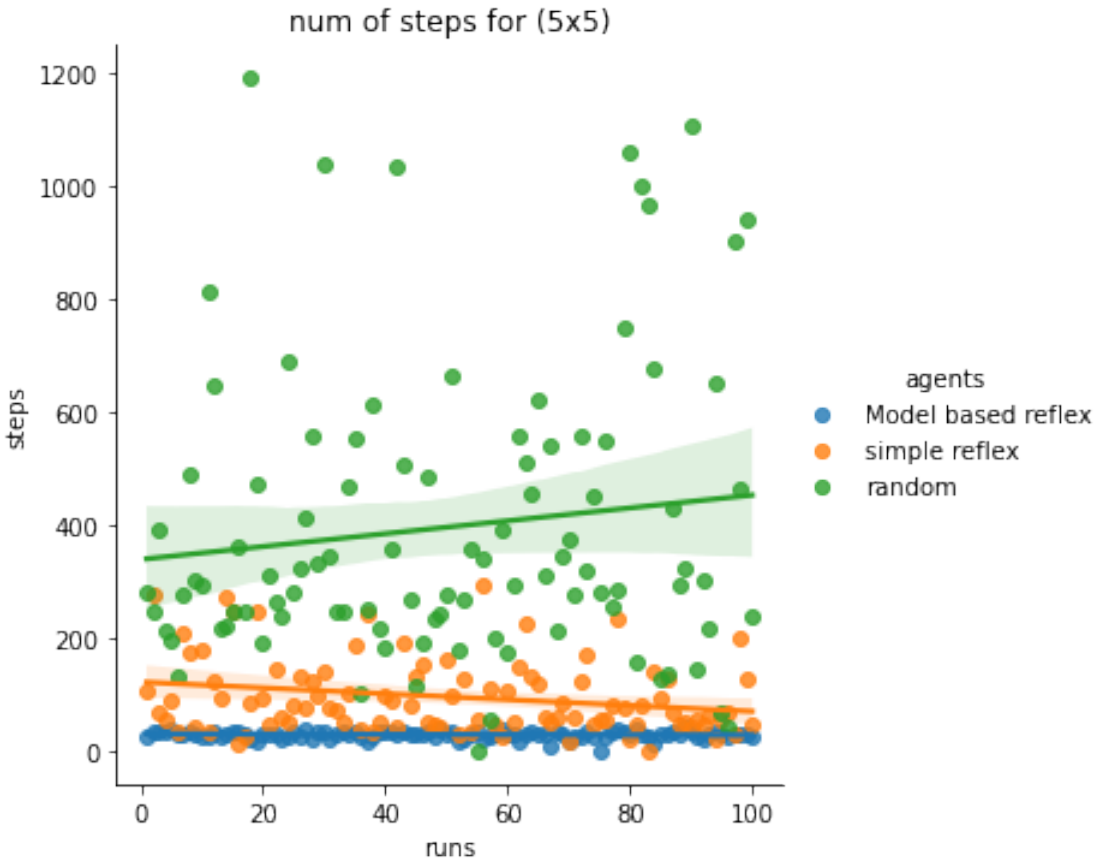
Add charts to compare the performance of the different agents.

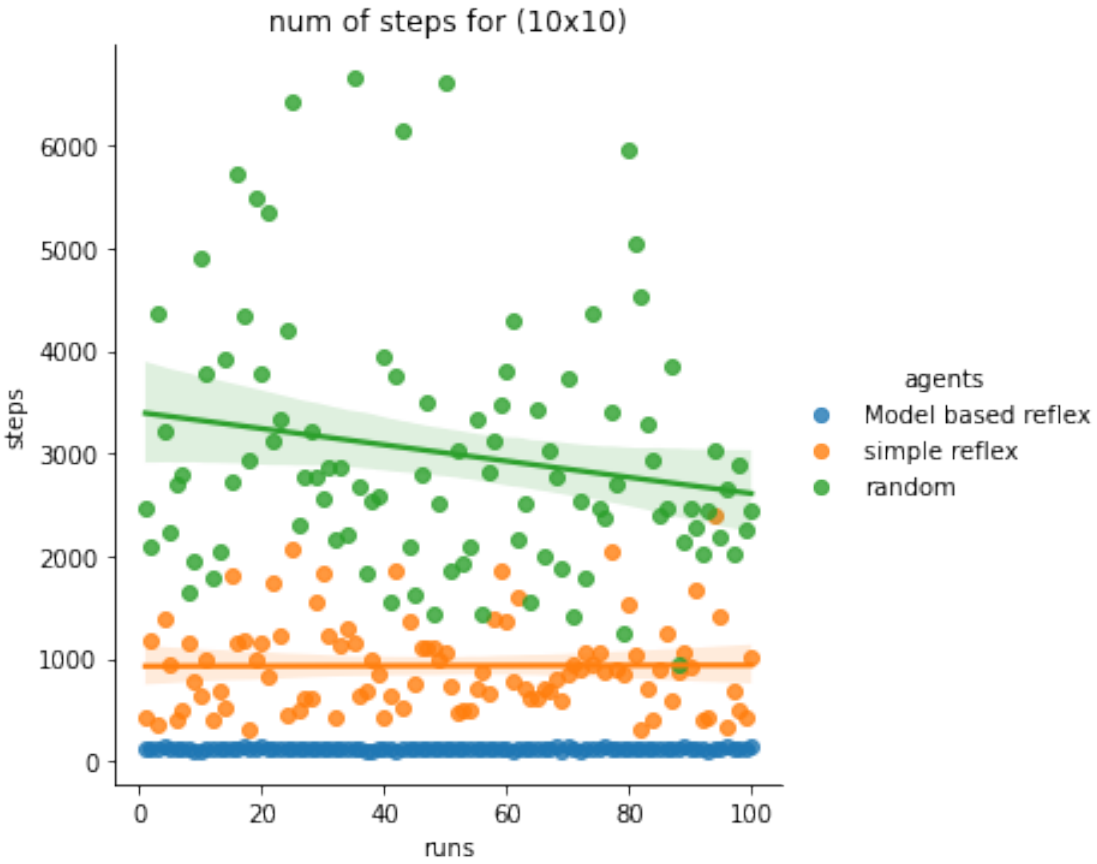
[257]:
```
# Your graphs and discussion of the results goes here
import matplotlib.pyplot as plt
import seaborn as sns
avgs = df.groupby(['Size']).mean()
size5 = df[0:100]
size10 = df[100:200]
size100 = df[200:300]
# scatter plot for 5x5 room
size5_m = size5.melt('Size', var_name='agents', value_name='steps')
size5_m = size5_m.drop('Size', 1)
arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,␣
 ↪21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,␣
 ↪40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58,␣
 ↪59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77,␣
 ↪78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96,␣
 ↪97, 98, 99, 100, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,␣
 ↪18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,␣
 ↪37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55,␣
 ↪56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74,␣
 ↪75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93,␣
 ↪94, 95, 96, 97, 98, 99, 100, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,␣
 ↪15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,␣
 ↪34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52,␣
 ↪53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71,␣
 ↪72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90,␣
 ↪91, 92, 93, 94, 95, 96, 97, 98, 99, 100]
size5_m['runs']= arr
ax5 = sns.lmplot(x='runs', y='steps', hue='agents', data=size5_m)
ax5 = plt.gca()
ax5.set_title('num of steps for (5x5)')
# scatter plot for 10x10 room
size10m = size10.melt('Size', var_name='agents', value_name='steps')
size10m = size10m.drop('Size', 1)
size10m['runs']= arr
sns.lmplot(x='runs', y='steps', hue='agents', data=size10m)
ax10 = plt.gca()
ax10.set_title('num of steps for (10x10)')
# scatter plot for 100x100 room
size100m = size100.melt('Size', var_name='agents', value_name='steps')
size100m = size100m.drop('Size', 1)
size100m['runs']= arr
sns.lmplot(x='runs', y='steps', hue='agents', data=size100m)
ax100 = plt.gca()
ax100.set_title('num of steps for (100x100)')
```

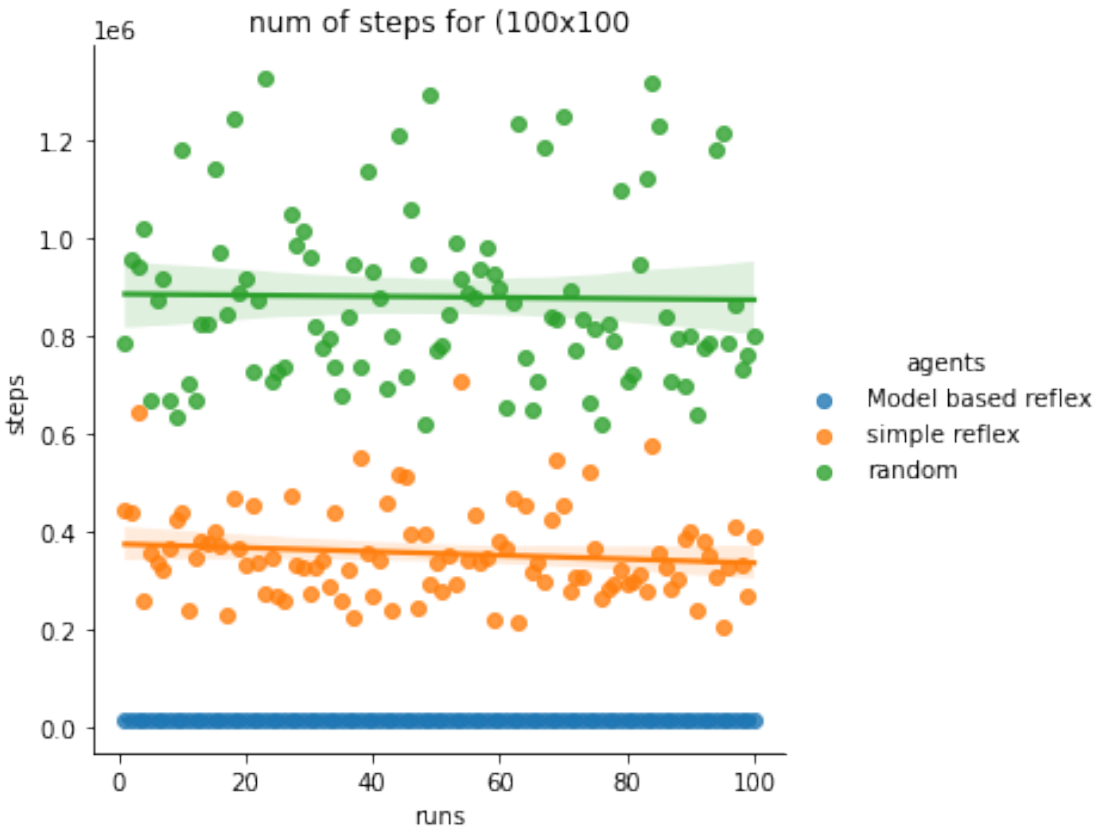
```
# bar graph cmoparing the timing of each agent at each size with log scale for␣
 ↪y axis
avgs.plot(kind='bar', stacked=False,logy= True)

# Looking at my bar graph it is obviously the model based reflex agent is the␣
 ↪quickest in all size rooms, then the simple reflex agent, and finally the␣
 ↪random agent took the longest
# in every room. I had to use a logarithmic scale on the y axis in order to␣
 ↪show the smaller data sets.
# If you look at the three scatter plots you can see that the distribution for␣
 ↪the random agent is much more spread out than the simple and model based␣
 ↪reflex agent. The model based
# agent is the most consistent meaning everytime you run it you will get a␣
 ↪result in a narrow range giving it a high accuracy.
```

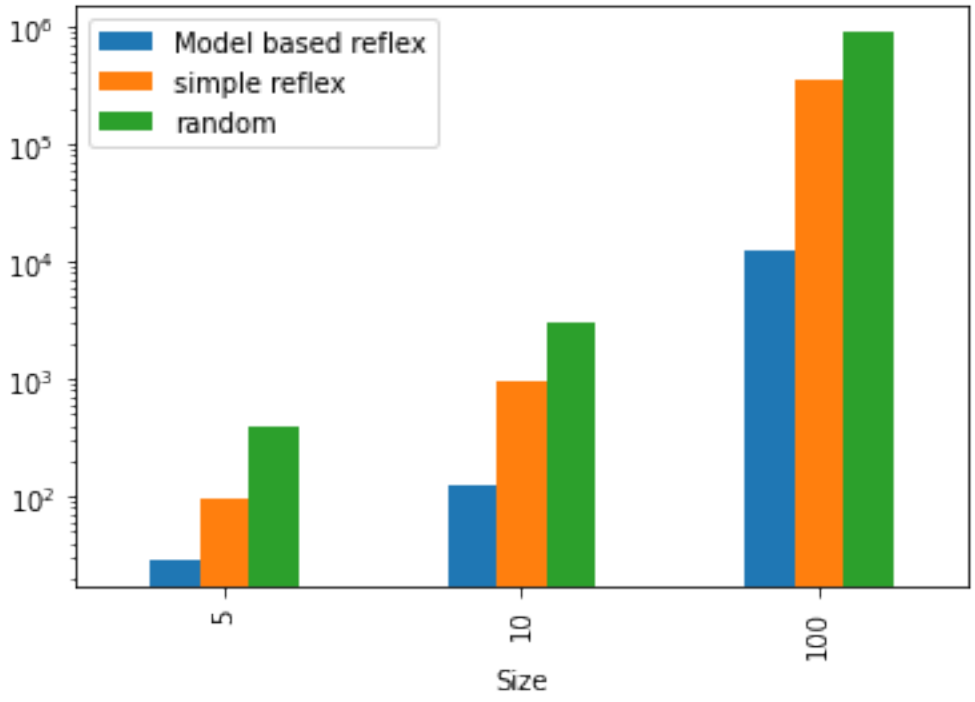
```
/var/folders/6j/zq70vzjs0ggcjv4q8z1llhmm0000gn/T/ipykernel_73908/1142797608.py:1
0: FutureWarning: In a future version of pandas all arguments of DataFrame.drop
except for the argument 'labels' will be keyword-only.
  size5_m = size5_m.drop('Size', 1)
/var/folders/6j/zq70vzjs0ggcjv4q8z1llhmm0000gn/T/ipykernel_73908/1142797608.py:1
8: FutureWarning: In a future version of pandas all arguments of DataFrame.drop
except for the argument 'labels' will be keyword-only.
  size10m = size10m.drop('Size', 1)
/var/folders/6j/zq70vzjs0ggcjv4q8z1llhmm0000gn/T/ipykernel_73908/1142797608.py:2
5: FutureWarning: In a future version of pandas all arguments of DataFrame.drop
except for the argument 'labels' will be keyword-only.
  size100m = size100m.drop('Size', 1)
```

[257]: <AxesSubplot:xlabel='Size'>


num of steps for (5x5)
steps
runs
0
200
400
600
800
1000
1200
0
20
40
60
80
100
agents
Model based reflex
simple reflex
random


num of steps for (10x10)
steps
runs
0
1000
2000
3000
4000
5000
6000
0
20
40
60
80
100
agents
Model based reflex
simple reflex
random


num of steps for (100x100
1e6
steps
runs
agents
Model based reflex
simple reflex
random


Model based reflex
simple reflex
random
Size
5
10
100

3.2 Task 5: Robustness of the agent implementations [10 Points]

Describe how your agent implementations will perform

- if it is put into a rectangular room with unknown size,
- if the cleaning area can have an iregular shape (e.g., a hallway connecting two rooms), or
- if the room contains obstacles (i.e., squares that it cannot pass through and trigger the bumper sensors).

4 Answer goes here

If it was put into a rectanglular room my agent will only clean a square in the rectangle and not the entire rectangle. With a couple simple changes to my environment and by taking in one more parameter I can easily make it work with a rectangular room.

My agent will not perform well with an irregular shape room as the environemnt is only designed for square rooms.

If the the room contained obstacles my agent would get stuck on the obstacle as it is not designed to have to go around obstacles.

4.1 Graduate student advanced task: Obstacles [10 Points]

Undergraduate students: This is a bonus task you can attempt if you like [+5 Bonus Points].

1. Change your simulation environment tor run experiments for the following problem: Add random obstacle squares that also trigger the bumper sensor. The agent does not know where the obstacles are. Observe how this changes the performance of the three implementations.
2. Describe what would need to be done to perform better with obstacles. Add code if you can.

[12]:
```
# Your code and discussion goes here
```

4.2 More advanced implementation tasks

- **Agent for and environment with obstacles:** Implement an agent for an environment where the agent does not know how large the environment is (we assume it is rectangular), where it starts or where the obstacles are. An option would be to always move to the closest unchecked/uncleaned square (note that this is actualy depth-first search).
- **Utility-based agent:** Change the environment for a 5×5 room, so each square has a fixed probability of getting dirty again. For the implementation, we give the environment a 2-dimensional array of probabilities. The utility of a state is defined as the number of currebntly clean squares in the room. Implement a utility-based agent that maximizes the

expected utility over one full charge which lasts for 100000 time steps. To do this, the agent needs to learn the probabilities with which different squares get dirty again. This is very tricky!

[13]:
```
# Your ideas/code
```