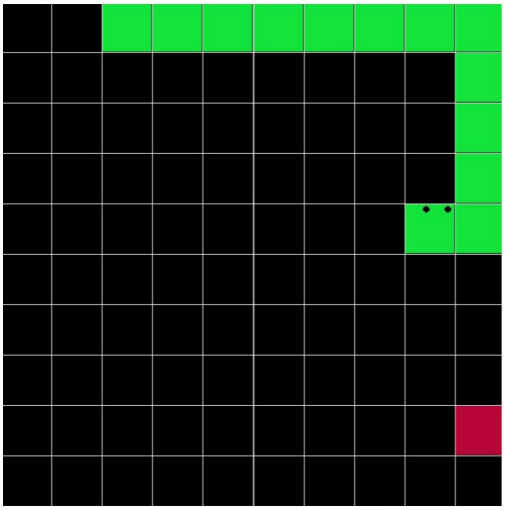
Snakey AI

Emily Traynor

April 9th, 2020



Contents

1 Introduction

The purpose of this project is to optimize the points achieved in the classic snake game. The game involves a "snake" which is a string of blocks that has a set amount of time to eat pieces of food on the playing grid. Every time the snake eats a piece of food it grows a block in length and another piece of food generates randomly elsewhere on the grid. The snake can move in four directions—up, down, left, and right—while remaining at a constant pace. For each move, the snake makes its score increases by 1. The game ends if the snake bumps into itself or the edge of the grid, or it runs out of time to eat the food. The success of the optimal algorithm will be determined based on the objectives mentioned below and data analysis. The data will be collected by running the program over 100 times and storing the data in a text file using Python. It will be run on different grid sizes to investigate its efficiency on varying grid sizes.

Primary and Secondary research was conducted to develop a plan to approach this problem. The primary research was done by playing the classic snake game and developing a strategy to score well. This strategy involved unwinding the snake before targeting the next snack. This could easily be done by going up and down or left and right a few times. The secondary research regarded which type of shortest-path algorithm would best suit this project. Algorithms such as Dijkstra's, A* and Bellman Ford's were investigated. These are discussed in the conceptual framework section.

This document will cover the objectives of this project, the framework of the solution including illustrations, results, performance analysis, a video of the snake algorithm and discussion of some future improvements.

2 Objectives

The following are the objectives of this project, other than optimizing the score.

1. The algorithm shouldn't require adjustment for different game board sizes
    - Metric: I will see if the algorithm executes on different board sizes.
    - Criteria: The more board sizes it works on the better.
2. The algorithm should incorporate a feature that prevents the snake from closing in on itself
    - Metric: The ratio of death by bumping into itself and the border (which will also be collected while running the trials).
    - Criteria: the lower the ratio, the better.
3. Minimizes worst-case space complexity
    - Metric: Calculate worst-case space complexity.
    - Criteria: The lower the better.

3 Detailed Framework

This program uses a graphical user interface called Pygame because it makes it easier to see the snake's movement. This program was written in Python and has two classes: cube and snake. The cube class builds the shapes of the snake and snack in Pygame, while the snake class controls the movement of the snake. The moves of the snake are determined by the functions called in the main function. Some of the starter code was derived from a tutorial [1].

I included a function, based on my primary research, that makes the snake execute a back and forth motion after it eats a target. This helps keep the snake's body in a concentrated area to prevent it from closing in on itself. As shown in Figure 1, the snake analyzes the free space in the column and row of its head and executes a back and forth motion. This function is called "cluster" or "clustering" because it clusters the snake's body in one section. Clustering occurs before Dijkstra's algorithm in order to prevent the snake from closing in on itself. The cluster function pseudo code is provided below, in Figure 1. It simply tells the snake to organize itself, given there is sufficient room, before generating a path to the snack.

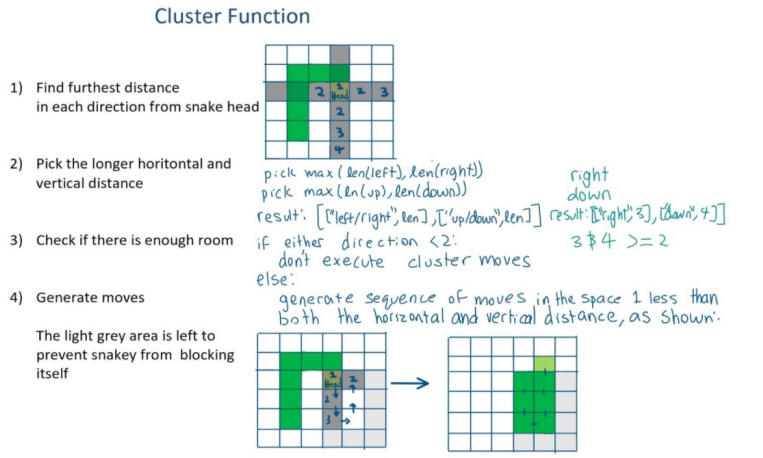


Figure 1: Cluster Function Pseudo Code and Illustration

In order for the snakey to get to a snack, the program must generate a path for the snake to follow. This can be done using a shortest path algorithm. As mentioned prior, three shortest path algorithms were investigated. One algorithm, called Bellman-Ford's algorithm, finds the shortest path however it's worst-case time complexity of O(VE) is worse than that of Dijkstra's which is O(V2), where V is the number of vertices and E is the number of edges [2]. Comparing Dijkstra's to the A* algorithm, A* has a better worst-case time complexity, although it is not expressed in terms of vertices and edges. A* is essentially a smart version of Dijkstra's algorithm because it uses a heuristic to pick nodes to explore [3]. It has a bias to choose nodes closest to the target, whereas Dijkstra explores all nodes regardless of its distance from the target. However, A* and Dijkstra's algorithm produce the same result and considering that the snake's body intercepts many of the shortest paths, I felt that the A* algorithm wouldn't significantly improve the time complexity to find the shortest path. Through my research, it appeared that A* significantly better than Dijkstra's when the shortest path total distance is similar to the euclidean distance. I demonstrated my logic in Figure 2, below.

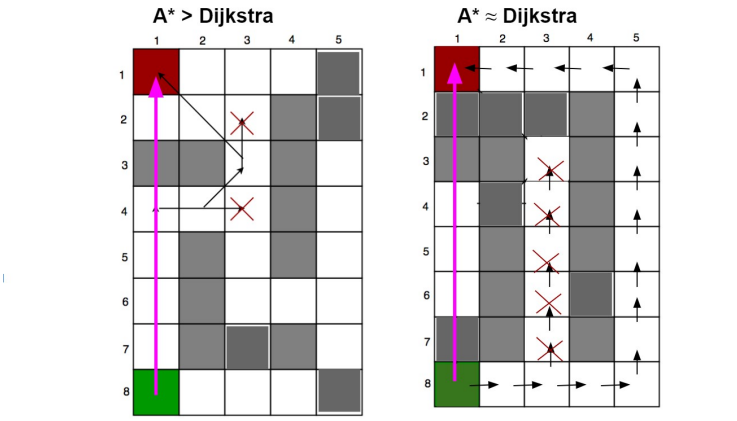


Figure 2: Example of when A* isn't more efficient than Dijkstra [4]

Additionally, considering the nature of this problem, the adjacency graph tends to have many edges relative to the number of nodes, this can be seen in Figure 3, below. This makes Dijkstra's algorithm appropriate to use as it only visits each node once. Figure 3 also shows the inputs and outputs of my Dijkstra's algorithm function. The whole function can be found in Appendix C. Instead of creating an adjacency matrix which is a $(rows^2)$ by $(rows^2)$ matrix consisting of the weights of connections, I made an adjacency dictionary. This has a far better space complexity considering the dictionary only has a maximum of $(rows^2)$ entries with a maximum of four values (the adjacent boxes on the left, right, above and below). This was also successful because each move has an equal weight, which I assigned the value of one.

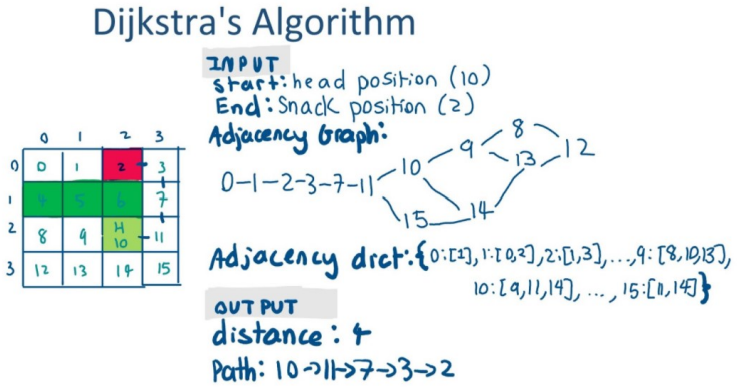


Figure 3: Dijkstra's Algorithm Inputs and Outputs

Notice that the grid is also converted to a single number from 0 to $(rows^2 - 1)$. This was done using a dictionary called a change of basis. It maps the box numbers to their coordinates. This was helpful because it simplifies Dijkstra's algorithm from comparing tuples of coordinates to a single number. The change of basis variable is a global variable because it is consistent throughout the game. As can be seen in Figure 4, it is only generated once to improve the space and time

complexity.

```
def main():
    global width, rows, s, snack, change_of_basis

    width = 700
    rows = 10
    cube.rows = rows
    cube.w = width

    change_of_basis = generate_box_to_coord_dict(rows)
```

Figure 4: Change of Basis

I also made a backup function for when there isn't enough space for clustering to occur and there isn't a path between the head and tail. Backup uses subfunctions of the clustering algorithm to determine which of the four directions (left, right, up, and down) from the head has the most space before meeting a boundary of part of the snake. It is then commanded to go in this direction before reattempting Dijkstra's algorithm. This happens in hopes that a connection between the snake's head and tail will appear after completing the sequence of backup moves so that it can eat the snack.

The following summarises the possible sequences of function calls.

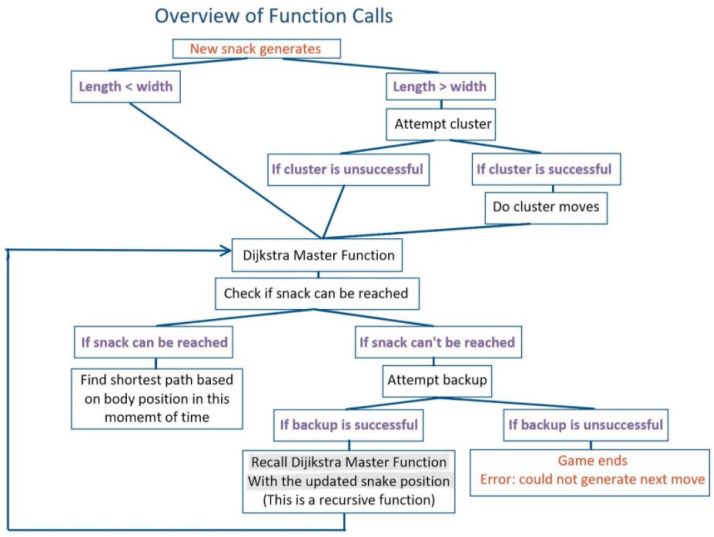


Figure 5: Overview of Function Calls

As can be seen in the figure above, Dijkstra's algorithm only occurs if there is a connection between the head and snack. This was done using depth-first search (DFS). An alternative method to find a connection is breadth-first search (BFS) however because the snack appears randomly it could generate close or far away from the head. Thus, neither search algorithm is significantly more efficient.

Finally, it should be mentioned that to improve the time complexity, Dijkstra's algorithm is based on the grid when the function is called. Thus as the snake heads towards the snack, a shorter path may appear, but the snake will continue on the original shortest-path so Dijkstra's algorithm only runs once per snack. The consideration of time complexity was taken so that the program could run on much larger grid sizes.

4 Results

The program was run over 100 times on a 10 x 10 grid. It was also run on a square grid with 3, 4, 5, 20, 40 and 80 rows. The results are in Appendix B. The following figure shows the box plot of the results from the 100 trials.

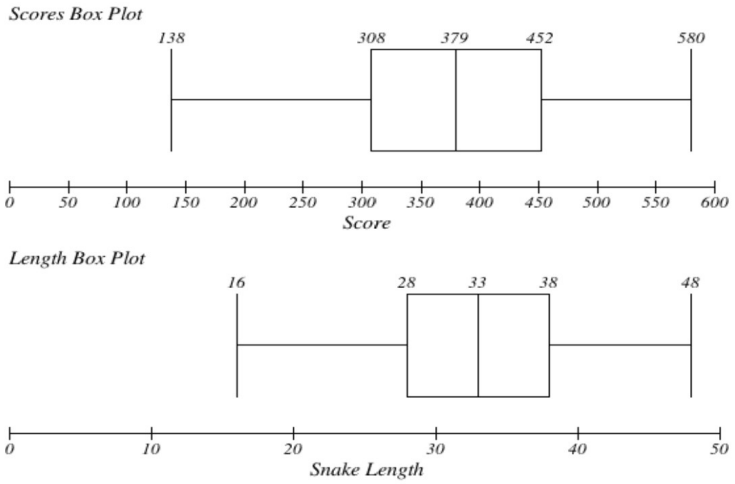


Figure 6: Box Plot of Scores and Snake Length on a 10 x 10 Grid

As can be seen, in Figure 6, 50% of trials scored between 308 and 452 and obtained a length of 28 to 38 snakey blocks. This shows that snakey AI is consistent with a few outlying trials. The standard deviation of the score and length were 99 and 6, respectively. Considering the ranges of scores and lengths this also suggests that snakey was consistent, in particular the length. Scores may vary more because clustering only occurs sometimes, but can change the score by 4 to 30 each time. The data doesn't confirm the lower or upper bound, however, it suggests that 138 and 580 are the bound for scores and 16 and 48 are the bound for length. Contrary to this, I believe the lower bound length is 5, because when the snake is less than the number of rows it only executes Dijkstra's algorithm. Thus if it makes a U-shape into a corner it can block itself in. Its score would depend on how far apart the snacks appear but would be less than 138 which corresponds with the smallest recorded length of 16. Thus, the guaranteed length is 4 and the score would be less than 138.

Thus the expected result is:
Score: 379 and Length: 33

A video of the snake AI is available via this link:
https://drive.google.com/file/d/1JPKnVk_WvulpWVCCN__C2dTaoGvn0N60/view?usp=sharing

5 Conclusion

In conclusion, snakey AI performs very well based on my objectives. Theoretically, the program doesn't have a maximum grid size, however up to a grid size of 80 was tested and shown to work. It works the same way it does on a 10 by 10 grid and so the program is reliable on a grid size of at least 80 x 80. The smallest grid size it worked on was a 3 by 3 grid. Below that, the snake obtained snacks, however, the program did not quit and went into an infinite loop. All in all, the confirmed grid sizes it is reliable on is 3 x 3 to 80 x 80.

The cluster feature dramatically improved the snake's length because it prevents it from closing in on itself. Without it, the snake was averaging a length of around 15 which is half of its new average. However, I believe this is also an area for growth because there are cases in which there is space available for clustering that the cluster function doesn't detect, this is further discussed in future improvements. In the data collected, the most common reason that the game ended was that the snake couldn't generate another move which means it trapped itself. This can be avoided with more clustering.

The worst-case space complexity is also an area for improvement. Measures were taken to improve it however, DFS and Dijkstra's algorithm, in particular, increased the space complexity. It becomes apparent on a grid size over 20 because the snake pauses before executing either function. The following shows my space complexity calculations:

The maximum space complexity of depth-first search in my algorithm is less than O(4 x V), where V is every box in the grid, because each vertex can have a maximum of 4 edges but my program pop nodes once they have been visited. Whereas the worst-case space complexity of Dijkstra's algorithm is O(V) because it tracks the shortest distance to each node, and the maximum number of nodes is V. Thus, it's difficult to conclude which has worse space complexity, but they both have room for improvement. One improvement would be to find a path to the snack, instead of the shortest path so that not every box in the grid must be visited.

6 Future Improvement

The cluster function occasionally causes the snake to bump into itself because it only considers the row and column that the head is in. When the furthest distance from the head is to the border in a horizontal and vertical direction, it guarantees that there is no snake in that region, however, this is not the case if the furthest distance reaches the body of the snake, there is a chance that some of the snake is in the cluster region. This rarely causes the snake to bump into itself however, it is a flaw in the program that causes the game to end early even with space to execute a smaller cluster or Dijkstra's algorithm. To fix this an additional step in deciding whether to execute the cluster should be taken to ensure no snake interferes with the cluster sequence.

From observation, I noticed that there is often an opportunity for clusters to occur when the head is facing a border. Cluster doesn't occur because it only analyzes the row and column of the head. This is shown in Figure 7. To improve, a secondary cluster feature should be implemented that analyzes the grid for regions of unoccupied space beside the head, also shown in Figure 7.

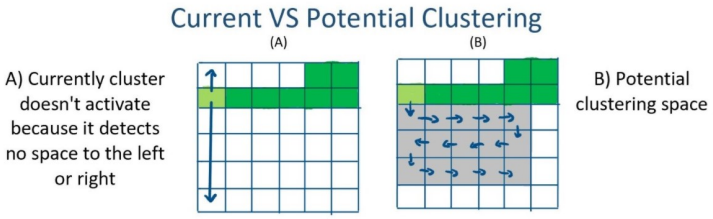


Figure 7: Potential for Cluster Improvement

In general, future improvements would encourage more clustering and for longer periods of time to increase the score since the more frames the snake lives, the higher its score, and it helps concentrate the snake's body in one region, to prevent it from getting in the way.

References

[1] Tim. Snake game python tutorial. urlwww.youtube.com/watch?v=CD4qAhfFuLo&t=587s.

[2] D.W. Shortest path algorithms tutorials & notes: Algorithms. urlwww.hackerearth.com/practice/algorithms/graphs/shortest-path-algorithms/tutorial/.

[3] D.W. A* graph search time-complexity. urlcs.stackexchange.com/questions/56176/a-graph-search-time-complexity., Apr 2016.

[4] GeeksforGeeks. A* search algorithm. Sep 2018.

A Data Collection

```
Trial: 0, Board width: 10, Score: 498, Body length: 47, Reason for end of game: Could not generate another move
Trial: 1, Board width: 10, Score: 499, Body length: 48, Reason for end of game: Could not generate another move
Trial: 2, Board width: 10, Score: 409, Body length: 38, Reason for end of game: Could not generate another move
Trial: 3, Board width: 10, Score: 344, Body length: 29, Reason for end of game: Bumped into self
Trial: 4, Board width: 10, Score: 376, Body length: 31, Reason for end of game: Bumped into self
Trial: 5, Board width: 10, Score: 374, Body length: 35, Reason for end of game: Could not generate another move
Trial: 6, Board width: 10, Score: 286, Body length: 29, Reason for end of game: Bumped into self
Trial: 7, Board width: 10, Score: 385, Body length: 33, Reason for end of game: Bumped into self
Trial: 8, Board width: 10, Score: 296, Body length: 28, Reason for end of game: Bumped into self
Trial: 9, Board width: 10, Score: 383, Body length: 33, Reason for end of game: Could not generate another move
Trial: 10, Board width: 10, Score: 424, Body length: 37, Reason for end of game: Bumped into self
Trial: 11, Board width: 10, Score: 263, Body length: 32, Reason for end of game: Could not generate another move
Trial: 12, Board width: 10, Score: 317, Body length: 27, Reason for end of game: Bumped into self
Trial: 13, Board width: 10, Score: 452, Body length: 37, Reason for end of game: Could not generate another move
Trial: 14, Board width: 10, Score: 436, Body length: 37, Reason for end of game: Could not generate another move
Trial: 15, Board width: 10, Score: 460, Body length: 41, Reason for end of game: Could not generate another move
Trial: 16, Board width: 10, Score: 359, Body length: 33, Reason for end of game: Could not generate another move
Trial: 17, Board width: 10, Score: 427, Body length: 39, Reason for end of game: Could not generate another move
Trial: 18, Board width: 10, Score: 288, Body length: 25, Reason for end of game: Bumped into self
Trial: 19, Board width: 10, Score: 446, Body length: 30, Reason for end of game: Could not generate another move
Trial: 20, Board width: 10, Score: 360, Body length: 31, Reason for end of game: Bumped into self
Trial: 21, Board width: 10, Score: 237, Body length: 22, Reason for end of game: Bumped into self
Trial: 22, Board width: 10, Score: 259, Body length: 26, Reason for end of game: Could not generate another move
Trial: 23, Board width: 10, Score: 402, Body length: 34, Reason for end of game: Bumped into self
Trial: 24, Board width: 10, Score: 299, Body length: 27, Reason for end of game: Could not generate another move
Trial: 25, Board width: 10, Score: 405, Body length: 36, Reason for end of game: Bumped into self
Trial: 26, Board width: 10, Score: 455, Body length: 38, Reason for end of game: Bumped into self
Trial: 27, Board width: 10, Score: 279, Body length: 28, Reason for end of game: Bumped into self
Trial: 28, Board width: 10, Score: 494, Body length: 39, Reason for end of game: Could not generate another move
Trial: 29, Board width: 10, Score: 341, Body length: 30, Reason for end of game: Could not generate another move
Trial: 30, Board width: 10, Score: 419, Body length: 37, Reason for end of game: Bumped into self
Trial: 31, Board width: 10, Score: 456, Body length: 39, Reason for end of game: Bumped into self
Trial: 32, Board width: 10, Score: 280, Body length: 28, Reason for end of game: Bumped into self
Trial: 33, Board width: 10, Score: 364, Body length: 34, Reason for end of game: Could not generate another move
Trial: 34, Board width: 10, Score: 441, Body length: 35, Reason for end of game: Bumped into self
Trial: 35, Board width: 10, Score: 449, Body length: 37, Reason for end of game: Could not generate another move
Trial: 36, Board width: 10, Score: 495, Body length: 40, Reason for end of game: Bumped into self
Trial: 37, Board width: 10, Score: 314, Body length: 36, Reason for end of game: Could not generate another move
```

```
Trial: 38, Board width: 10, Score: 395, Body length: 36, Reason for end of game: Could not generate another move
Trial: 39, Board width: 10, Score: 167, Body length: 21, Reason for end of game: Could not generate another move
Trial: 40, Board width: 10, Score: 536, Body length: 36, Reason for end of game: Could not generate another move
Trial: 41, Board width: 10, Score: 248, Body length: 25, Reason for end of game: Could not generate another move
Trial: 42, Board width: 10, Score: 310, Body length: 30, Reason for end of game: Could not generate another move
Trial: 43, Board width: 10, Score: 410, Body length: 33, Reason for end of game: Could not generate another move
Trial: 44, Board width: 10, Score: 437, Body length: 35, Reason for end of game: Bumped into self
Trial: 45, Board width: 10, Score: 309, Body length: 27, Reason for end of game: Could not generate another move
Trial: 46, Board width: 10, Score: 376, Body length: 33, Reason for end of game: Could not generate another move
Trial: 47, Board width: 10, Score: 311, Body length: 33, Reason for end of game: Could not generate another move
Trial: 48, Board width: 10, Score: 409, Body length: 31, Reason for end of game: Could not generate another move
Trial: 49, Board width: 10, Score: 308, Body length: 28, Reason for end of game: Bumped into self
Trial: 50, Board width: 10, Score: 512, Body length: 42, Reason for end of game: Bumped into self
Trial: 51, Board width: 10, Score: 541, Body length: 41, Reason for end of game: Bumped into self
Trial: 52, Board width: 10, Score: 465, Body length: 33, Reason for end of game: Bumped into self
Trial: 53, Board width: 10, Score: 456, Body length: 39, Reason for end of game: Could not generate another move
Trial: 54, Board width: 10, Score: 580, Body length: 42, Reason for end of game: Bumped into self
Trial: 55, Board width: 10, Score: 373, Body length: 32, Reason for end of game: Could not generate another move
Trial: 56, Board width: 10, Score: 339, Body length: 30, Reason for end of game: Could not generate another move
Trial: 57, Board width: 10, Score: 563, Body length: 44, Reason for end of game: Bumped into self
Trial: 58, Board width: 10, Score: 279, Body length: 26, Reason for end of game: Bumped into self
Trial: 59, Board width: 10, Score: 326, Body length: 28, Reason for end of game: Bumped into self
Trial: 60, Board width: 10, Score: 287, Body length: 29, Reason for end of game: Bumped into self
Trial: 61, Board width: 10, Score: 357, Body length: 32, Reason for end of game: Could not generate another move
Trial: 62, Board width: 10, Score: 454, Body length: 38, Reason for end of game: Could not generate another move
Trial: 63, Board width: 10, Score: 477, Body length: 37, Reason for end of game: Could not generate another move
Trial: 64, Board width: 10, Score: 138, Body length: 16, Reason for end of game: Could not generate another move
Trial: 65, Board width: 10, Score: 306, Body length: 26, Reason for end of game: Could not generate another move
Trial: 66, Board width: 10, Score: 417, Body length: 40, Reason for end of game: Bumped into self
Trial: 67, Board width: 10, Score: 299, Body length: 29, Reason for end of game: Bumped into self
Trial: 68, Board width: 10, Score: 550, Body length: 40, Reason for end of game: Bumped into self
Trial: 69, Board width: 10, Score: 188, Body length: 19, Reason for end of game: Could not generate another move
Trial: 70, Board width: 10, Score: 401, Body length: 33, Reason for end of game: Could not generate another move
Trial: 71, Board width: 10, Score: 348, Body length: 33, Reason for end of game: Could not generate another move
Trial: 72, Board width: 10, Score: 571, Body length: 42, Reason for end of game: Could not generate another move
Trial: 73, Board width: 10, Score: 356, Body length: 28, Reason for end of game: Bumped into self
Trial: 74, Board width: 10, Score: 248, Body length: 22, Reason for end of game: Could not generate another move
Trial: 75, Board width: 10, Score: 459, Body length: 39, Reason for end of game: Could not generate another move
Trial: 76, Board width: 10, Score: 491, Body length: 40, Reason for end of game: Could not generate another move
Trial: 77, Board width: 10, Score: 309, Body length: 23, Reason for end of game: Bumped into border
Trial: 78, Board width: 10, Score: 317, Body length: 32, Reason for end of game: Could not generate another move
Trial: 79, Board width: 10, Score: 428, Body length: 36, Reason for end of game: Could not generate another move
Trial: 80, Board width: 10, Score: 286, Body length: 30, Reason for end of game: Could not generate another move
Trial: 81, Board width: 10, Score: 547, Body length: 40, Reason for end of game: Bumped into self
Trial: 82, Board width: 10, Score: 173, Body length: 20, Reason for end of game: Bumped into self
Trial: 83, Board width: 10, Score: 374, Body length: 31, Reason for end of game: Could not generate another move
Trial: 84, Board width: 10, Score: 202, Body length: 27, Reason for end of game: Could not generate another move
Trial: 85, Board width: 10, Score: 366, Body length: 26, Reason for end of game: Bumped into self
Trial: 86, Board width: 10, Score: 353, Body length: 31, Reason for end of game: Could not generate another move
Trial: 87, Board width: 10, Score: 224, Body length: 26, Reason for end of game: Could not generate another move
Trial: 88, Board width: 10, Score: 380, Body length: 29, Reason for end of game: Bumped into self
Trial: 89, Board width: 10, Score: 175, Body length: 19, Reason for end of game: Could not generate another move
Trial: 90, Board width: 10, Score: 400, Body length: 34, Reason for end of game: Bumped into self
Trial: 91, Board width: 10, Score: 264, Body length: 25, Reason for end of game: Could not generate another move
Trial: 92, Board width: 10, Score: 417, Body length: 39, Reason for end of game: Could not generate another move
Trial: 93, Board width: 10, Score: 527, Body length: 41, Reason for end of game: Bumped into self
Trial: 94, Board width: 10, Score: 426, Body length: 38, Reason for end of game: Could not generate another move
Trial: 95, Board width: 10, Score: 295, Body length: 27, Reason for end of game: Bumped into self
Trial: 96, Board width: 10, Score: 486, Body length: 39, Reason for end of game: Bumped into self
Trial: 97, Board width: 10, Score: 557, Body length: 42, Reason for end of game: Bumped into self
Trial: 98, Board width: 10, Score: 403, Body length: 37, Reason for end of game: Could not generate another move
Trial: 99, Board width: 10, Score: 435, Body length: 36, Reason for end of game: Bumped into self
Trial: 100, Board width: 10, Score: 477, Body length: 39, Reason for end of game: Bumped into self
Trial: 101, Board width: 5, Score: 45, Body length: 15, Reason for end of game: Could not generate another move
Trial: 102, Board width: 5, Score: 44, Body length: 11, Reason for end of game: Bumped into self
Trial: 103, Board width: 5, Score: 32, Body length: 10, Reason for end of game: Could not generate another move
```

```
Trial: 104, Board width: 5, Score: 61, Body length: 17, Reason for end of game: Bumped into self
Trial: 105, Board width: 5, Score: 63, Body length: 13, Reason for end of game: Bumped into self
Trial: 106, Board width: 5, Score: 56, Body length: 15, Reason for end of game: Could not generate another move
Trial: 107, Board width: 5, Score: 47, Body length: 10, Reason for end of game: Bumped into self
Trial: 108, Board width: 5, Score: 56, Body length: 12, Reason for end of game: Could not generate another move
Trial: 109, Board width: 4, Score: 30, Body length: 9, Reason for end of game: Bumped into self
Trial: 110, Board width: 4, Score: 28, Body length: 12, Reason for end of game: Could not generate another move
Trial: 111, Board width: 4, Score: 32, Body length: 12, Reason for end of game: Could not generate another move
Trial: 112, Board width: 4, Score: 21, Body length: 7, Reason for end of game: Could not generate another move
Trial: 113, Board width: 4, Score: 32, Body length: 14, Reason for end of game: Could not generate another move
Trial: 114, Board width: 4, Score: 26, Body length: 9, Reason for end of game: Bumped into self
Trial: 115, Board width: 4, Score: 43, Body length: 12, Reason for end of game: Could not generate another move
Trial: 116, Board width: 3, Score: 14, Body length: 7, Reason for end of game: Could not generate another move
Trial: 117, Board width: 3, Score: 11, Body length: 5, Reason for end of game: Bumped into self
Trial: 118, Board width: 3, Score: 17, Body length: 7, Reason for end of game: Bumped into self
Trial: 119, Board width: 3, Score: 20, Body length: 8, Reason for end of game: Could not generate another move
Trial: 120, Board width: 3, Score: 14, Body length: 8, Reason for end of game: Could not generate another move
Trial: 121, Board width: 3, Score: 14, Body length: 8, Reason for end of game: Could not generate another move
Trial: 122, Board width: 3, Score: 8, Body length: 6, Reason for end of game: Could not generate another move
Trial: 123, Board width: 20, Score: 3148, Body length: 94, Reason for end of game: Bumped into self
Trial: 124, Board width: 20, Score: 2316, Body length: 74, Reason for end of game: Bumped into self
Trial: 125, Board width: 40, Score: 15077, Body length: 189, Reason for end of game: Bumped into self
Trial: 126, Board width: 40, Score: 16064, Body length: 201, Reason for end of game: Bumped into self
Trial: 127, Board width: 80, Score: 5034, Body length: 85, Reason for end of game: Bumped into self
Trial: 128 Board width: 10, Score: 365, Body length: 36, Reason for end of game: Could not generate another move
Trial: 129 Board width: 10, Score: 572, Body length: 46, Reason for end of game: Could not generate another move
Trial: 130 Board width: 10, Score: 148, Body length: 20, Reason for end of game: Bumped into self
Trial: 131 Board width: 10, Score: 361, Body length: 31, Reason for end of game: Bumped into self
Trial: 132 Board width: 10, Score: 100, Body length: 15, Reason for end of game: Could not generate another move
Trial: 133 Board width: 10, Score: 262, Body length: 25, Reason for end of game: Could not generate another move
Trial: 134 Board width: 10, Score: 149, Body length: 17, Reason for end of game: Could not generate another move
Trial: 135 Board width: 10, Score: 465, Body length: 40, Reason for end of game: Could not generate another move
Trial: 136 Board width: 10, Score: 195, Body length: 20, Reason for end of game: Bumped into self
Trial: 137 Board width: 10, Score: 189, Body length: 21, Reason for end of game: Bumped into self
Trial: 138 Board width: 10, Score: 407, Body length: 35, Reason for end of game: Could not generate another move
Trial: 139 Board width: 10, Score: 499, Body length: 36, Reason for end of game: Could not generate another move
```