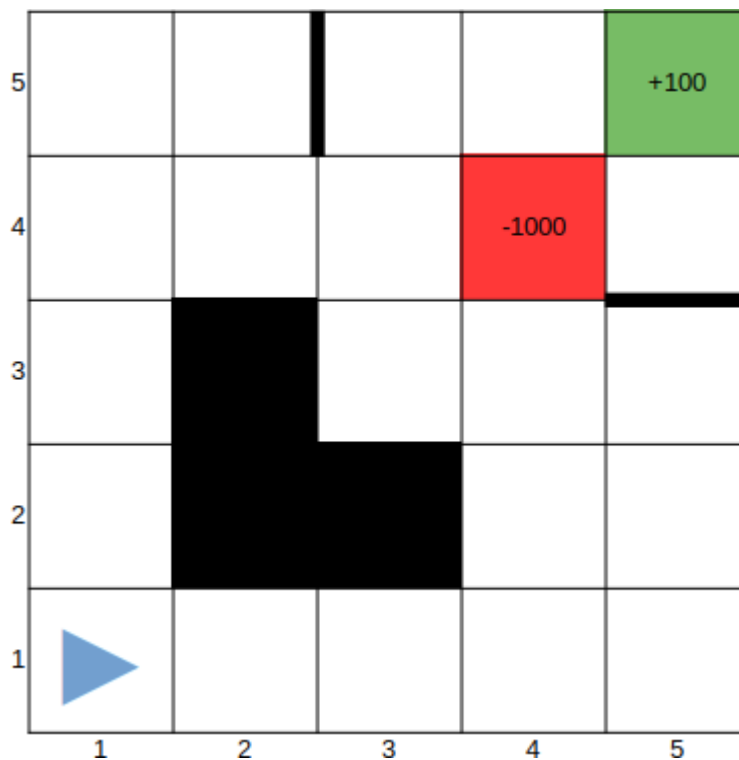


# CPSC 4420/6420: ARTIFICIAL INTELLIGENCE

## ASSIGNMENT 2, NAME:

PARAMPREET SINGH

Consider the following puzzle. The green and red states are both terminal states, with the rewards as shown (so we can consider the green state the “goal”, and the red a “game over” state with a large negative reward). Thick borders between cells represent walls that the robot cannot cross, and the black squares contain obstacles and cannot be entered. The robot player is represented by the blue triangle, and the direction the triangle points is the way the robot is facing.



Let's represent a state with  $(x,y,d)$ , where  $x$  and  $y$  represent the horizontal and vertical positions (i.e. location), and  $d$  represents the direction the robot is facing (1: up, 2: down, 3: left, and 4: right).

The robot can take the following actions:

- $A_1$ : Move one cell forward in the direction it is facing. Cost: 1.5
- $A_2$ : Move two cells forward in the direction it is facing. Cost: 2
- $A_3$ : Turn to its left, and stay in the same cell. Cost: 0.5
- $A_4$ : Turn to its right, and stay in the same cell. Cost: 0.5

Note that each action has a different cost value. This can also be considered an immediate negative reward. For example, we have  $R(s,A_1,s') = -1.5$ . The cost is evaluated on the current state, (the state the robot is in when it begins the action, not the one it lands on after performing the action). In the same way, the value of state  $V(s)$  represents the value of the current state and

you should initialize the algorithm with  $V_1(5,5,x)=+100$ ,  $V_1(4,4,x)=-1000$  (for  $x=1,2,3,4$  representing the robot orientation/direction), and zero for all other states.

So, for example, if the robot is in state  $(4,1,4)$ , it means that it is in location  $(4,1)$  and facing right. The result of possible actions for this state are as follows:

$A_1$  (move 1 cell forward) -->  $(5,1,4)$

$A_2$  (move 2 cells forward) --> impossible remains in the current state  $(4,1,4)$

$A_3$  (turn left) -->  $(4,1,1)$  : the robot stays in  $(4,1)$  but now faces up

$A_4$  (turn right) -->  $(4,1,2)$  : the robot stays in  $(4,1)$  but now faces down

A move is impossible if it would result in landing on a blocked cell, like  $(2,2)$ ,  $(2,3)$ , or  $(3,2)$ , or if it would result in crossing a barrier, like moving from state  $(2,5)$  to  $(3,5)$ , or  $(5,3)$  to  $(5,4)$ . A move that would take the robot outside of our 5x5 grid is also impossible.

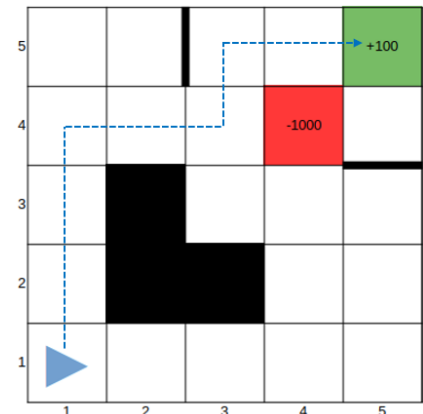
Note that we have more states than the number of cells, because the robot facing a different direction produces a new state, even if it does not change location. In the example above, if we move to  $(4,1,1)$ , where the robot is facing up, this is a different state from the one we were in,  $(4,1,4)$ , even though the robot has not moved cells.

A) If there is no living reward/penalty, no noise, and no discount ( $\gamma = 1$ ), use your common sense to find the best possible route from  $(1,1)$  to  $(5,5)$ .

ANSWER – With no living reward/penalty, no noise, and no discount, the best possible route is the one costing least.

[State  $(1,1,4)$ , Action:  $A_3$ ] → State  $(1,1,1)$  Action Cost = 0.5  
 [State  $(1,1,1)$ , Action:  $A_1$ ] → State  $(1,2,1)$  Action Cost = 1.5  
 [State  $(1,2,1)$ , Action:  $A_2$ ] → State  $(1,4,1)$  Action Cost = 2.0  
 [State  $(1,4,1)$ , Action:  $A_4$ ] → State  $(1,4,4)$  Action Cost = 0.5  
 [State  $(1,4,4)$ , Action:  $A_2$ ] → State  $(3,4,4)$  Action Cost = 2.0  
 [State  $(3,4,4)$ , Action:  $A_3$ ] → State  $(3,4,1)$  Action Cost = 0.5  
 [State  $(3,4,1)$ , Action:  $A_1$ ] → State  $(3,5,1)$  Action Cost = 1.5  
 [State  $(3,5,1)$ , Action:  $A_4$ ] → State  $(3,5,4)$  Action Cost = 0.5  
 [State  $(3,5,4)$ , Action:  $A_2$ ] → State  $(5,5,4)$  Action Cost = 2.0

TOTAL COST = 11



B) With no discount ( $\gamma = 1$ ), no living reward, and no noise, use the Value Iteration Algorithm with 100 iterations to update the optimal values for each state and print the result [only for the first 10 iterations] in the following format:

iter 1:

state  $(1,1,1)$   $V =$  (some value)

Best Action:  $A_i$  (where  $i$  is some number 1-4)

state  $(1,1,2)$   $V =$  (some value)

Best Action:  $A_j$

...

state  $(5,5,4)$   $V =$  (some value)

iter 2:

state  $(1,1,1)$   $V =$  (some value)

Best Action:  $A_i$  (where  $i$  is some number 1-4)

state  $(1,1,2)$   $V =$  (some value)

Best Action:  $A_j$

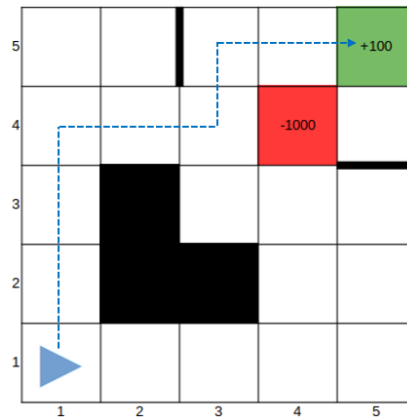
...

state  $(5,5,4)$   $V =$  (some value)

If two actions are tied for best, you can select one at random or always choose the one with the smallest index.

ANSWER – For iterations, please refer python file – HW2\_B.py(may also refer – ITERATIONS(B))  
Resulting path –

$[(1, 1, 4), A3] \rightarrow [(1, 1, 1), A1] \rightarrow [(1, 2, 1), A2] \rightarrow [(1, 4, 1), A4] \rightarrow [(1, 4, 4), A2] \rightarrow [(3, 4, 4), A3] \rightarrow [(3, 4, 1), A1] \rightarrow [(3, 5, 1), A4] \rightarrow [(3, 5, 4), A2] \rightarrow (5, 5, 4)]$



- C) If you start from state (1,1,4) and follow the optimal policy you found in part B, does it follow the same path you proposed in part A?

ANSWER – From ANSWER A and ANSWER B,

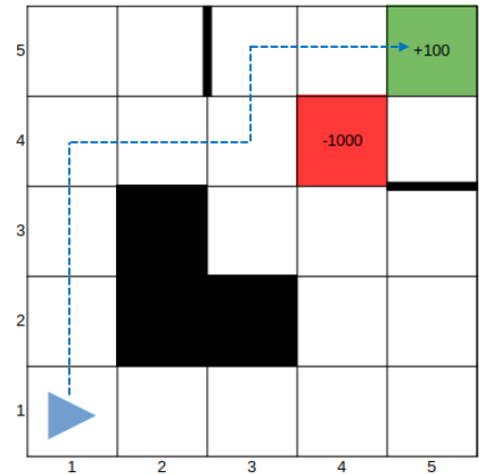
**Yes**, starting from state (1,1,4), part A and part B follow the same path.

- D) Repeat part B with the same assumptions, except for gamma = 0.8 (discount factor). Compare the results with that from part B. Do they match?

ANSWER – From ANSWER B,

Resulting path of part B is –

$[(1, 1, 4), A3] \rightarrow [(1, 1, 1), A1] \rightarrow [(1, 2, 1), A2] \rightarrow [(1, 4, 1), A4] \rightarrow [(1, 4, 4), A2] \rightarrow [(3, 4, 4), A3] \rightarrow [(3, 4, 1), A1] \rightarrow [(3, 5, 1), A4] \rightarrow [(3, 5, 4), A2] \rightarrow (5, 5, 4)]$

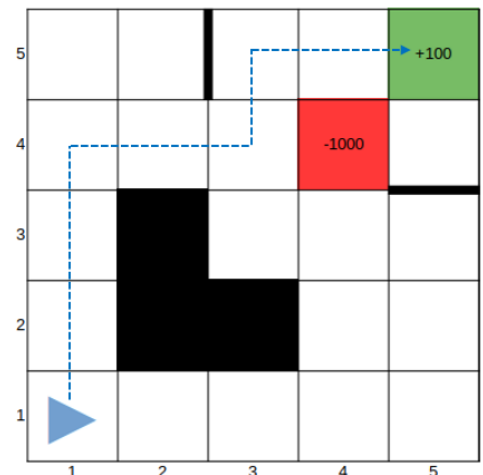


When gamma = 0.8 (discount factor),

[Please refer python file – HW2\_D.py (may also refer – ITERATIONS(D))]

Resulting path –

$[(1, 1, 4), A3] \rightarrow [(1, 1, 1), A1] \rightarrow [(1, 2, 1), A2] \rightarrow [(1, 4, 1), A4] \rightarrow [(1, 4, 4), A2] \rightarrow [(3, 4, 4), A3] \rightarrow [(3, 4, 1), A1] \rightarrow [(3, 5, 1), A4] \rightarrow [(3, 5, 4), A2] \rightarrow (5, 5, 4)]$



Therefore, for similar assumptions except gamma = 1 (part B) and gamma = 0.8 (part D), the resulting paths for both the cases are **SAME**.

- E) Repeat part B with the same assumptions, except for  $\gamma = 0.2$ . Compare the results with that from parts B and D. Do they match?

ANSWER – The value of a state is calculated as -

$$Q^*(s, a) = \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma V^*(s')]$$

$$V^*(s) = \max_a Q^*(s, a)$$

$$V^*(s) = \max_a \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma V^*(s')]$$

Therefore, for  $\gamma = 0.2$ ,  $P = 1$  and  $i = 100$  iterations, value of a state largely depends upon the reward of the action. In our case, as reward is negative, the best action for a state would be the one that gives minimum negative reward. The rewards for various actions are as follows –

$A_1$ : - 1.5;  $A_2$ : - 2;  $A_3$ : - 0.5;  $A_4$ : - 0.5

Now, as  $A_3$  and  $A_4$  have least negative reward, the best action for most of the states would converge to either  $A_3$  or  $A_4$ , hence resulting in an infinite loop.

Please refer python file – HW2\_E.py (may also refer – ITERATIONS(E)).

- F) **(Optional for 4420)** Repeat part B, but this time with noise = 0.2, and  $\gamma = 0.9$  and no living reward. With a noise of 0.2, every time you take an action, the result will be the expected action with Probability 0.8 (80%), but 20% of the time, the robot will instead take a different action (taken randomly out of unexpected actions, with equal probability). If the action is impossible, it remains in the same cell.

For example, if we are in state (4,1,4), location (4,1) and facing right, and we take action  $A_1$  (moving one cell forward), the resulting state will be:

$s' = (5, 1, 2)$  with probability 0.8 [because  $A_1$  is rendered]  
 $s' = (4, 1, 4)$  with probability 0.2/3 [renders  $A_2$  which is impossible]  
 $s' = (4, 1, 1)$  with probability 0.2/3 [because  $A_3$  is rendered]  
 $s' = (4, 1, 2)$  with probability 0.2/3 [because  $A_4$  is rendered]

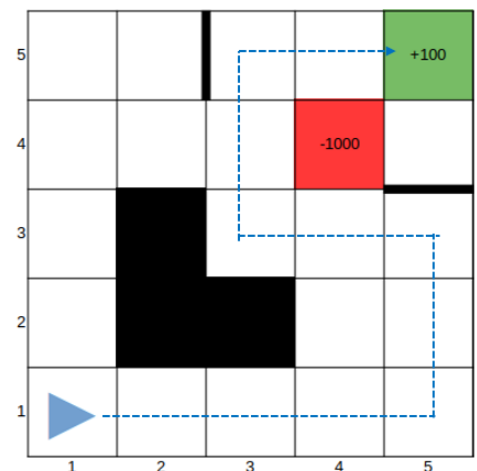
Compare the results with that of the previous parts and explain your observations.

ANSWER – For noise = 0.2 and  $\gamma = 0.9$ ,

For iterations, please refer python file – HW2\_F.py (may also refer – ITERATIONS(F))

Resulting path –

$[(1, 1, 4), A_2] \rightarrow [(3, 1, 4), A_2] \rightarrow [(5, 1, 4), A_3] \rightarrow [(5, 1, 1), A_2] \rightarrow [(5, 3, 1), A_3] \rightarrow [(5, 3, 3), A_2] \rightarrow [(3, 3, 3), A_4] \rightarrow [(3, 3, 1), A_2] \rightarrow [(3, 5, 1), A_4] \rightarrow [(3, 5, 4), A_2] \rightarrow [(5, 5, 4)]$



Form ANSWER B and ANSWER D, the resulting path in previous cases [ $\gamma = 1$ , noise = 0], [ $\gamma = 0.8$ , noise = 0] is different from the resulting path in this case [ $\gamma = 0.9$ , noise = 0.2].

### Observations –

In part B, D and E, as noise was zero, the problems were deterministic. From a state  $S$ , taking action  $A$  would always result in state  $S'$ .

But when noise is nonzero, or 0.2 as in our case, the problem becomes probabilistic. From a state  $S$ , taking action  $A$  may or may not result in state  $S'$ .

So, in parts B and D, taking action 'A3' from state  $(3, 4, 4)$  will always result in state  $(3, 4, 1)$ . But in this case when noise is 0.2, taking action 'A3' from state  $(3, 4, 4)$  has a probability =  $0.2/3$  of resulting in terminal state  $(4, 4, 4)$  with high negative value of -1000.

To avoid that possibility of landing in a terminal state, the robot takes a different approach. It finds a path to land itself in state  $(3,3,1)$  from where an action  $A2$  will ensure zero probability of robot landing in terminal state  $(4,4)$ .

Hence, deterministic and probabilistic model problems have **different optimal policies**.

```

1 # MASTER CODE
2
3 # Using class defined attributes and transition model
  function
4 class mdpclass:
5     def __init__(self, states, prob, discount,
6         actions):
7         self.states = states[:]
8         self.prob = prob
9         self.discount = discount
10        self.actions = actions[:]
11
12        # Transition model returns (probability, reward
13        and resulting state) for a given (state, action) pair
14    def transition_model(self, s, action, pr):
15        q = [1, 2, 3, 4]
16        r = 0
17        temp_list = []
18        prp = (1 - pr) / 3
19        q.remove(action)
20        while True:
21            x, y, z = s
22            if action == 1:
23                r = -1.5
24                if s[2] == 1:
25                    y = y + 1
26                elif s[2] == 2:
27                    y = y - 1
28                elif s[2] == 3:
29                    x = x - 1
30                elif s[2] == 4:
31                    x = x + 1
32            elif action == 2:
33                r = -2
34                if s[2] == 1:
35                    y = y + 2
36                elif s[2] == 2:
37                    y = y - 2
38                elif s[2] == 3:
39                    x = x - 2
40                elif s[2] == 4:

```

```

39         x = x + 2
40     elif action == 3:
41         r = -0.5
42         if s[2] == 1:
43             z = z + 2
44         elif s[2] == 2:
45             z = z + 2
46         elif s[2] == 3:
47             z = z - 1
48         elif s[2] == 4:
49             z = z - 3
50     elif action == 4:
51         r = -0.5
52         if s[2] == 1:
53             z = z + 3
54         elif s[2] == 2:
55             z = z + 1
56         elif s[2] == 3:
57             z = z - 2
58         elif s[2] == 4:
59             z = z - 2
60     s_prime = (x, y, z)
61     # checking for prohibited moves
62     if (x < 1 or x > 5) or (y < 1 or y > 5)
    or \
63         (s[:2] in [(2, 2), (2, 3), (3, 2
64         )] or s_prime[:2] in [(2, 2), (2, 3), (3, 2)]) or \
65         (s[:2] in [(2, 5), (5, 3)] and
66         s_prime[:2] in [(3, 5), (4, 5), (5, 4), (5, 5)]) or \
67         (s[:2] in [(1, 5), (5, 2)] and
68         s_prime[:2] in [(3, 5), (5, 4)]) or \
69         (s[:2] in [(3, 5), (5, 4)] and
70         s_prime[:2] in [(1, 5), (2, 5), (5, 2), (5, 3)]) or \
71         (s[:2] in [(4, 5)] and s_prime[:2
72         ] in [(2, 5)]) or \
73         (s[:2] in [(4, 3), (3, 4)] and
74         s_prime[:2] in [(5, 4), (4, 5)]) or \
75         (s[:2] in [(5, 4), (4, 5)] and
76         s_prime[:2] in [(4, 3), (3, 4)]) or \
77         (s[:2] in [(3, 1), (1, 3)] and
78         s_prime[:2] in [(3, 3)]) or \

```

```

71         (s[:2] in [(3, 3)] and s_prime[:
72 2] in [(3, 1), (1, 3)]):
73             s_prime = s # returning s' as s if
move is prohibited
74             temp_list.append((pr, r, s_prime))
75             if pr == 1:
76                 return temp_list
77             else:
78                 if not q:
79                     return temp_list
80                 action = q.pop()
81                 pr = prp
82
83 # defined value iteration function printing required
iterations and path of robot
84 def valueiteration(mdp):
85     value = {}
86     pi = {}
87
88     # initializing default value +100 and -1000 for
terminal states and 0 for all other states
89     for state in mdp.states:
90         if state[:2] == (5, 5):
91             value[state] = 100.
92             pi[state] = None
93         elif state[:2] == (4, 4):
94             value[state] = -1000.
95             pi[state] = None
96         else:
97             value[state] = 0.
98
99     # function qvalue first extracts tuple triplets
of (probability, reward and resulting state) from
transition model
100     # then returns the expected utility of a given (
state, action) pair
101     def qvalue(st, action, v):
102         return sum(p * (r + mdp.discount * v[s_prime
]) for p, r, s_prime in mdp.transition_model(st,
action, mdp.prob))

```



```

103
104     # initiated n = 100 iterations
105     for i in range(100):
106         if i < 11:
107             print("iter {}".format(i))
108             for state in mdp.states:
109                 if state[:2] not in [(4, 4), (5, 5), (3
110 , 2), (2, 2), (2, 3)]:
111                     # maximizing over Q value of a given
112                     (state, action) pair to assign value to the state
113                     value[state] = max(qvalue(state,
114 action, value) for action in mdp.actions)
115
116                     # maximizing over Q value of a given
117                     (state, action) pair to extract policy
118                     pi[state] = max((qvalue(state,
119 action, value), action) for action in mdp.actions),
120 key=lambda tup: tup[
121 0])
122
123                     if i < 11:
124                         print("state({}) V = {}      Best
125 Action: {}".format(state, pi[state][0], pi[state][1
126 ]))
127
128 s = (1, 1, 4) # initial state of robot
129 result = [] # list to capture the resulting
130 best path of robot to end state
131     while pi[s] is not None:
132         q, act = pi[s] # extracting best action for
133         a state from already extracted policy over 100
134         iterations
135
136         # extracting next state from best action for
137         a given state
138         j_list = mdp.transition_model(s, act, 1)
139         x, y, z = j_list[0]
140
141         result.append((s, pi[s]))
142         s = z
143

```

```
132     result.append((s, pi[s]))
133     print("Result:", result)
134
135
136
137 # initializing states as tuple triplets of (x, y, d
) in a list
138 states = []
139 for i in range(1, 6):
140     for j in range(1, 6):
141         for k in range(1, 5):
142             a = (i, j, k)
143             states.append(a)
144
145
146 # initialize actions, noise and discount factor (
gamma)
147 actions = [1, 2, 3, 4]
148 noise = 0
149 discount = 1
150
151 # created instance of class mdpclass
152 prob = 1 - noise
153 mdp = mdpclass(states, prob, discount, actions)
154
155 # called function for value iteration
156 valueiteration(mdp)
157
```

[illegible]





**iter 1:**

```

state((5, 1, 1)) V = -0.5      Best Action: 3
state((5, 1, 2)) V = -0.5      Best Action: 3
state((5, 1, 3)) V = -0.9      Best Action: 3
state((5, 1, 4)) V = -0.9      Best Action: 3
state((5, 2, 1)) V = -0.5      Best Action: 3
state((5, 2, 2)) V = -0.5      Best Action: 3
state((5, 2, 3)) V = -0.9      Best Action: 3
state((5, 2, 4)) V = -0.9      Best Action: 3
state((5, 3, 1)) V = -0.5      Best Action: 3
state((5, 3, 2)) V = -0.5      Best Action: 3
state((5, 3, 3)) V = -0.9      Best Action: 3
state((5, 3, 4)) V = -0.9      Best Action: 3
state((5, 4, 1)) V = 78.5      Best Action: 1
state((5, 4, 2)) V = -0.5      Best Action: 3
state((5, 4, 3)) V = 62.30000000000000
state((5, 4, 4)) V = 62.30000000000000

iter 2:

state((1, 1, 1)) V = -1.2200000000000000
state((1, 1, 2)) V = -1.2200000000000000
state((1, 1, 3)) V = -1.4760000000000000
state((1, 1, 4)) V = -1.4760000000000000
state((1, 2, 1)) V = -1.2200000000000000
state((1, 2, 2)) V = -1.2200000000000000
state((1, 2, 3)) V = -1.4760000000000000
state((1, 2, 4)) V = -1.4760000000000000
state((1, 3, 1)) V = -1.2200000000000000
state((1, 3, 2)) V = -1.2200000000000000
state((1, 3, 3)) V = -1.4760000000000000
state((1, 3, 4)) V = -1.4760000000000000
state((1, 4, 1)) V = -1.2200000000000000
state((1, 4, 2)) V = -1.2200000000000000
state((1, 4, 3)) V = -1.4760000000000000
state((1, 4, 4)) V = -1.4760000000000000
state((1, 5, 1)) V = -1.2200000000000000
state((1, 5, 2)) V = -1.2200000000000000
state((1, 5, 3)) V = -1.4760000000000000
state((1, 5, 4)) V = -1.4760000000000000
state((2, 1, 1)) V = -1.2200000000000000
state((2, 1, 2)) V = -1.2200000000000000
state((2, 1, 3)) V = -1.4760000000000000
state((2, 1, 4)) V = -1.4760000000000000
state((2, 4, 1)) V = -1.2200000000000000
state((2, 4, 2)) V = -1.2200000000000000
state((2, 4, 3)) V = -1.4760000000000000
state((2, 4, 4)) V = -1.4760000000000000
state((2, 5, 1)) V = -1.2200000000000000
state((2, 5, 2)) V = -1.2200000000000000
state((2, 5, 3)) V = -1.4760000000000000
state((2, 5, 4)) V = -1.4760000000000000
state((3, 1, 1)) V = -1.2200000000000000
state((3, 1, 2)) V = -1.2200000000000000
state((3, 1, 3)) V = -1.4760000000000000
state((3, 1, 4)) V = -1.4760000000000000
state((3, 3, 1)) V = -1.2200000000000000
state((3, 3, 2)) V = -1.2200000000000000
state((3, 3, 3)) V = -1.4760000000000000
state((3, 3, 4)) V = -1.4760000000000000
state((3, 4, 1)) V = -1.2200000000000000
state((3, 4, 2)) V = -1.2200000000000000
state((3, 4, 3)) V = -1.4760000000000000
state((3, 4, 4)) V = -1.4760000000000000
state((3, 5, 1)) V = 61.90000000000000
state((3, 5, 2)) V = 61.90000000000000
state((3, 5, 3)) V = 49.020000000000001
state((3, 5, 4)) V = 78.0      Best Action: 2

```

Best Action: 3

Best Action: 3

Best Action: 3

state((1, 1)) V = -2.1644556800000005	Best Action: 3	state((5, 1, 2)) V = -2.1644556800000005	Best Action: 3	state((4, 1, 2)) V = 9.2428428800000007	Best Action: 3	state((3, 1, 2)) V = 6.0942743040000006	Best Action: 3	state((1, 5, 2)) V = 16.2329920000000007	Best Action: 1
state((1, 1, 2)) V = -2.1644556800000005	Best Action: 3	state((5, 1, 3)) V = 11.922553600000001	Best Action: 4	state((4, 1, 3)) V = 12.1785536000000008	Best Action: 4	state((3, 1, 3)) V = 4.6952919040000005	Best Action: 2	state((1, 5, 3)) V = 12.4863936000000007	Best Action: 3
state((1, 1, 3)) V = -2.2315645440000003	Best Action: 3	state((5, 1, 4)) V = 11.922553600000001	Best Action: 3	state((4, 1, 4)) V = 12.1785536000000008	Best Action: 3	state((3, 1, 4)) V = 8.2428428800000007	Best Action: 1	state((1, 5, 4)) V = 12.4863936000000007	Best Action: 4
state((1, 1, 4)) V = -2.2315645440000003	Best Action: 3	state((5, 2, 1)) V = 16.028192000000001	Best Action: 1	state((4, 2, 1)) V = 16.3481920000000008	Best Action: 1	state((3, 3, 1)) V = 47.520000000000001	Best Action: 2	state((2, 1, 1)) V = 5.6942743040000006	Best Action: 4
state((1, 2, 1)) V = -2.1644556800000005	Best Action: 3	state((5, 2, 2)) V = -2.1644556800000005	Best Action: 3	state((4, 2, 2)) V = 9.5628428800000005	Best Action: 3	state((3, 3, 2)) V = 29.5128000000000013	Best Action: 3	state((2, 1, 2)) V = 5.6942743040000006	Best Action: 3
state((1, 2, 2)) V = -2.1644556800000005	Best Action: 3	state((5, 2, 3)) V = 12.322553600000001	Best Action: 4	state((4, 2, 3)) V = 12.5785536000000006	Best Action: 4	state((3, 3, 3)) V = 37.516000000000001	Best Action: 4	state((2, 1, 3)) V = 5.1952919040000005	Best Action: 1
state((1, 2, 3)) V = -2.2315645440000003	Best Action: 3	state((5, 2, 4)) V = 12.322553600000001	Best Action: 3	state((4, 2, 4)) V = 12.5785536000000006	Best Action: 3	state((3, 3, 4)) V = 37.516000000000001	Best Action: 3	state((2, 1, 4)) V = 7.7428428800000007	Best Action: 2
state((1, 2, 4)) V = -2.2315645440000003	Best Action: 3	state((5, 3, 1)) V = 21.9102400000000012	Best Action: 3	state((4, 3, 1)) V = 22.310240000000001	Best Action: 3	state((3, 4, 1)) V = 48.020000000000001	Best Action: 1	state((2, 4, 1)) V = 22.5662400000000008	Best Action: 4
state((1, 3, 1)) V = -2.1644556800000005	Best Action: 3	state((5, 3, 2)) V = 21.9102400000000012	Best Action: 4	state((4, 3, 2)) V = 22.310240000000001	Best Action: 4	state((3, 4, 2)) V = 29.832800000000001	Best Action: 3	state((2, 4, 2)) V = 22.5662400000000008	Best Action: 3
state((1, 3, 2)) V = -2.1644556800000005	Best Action: 3	state((5, 3, 3)) V = 28.0128000000000013	Best Action: 2	state((4, 3, 3)) V = 28.5128000000000013	Best Action: 1	state((3, 4, 3)) V = 37.916000000000001	Best Action: 4	state((2, 4, 3)) V = 17.5529920000000007	Best Action: 3
state((1, 3, 3)) V = -2.2315645440000003	Best Action: 3	state((5, 3, 4)) V = 17.028192000000001	Best Action: 3	state((4, 3, 4)) V = 17.3481920000000008	Best Action: 3	state((3, 4, 4)) V = 37.916000000000001	Best Action: 3	state((2, 4, 4)) V = 28.832800000000001	Best Action: 1
state((1, 3, 4)) V = -2.2315645440000003	Best Action: 3	state((5, 4, 1)) V = 78.5 Best Action: 1		state((4, 5, 1)) V = 62.300000000000004	Best Action: 4	state((3, 5, 1)) V = 61.900000000000006	Best Action: 4	state((2, 5, 1)) V = 9.6939148800000007	Best Action: 3
state((1, 4, 1)) V = 22.166240000000001	Best Action: 4	state((5, 4, 2)) V = 49.34 Best Action: 3		state((4, 5, 2)) V = 62.300000000000004	Best Action: 3	state((3, 5, 2)) V = 61.900000000000006	Best Action: 3	state((2, 5, 2)) V = 16.5529920000000007	Best Action: 1
state((1, 4, 2)) V = 22.166240000000001	Best Action: 3	state((5, 4, 3)) V = 62.300000000000004	Best Action: 4	state((4, 5, 3)) V = 49.34 Best Action: 3		state((3, 5, 3)) V = 49.020000000000001	Best Action: 3	state((2, 5, 3)) V = 12.7423936000000007	Best Action: 3
state((1, 4, 3)) V = 17.2329920000000007	Best Action: 3	state((5, 4, 4)) V = 62.300000000000004	Best Action: 3	state((4, 5, 4)) V = 78.5 Best Action: 1		state((3, 5, 4)) V = 78.0 Best Action: 2		state((2, 5, 4)) V = 12.7423936000000007	Best Action: 4
state((1, 4, 4)) V = 28.332800000000001	Best Action: 2	<b>iter 6:</b>		state((5, 1, 1)) V = 15.5281920000000011	Best Action: 2	state((4, 1, 1)) V = 15.8481920000000008	Best Action: 2	state((3, 1, 1)) V = 6.0942743040000006	Best Action: 4
state((1, 5, 1)) V = -2.1644556800000005	Best Action: 3	state((1, 1, 1)) V = -2.2852516352000003	Best Action: 3	state((5, 1, 2)) V = 9.0380428800000008	Best Action: 3	state((4, 1, 2)) V = 9.2428428800000007	Best Action: 3	state((3, 1, 2)) V = 6.0942743040000006	Best Action: 3
state((1, 5, 2)) V = 16.23299200000000									









[illegible]

[illegible]



state((1, 1, 1)) V = -0.6666666666666666	Best Action: 3	state((5, 1, 1)) V = -0.6666666666666666	Best Action: 3	state((4, 1, 1)) V = -8.881595005866664	Best Action: 4	state((3, 1, 1)) V = -2.6308954853649067	Best Action: 4	state((1, 5, 1)) V = -3.316681500057933	Best Action: 3
state((1, 1, 2)) V = -0.7466666666666667	Best Action: 3	state((5, 1, 2)) V = -0.7466666666666667	Best Action: 3	state((4, 1, 2)) V = -1.8033864065706668	Best Action: 3	state((3, 1, 2)) V = -2.6308954853649067	Best Action: 3	state((1, 5, 2)) V = -3.3715824504679723	Best Action: 4
state((1, 1, 3)) V = -1.3290666666666668	Best Action: 3	state((5, 1, 3)) V = -1.3471953258666667	Best Action: 3	state((4, 1, 3)) V = -2.6750276870826672	Best Action: 3	state((3, 1, 3)) V = -2.988612307146974	Best Action: 3	state((1, 5, 3)) V = -3.621227389423843	Best Action: 4
state((1, 1, 4)) V = -1.1866666666666667	Best Action: 3	state((5, 1, 4)) V = -1.3290666666666668	Best Action: 3	state((4, 1, 4)) V = -2.634673417517227	Best Action: 4	state((3, 1, 4)) V = -2.91980611146682	Best Action: 3	state((1, 5, 4)) V = -3.5842078846266	Best Action: 3
state((1, 2, 1)) V = -0.6666666666666666	Best Action: 3	state((5, 2, 1)) V = -0.7066666666666667	Best Action: 3	state((4, 2, 1)) V = -68.06273066666664	Best Action: 3	state((3, 3, 1)) V = 33.41957829457013	Best Action: 2	state((2, 1, 1)) V = -3.3307556617576646	Best Action: 4
state((1, 2, 2)) V = -0.7490666666666667	Best Action: 3	state((5, 2, 2)) V = -0.7490666666666667	Best Action: 3	state((4, 2, 2)) V = -3.3658575062451197	Best Action: 1	state((3, 3, 2)) V = -2.6318181761706665	Best Action: 4	state((2, 1, 2)) V = -3.3307556617576646	Best Action: 3
state((1, 2, 3)) V = -1.3317546666666666	Best Action: 4	state((5, 2, 3)) V = -1.5666743466666664	Best Action: 4	state((4, 2, 3)) V = -7.772106578630516	Best Action: 3	state((3, 3, 3)) V = 26.002498400304447	Best Action: 4	state((2, 1, 3)) V = -3.623508863012826	Best Action: 3
state((1, 2, 4)) V = -1.3317546666666666	Best Action: 3	state((5, 2, 4)) V = -0.7466666666666667	Best Action: 3	state((4, 2, 4)) V = -6.925509318542384	Best Action: 1	state((3, 3, 4)) V = 22.70366855668252	Best Action: 3	state((2, 1, 4)) V = -3.59197856111557	Best Action: 3
state((1, 3, 1)) V = -0.6666666666666666	Best Action: 3	state((5, 3, 1)) V = -0.7490666666666667	Best Action: 3	state((4, 3, 1)) V = -68.53626337706665	Best Action: 4	state((3, 4, 1)) V = 31.795904398780376	Best Action: 1	state((2, 4, 1)) V = -8.23639963669819	Best Action: 1
state((1, 3, 2)) V = -0.7490666666666667	Best Action: 3	state((5, 3, 2)) V = -0.7490666666666667	Best Action: 3	state((4, 3, 2)) V = -3.7404595329117867	Best Action: 2	state((3, 4, 2)) V = -8.0568594817229074	Best Action: 1	state((2, 4, 2)) V = -9.50172827969838	Best Action: 4
state((1, 3, 3)) V = -1.3346039466666666	Best Action: 4	state((5, 3, 3)) V = -1.5578423466666664	Best Action: 4	state((4, 3, 3)) V = -7.683144267317188	Best Action: 1	state((3, 4, 3)) V = 19.75202439432462	Best Action: 4	state((2, 4, 3)) V = -5.802596061268631	Best Action: 1
state((1, 3, 4)) V = -1.3346039466666666	Best Action: 3	state((5, 3, 4)) V = -1.3346039466666666	Best Action: 3	state((4, 3, 4)) V = -6.9793518705171875	Best Action: 1	state((3, 4, 4)) V = -42.395773478592666	Best Action: 3	state((2, 4, 4)) V = -69.7486961118888	Best Action: 3
state((1, 4, 1)) V = -0.7066666666666667	Best Action: 3	state((5, 4, 1)) V = 74.836	Best Action: 1	state((4, 5, 1)) V = 52.88155275912533	Best Action: 4	state((3, 5, 1)) V = 67.18303094176058	Best Action: 4	state((2, 5, 1)) V = -3.342189479175216	Best Action: 4
state((1, 4, 2)) V = -0.7540106666666666	Best Action: 3	state((5, 4, 2)) V = -0.7466666666666667	Best Action: 3	state((4, 5, 2)) V = -13.760701852970655	Best Action: 3	state((3, 5, 2)) V = 58.71972073704706	Best Action: 3	state((2, 5, 2)) V = -3.7324338541270907	Best Action: 3
state((1, 4, 3)) V = -1.3349361834666666	Best Action: 4	state((5, 4, 3)) V = -10.46714666666665	Best Action: 4	state((4, 5, 3)) V = 36.62442624978109	Best Action: 4	state((3, 5, 3)) V = 56.41032693503395	Best Action: 4	state((2, 5, 3)) V = -3.662202606668337	Best Action: 4
state((1, 4, 4)) V = -1.1919073066666668	Best Action: 3	state((5, 4, 4)) V = 56.140373333333336	Best Action: 3	state((4, 5, 4)) V = 77.03124377764351	Best Action: 1	state((3, 5, 4)) V = 82.26256928368771	Best Action: 2	state((2, 5, 4)) V = -3.662152283026832	Best Action: 3
state((1, 5, 1)) V = -0.7466666666666667	Best Action: 3	iter 2:		state((5, 1, 1)) V = -1.681898386218667	Best Action: 4	state((4, 1, 1)) V = -10.834332793586343	Best Action: 4	state((3, 1, 1)) V = -3.338452507299336	Best Action: 4
state((1, 5, 2)) V = -0.7568513066666666	Best Action: 3	state((1, 1, 1)) V = -1.6722666666666667	Best Action: 3	state((5, 1, 2)) V = -1.803726192564907	Best Action: 3				

ANSWER F –

state((1, 1, 1)) V = -3.8043621592084516    Best Action: 3  
state((1, 1, 2)) V = -3.8700369307316946    Best Action: 4  
state((1, 1, 3)) V = -4.117416173751613    Best Action: 3  
state((1, 1, 4)) V = -4.06228958296474    Best Action: 4  
state((1, 2, 1)) V = -3.870522251704674    Best Action: 3  
state((1, 2, 2)) V = -3.894936355217257    Best Action: 3  
state((1, 2, 3)) V = -4.1701655787629175    Best Action: 3  
state((1, 2, 4)) V = -4.1701655787629175    Best Action: 4  
state((1, 3, 1)) V = -3.8751003574089533    Best Action: 3  
state((1, 3, 2)) V = -3.9009328325002697    Best Action: 3  
state((1, 3, 3)) V = -4.1849153792968075    Best Action: 4  
state((1, 3, 4)) V = -4.1849153792968075    Best Action: 3  
state((1, 4, 1)) V = -4.791444760685347    Best Action: 3  
state((1, 4, 2)) V = -4.778364977565765    Best Action: 4  
state((1, 4, 3)) V = -4.955477813021691    Best Action: 4  
state((1, 4, 4)) V = -10.327124338781314    Best Action: 3  
state((1, 5, 1)) V = -3.889325744153624    Best Action: 3  
state((1, 5, 2)) V = -3.9518167657860084    Best Action: 4  
state((1, 5, 3)) V = -4.142815182205588    Best Action: 4  
state((1, 5, 4)) V = -4.11323590641482    Best Action: 3  
state((2, 1, 1)) V = -3.9254612728413787    Best Action: 3  
state((2, 1, 2)) V = -3.9254612728413787    Best Action: 4  
state((2, 1, 3)) V = -4.162599234276065    Best Action: 3  
state((2, 1, 4)) V = -4.161202932078326    Best Action: 3  
state((2, 4, 1)) V = -8.800005170511456    Best Action: 1  
state((2, 4, 2)) V = -10.203564596586471    Best Action: 4  
state((2, 4, 3)) V = -6.423460037177684    Best Action: 1  
state((2, 4, 4)) V = -69.35537094461634    Best Action: 3  
state((2, 5, 1)) V = -3.926545282677319    Best Action: 4  
state((2, 5, 2)) V = -4.328081387734338    Best Action: 3  
state((2, 5, 3)) V = -4.1939748124301435    Best Action: 4  
state((2, 5, 4)) V = -4.195100214953283    Best Action: 3  
state((3, 1, 1)) V = -3.937586258733454    Best Action: 4  
state((3, 1, 2)) V = -3.937586258733454    Best Action: 3  
state((3, 1, 3)) V = -4.171676890005051    Best Action: 3  
state((3, 1, 4)) V = -4.1669188189405455    Best Action: 3  
state((3, 3, 1)) V = 55.61922421548264    Best Action: 2  
state((3, 3, 2)) V = 33.29470183163667    Best Action: 4  
state((3, 3, 3)) V = 46.785233877198785    Best Action: 4  
state((3, 3, 4)) V = 41.62078642919743    Best Action: 3  
state((3, 4, 1)) V = 52.04019154723005    Best Action: 1  
state((3, 4, 2)) V = 24.277957213297782    Best Action: 4  
state((3, 4, 3)) V = 37.292047764618495    Best Action: 4  
state((3, 4, 4)) V = -23.460749132126878    Best Action: 3  
state((3, 5, 1)) V = 71.16324357975995    Best Action: 4  
state((3, 5, 2)) V = 65.94865089099085    Best Action: 3  
state((3, 5, 3)) V = 61.803328871871194    Best Action: 4  
state((3, 5, 4)) V = 83.20344973424581    Best Action: 2  
state((4, 1, 1)) V = -11.971881604622798    Best Action: 4  
state((4, 1, 2)) V = -4.791954191391402    Best Action: 3  
state((4, 1, 3)) V = -5.262238300361165    Best Action: 3  
state((4, 1, 4)) V = -5.262402108395386    Best Action: 1  
state((4, 2, 1)) V = -70.90879027910444    Best Action: 4  
state((4, 2, 2)) V = -6.262725669079399    Best Action: 1  
state((4, 2, 3)) V = -10.668874330633782    Best Action: 3  
state((4, 2, 4)) V = -9.14972781119086    Best Action: 1  
state((4, 3, 1)) V = -46.15869257627553    Best Action: 3  
state((4, 3, 2)) V = 15.996410269153385    Best Action: 4  
state((4, 3, 3)) V = 31.70758528800305    Best Action: 1  
state((4, 3, 4)) V = 8.937374416494274    Best Action: 4  
state((4, 5, 1)) V = 67.68880627937362    Best Action: 4  
state((4, 5, 2)) V = -0.4466030128649905    Best Action: 3  
state((4, 5, 3)) V = 54.92247925281985    Best Action: 4  
state((4, 5, 4)) V = 79.3853435163084    Best Action: 2  
state((5, 1, 1)) V = 4.593842982216989    Best Action: 1

state((5, 1, 2)) V = -3.8960727562216078    Best Action: 3  
state((5, 1, 3)) V = 1.845328361574649    Best Action: 4  
state((5, 1, 4)) V = 2.649744450277043    Best Action: 3  
state((5, 2, 1)) V = 5.431063638500724    Best Action: 1  
state((5, 2, 2)) V = -3.9475397836253787    Best Action: 3  
state((5, 2, 3)) V = 2.1251753970983156    Best Action: 4  
state((5, 2, 4)) V = 2.9210156129650615    Best Action: 3  
state((5, 3, 1)) V = 23.12516599638467    Best Action: 3  
state((5, 3, 2)) V = 20.058369870653358    Best Action: 4  
state((5, 3, 3)) V = 35.69566168056099    Best Action: 2  
state((5, 3, 4)) V = 18.18459111925301    Best Action: 3  
state((5, 4, 1)) V = 79.3864622778244    Best Action: 1  
state((5, 4, 2)) V = 54.46489668618733    Best Action: 3  
state((5, 4, 3)) V = -0.27249344455385227    Best Action: 4  
state((5, 4, 4)) V = 67.88686477732749    Best Action: 3  
**iter 6:**  
state((1, 1, 1)) V = -4.293744225094134    Best Action: 3  
state((1, 1, 2)) V = -4.343208772233966    Best Action: 4  
state((1, 1, 3)) V = -4.549015127905341    Best Action: 3  
state((1, 1, 4)) V = -4.509190047353352    Best Action: 4  
state((1, 2, 1)) V = -4.384830042451253    Best Action: 3  
state((1, 2, 2)) V = -4.386008409047592    Best Action: 3  
state((1, 2, 3)) V = -4.613263887616085    Best Action: 3  
state((1, 2, 4)) V = -4.613263887616085    Best Action: 4  
state((1, 3, 1)) V = -4.392372106716072    Best Action: 3  
state((1, 3, 2)) V = -4.39692307127116    Best Action: 3  
state((1, 3, 3)) V = -4.643443222153411    Best Action: 4  
state((1, 3, 4)) V = -4.643443222153411    Best Action: 3  
state((1, 4, 1)) V = -5.341769416744717    Best Action: 3  
state((1, 4, 2)) V = -5.31933924468749    Best Action: 4  
state((1, 4, 3)) V = -5.454714163402578    Best Action: 4  
state((1, 4, 4)) V = -10.397195474617963    Best Action: 3  
state((1, 5, 1)) V = -4.364934923726196    Best Action: 3  
state((1, 5, 2)) V = -4.431139044394667    Best Action: 4  
state((1, 5, 3)) V = -4.575877096683861    Best Action: 4  
state((1, 5, 4)) V = -4.552485399949584    Best Action: 3  
state((2, 1, 1)) V = -4.412116397838653    Best Action: 3  
state((2, 1, 2)) V = -4.412116397838653    Best Action: 4  
state((2, 1, 3)) V = -4.605836512127454    Best Action: 3  
state((2, 1, 4)) V = -4.624733053907257    Best Action: 3  
state((2, 4, 1)) V = -9.274561242228721    Best Action: 1  
state((2, 4, 2)) V = -10.701991068620039    Best Action: 4  
state((2, 4, 3)) V = -6.887016529438795    Best Action: 1  
state((2, 4, 4)) V = -69.39012660791721    Best Action: 3  
state((2, 5, 1)) V = -4.41190088055347    Best Action: 4  
state((2, 5, 2)) V = -4.8150772608662695    Best Action: 3  
state((2, 5, 3)) V = -4.63524066361072    Best Action: 4  
state((2, 5, 4)) V = -4.637329181352267    Best Action: 3  
state((3, 1, 1)) V = -4.442849767334564    Best Action: 4  
state((3, 1, 2)) V = -4.442849767334564    Best Action: 3  
state((3, 1, 3)) V = -4.628212714711641    Best Action: 3  
state((3, 1, 4)) V = -1.175451906705279    Best Action: 2  
state((3, 3, 1)) V = 57.711044329998884    Best Action: 2  
state((3, 3, 2)) V = 39.55507219247813    Best Action: 4  
state((3, 3, 3)) V = 49.053550905639284    Best Action: 4  
state((3, 3, 4)) V = 44.72676439356404    Best Action: 3  
state((3, 4, 1)) V = 53.74924381129561    Best Action: 1  
state((3, 4, 2)) V = 28.991904444660754    Best Action: 1  
state((3, 4, 3)) V = 38.93508971738373    Best Action: 4  
state((3, 4, 4)) V = -21.635341803781785    Best Action: 3  
state((3, 5, 1)) V = 71.51005248468738    Best Action: 4  
state((3, 5, 2)) V = 66.97820790772698    Best Action: 3  
state((3, 5, 3)) V = 62.263075871916904    Best Action: 4  
state((3, 5, 4)) V = 83.30191727531499    Best Action: 2  
state((4, 1, 1)) V = -11.807143683992649    Best Action: 4

state((4, 1, 2)) V = -5.379583538006944    Best Action: 3  
state((4, 1, 3)) V = -5.747878907908125    Best Action: 3  
state((4, 1, 4)) V = -1.0750972245036552    Best Action: 1  
state((4, 2, 1)) V = -70.67578133288319    Best Action: 4  
state((4, 2, 2)) V = -6.809384373997222    Best Action: 1  
state((4, 2, 3)) V = -11.112673350366864    Best Action: 3  
state((4, 2, 4)) V = -4.744745849175744    Best Action: 1  
state((4, 3, 1)) V = -40.070484348891334    Best Action: 3  
state((4, 3, 2)) V = 21.631782743581997    Best Action: 4  
state((4, 3, 3)) V = 34.61510234333389    Best Action: 1  
state((4, 3, 4)) V = 14.337356079665257    Best Action: 4  
state((4, 5, 1)) V = 67.92026496285396    Best Action: 4  
state((4, 5, 2)) V = -0.2406667605273185    Best Action: 3  
state((4, 5, 3)) V = 55.23600968155082    Best Action: 4  
state((4, 5, 4)) V = 79.42269213715598    Best Action: 1  
state((5, 1, 1)) V = 14.374725085777266    Best Action: 2  
state((5, 1, 2)) V = 0.9241720287476741    Best Action: 3  
state((5, 1, 3)) V = 9.085485862754448    Best Action: 4  
state((5, 1, 4)) V = 10.901623264598678    Best Action: 3  
state((5, 2, 1)) V = 15.340407756968904    Best Action: 1  
state((5, 2, 2)) V = 1.1166835432156086    Best Action: 3  
state((5, 2, 3)) V = 9.907853540849988    Best Action: 4  
state((5, 2, 4)) V = 11.236972073612353    Best Action: 3  
state((5, 3, 1)) V = 29.488852300046737    Best Action: 3  
state((5, 3, 2)) V = 26.12532769107228    Best Action: 4  
state((5, 3, 3)) V = 38.66441330454346    Best Action: 2  
state((5, 3, 4)) V = 24.3586596814487    Best Action: 3  
state((5, 4, 1)) V = 79.41900388119112    Best Action: 1  
state((5, 4, 2)) V = 54.732083352386276    Best Action: 3  
state((5, 4, 3)) V = -0.21876257320248982    Best Action: 4  
state((5, 4, 4)) V = 67.94700862680531    Best Action: 3  
**iter 7:**  
state((1, 1, 1)) V = -4.700924512136637    Best Action: 3  
state((1, 1, 2)) V = -4.7369116976113865    Best Action: 4  
state((1, 1, 3)) V = -4.90811207309041    Best Action: 3  
state((1, 1, 4)) V = -3.4538411506694966    Best Action: 2  
state((1, 2, 1)) V = -4.804452398434587    Best Action: 3  
state((1, 2, 2)) V = -4.789641754165829    Best Action: 3  
state((1, 2, 3)) V = -4.977753414643203    Best Action: 3  
state((1, 2, 4)) V = -4.977753414643203    Best Action: 4  
state((1, 3, 1)) V = -4.821483060110024    Best Action: 3  
state((1, 3, 2)) V = -4.811828933099705    Best Action: 3  
state((1, 3, 3)) V = -5.016983570482651    Best Action: 3  
state((1, 3, 4)) V = -5.016983570482651    Best Action: 4  
state((1, 4, 1)) V = -5.776316782479569    Best Action: 3  
state((1, 4, 2)) V = -5.747558417179814    Best Action: 4  
state((1, 4, 3)) V = -5.845534048561787    Best Action: 3  
state((1, 4, 4)) V = -10.618790439173775    Best Action: 4  
state((1, 5, 1)) V = -4.7598404185835586    Best Action: 3  
state((1, 5, 2)) V = -4.828052369935408    Best Action: 4  
state((1, 5, 3)) V = -4.935397937788909    Best Action: 4  
state((1, 5, 4)) V = -4.9171318164048685    Best Action: 3  
state((2, 1, 1)) V = -4.81254148429273    Best Action: 3  
state((2, 1, 2)) V = -4.81254148429273    Best Action: 4  
state((2, 1, 3)) V = -4.9707234575291155    Best Action: 3  
state((2, 1, 4)) V = -2.8965036412614737    Best Action: 1  
state((2, 4, 1)) V = -9.68830384840391    Best Action: 1  
state((2, 4, 2)) V = -11.093664905945    Best Action: 4  
state((2, 4, 3)) V = -7.24997822668428    Best Action: 1  
state((2, 4, 4)) V = -69.59238732825366    Best Action: 3  
state((2, 5, 1)) V = -4.814694567300199    Best Action: 4  
state((2, 5, 2)) V = -5.217133678524488    Best Action: 3  
state((2, 5, 3)) V = -5.001347191234482    Best Action: 4  
state((2, 5, 4)) V = -5.004220349538073    Best Action: 3  
state((3, 1, 1)) V = -2.0686510911780744    Best Action: 4

state((3, 1, 2)) V = -2.0686510911780744    Best Action: 3  
state((3, 1, 3)) V = -3.062197241172572    Best Action: 3  
state((3, 1, 4)) V = 5.038205909722947    Best Action: 2  
state((3, 3, 1)) V = 58.50625617874828    Best Action: 2  
state((3, 3, 2)) V = 42.10446755344829    Best Action: 4  
state((3, 3, 3)) V = 49.93923824764765    Best Action: 4  
state((3, 3, 4)) V = 46.21605651378948    Best Action: 3  
state((3, 4, 1)) V = 54.33652449944506    Best Action: 1  
state((3, 4, 2)) V = 31.56718554866535    Best Action: 1  
state((3, 4, 3)) V = 39.529710972111936    Best Action: 4  
state((3, 4, 4)) V = -20.94845840237319    Best Action: 3  
state((3, 5, 1)) V = 71.636212217586    Best Action: 4  
state((3, 5, 2)) V = 67.43078743556612    Best Action: 3  
state((3, 5, 3)) V = 62.432081053359695    Best Action: 4  
state((3, 5, 4)) V = 83.34139522466347    Best Action: 2  
state((4, 1, 1)) V = -8.62238061699112    Best Action: 4  
state((4, 1, 2)) V = -2.2656153005804907    Best Action: 3  
state((4, 1, 3)) V = -3.530444962832516    Best Action: 3  
state((4, 1, 4)) V = 5.389206817504209    Best Action: 1  
state((4, 2, 1)) V = -67.3516457857542    Best Action: 4  
state((4, 2, 2)) V = -4.506849711493863    Best Action: 1  
state((4, 2, 3)) V = -9.169810128919561    Best Action: 3  
state((4, 2, 4)) V = 1.758781589028089    Best Action: 1  
state((4, 3, 1)) V = -37.287780675619814    Best Action: 3  
state((4, 3, 2)) V = 24.52663643260091    Best Action: 4  
state((4, 3, 3)) V = 35.83090455685007    Best Action: 1  
state((4, 3, 4)) V = 17.17821122272772    Best Action: 4  
state((4, 5, 1)) V = 67.9855007440958    Best Action: 4  
state((4, 5, 2)) V = -0.1826077526529417    Best Action: 3  
state((4, 5, 3)) V = 55.32718398049131    Best Action: 4  
state((4, 5, 4)) V = 79.43319638206599    Best Action: 1  
state((5, 1, 1)) V = 20.890520321065875    Best Action: 2  
state((5, 1, 2)) V = 7.890661013067865    Best Action: 3  
state((5, 1, 3)) V = 14.404332982783952    Best Action: 4  
state((5, 1, 4)) V = 16.71910712536561    Best Action: 3  
state((5, 2, 1)) V = 21.691221286625275    Best Action: 1  
state((5, 2, 2)) V = 8.187226345874903    Best Action: 3  
state((5, 2, 3)) V = 15.480463444149581    Best Action: 4  
state((5, 2, 4)) V = 17.0984851861928    Best Action: 3  
state((5, 3, 1)) V = 32.42839290980924    Best Action: 3  
state((5, 3, 2)) V = 29.477163476362897    Best Action: 4  
state((5, 3, 3)) V = 39.93071969305372    Best Action: 2  
state((5, 3, 4)) V = 27.3975784164057    Best Action: 3  
state((5, 4, 1)) V = 79.42852959063094    Best Action: 1  
state((5, 4, 2)) V = 54.81012982475315    Best Action: 3  
state((5, 4, 3)) V = -0.2030392716373015    Best Action: 4  
state((5, 4, 4)) V = 67.96460472189635    Best Action: 3  
**iter 8:**  
state((1, 1, 1)) V = -4.022825447234784    Best Action: 4  
state((1, 1, 2)) V = -3.9262159558197354    Best Action: 3  
state((1, 1, 3)) V = -4.31900281719452    Best Action: 3  
state((1, 1, 4)) V = 1.2052085249752666    Best Action: 2  
state((1, 2, 1)) V = -5.148568273521526    Best Action: 3  
state((1, 2, 2)) V = -5.057204078988098    Best Action: 3  
state((1, 2, 3)) V = -5.228480053486413    Best Action: 3  
state((1, 2, 4)) V = -5.228480053486413    Best Action: 4  
state((1, 3, 1)) V = -5.171653133878487    Best Action: 3  
state((1, 3, 2)) V = -5.086902278119199    Best Action: 3  
state((1, 3, 3)) V = -5.2678610849439345    Best Action: 3  
state((1, 3, 4)) V = -5.2678610849439345    Best Action: 4  
state((1, 4, 1)) V = -6.126605855096233    Best Action: 3  
state((1, 4, 2)) V = -6.0852678902575645    Best Action: 4  
state((1, 4, 3)) V = -6.1475777589222815    Best Action: 3  
state((1, 4, 4)) V = -10.849576657524619    Best Action: 4  
state((1, 5, 1)) V = -5.0879098141623    Best Action: 3

state((1, 5, 2)) V = -5.152338091045856    Best Action: 4  
state((1, 5, 3)) V = -5.233527068475769    Best Action: 4  
state((1, 5, 4)) V = -5.219519396478091    Best Action: 3  
state((2, 1, 1)) V = -3.4835396812379402    Best Action: 4  
state((2, 1, 2)) V = -3.4835396812379402    Best Action: 3  
state((2, 1, 3)) V = -3.996434926537388    Best Action: 3  
state((2, 1, 4)) V = 2.0936313417946257    Best Action: 1  
state((2, 4, 1)) V = -10.041309330965609    Best Action: 1  
state((2, 4, 2)) V = -11.41084021969782    Best Action: 4  
state((2, 4, 3)) V = -7.533684038730996    Best Action: 1  
state((2, 4, 4)) V = -69.82295689997841    Best Action: 3  
state((2, 5, 1)) V = -5.1488839169163    Best Action: 4  
state((2, 5, 2)) V = -5.550038328504994    Best Action: 3  
state((2, 5, 3)) V = -5.305055919946513    Best Action: 4  
state((2, 5, 4)) V = -5.30858543327148    Best Action: 3  
state((3, 1, 1)) V = 3.0770069696710123    Best Action: 4  
state((3, 1, 2)) V = 3.0770069696710123    Best Action: 3  
state((3, 1, 3)) V = 0.77700215709812    Best Action: 3  
state((3, 1, 4)) V = 10.3491250181195    Best Action: 2  
state((3, 3, 1)) V = 58.81676626523059    Best Action: 2  
state((3, 3, 2)) V = 43.12580727214642    Best Action: 4  
state((3, 3, 3)) V = 50.28849014181793    Best Action: 4  
state((3, 3, 4)) V = 46.89914929443976    Best Action: 3  
state((3, 4, 1)) V = 54.54831735219814    Best Action: 1  
state((3, 4, 2)) V = 32.60815021516586    Best Action: 1  
state((3, 4, 3)) V = 39.732683716183274    Best Action: 4  
state((3, 4, 4)) V = -20.69229666431643    Best Action: 3  
state((3, 5, 1)) V = 71.68481911922666    Best Action: 4  
state((3, 5, 2)) V = 67.61454928866338    Best Action: 3  
state((3, 5, 3)) V = 62.4964488682418    Best Action: 4  
state((3, 5, 4)) V = 83.35680572962451    Best Action: 2  
state((4, 1, 1)) V = -3.5603992739278842    Best Action: 4  
state((4, 1, 2)) V = 3.018846457938136    Best Action: 3  
state((4, 1, 3)) V = 0.6640906661448737    Best Action: 3  
state((4, 1, 4)) V = 10.66757491014558    Best Action: 1  
state((4, 2, 1)) V = -62.48581464057199    Best Action: 4  
state((4, 2, 2)) V = -0.1149837184605918    Best Action: 3  
state((4, 2, 3)) V = -5.3797234148484065    Best Action: 3  
state((4, 2, 4)) V = 7.014574380871991    Best Action: 1  
state((4, 3, 1)) V = -36.07498955290812    Best Action: 3  
state((4, 3, 2)) V = 26.217051300960144    Best Action: 4  
state((4, 3, 3)) V = 36.35366887864447    Best Action: 1  
state((4, 3, 4)) V = 18.780360659444604    Best Action: 4  
state((4, 5, 1)) V = 68.00404681181614    Best Action: 4  
state((4, 5, 2)) V = -0.16609069790883524    Best Action: 3  
state((4, 5, 3)) V = 55.35425887192411    Best Action: 4  
state((4, 5, 4)) V = 79.43617324536395    Best Action: 1  
state((5, 1, 1)) V = 24.392305877986676    Best Action: 2  
state((5, 1, 2)) V = 13.220525463395967    Best Action: 3  
state((5, 1, 3)) V = 17.73607837940205    Best Action: 4  
state((5, 1, 4)) V = 19.997872220449327    Best Action: 3  
state((5, 2, 1)) V = 25.03708740601791    Best Action: 1  
state((5, 2, 2)) V = 13.574498501848149    Best Action: 3  
state((5, 2, 3)) V = 1



state((1, 1, 2)) V = -0.12666072043880633    Best Action: 3  
state((1, 1, 3)) V = -1.2929263403869213    Best Action: 3  
state((1, 1, 4)) V = 5.736924439785937    Best Action: 2  
state((1, 2, 1)) V = -5.3967908399476165    Best Action: 3  
state((1, 2, 2)) V = -2.577221088888283    Best Action: 1  
state((1, 2, 3)) V = -3.374880933918631    Best Action: 3  
state((1, 2, 4)) V = -3.374880933918631    Best Action: 4  
state((1, 3, 1)) V = -5.419941227537648    Best Action: 3  
state((1, 3, 2)) V = -2.9485966825374503    Best Action: 2  
state((1, 3, 3)) V = -3.564034152755245    Best Action: 3  
state((1, 3, 4)) V = -3.564034152755245    Best Action: 4  
state((1, 4, 1)) V = -6.40285255246511    Best Action: 3  
state((1, 4, 2)) V = -4.703564675362225    Best Action: 1  
state((1, 4, 3)) V = -5.056745190181273    Best Action: 3  
state((1, 4, 4)) V = -9.868334595244598    Best Action: 4  
state((1, 5, 1)) V = -5.359602442912662    Best Action: 3  
state((1, 5, 2)) V = -4.795749873773474    Best Action: 2  
state((1, 5, 3)) V = -5.04671107910175    Best Action: 3  
state((1, 5, 4)) V = -5.059494521447843    Best Action: 4  
state((2, 1, 1)) V = 0.6173182844585087    Best Action: 4  
state((2, 1, 2)) V = 0.6173182844585087    Best Action: 3  
state((2, 1, 3)) V = -0.6895208629060119    Best Action: 3  
state((2, 1, 4)) V = 6.691800208437956    Best Action: 1  
state((2, 4, 1)) V = -10.337101998302796    Best Action: 1  
state((2, 4, 2)) V = -11.663722691807147    Best Action: 4  
state((2, 4, 3)) V = -6.852614235973427    Best Action: 1  
state((2, 4, 4)) V = -70.03696209421157    Best Action: 3  
state((2, 5, 1)) V = -5.426124600608588    Best Action: 4  
state((2, 5, 2)) V = -5.8255574636622445    Best Action: 3  
state((2, 5, 3)) V = -5.5323189802273935    Best Action: 4  
state((2, 5, 4)) V = -5.561044498891482    Best Action: 3  
state((3, 1, 1)) V = 7.687069958798792    Best Action: 4  
state((3, 1, 2)) V = 7.687069958798792    Best Action: 3  
state((3, 1, 3)) V = 4.745495883541919    Best Action: 3  
state((3, 1, 4)) V = 13.799476413906856    Best Action: 2  
state((3, 3, 1)) V = 58.941379701300335    Best Action: 2  
state((3, 3, 2)) V = 43.53501033378688    Best Action: 4  
state((3, 3, 3)) V = 50.42861275619136    Best Action: 4  
state((3, 3, 4)) V = 47.21638740881005    Best Action: 3  
state((3, 4, 1)) V = 54.62650189020978    Best Action: 1  
state((3, 4, 2)) V = 33.023337051806664    Best Action: 1  
state((3, 4, 3)) V = 39.91681969035181    Best Action: 4  
state((3, 4, 4)) V = -20.59572288246619    Best Action: 3  
state((3, 5, 1)) V = 71.70353015269333    Best Action: 4  
state((3, 5, 2)) V = 67.68771443923833    Best Action: 3  
state((3, 5, 3)) V = 62.52123856289999    Best Action: 4  
state((3, 5, 4)) V = 83.36279867821719    Best Action: 2  
state((4, 1, 1)) V = 0.9048847367446609    Best Action: 4  
state((4, 1, 2)) V = 7.676867256246849    Best Action: 3  
state((4, 1, 3)) V = 4.741212039852571    Best Action: 3  
state((4, 1, 4)) V = 13.998142565576275    Best Action: 1  
state((4, 2, 1)) V = -58.36361986312709    Best Action: 4  
state((4, 2, 2)) V = 4.485019907223545    Best Action: 3  
state((4, 2, 3)) V = -1.357256933622724    Best Action: 3  
state((4, 2, 4)) V = 10.334352220650388    Best Action: 1  
state((4, 3, 1)) V = -35.52970280765045    Best Action: 3  
state((4, 3, 2)) V = 27.2836118600492    Best Action: 4  
state((4, 3, 3)) V = 36.59150780299819    Best Action: 1  
state((4, 3, 4)) V = 19.88376393911449    Best Action: 1  
state((4, 5, 1)) V = 68.00938462295525    Best Action: 4  
state((4, 5, 2)) V = -0.1613318395637009    Best Action: 3  
state((4, 5, 3)) V = 55.362500269239305    Best Action: 4  
state((4, 5, 4)) V = 79.43702612900441    Best Action: 1  
state((5, 1, 1)) V = 26.113934182517525    Best Action: 2  
state((5, 1, 2)) V = 16.406417657284138    Best Action: 3

state((5, 1, 3)) V = 19.664249999417382    Best Action: 4  
state((5, 1, 4)) V = 21.668244356148385    Best Action: 3  
state((5, 2, 1)) V = 26.65774779402981    Best Action: 1  
state((5, 2, 2)) V = 16.77725351424365    Best Action: 3  
state((5, 2, 3)) V = 20.563870393035632    Best Action: 4  
state((5, 2, 4)) V = 22.068022520017962    Best Action: 3  
state((5, 3, 1)) V = 34.29410598679752    Best Action: 3  
state((5, 3, 2)) V = 32.142761699273855    Best Action: 4  
state((5, 3, 3)) V = 40.703125859378545    Best Action: 2  
state((5, 3, 4)) V = 29.41525092095759    Best Action: 3  
state((5, 4, 1)) V = 79.4321315576298    Best Action: 1  
state((5, 4, 2)) V = 54.839621103033366    Best Action: 3  
state((5, 4, 3)) V = -0.19709493607776096    Best Action: 4  
state((5, 4, 4)) V = 67.97125668405874    Best Action: 3  
**iter 10:**  
state((1, 1, 1)) V = 2.717510730171436    Best Action: 4  
state((1, 1, 2)) V = 3.762684318142978    Best Action: 3  
state((1, 1, 3)) V = 2.0748331697774813    Best Action: 3  
state((1, 1, 4)) V = 8.931599839624045    Best Action: 2  
state((1, 2, 1)) V = -4.159691631958576    Best Action: 3  
state((1, 2, 2)) V = 0.5717439965231119    Best Action: 1  
state((1, 2, 3)) V = -0.7666530449996533    Best Action: 3  
state((1, 2, 4)) V = -0.7666530449996533    Best Action: 4  
state((1, 3, 1)) V = -4.289075163968859    Best Action: 3  
state((1, 3, 2)) V = 0.18430638566154015    Best Action: 2  
state((1, 3, 3)) V = -0.9401696164090305    Best Action: 3  
state((1, 3, 4)) V = -0.9401696164090305    Best Action: 4  
state((1, 4, 1)) V = -5.641460504779744    Best Action: 3  
state((1, 4, 2)) V = -2.1464387256894026    Best Action: 1  
state((1, 4, 3)) V = -2.9339352659523548    Best Action: 3  
state((1, 4, 4)) V = -7.991465973055222    Best Action: 4  
state((1, 5, 1)) V = -5.246539516332713    Best Action: 3  
state((1, 5, 2)) V = -2.3705709730034585    Best Action: 2  
state((1, 5, 3)) V = -3.0843400106748002    Best Action: 3  
state((1, 5, 4)) V = -3.2200587942237333    Best Action: 4  
state((2, 1, 1)) V = 4.592832277563665    Best Action: 4  
state((2, 1, 2)) V = 4.592832277563665    Best Action: 3  
state((2, 1, 3)) V = 2.801880817025552    Best Action: 3  
state((2, 1, 4)) V = 9.857300712479702    Best Action: 1  
state((2, 4, 1)) V = -10.528901387849027    Best Action: 1  
state((2, 4, 2)) V = -11.176902971578384    Best Action: 4  
state((2, 4, 3)) V = -5.313635293599931    Best Action: 1  
state((2, 4, 4)) V = -70.14859917758605    Best Action: 3  
state((2, 5, 1)) V = -5.645972936782473    Best Action: 3  
state((2, 5, 2)) V = -6.004891482581703    Best Action: 4  
state((2, 5, 3)) V = -4.758174393294599    Best Action: 1  
state((2, 5, 4)) V = -5.7600519228649185    Best Action: 3  
state((3, 1, 1)) V = 10.802818014841062    Best Action: 4  
state((3, 1, 2)) V = 10.802818014841062    Best Action: 3  
state((3, 1, 3)) V = 7.709015535463161    Best Action: 3  
state((3, 1, 4)) V = 15.749135823444668    Best Action: 2  
state((3, 3, 1)) V = 58.99261495330095    Best Action: 2  
state((3, 3, 2)) V = 43.70140311679761    Best Action: 4  
state((3, 3, 3)) V = 50.485960543233695    Best Action: 4  
state((3, 3, 4)) V = 47.37725390385102    Best Action: 3  
state((3, 4, 1)) V = 54.66305800556723    Best Action: 1  
state((3, 4, 2)) V = 33.19761615857878    Best Action: 1  
state((3, 4, 3)) V = 40.165623551532306    Best Action: 4  
state((3, 4, 4)) V = -20.553151306091493    Best Action: 3  
state((3, 5, 1)) V = 71.71076410450532    Best Action: 4  
state((3, 5, 2)) V = 67.71740373658155    Best Action: 3  
state((3, 5, 3)) V = 62.53089010326092    Best Action: 4  
state((3, 5, 4)) V = 83.36517695347237    Best Action: 2  
state((4, 1, 1)) V = 3.911962231171306    Best Action: 4  
state((4, 1, 2)) V = 10.795835652942115    Best Action: 3

state((4, 1, 3)) V = 7.675128510967352    Best Action: 3  
state((4, 1, 4)) V = 15.841503939831755    Best Action: 1  
state((4, 2, 1)) V = -55.61818121546448    Best Action: 4  
state((4, 2, 2)) V = 7.583342174717241    Best Action: 3  
state((4, 2, 3)) V = 1.4506182636535454    Best Action: 3  
state((4, 2, 4)) V = 12.15577064064057    Best Action: 1  
state((4, 3, 1)) V = -35.25953738062012    Best Action: 3  
state((4, 3, 2)) V = 27.924694092787195    Best Action: 4  
state((4, 3, 3)) V = 36.704595835361374    Best Action: 1  
state((4, 3, 4)) V = 20.5027630329464    Best Action: 1  
state((4, 5, 1)) V = 68.0109460886323    Best Action: 4  
state((4, 5, 2)) V = -0.15993774800293892    Best Action: 3  
state((4, 5, 3)) V = 55.36508328866928    Best Action: 4  
state((4, 5, 4)) V = 79.43727410996111    Best Action: 1  
state((5, 1, 1)) V = 26.927037464284204    Best Action: 2  
state((5, 1, 2)) V = 18.09697722426342    Best Action: 3  
state((5, 1, 3)) V = 20.716152899482307    Best Action: 4  
state((5, 1, 4)) V = 22.47706038735621    Best Action: 3  
state((5, 2, 1)) V = 27.413505950816695    Best Action: 1  
state((5, 2, 2)) V = 18.460778329752042    Best Action: 3  
state((5, 2, 3)) V = 21.489015726366947    Best Action: 4  
state((5, 2, 4)) V = 22.86829104947795    Best Action: 3  
state((5, 3, 1)) V = 34.53984638598119    Best Action: 3  
state((5, 3, 2)) V = 32.572809717004695    Best Action: 4  
state((5, 3, 3)) V = 40.8037212837485    Best Action: 2  
state((5, 3, 4)) V = 29.689317770502647    Best Action: 3  
state((5, 4, 1)) V = 79.43236996264872    Best Action: 1  
state((5, 4, 2)) V = 54.84157264669149    Best Action: 3  
state((5, 4, 3)) V = -0.19670153078667596    Best Action: 4  
state((5, 4, 4)) V = 67.97169691504925    Best Action: 3

**Result: [((1, 1, 4), (13.037595531961905, 2)), ((3, 1, 4), (17.735035194058202, 2)), ((5, 1, 4), (23.187304977843954, 3)), ((5, 1, 1), (27.624299272635582, 2)), ((5, 3, 1), (34.734895191420236, 3)), ((5, 3, 3), (40.88731531557343, 2)), ((3, 3, 3), (50.53366647701077, 4)), ((3, 3, 1), (59.03692218472277, 2)), ((3, 5, 1), (71.7157987204508, 4)), ((3, 5, 4), (83.36708497268832, 2)), ((5, 5, 4), None)]**