CS520: Question 1

Keshav Shivkumar

ks1830

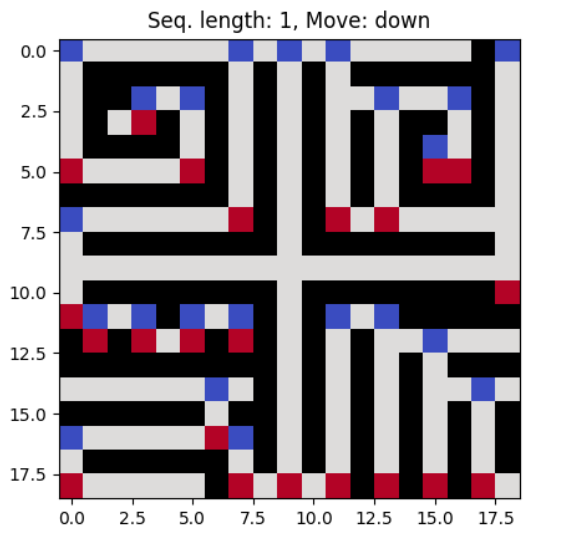
Repository: <https://github.com/keshavshivkumar/cs520finals>

## Question 1

In the beginning, the likelihood of the drone’s position is equally spread across all non-blocked positions in the reactor. So, the probability of the drone being in the top left corner is 1/(# of ‘\_’ spaces in the reactor) = 1/199

## Question 2

Upon issuing the ‘DOWN’ command, the likelihoods shift downwards. The spaces that are just below a blocked space or just below the out of bounds region have zero likelihood, while the spaces that have blocked spaces/the out of bounds region below them have double the prior probability.



DOWN (Red is the highest probability, blue is zero probability)

Background pattern

Description automatically generated

Probability Matrix of the reactor

## Question 3

Design:

* The probabilities shift by moving in the specified direction.
* A sequence of commands can be generated by defining a heuristic to select the direction to choose.
* The selected heuristic is to take the next direction in the actual path between the first node of the lowest non-zero probability and the first node of the highest probability.

Implementation:

* The environment is made using the ‘Reactor’ class. Each node in the ‘Reactor' class is made of ‘Space’ objects.
* The probability of 1 is evenly spread across all 199 unblocked spaces.
* All non-zero probabilities are stored in a list.
* The first space of lowest (non-zero) probability and the first space of highest probability are considered. If they tie, then the next node in the list of probabilities is taken.
* The BFS path between the two is computed. The next node in the path determines the direction to take.
* Once the direction is picked, the probabilities are shifted in that direction.
* The process is repeated until the probabilities converge into 1 space with probability 1; we know for sure the drone exists in this node

Results:

* The probability converges in 219 commands. It can also converge to 245 commands since the way the max and min nodes are picked can alter the outcome.

219 Commands

Description automatically generated

Chart

Description automatically generated with medium confidence

219 command sequence

Text, shape, arrow

Description automatically generated

245 command sequence

Does this methodology find the shortest sequence of commands?

* No, it is not the most optimal solution. Since the used heuristic does not consider every possible state, it simply follows the BFS path between the found min and max probability spaces.
* On trialing with different heuristics, I believe that the implementation finds the shortest possible sequence.

## Question 4

The attempted implementation does not give the shortest possible sequence. Based on the implementation, I believe a reactor configuration that has many warped, spiraling blocks in different directions would make the sequence very long.

Qr code

Description automatically generated

Probably a bad schematic of the reactor (it took 282 commands)

Making the spirals warp in different directions would likely make the sequence longer.