June 24, 2022

5 Exercises

5.1 Question

Consider the diagrams on the right in Figure 5.1. Why does the estimated value function jump up for the last two rows in the rear? Why does it drop off for the whole last row on the left? Why are the frontmost values higher in the upper diagrams than in the lower?

Answer

No answer provided.

5.2 Question

Suppose every-visit MC was used instead of first-visit MC on the blackjack task. Would you expect the results to be very different? Why or why not?

Answer

The results would be the same because a state can be visited only once.

5.3 Question

What is the backup diagram for Monte Carlo estimation of q pi ?

Answer

The backup diagram for MC is given at page 95. An empty circle and a black filled circle represent a state action pair.

5.4 Question

The pseudocode for Monte Carlo ES is inefficient because, for each state–action pair, it maintains a list of all returns and repeatedly calculates their mean. It would be more efficient to use techniques similar to those explained in Section 2.4 to maintain just the mean and a count (for each state–action pair) and update them incrementally. Describe how the pseudocode would be altered to achieve this.

Answer

In section 2.4 average return is calculated iteratively using only current average and number of occurrences.

Average return for each state-action pair can be calculated similarly.

$$Q_(S, A) = Q_(S, A) + (G - Q_(S, A))/N(S, A) \tag{1}$$

5.5 Question

Consider an MDP with a single nonterminal state and a single action that transitions back to the nonterminal state with probability p and transitions to the terminal state with probability 1-p. Let the reward be +1 on all transitions, and let $\gamma = 1$. Suppose you observe one episode that lasts 10 steps, with a return of 10. What are the first-visit and every-visit estimators of the value of the nonterminal state?

Answer

There is no separate behaviour policy. Thus importing sampling ratio is 1.

First visit case:

$v(s) = 1 * 10/1 = 10$

Every visit case:

$v(s) = 1*10+1*9+1*8+1*7+1*6+1*5+1*4+1*3+1*2+1*1/10 = 55/10 = 5.5$

Results are similar if weighted estimator is used.

5.6 Question

What is the equation analogous to (5.6) for action values Q(s, a) instead of state values V (s), again given returns generated using b?

Answer

$$Q(s,a) = \frac{\sum_{t\in\tau(s,a)} Pt : (T(t) - 1)G_t]}{\sum_{t\in\tau(s,a)} Pt : (T(t) - 1)} \tag{2}$$

5.7 Question

In learning curves such as those shown in Figure 5.3 error generally decreases with training, as indeed happened for the ordinary importance-sampling method. But for the weighted importance-sampling method error first increased and then decreased. Why do you think this happened?

Answer

Weightet importance-sampling initially produces biased estimations. As the number of samples increases the bias disappears.

5.8 Question

The results with Example 5.5 and shown in Figure 5.4 used a first-visit MC method. Suppose that instead an every-visit MC method was used on the same problem. Would the variance of the estimator still be infinite? Why or why not?

Answer

The only difference would be the added 1/k term to the sum. The variance would still be infinite.

5.9 Question

Modify the algorithm for first-visit MC policy evaluation (Section 5.1) to use the incremental implementation for sample averages described in Section 2.4.

Answer

Update rule should be replaced with the following expression.

$$V(S_t) = V(S_t) + (G - V(S_t))/|\tau(S_t)| \tag{3}$$

5.10 Question

Derive the weighted-average update rule (5.8) from (5.7). Follow the pattern of the derivation of the unweighted rule (2.3).

Answer

$$C_n = \sum_{k=1}^{n} W_k \tag{4}$$

$$V_{n+1} = \frac{\sum_{k=1}^{n} W_k G_k}{\sum_{k=1}^{n} W_k} = \frac{1}{\sum_{k=1}^{n} W_k} \sum_{k=1}^{n} W_k G_k = \frac{1}{C_n}(W_n G_n + \sum_{k=1}^{n-1} W_k G_k) \tag{5}$$

$$V_{n+1} = \frac{1}{C_n}(W_n G_n + \frac{\sum_{k=1}^{n-1} W_k}{\sum_{k=1}^{n-1} W_k} \sum_{k=1}^{n-1} W_k G_k) = \frac{1}{C_n}(W_n G_n + V_n \sum_{k=1}^{n-1} W_k) \tag{6}$$

$$V_{n+1} = \frac{1}{C_n}(W_n G_n + V_n(\sum_{k=1}^{n} W_k - W_n)) = \frac{1}{C_n}(W_n G_n + V_n(C_n - W_n)) \tag{7}$$

$$V_{n+1} = \frac{1}{C_n}(W_n G_n + V_n C_n - V_n W_n) = \frac{1}{C_n}(V_n C_n + W_n(G_n - V_n)) \tag{8}$$

$$V_{n+1} = V_n + \frac{W_n(G_n - V_n)}{C_n} \tag{9}$$

5.11 Question

In the boxed algorithm for off-policy MC control, you may have been expecting the W update to have involved the importance-sampling ratio $\frac{\pi(A_t|S_t)}{b(A_t|S_t)}$, but instead it involves $\frac{1}{b(A_t|S_t)}$. Why is this nevertheless correct?

Answer

π is a deterministic policy (e.g. argmax is used.). π returns 1 for the greedy action and 0 otherwise. The importance-sampling ratio will become 0 after policy π and b diverges. As a matter of the fact, algorithm will quit the actions loop once $A_t \neq \pi(S_t)$.

In this setting it is OK to use 1 instead of $\pi(A_t|S_t)$ which already equals to 1.

5.12 Question

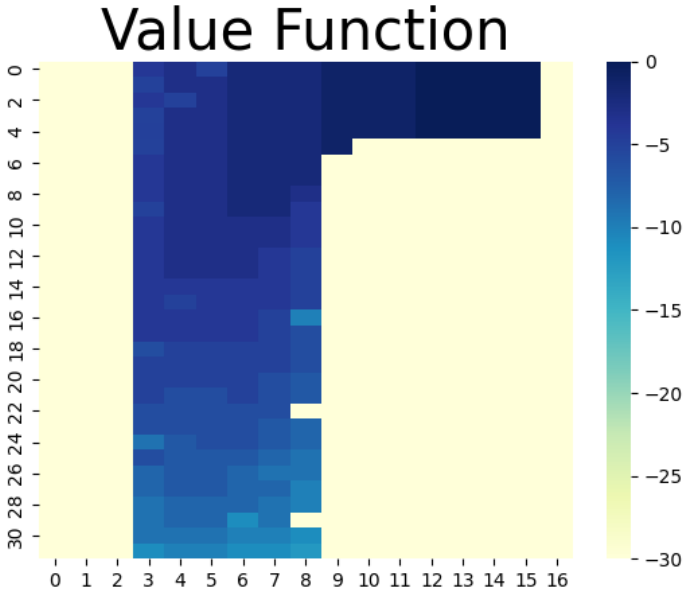
Racetrack (programming)

Consider driving a race car around a turn like those shown in Figure 5.5. You want to go as fast as possible, but not so fast as to run off the track.

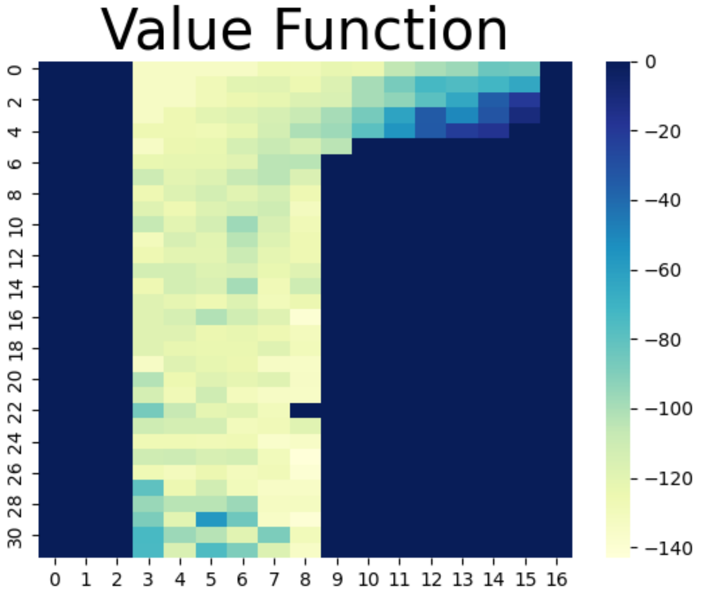
Apply a Monte Carlo control method to this task to compute the optimal policy from each starting state. Exhibit several trajectories following the optimal policy (but turn the noise o for these trajectories).

Answer

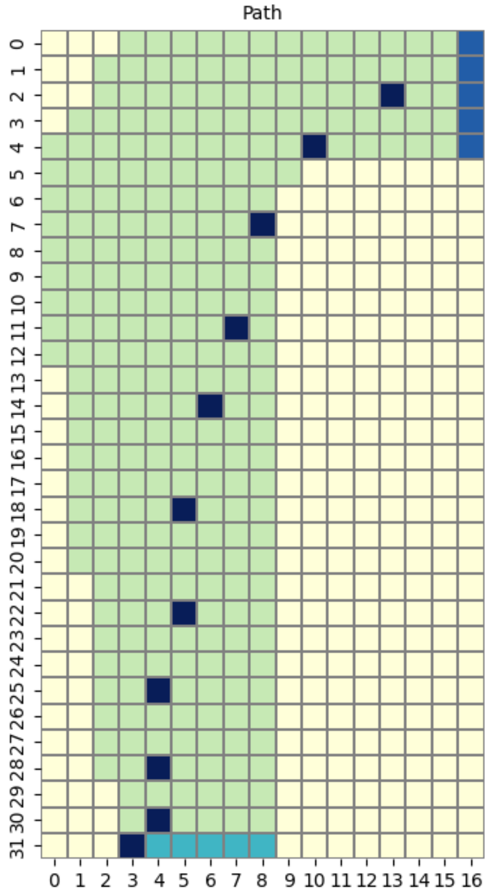
Grid 1 (env2)



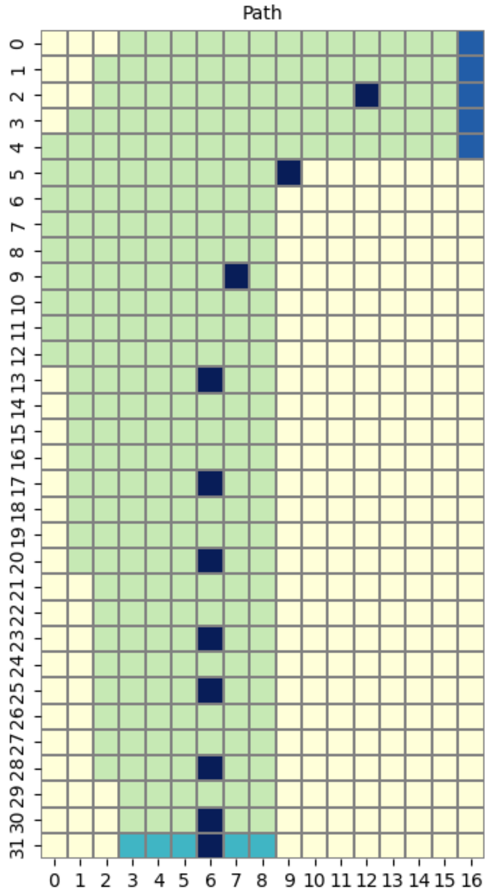
Maximum values over speeds.



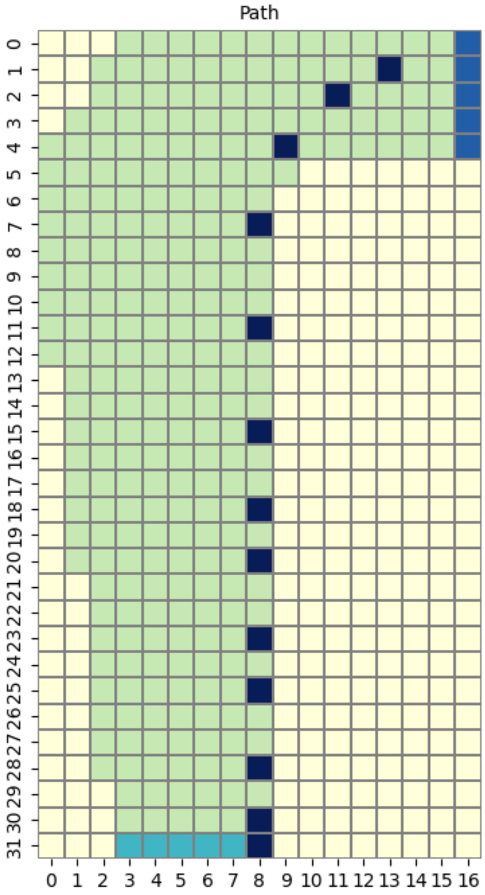
Average values over speeds.


Path
0
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
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Demo path.

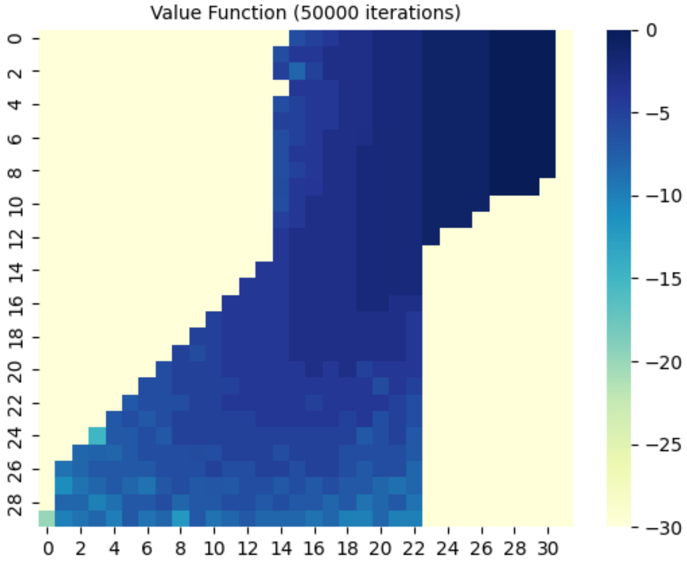

Path
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Demo path.

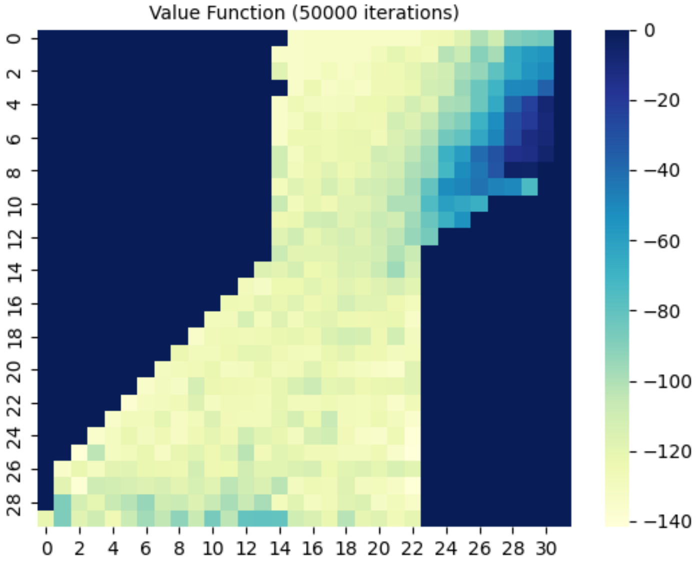


Demo path.
Average values over speeds.

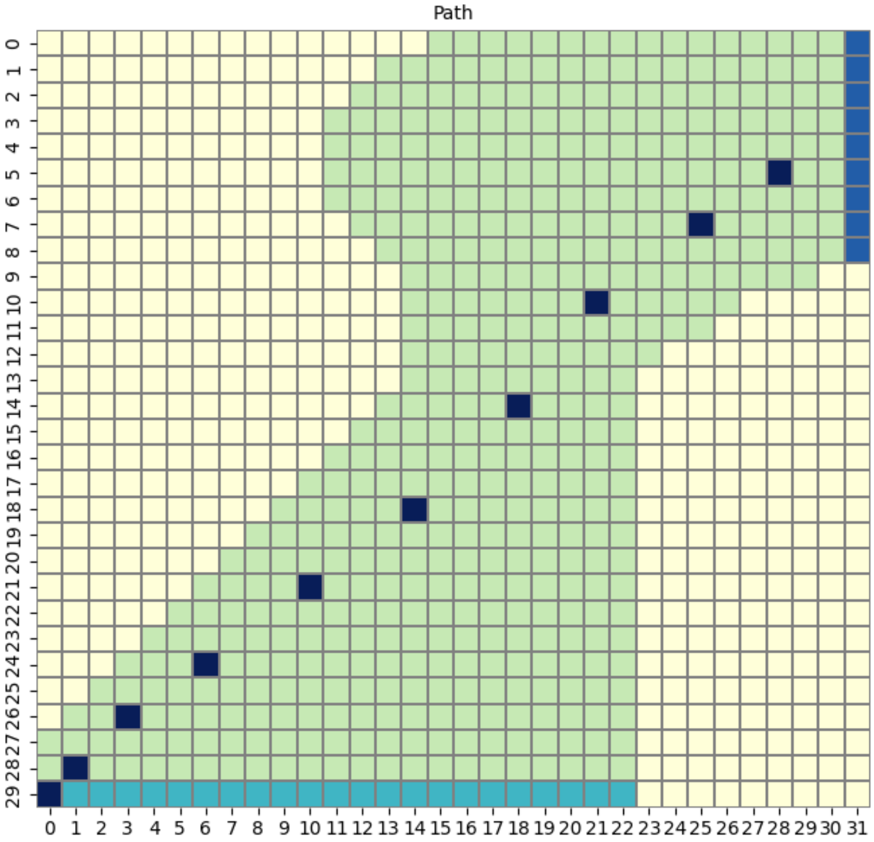
5.12.1 Grid 2 (env3)



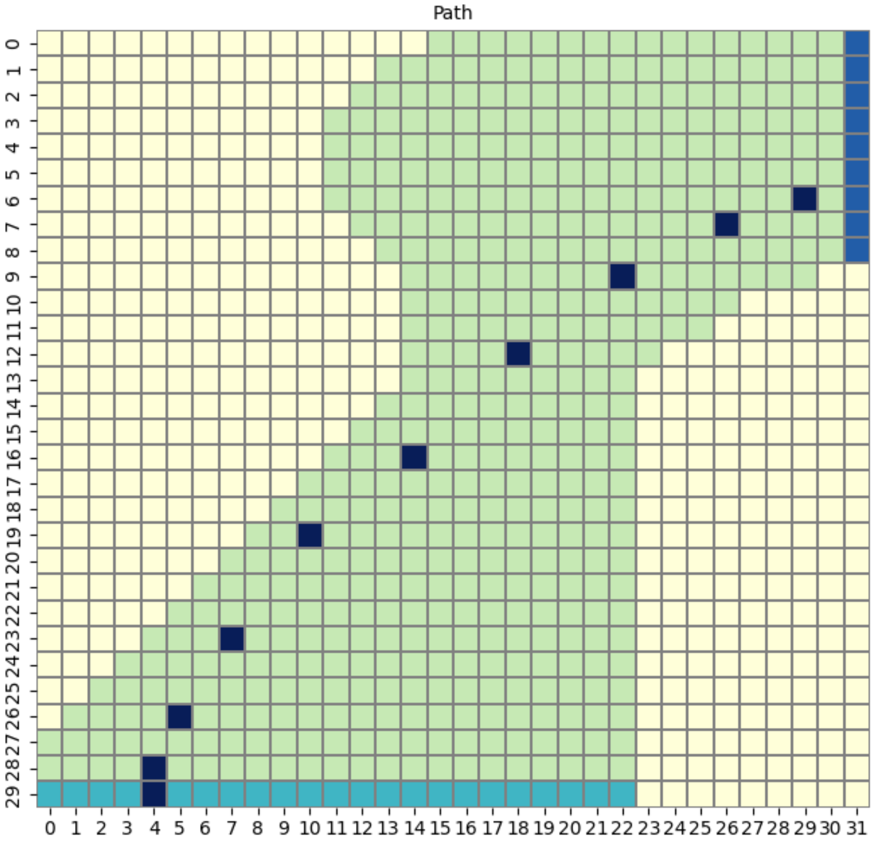
Maximum values over speeds.



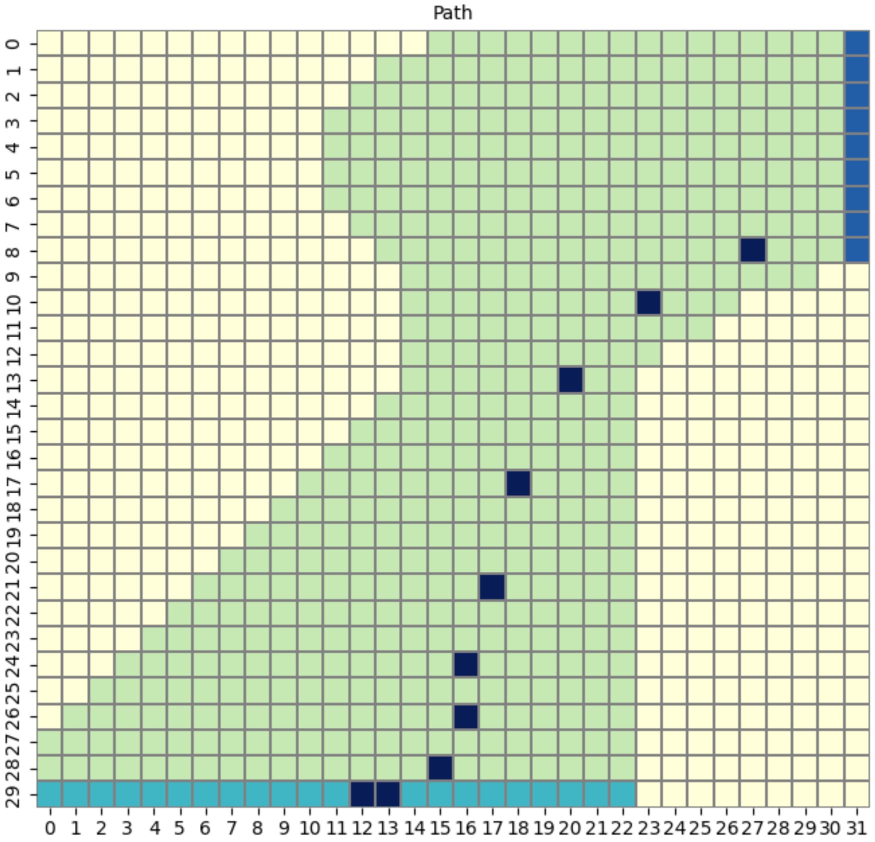
Average values over speeds.


Path
0
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
0 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

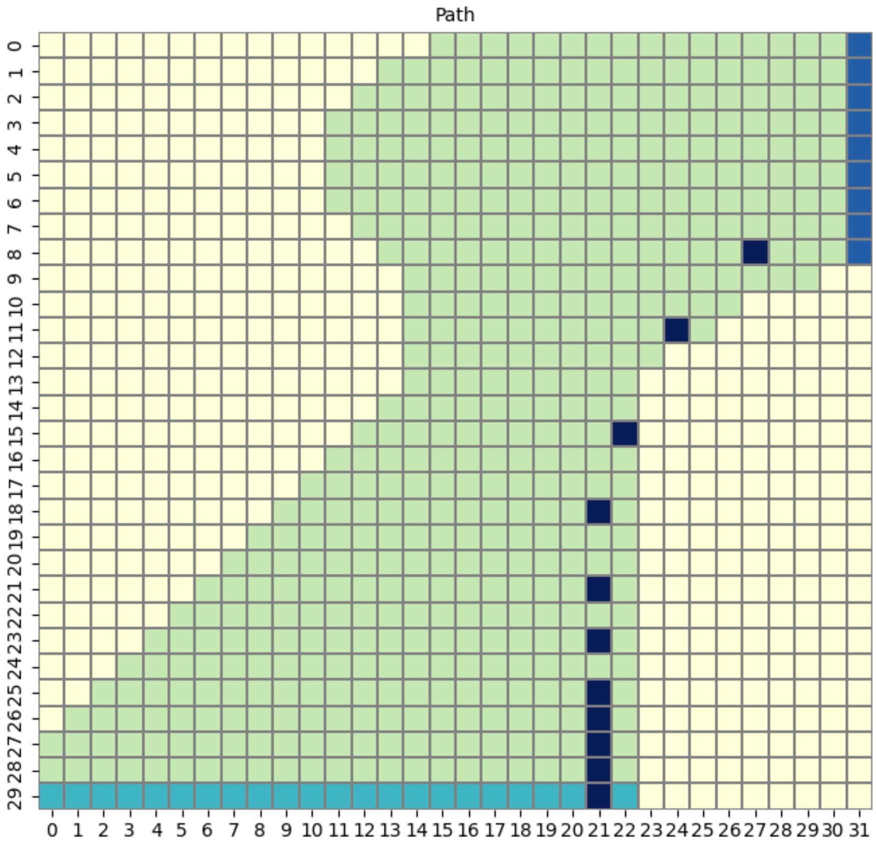
Demo path.


Path
0
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

Demo path.


Path
0
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
0 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

Demo path.


Path
0
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
0
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

Demo path.