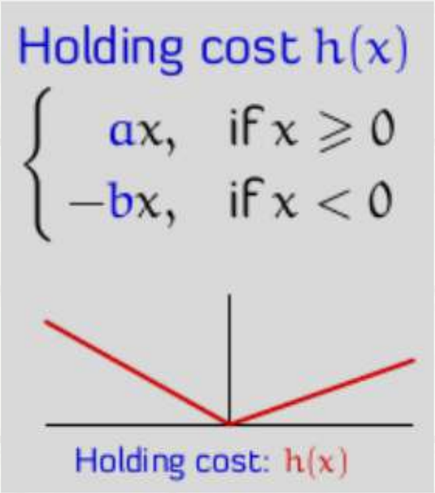
RL Project presentation

Team ID 4

Inventory Management problem

- Retail stores stockpile products in warehouses to meet the random demand. Additional stocks are procured at regular intervals.
- Let Xt denote the amount of stock before the t-th procurement.
- At time t, the store may procure an additional stock Ut ($\leq$ U) units for price p per unit. Thus the total procurement cost is pUt.

- The random demand Wt is i.i.d. with distribution Pw . Pw is uniform[0,10].
- The stock available at the next time is X(t+1) = Xt + Ut - Wt, where a negative stock denotes backlogged demand.
- Per-stage cost is c(Xt+1, Ut) = h(Xt+1) + pUt. The holding cost for the stock is given by h(x) where a is the per-unit storage cost and b is the per-unit backlog cost.
- We want to find the optimal inventory control strategy to minimize the expected total cost over a finite horizon.



State Space

Our states are X's (the amount of inventory). Our state's range is X's range.

$$-10 <= X <= 10$$

Actions

The amount of inventory produced in that day is taken as the action. U(t) is our action on t^{th} day.

$$0 <= U <= 10$$

Value Iteration

Pseudo code value iteration

Algorithm 1 Finite Horizon Value Iteration

for $t = T-1, T-2, \ldots, 0$ **do**
  for $s \in S$ **do**
    $\pi_t(s), V_t(s) = maximize_a E[r_t + V_{t+1}(s_{t+1})]$
  end for
end for

Experimenting with various parameters

a = 0, b = 5, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9
1	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	
2	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	
3	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	
4	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	
5	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	
6	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0
7	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0
8	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0
9	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0
10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0

a = 0, b = 5, p = 0

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3
1	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	
2	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	
3	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	
4	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	
5	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	
6	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	
7	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	
8	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	
9	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0
10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0

a = 5, b = 0, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
8	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

a = 5, b = 0, p = 0

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
3	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0

a = 5, b = 5, p = 10

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
1	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0
2	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0
3	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
4	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
9	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

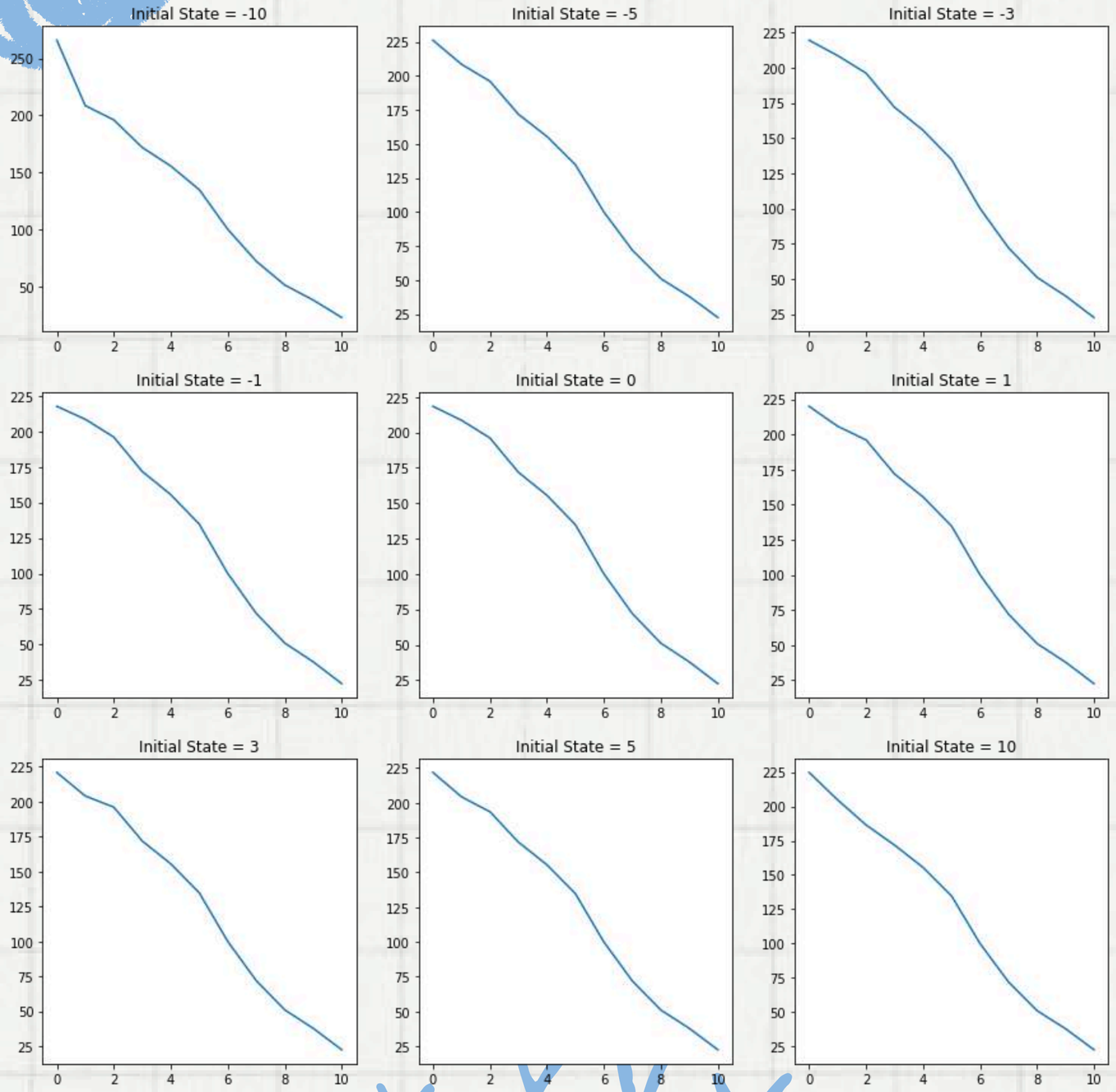
a = 5, b = 5, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0
1	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
2	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
3	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
4	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
6	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
7	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
8	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0
9	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0

a = 5, b = 5, p = 0

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0
1	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
2	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
3	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
4	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
6	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
7	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
8	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0
9	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0

Cost vs Time:


Initial State = -10
Initial State = -5
Initial State = -3
Initial State = -1
Initial State = 0
Initial State = 1
Initial State = 3
Initial State = 5
Initial State = 10

Policy Iteration

Pseudo code policy iteration

Algorithm 1 Policy Iteration

1: Initialize $\pi_t^{(0)}$.
2: **for** $n = 1, 2, \ldots$ **do**
3: Policy evaluation step:
4: $V_t(s) = Solve[V_{(t+1)}(s'), \pi_t]$
5: Policy improvement step:
6: $\pi_t(s) = argmax_a(r(s,a) + V_{(t+1)}(s'))$
7: **end for**

a = 0, b = 5, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
1	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3
2	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3
3	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3
4	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8
5	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5
6	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6
7	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2
8	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1
9	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0
10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0

a = 0, b = 5, p = 0

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	
1	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2
2	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	
3	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	
4	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	8	7	6	
5	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	
6	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	
7	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	
8	10	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	
9	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0
10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	

a = 5, b = 0, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
3	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

a = 5, b = 0, p = 0

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
8	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0

a = 5, b = 5, p = 10

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
1	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0
2	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0
3	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
6	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
7	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
8	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
9	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

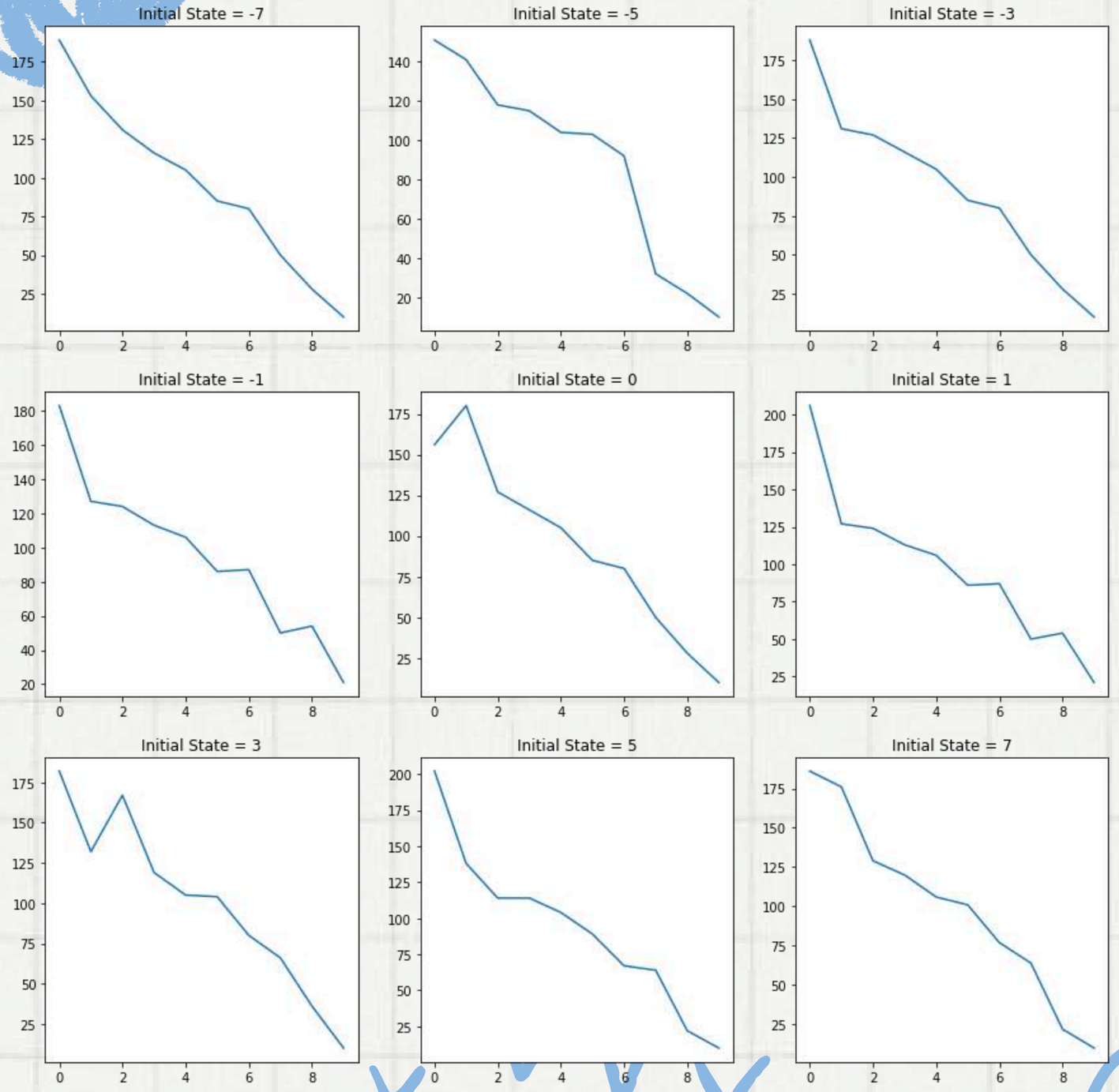
a = 5, b = 5, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
1	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
2	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
3	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0
4	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0
5	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0
6	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0

a = 5, b = 5, p = 0

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0
1	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0
2	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
3	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0
4	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0
5	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0
6	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
7	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
8	10	10	10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0
9	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0
10	10	10	10	10	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0

Cost vs Time:


Initial State = -7
Initial State = -5
Initial State = -3
Initial State = -1
Initial State = 0
Initial State = 1
Initial State = 3
Initial State = 5
Initial State = 7

MC control

Pseudo code MC control

Algorithm 1 Monte Carlo Control

```
procedure MonteCarlo(num_episodes, ε)
    Input: Number of episodes num_episodes, exploration rate ε
    Output: Q-value function Q
    Initialize Q arbitrarily
    Initialize empty list rewards_per_episode
    for episode_num = 1 to num_episodes do
        Initialize empty episode list
        Initialize total_episode_cost = 0
        Initialize initial state
        while episode is not terminated do
            Choose action using exploration-exploitation strategy based on Q and ε
            Take action, observe next state and reward
            Add (state, action, reward) tuple to episode
            Update total_episode_cost
            Update Q-value using episode (backward pass)
        end while
        Update ε (if applicable)
        Append total_episode_cost to rewards_per_episode
    end for
    return Q, rewards_per_episode
end procedure
```

a = 0, b = 5, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	9	9	10	10	9	10	10	5	8	10	0	10	5	4	9	1	1	0	5	1	4
1	1	1	0	9	8	0	10	9	10	9	10	7	10	9	5	8	7	5	9	9	8
2	10	9	9	8	9	5	7	7	6	5	6	6	6	6	3	1	0	9	2	0	1
3	9	8	10	6	10	10	8	9	8	8	9	5	9	4	10	10	2	1	3	3	0
4	1	9	10	9	8	10	9	9	9	8	5	10	5	7	4	6	1	1	4	0	1
5	10	9	1	9	5	10	8	8	10	9	8	8	5	8	3	0	0	2	4	0	5
6	10	9	9	9	6	7	6	9	10	8	8	5	1	1	0	3	0	0	0	0	0
7	0	1	10	10	7	8	7	10	9	8	10	9	10	5	4	6	2	5	0	2	0
8	9	9	9	10	9	9	7	6	5	5	4	0	2	0	0	1	0	3	1	2	2
9	9	10	8	7	6	7	6	4	3	3	2	1	0	0	0	0	0	0	0	0	0

a = 0, b = 5, p = 0

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	9	10	9	9	5	7	6	8	3	2	5	3	4	3	2	0	5	0	0
1	7	9	10	9	6	6	10	5	3	2	4	1	0	7	2	3	2	3	1	0	3
2	8	10	10	10	9	6	6	6	7	7	6	3	6	3	3	0	1	1	0	1	1
3	10	10	10	10	10	10	9	7	5	5	8	10	6	3	6	0	1	1	2	0	2
4	10	10	10	9	10	8	10	9	7	8	9	7	5	7	5	5	2	6	2	0	2
5	10	8	8	9	10	8	10	10	8	6	6	6	6	4	4	2	2	3	1	0	1
6	9	9	10	8	8	8	8	10	10	6	7	7	4	4	2	3	3	3	0	0	1
7	10	9	8	10	10	3	10	6	7	4	9	7	2	7	5	4	5	3	0	1	0
8	10	7	9	9	10	8	3	9	9	7	10	8	3	8	6	7	5	4	3	2	1
9	10	8	5	10	10	10	10	9	9	7	6	5	4	3	2	1	0	0	0	2	0

a = 5, b = 0, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	5	0	9	8	9	9	8	8	10	10	4	6	1	6	5	3	3	3	0
1	0	4	0	0	0	7	10	10	9	8	7	1	9	10	9	10	6	7	7	9	6
2	3	0	0	0	10	1	10	8	9	10	6	8	10	8	10	8	8	10	2	6	6
3	0	0	10	9	9	8	10	10	10	8	5	10	10	5	9	10	3	4	9	3	4
4	0	10	8	0	10	9	7	10	8	7	6	5	10	3	4	4	8	8	6	0	5
5	10	7	10	9	10	10	7	10	9	10	5	8	6	5	4	3	1	8	0	0	1
6	10	3	9	10	10	8	10	10	7	10	7	7	7	6	5	4	1	3	0	4	0
7	2	0	1	0	0	10	10	10	9	8	7	7	6	6	3	3	1	1	1	0	0
8	0	2	0	2	0	6	2	1	8	4	3	3	1	0	0	6	0	2	1	0	0
9	0	0	0	1	0	0	0	1	0	7	0	0	0	0	1	2	0	0	0	0	0

a = 5, b = 0, p = 0

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	5	2	4	2	7	1	0	2	0	4	0	5	4	1	0	3	2	2	0	2	0
1	1	1	1	0	1	2	2	4	0	1	1	2	3	0	0	3	2	0	0	0	0
2	0	1	0	1	0	3	5	0	6	2	0	0	1	0	1	0	1	0	0	0	6
3	0	0	2	1	9	2	0	2	3	2	0	2	0	0	0	2	0	1	0	2	1
4	0	0	0	0	0	0	0	1	3	5	0	0	0	1	1	0	0	2	1	1	0
5	0	0	0	1	0	2	0	1	0	0	0	0	2	1	7	1	1	0	2	3	0
6	0	1	0	0	0	0	0	1	5	0	1	0	1	0	2	1	2	0	2	1	0
7	0	0	0	0	0	0	0	0	0	0	1	2	0	0	2	0	2	1	0	0	1
8	0	0	0	0	0	0	0	1	0	0	0	3	1	1	0	0	0	3	0	1	1
9	0	0	2	0	1	0	0	2	0	0	0	0	0	1	1	0	1	0	0	1	0

a = 5, b = 5, p = 10

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	9	8	10	6	5	2	4	3	2	0	0	1	1	1	0	1	5	0	1	1	0
1	10	7	8	6	7	5	3	4	3	2	0	2	0	1	0	0	1	0	4	0	1
2	10	10	10	7	6	7	7	4	3	2	1	1	0	0	2	0	0	0	1	4	1
3	10	10	9	10	7	7	7	4	5	4	1	2	1	2	1	0	2	2	0	8	0
4	9	10	10	9	10	10	9	10	9	8	4	6	3	3	5	0	1	1	1	2	1
5	9	9	10	10	9	9	10	10	8	5	5	5	7	3	1	3	1	0	1	2	0
6	10	10	10	10	10	9	7	7	5	5	4	2	2	1	0	0	0	0	0	0	1
7	10	10	9	9	8	7	6	5	4	3	2	1	1	0	1	0	0	1	0	1	2
8	2	0	1	0	10	10	10	10	10	9	8	7	6	5	4	2	2	1	0	1	0
9	0	0	0	0	0	10	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0

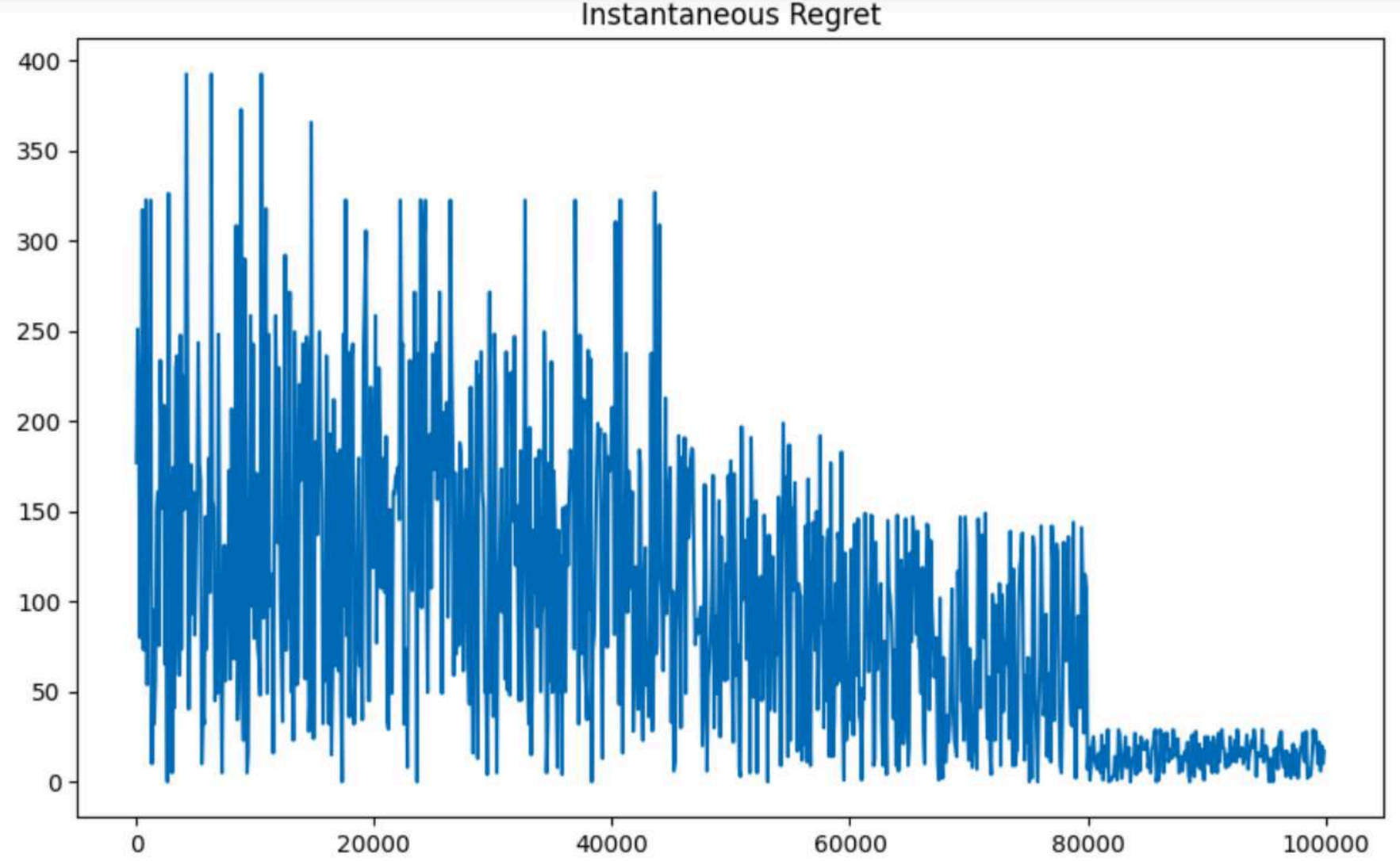
a = 5, b = 5, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	9	10	9	7	4	10	8	6	7	5	3	5	3	3	2	0	5	0	0	0
1	0	10	10	9	10	10	10	10	9	9	7	7	7	4	4	2	0	0	1	7	0
2	10	10	9	10	9	8	7	8	5	6	5	2	1	1	1	1	0	0	0	6	1
3	8	7	6	9	5	7	2	3	4	3	2	1	0	3	3	2	6	0	0	0	1
4	3	8	7	9	9	8	5	7	9	7	3	2	6	5	2	7	1	1	0	0	2
5	1	9	10	9	10	10	9	9	8	7	7	6	3	2	6	1	4	0	0	0	0
6	10	9	7	10	9	10	9	10	6	6	4	4	5	2	1	1	1	0	3	0	6
7	7	6	10	8	10	9	7	6	4	5	4	3	2	1	0	0	0	0	0	3	0
8	10	10	8	7	7	5	4	2	2	1	0	0	0	0	0	0	0	0	0	1	0
9	1	0	4	1	0	0	10	8	9	8	7	6	4	4	3	2	1	0	0	0	0

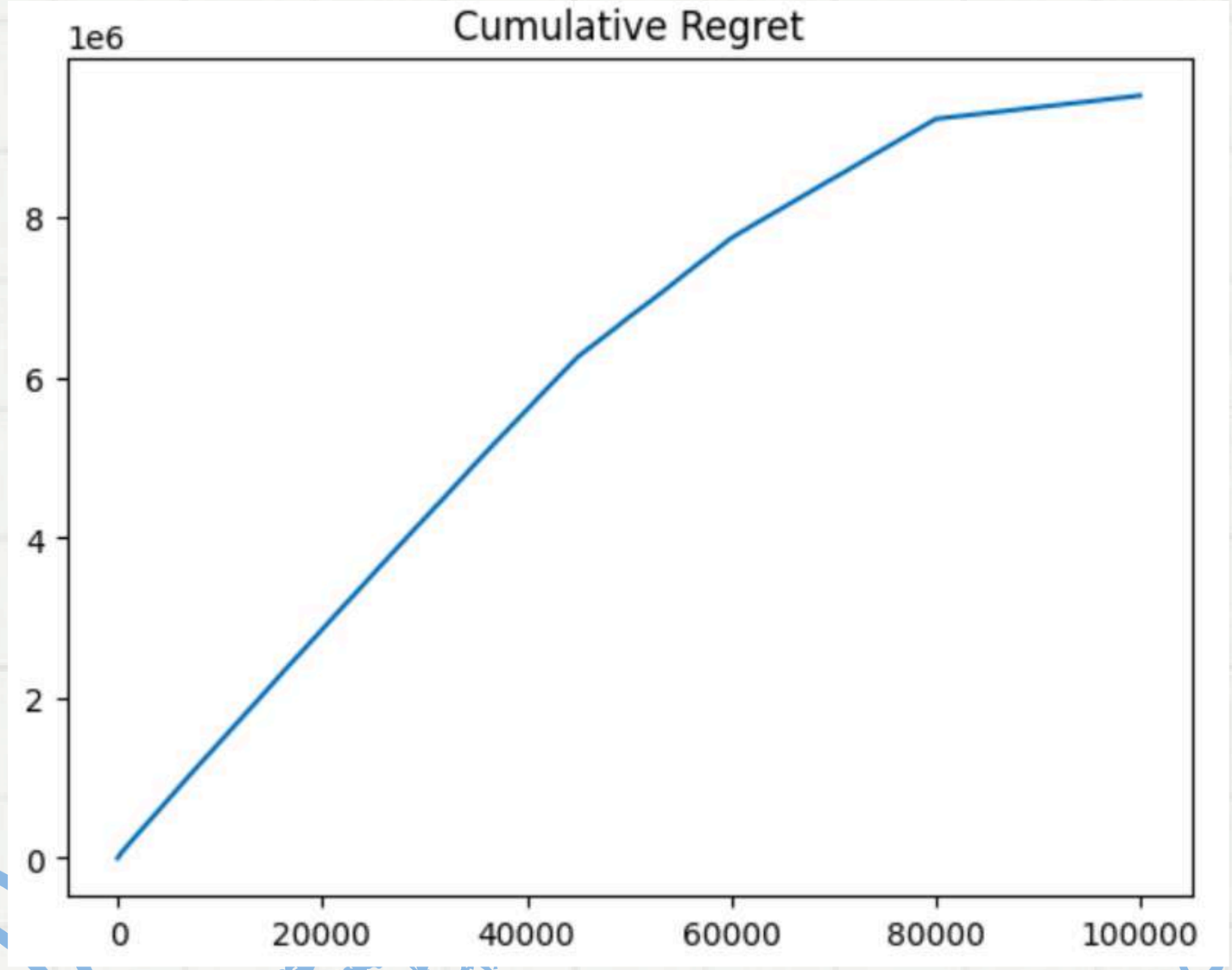
a = 5, b = 5, p = 0

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	5	0	9	8	9	9	8	8	10	10	4	6	1	6	5	3	3	3	0
1	0	4	0	0	0	7	10	10	9	8	7	1	9	10	9	10	6	7	7	9	6
2	3	0	0	0	10	1	10	8	9	10	6	8	10	8	10	8	8	10	2	6	6
3	0	0	10	9	9	8	10	10	10	8	5	10	10	5	9	10	3	4	9	3	4
4	0	10	8	0	10	9	7	10	8	7	6	5	10	3	4	4	8	8	6	0	5
5	10	7	10	9	10	10	7	10	9	10	5	8	6	5	4	3	1	8	0	0	1
6	10	3	9	10	10	8	10	10	7	10	7	7	7	6	5	4	1	3	0	4	0
7	2	0	1	0	0	10	10	10	9	8	7	7	6	6	3	3	1	1	1	0	0
8	0	2	0	2	0	6	2	1	8	4	3	3	1	0	0	6	0	2	1	0	0
9	0	0	0	1	0	0	0	1	0	7	0	0	0	0	1	2	0	0	0	0	0

Instantaneous regrets plot



Cumulative regrets plot



TD control using SARSA(λ)

Pseudo code TD control using SARSA(λ)

Algorithm 1 TD Control with SARSA(λ)

Initialize $Q(t, s, a)$ arbitrarily, for all $s \in S$, $a \in A(s)$, $t \leq Horizon$
repeat
    For each episode:
    $E(t, s, a) = 0$, for all $s \in S$, $a \in A(s)$
    Initialize T, S, A
    repeat
        For each step of the episode:
        Take action A, observe R, S'
        Choose A' from S' using policy derived from Q (e.g., ϵ-greedy)
        $\delta \leftarrow R + \gamma Q(t+1, S', A') - Q(t, S, A)$
        $E(t, S, A) \leftarrow E(t, S, A) + 1$
        for all $s \in S$, $a \in A(s)$ **do**
            $Q(t, s, a) \leftarrow Q(t, s, a) + \alpha\delta E(t, s, a)$
            $E(t, s, a) \leftarrow \gamma\lambda E(t, s, a)$
        end for
        $S \leftarrow S'$; $A \leftarrow A'$; $t \leftarrow t + 1$
    until S is terminal
until termination condition is met

Optimal policy :

	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
0	7	7	8	10	10	7	9	5	10	5	8	4	9	10	5	5	6	3	7	2	0
1	3	1	0	6	10	9	6	7	6	6	9	8	4	5	7	5	2	7	1	2	2
2	10	10	9	10	9	9	10	4	9	4	6	8	10	5	9	8	4	1	3	1	1
3	10	7	4	8	7	5	6	5	10	9	10	9	1	3	7	4	5	7	2	3	3
4	9	8	7	9	8	9	3	10	9	7	9	6	4	8	8	1	8	3	6	2	0
5	6	10	9	7	9	5	8	8	3	9	9	9	10	7	6	10	6	3	6	1	4
6	10	10	9	4	10	5	6	8	6	10	10	0	10	9	7	10	6	5	5	1	5
7	9	7	8	8	8	8	10	7	8	8	4	9	7	10	9	7	10	9	2	1	1
8	10	10	10	7	4	10	9	5	8	9	3	8	10	2	6	2	7	7	5	4	5
9	9	8	10	4	10	7	10	10	10	10	7	10	8	10	8	9	8	5	5	2	1

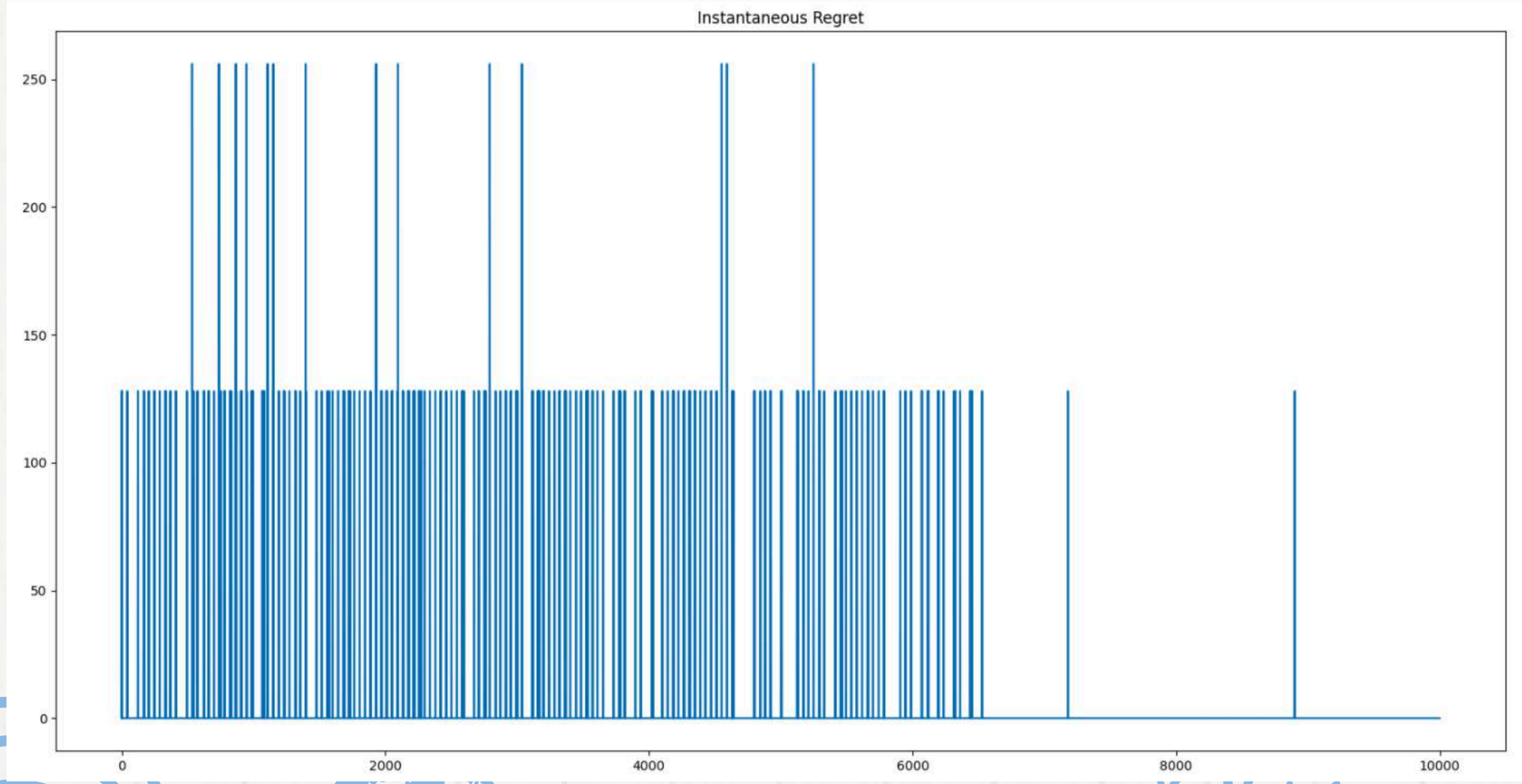
a = 2 , b = 3 , p = 1

Optimal policy :

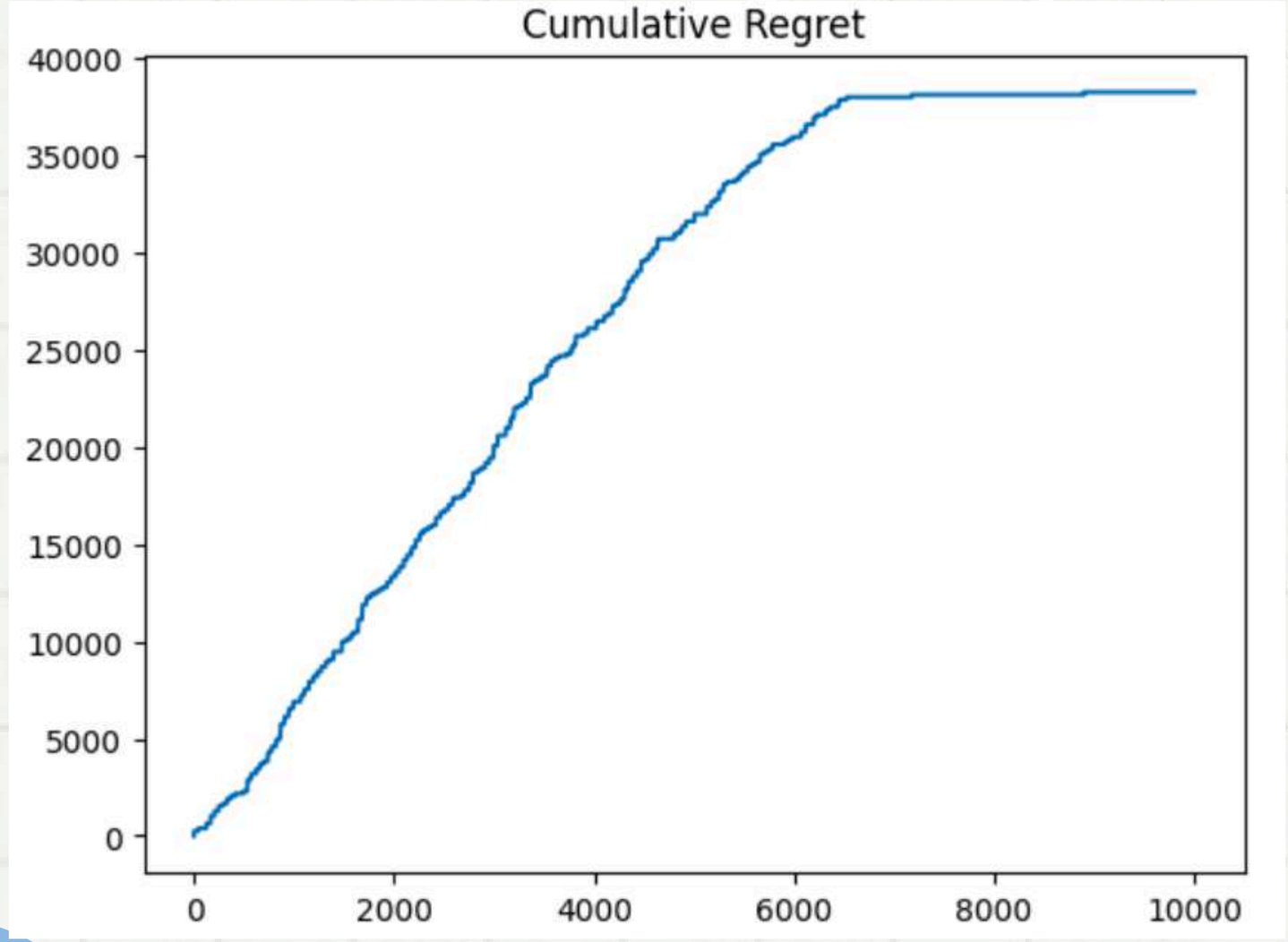
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	4	2	7	1	3	0	0	2	0	1	2	1	0	1	3	1	0	0	1	3
2	0	0	0	0	0	5	2	3	1	0	0	3	4	1	1	0	1	2	5	1
1	5	2	4	0	1	4	3	0	3	0	0	4	3	0	2	1	0	0	0	4
3	4	9	4	5	5	1	1	0	2	0	0	0	0	1	0	2	2	3	2	0
0	0	4	2	0	1	3	3	1	1	1	1	5	3	1	0	0	0	1	1	7
4	6	6	2	1	0	0	1	1	1	3	0	1	2	0	0	0	0	1	0	0
5	5	2	4	1	0	7	0	1	3	0	3	3	0	0	0	0	1	1	1	0
1	6	1	3	4	0	3	4	1	1	1	0	3	7	0	2	5	2	0	2	1
5	2	2	7	3	0	2	1	0	0	1	4	3	3	0	3	1	4	1	2	0
1	1	0	1	5	0	3	1	0	0	3	1	1	0	0	1	0	0	0	0	1

Not experimenting with various parameters since SARSA lambda for finite horizon is not clear and this algorithm is run by introducing time into states.

Instantaneous regrets plot



Cumulative regrets plot



Q Learning

Pseudo code Q Learning

Algorithm 1 Finite Horizon Q-Learning

Notation:

$Q_n^m(i,a)$: Q-value at state i, action a, stage n, recursion m.

$a(m)$: step-size at recursion index m

$Q_N(i,a)$: Q-value for state i and action a at terminal stage (N).

$g_n(i,a,j)$: Single stage reward for stage n where current state is i, action is a and next state is j.

$g_N(i)$: Terminal reward at the N^{th} stage when terminal state is i.

$A(j)$: Set of feasible actions in state j.

$\eta(i,a)$: Sampling function taking input (i,a) as state-action pair and returns the next state.

Input: Samples of the form $\big(i$ (current state), a (action), r (reward), j (next state)$\big)$.

Output: Updated Q-value $Q_n^{m+1}(i,a)$ estimated after m iterations of the algorithm.

Initialization: $Q_n^0(i,a) = 0, \forall (i,a), n = 0,\ldots,N-1,$ and $Q_N^0(i,a) = g_N(i), \forall (i,a)$

1: **procedure** FINITE HORIZON Q-LEARNING:

2: $a(m) = \left\lceil \frac{1}{(m+1)/10} \right\rceil$

3: $j = \eta(i,a)$ (from samples)

4: $Q_n^{m+1}(i,a) = \big(1 - a(m)\big)\Big(Q_n^m(i,a)\Big) + a(m)$

5: $\times \Big(g_n\big(i,a\big) + \min\limits_{b \in A(j)} Q_{n+1}^m\big(j,b\big)\Big), n = 0,1,\ldots,N-1,$

6: $Q_N^{m+1}(i,a) = g_N(i), \forall (i,a)$ tuples.

7: **return** $Q_n^{m+1}(i,a)$

Implementation of Q Learning

```
def update_queue_values(n, i, a, m):
    '''
    n (stage), i (current state), a (action), m (recursion)
    '''
    am = 10 / (m+1)
    j = next_state(i, a)
    r = cost_function(i, a, j)
    minQ = min([ Q[(n+1, j, b)] for b in range(0, U+1)])
    Q[(n,i,a)] = (1-am) * Q[(n,i,a)] + am * (r + minQ)
```

```
for m in range(num_itetations):

    for n in range(T):
        for i in range(-S, S+1):
            for a in range(0, U+1):
                update_queue_values(n, i, a, m)
```

FINITE HORIZON Q-LEARNING: STABILITY CONVERGENCE paper by Vivek et al presents a stochastic approximation version of the finite horizon DP algorithm in Q-value. Hence iterating for all s,a pairs and applying update eqn converges to optimal Q values as shown in experiments in further slides.

Experimenting with various parameters

a = 0, b = 5, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	9	9	10	10	9	6	5	7	5	3	5	4	3	1	0	2	2	2	1	0	0
1	10	7	10	10	7	9	8	7	4	5	4	2	5	4	0	1	3	1	1	0	0
2	9	10	10	9	9	9	7	6	5	5	5	5	3	3	2	1	2	0	0	1	1
3	10	10	10	9	10	9	7	7	4	4	6	4	3	2	3	0	1	0	0	0	0
4	7	9	10	10	8	9	6	6	4	6	1	5	3	1	3	1	0	0	0	0	0
5	10	9	10	9	10	7	9	7	5	5	5	5	5	3	2	3	0	2	0	0	0
6	9	10	8	10	8	7	8	5	6	6	1	5	3	0	1	0	1	0	0	0	1
7	10	10	9	9	10	8	8	5	7	3	4	3	4	1	0	1	1	0	0	0	0
8	0	10	10	8	6	8	7	3	5	4	4	5	3	0	1	1	0	0	0	0	0
9	0	0	0	0	0	7	6	5	3	0	1	1	0	1	0	0	0	0	0	0	0

a = 0, b = 5, p = 0

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	10	10	10	10	8	8	7	4	5	4	4	4	4	0	2	0	1	1	0	0
1	9	10	9	9	8	9	9	8	6	5	5	2	4	2	1	1	0	0	0	0	0
2	10	9	10	9	8	8	8	6	5	6	6	4	0	2	0	2	1	1	0	0	0
3	10	10	9	9	9	9	7	8	5	6	3	4	4	2	2	1	0	1	0	0	0
4	9	9	10	9	9	9	8	7	7	7	5	3	4	1	2	2	0	1	1	0	0
5	10	9	10	10	9	9	6	6	6	3	5	3	2	3	2	0	1	0	0	0	0
6	10	10	10	9	9	9	7	5	7	6	4	4	4	3	2	2	0	0	1	0	0
7	9	10	10	10	8	7	7	6	5	3	5	4	2	2	2	0	1	0	0	0	0
8	10	10	10	9	8	9	8	8	7	5	5	3	2	1	2	1	2	0	0	0	0
9	10	10	9	9	8	9	8	6	6	4	3	3	2	1	2	1	0	0	0	0	0

a = 5, b = 0, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

a = 5, b = 0, p = 0

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	0	0	1	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	3	0	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	10	7	7	7	4	3	3	2	0	1	0	0	0	0	0	0	0	0	0	0	0
3	9	9	6	6	6	4	2	3	2	0	0	0	0	0	0	0	0	0	0	0	0
4	9	8	8	4	6	4	3	1	2	0	0	0	0	0	0	0	0	0	0	0	0
5	6	7	2	5	2	1	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0
6	10	1	7	5	4	5	4	3	2	0	0	0	0	0	0	0	0	0	0	0	0
7	7	8	8	6	4	4	4	2	2	1	0	0	0	0	0	0	0	0	0	0	0
8	7	3	6	5	5	1	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

a = 5, b = 5, p = 10

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	5	2	5	5	3	2	4	3	0	0	0	1	1	2	1
1	0	0	0	0	0	0	6	4	5	3	2	0	0	1	0	0	1	0	1	0	0
2	0	0	0	0	0	0	1	2	6	3	6	4	5	2	3	2	2	0	1	0	0
3	0	0	0	0	0	2	0	4	0	0	3	0	0	0	3	1	1	0	0	1	0
4	0	0	0	0	0	0	0	7	2	2	1	2	3	4	3	1	2	0	1	0	0
5	0	0	0	0	0	0	4	3	2	4	1	0	3	2	1	0	0	1	0	1	0
6	0	0	0	0	0	0	0	2	1	1	1	1	3	0	0	1	0	1	0	1	1
7	0	0	0	0	0	0	0	1	2	2	0	1	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	1	1	2	1	1	1	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

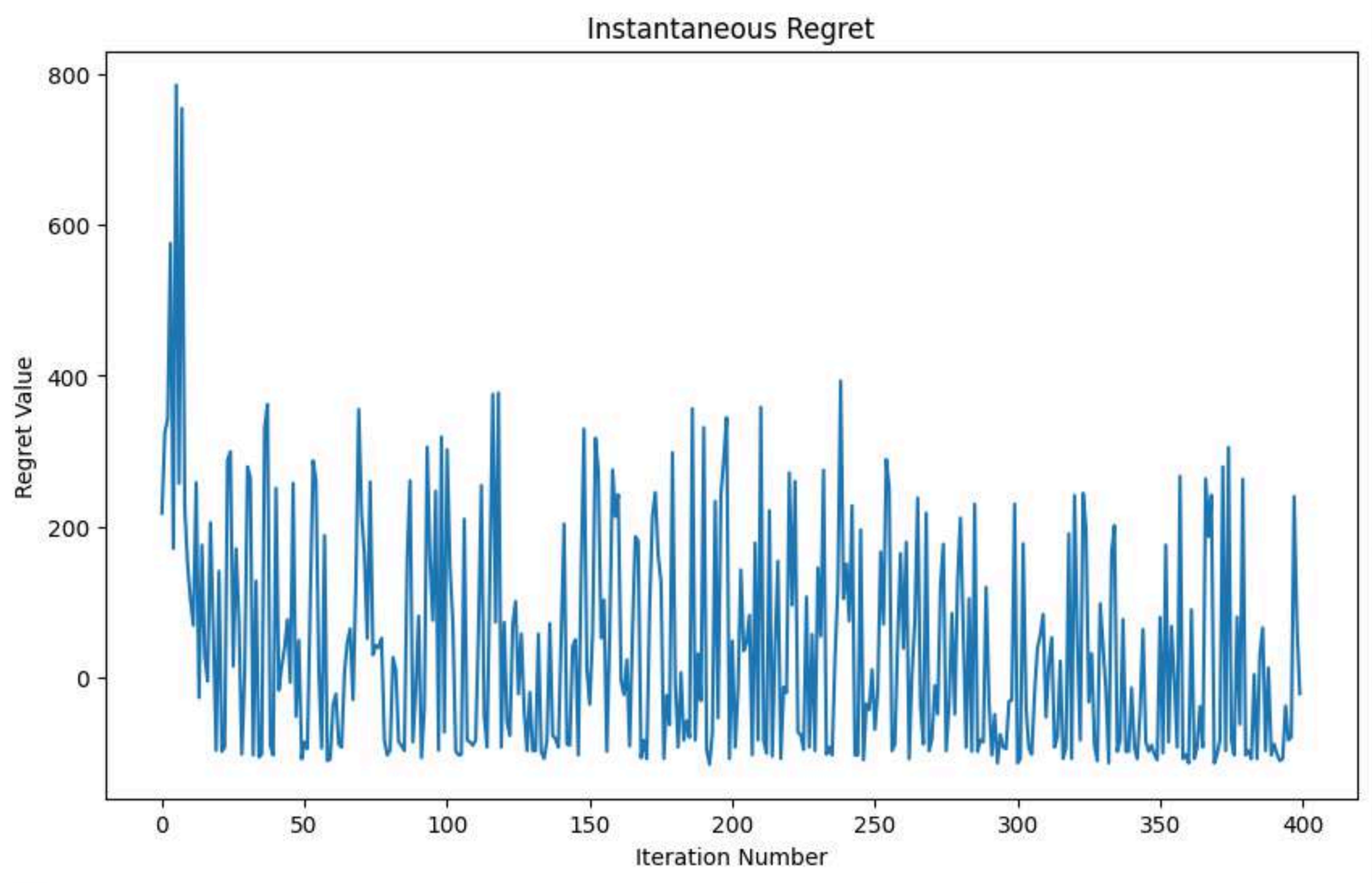
a = 5, b = 5, p = 4

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	10	9	9	9	10	8	7	8	2	6	2	4	5	0	3	0	1	1	1	0	1
1	10	10	10	9	10	7	9	8	7	6	5	5	4	3	2	2	1	1	1	0	0
2	8	10	10	9	10	7	9	6	4	4	3	5	5	2	4	1	1	0	1	0	0
3	9	10	9	9	8	8	9	7	8	4	3	5	0	2	1	2	2	0	0	0	0
4	9	10	9	10	9	9	9	8	6	6	5	2	4	4	3	1	2	0	1	0	0
5	7	10	10	10	10	7	9	6	5	7	4	4	4	2	0	1	2	1	0	1	0
6	10	10	9	10	8	7	9	7	6	6	5	5	2	1	1	1	0	1	1	0	1
7	9	9	10	8	9	6	7	7	8	2	2	3	3	4	1	2	1	1	0	0	0
8	10	10	8	10	7	8	8	6	4	4	2	3	1	2	2	0	0	0	0	0	0
9	0	0	0	7	1	0	4	2	0	2	2	0	0	1	0	0	0	0	0	0	0

a = 5, b = 5, p = 0

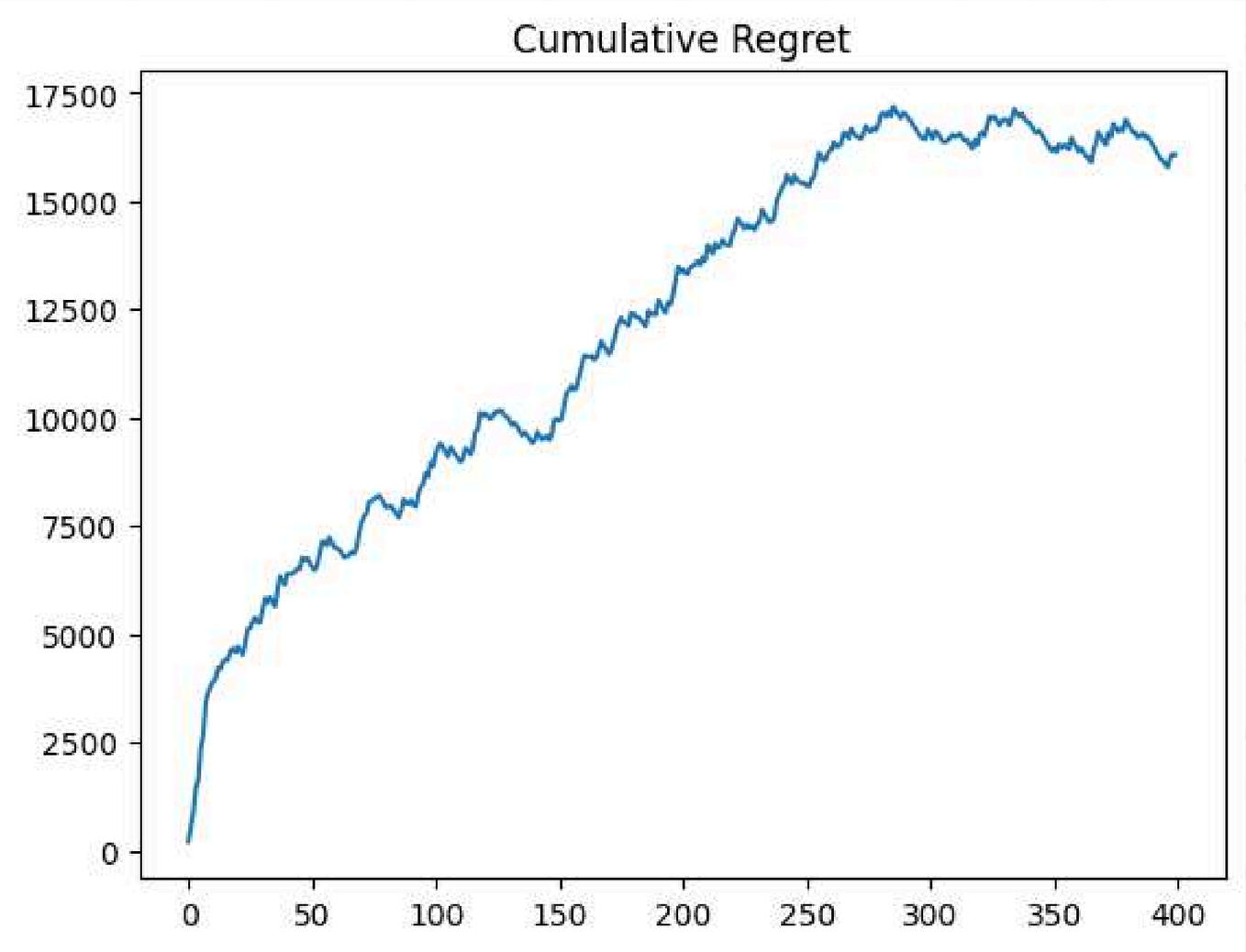
	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
0	9	10	10	8	8	8	7	7	7	6	6	4	3	4	3	1	1	0	0	0	0
1	10	10	10	10	10	9	9	8	6	7	5	5	4	3	1	0	0	0	0	0	0
2	10	10	10	9	9	10	7	6	7	5	7	5	4	3	1	0	1	1	0	0	0
3	10	9	10	10	8	10	8	8	6	6	6	3	3	3	2	1	0	0	0	0	0
4	10	10	10	9	10	10	9	6	6	3	4	2	2	2	1	1	1	0	0	0	0
5	10	10	9	10	10	9	7	6	6	5	7	4	5	3	2	2	0	0	1	0	0
6	10	10	9	9	9	9	8	8	6	5	5	5	1	1	2	1	2	1	0	0	0
7	10	9	9	10	10	6	8	7	7	3	6	3	3	3	1	0	1	1	1	0	0
8	10	10	9	10	9	8	8	7	6	5	4	3	3	2	2	2	0	1	0	0	0
9	9	10	8	9	9	8	9	7	4	5	4	3	2	0	1	0	0	0	0	0	0

Instantaneous regrets plot



Since finite Horizon Q learning is stochastic approximation algorithm it does not genrate episodes on go. Hence to find regret, episode is generated with the greedy policy corresponding to Q values learned till ith iteration

Cumulative regrets plot



Team members

1. Madhav Tank - 2021101108
2. Manav Shah - 2021101090
3. Nikunj Garg - 2021101021
4. Yash Adivarekar - 2021101008
5. Yash Kawade - 2021101032

Thank you very much!