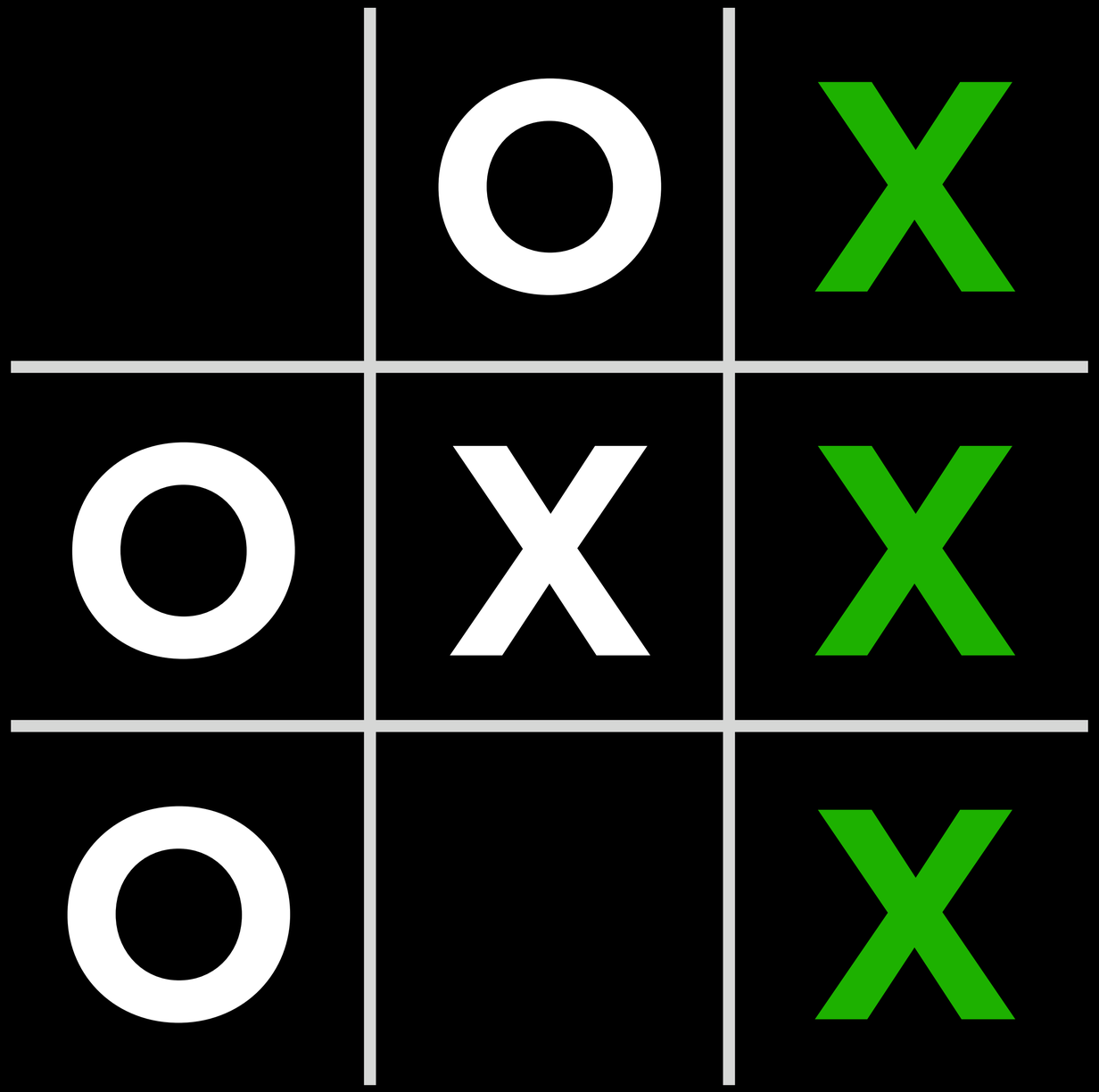
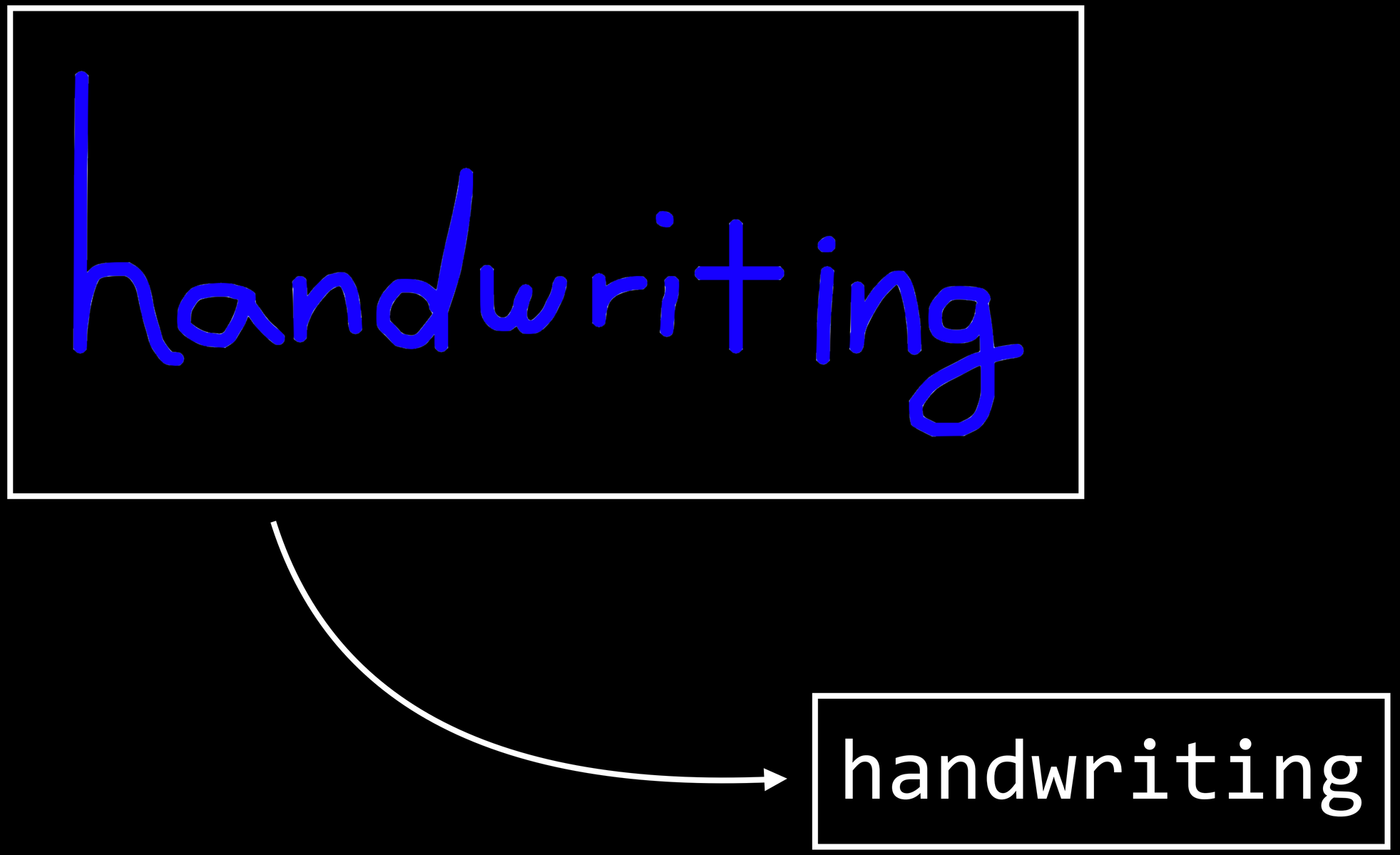
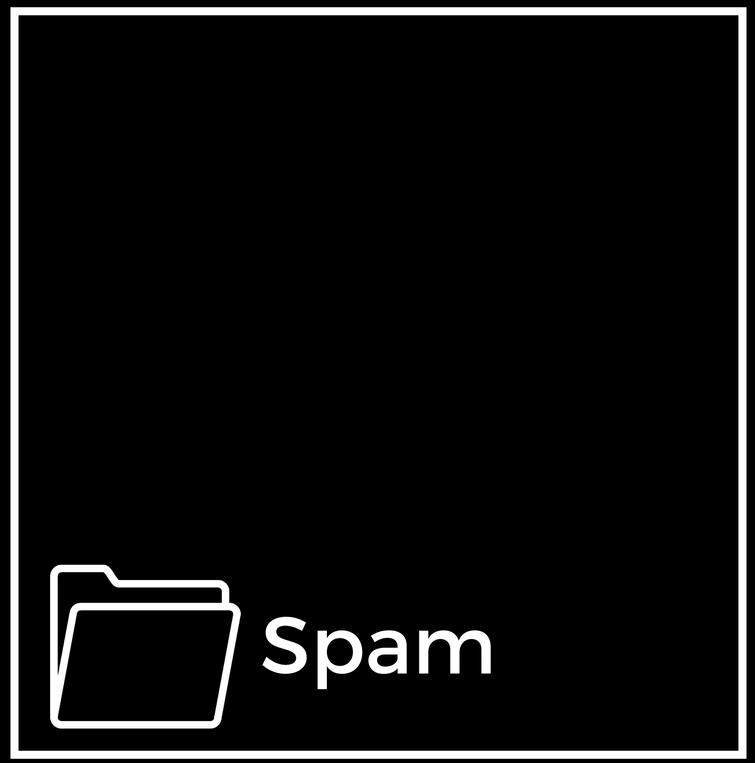
CS50 for JDs

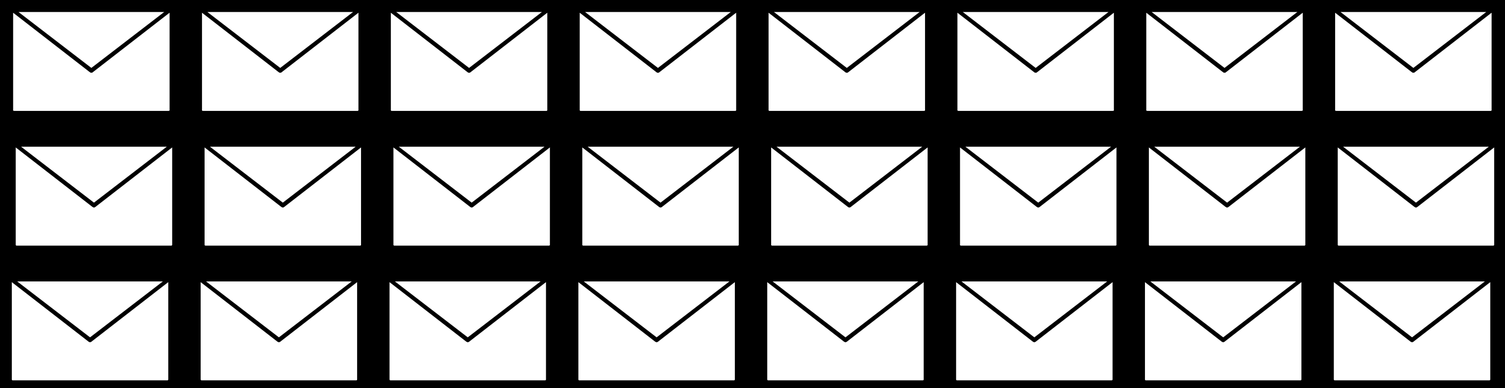
Artificial Intelligence


O X
O X X
O X


handwriting
handwriting

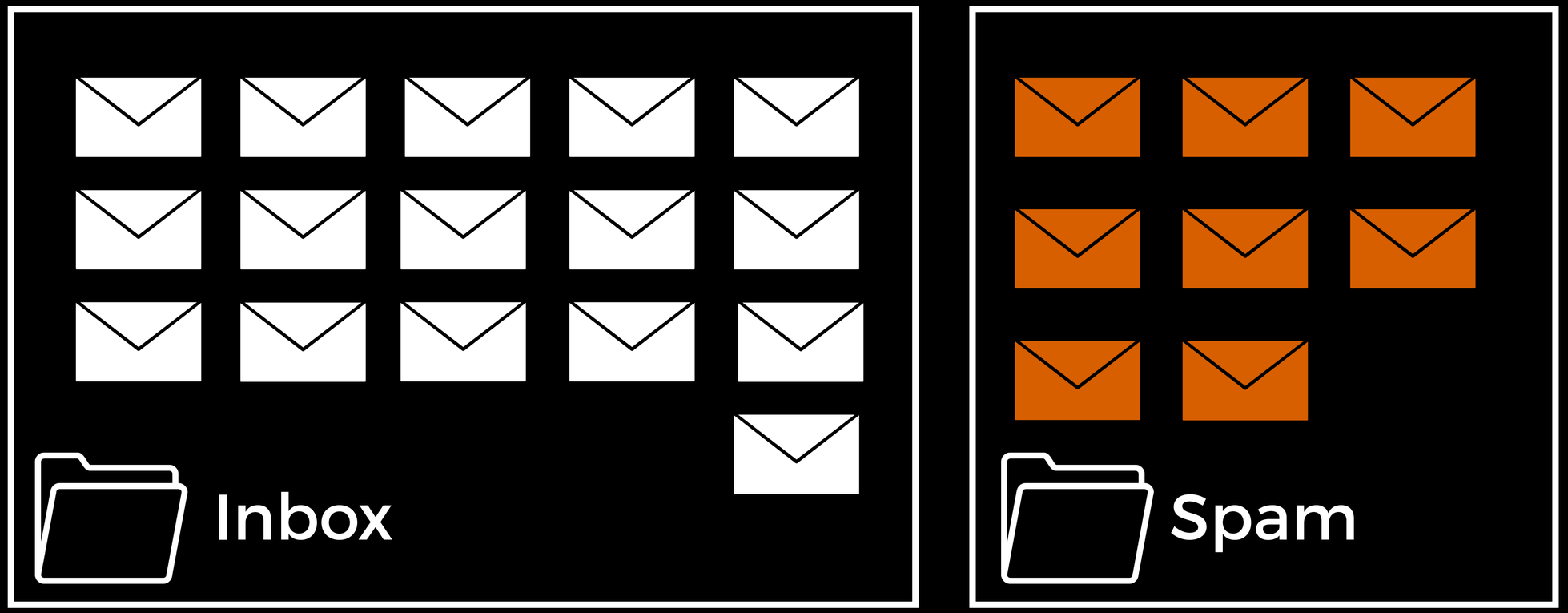

Inbox

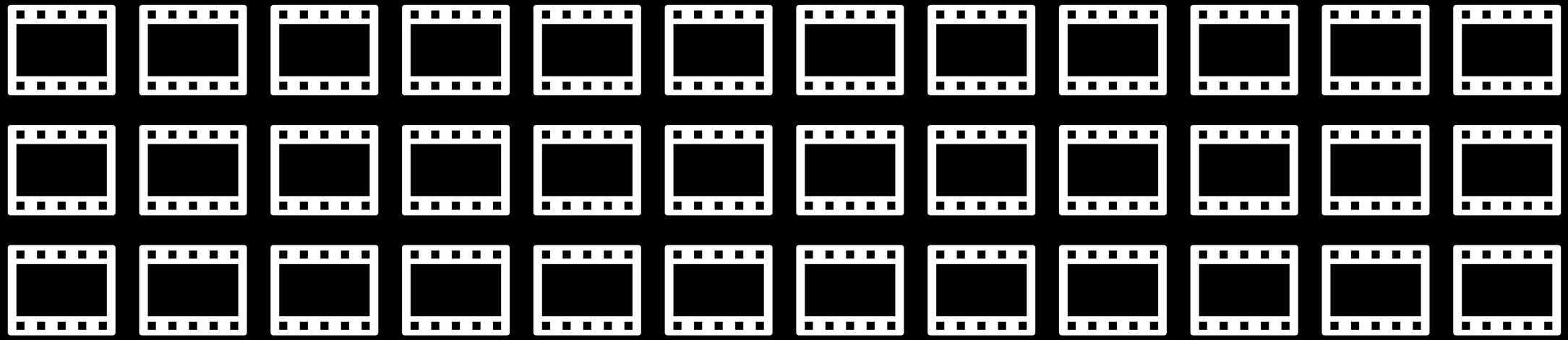

Spam



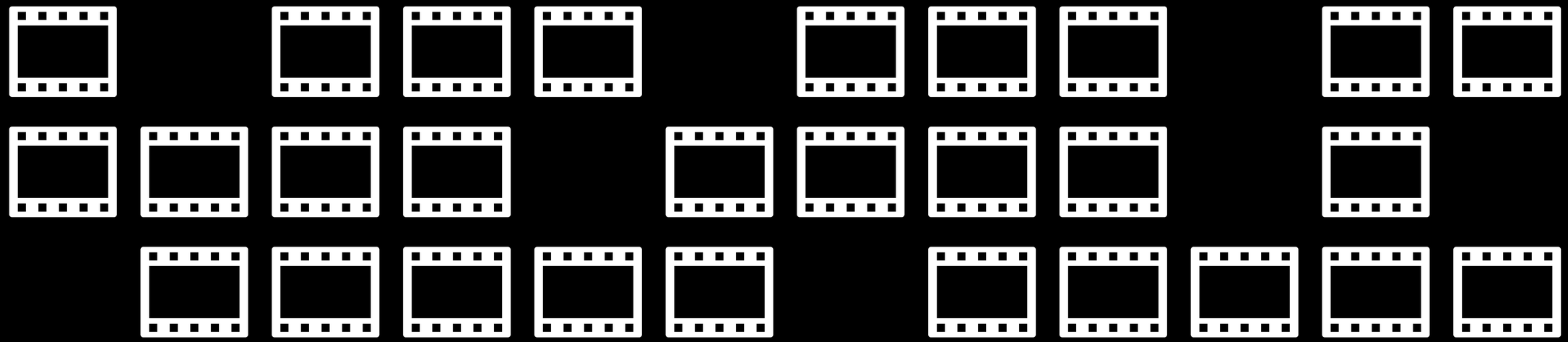
Inbox

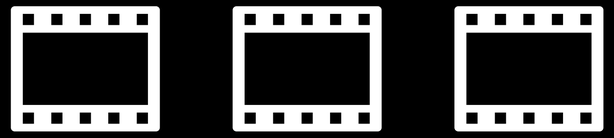
Spam


Inbox
Spam



Watch History

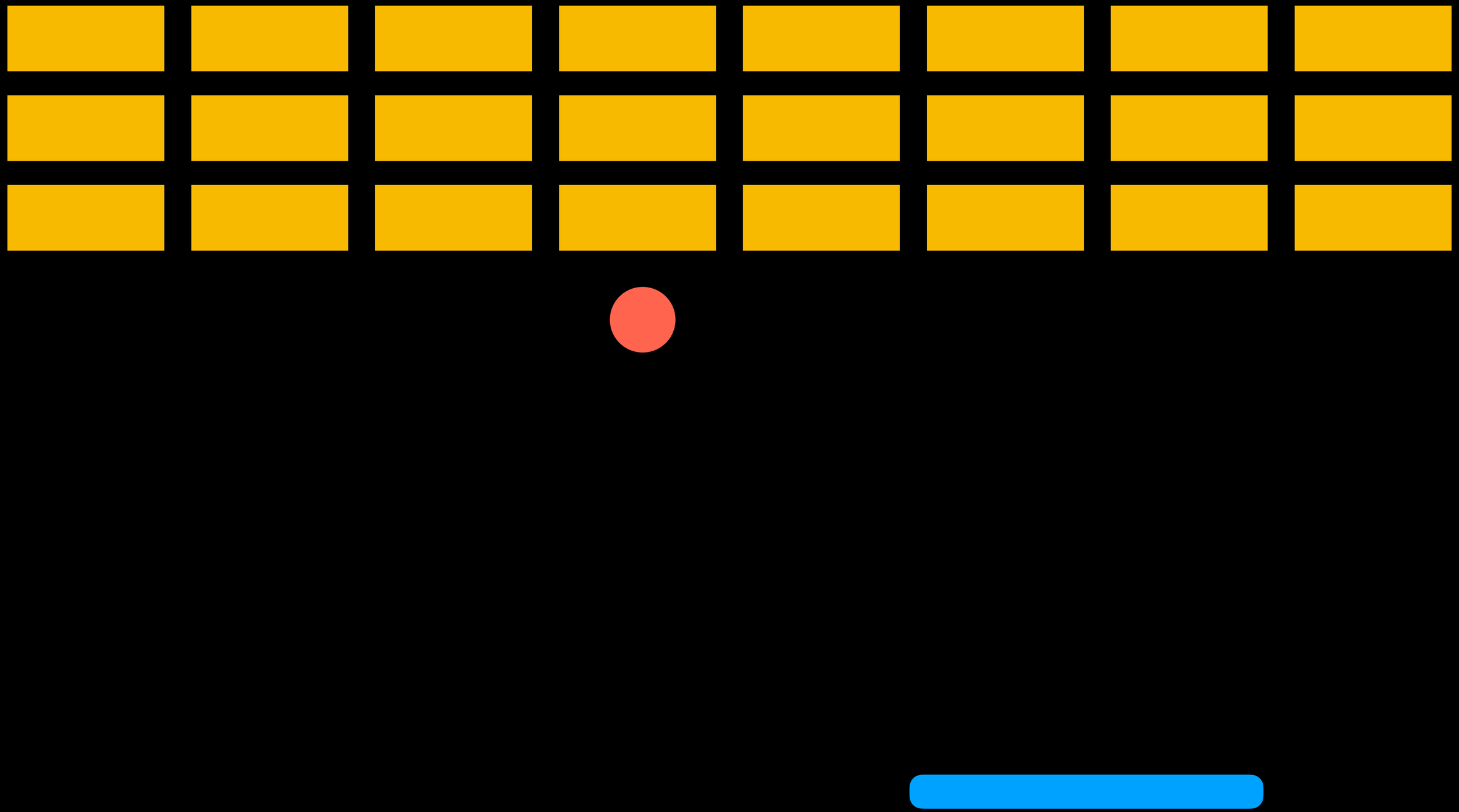


Watch History 

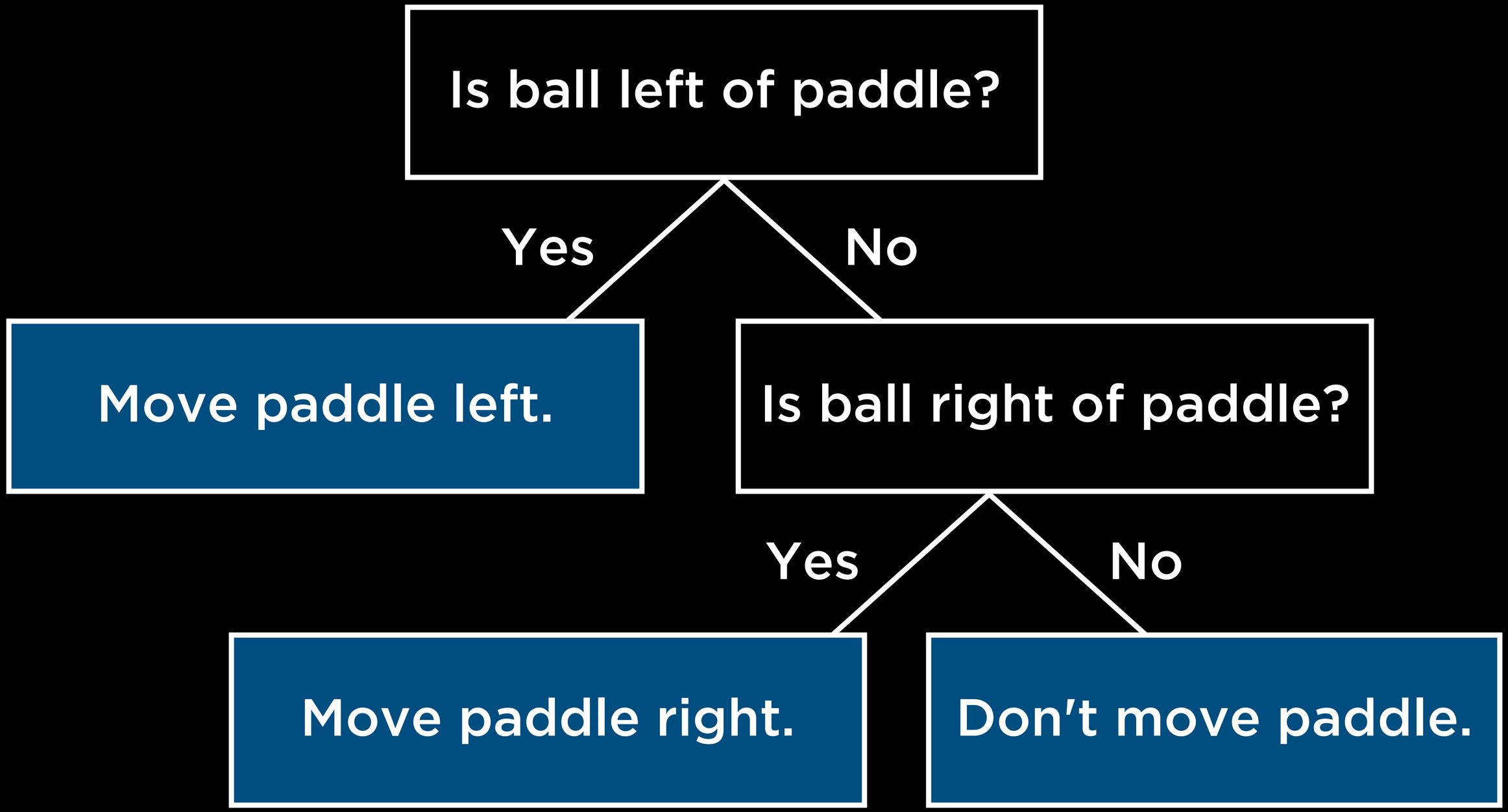
Recommended 

Artificial Intelligence

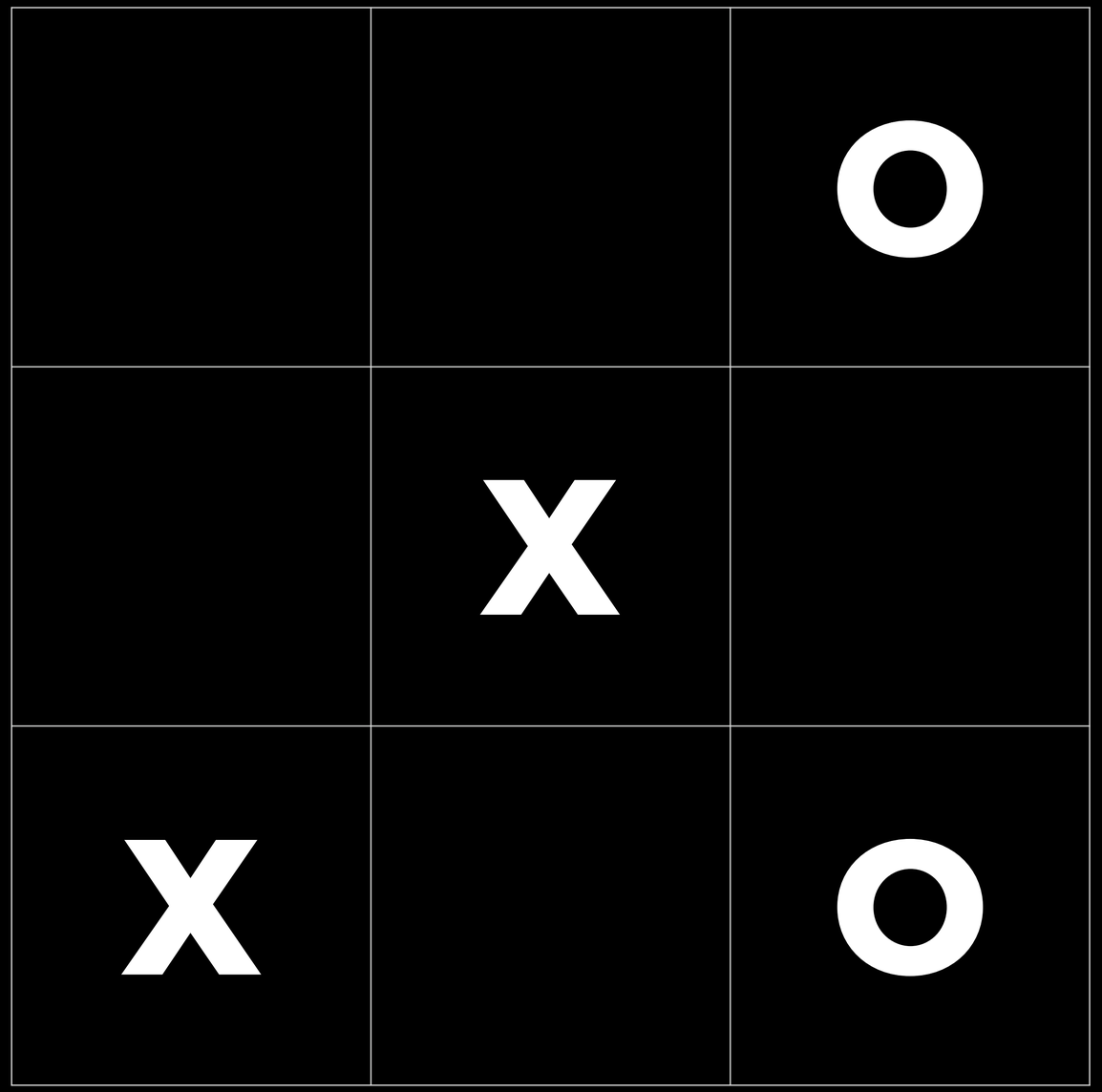
Decision-Making

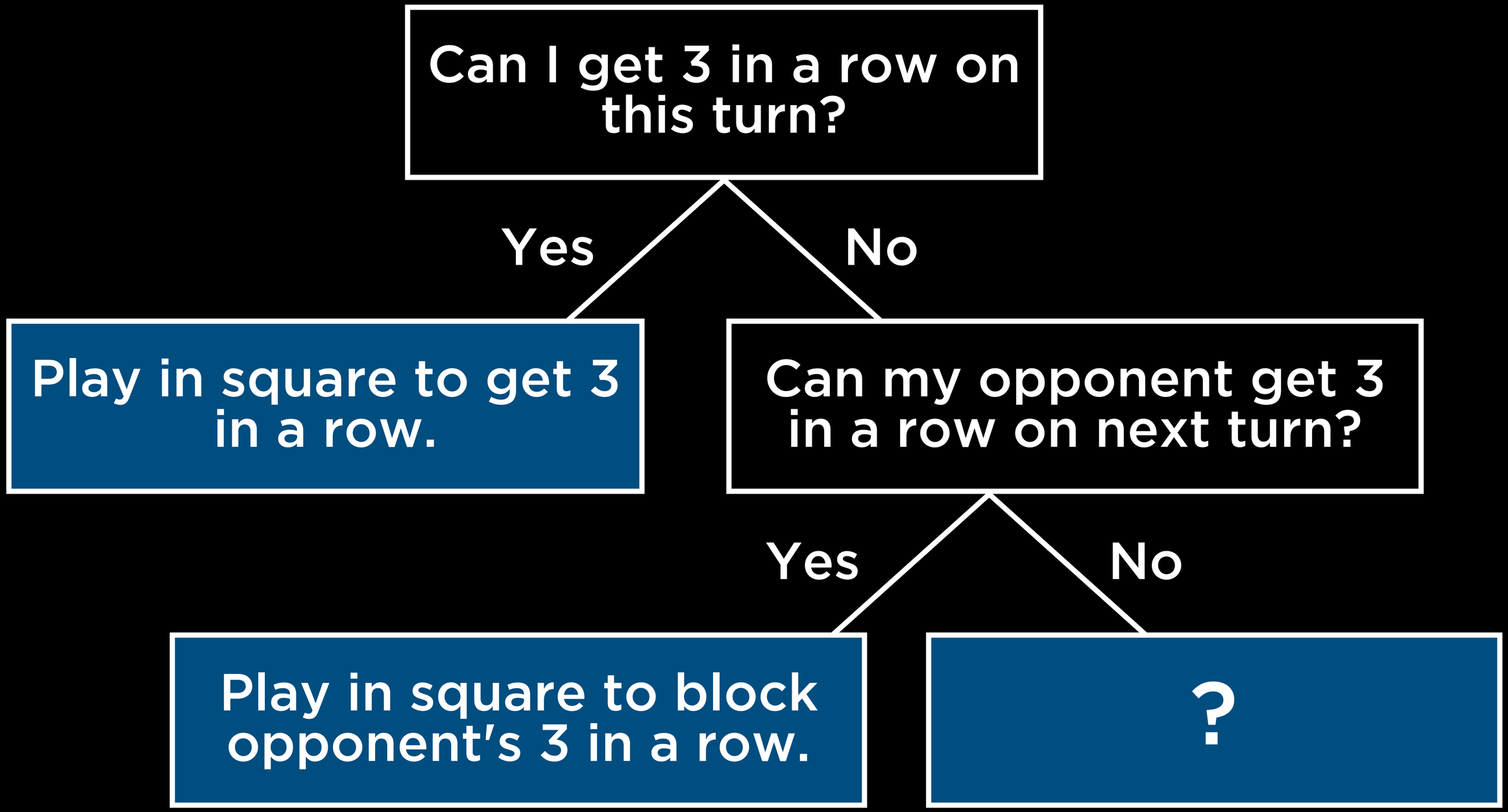


Decision Trees


Is ball left of paddle?
Yes
No
Move paddle left.
Is ball right of paddle?
Yes
No
Move paddle right.
Don't move paddle.

```
while game is ongoing:
    if ball left of paddle:
        move paddle left
    else if ball right of paddle:
        move paddle right
    else:
        don't move paddle
```

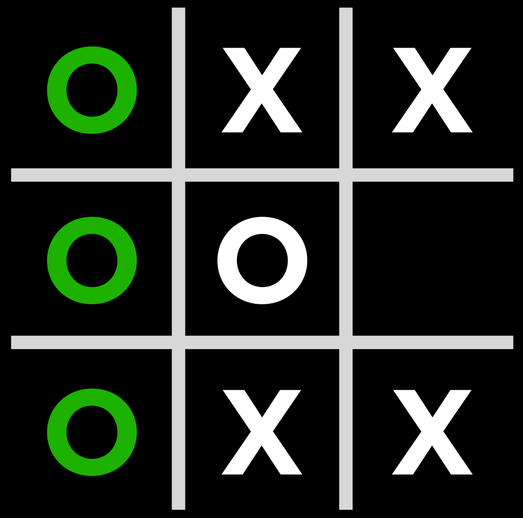

O
X
X
O


Can I get 3 in a row on this turn?
Yes
No
Play in square to get 3 in a row.
Can my opponent get 3 in a row on next turn?
Yes
No
Play in square to block opponent's 3 in a row.
?

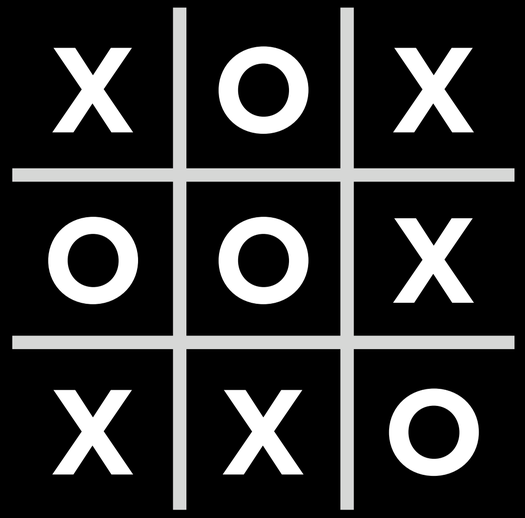
Optimal Decision-Making

Minimax

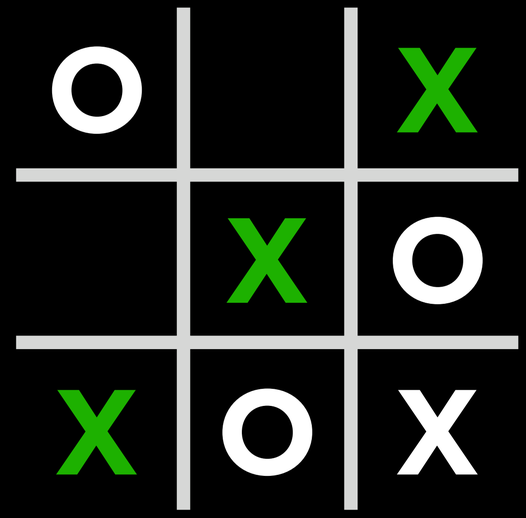
- MAX (X) aims to maximize score.
- MIN (O) aims to minimize score.



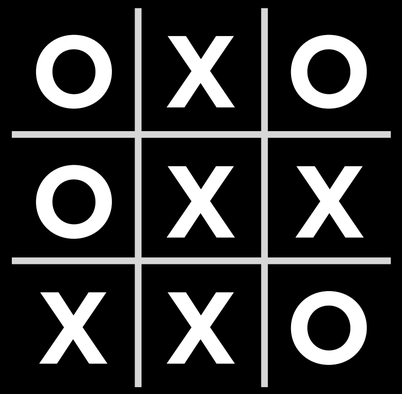
-1



0



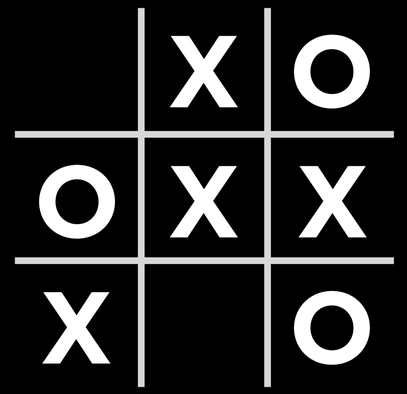
1



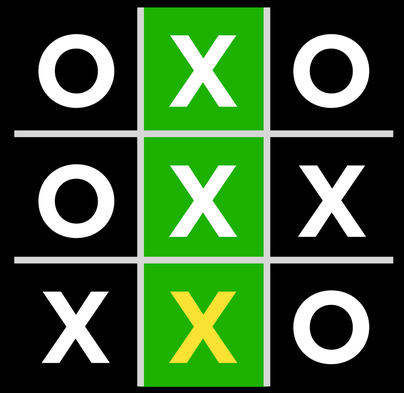
VALUE: 1

PLAYER(s) = O

VALUE: 0



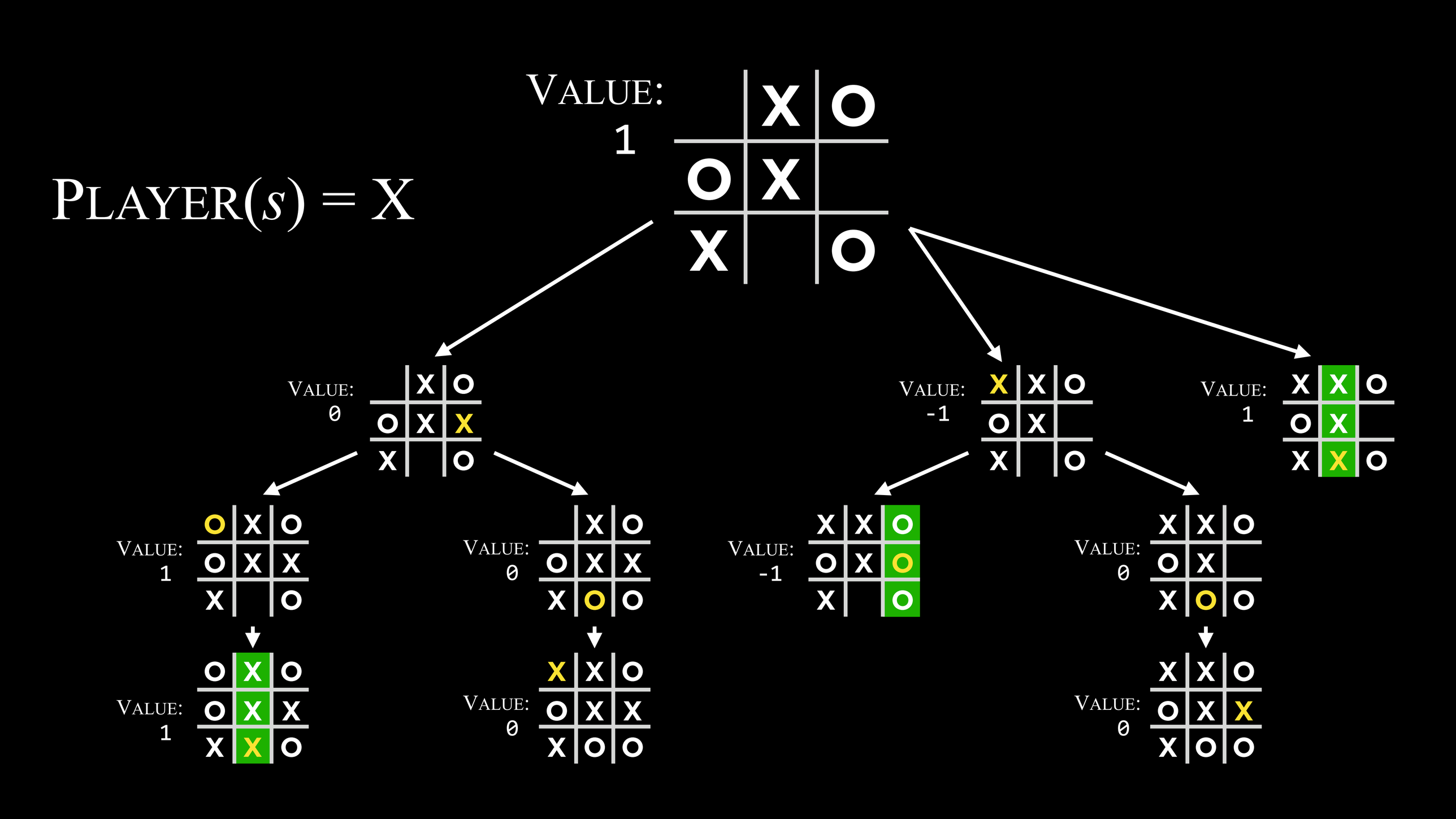
VALUE: 1



VALUE: 1

VALUE: 0

VALUE: 0


Player(s) = X
Value: 1
Value: 0
Value: -1
Value: 1
Value: 1
Value: 0
Value: -1
Value: 0
Value: 1
Value: 0
Value: 0

Minimax

```
if player is X:
    for all possible moves:
        calculate score for board
    choose move with highest score

else:
    for all possible moves:
        calculate score for board
    choose move with lowest score
```

255,168

total possible Tic-Tac-Toe games

288,000,000,000

total possible chess games
after four moves each

10^{29241}

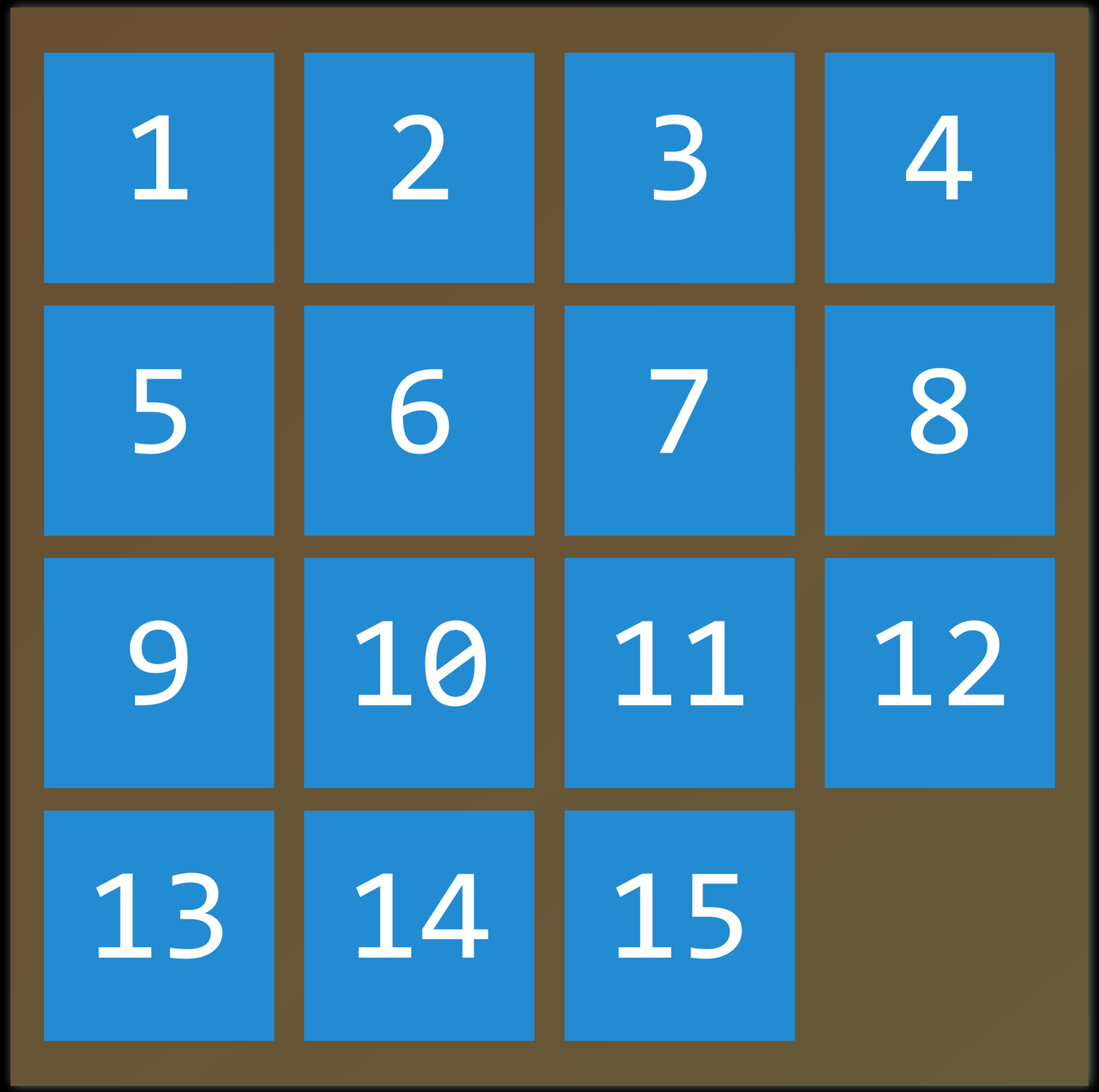
total possible chess games
(lower bound)

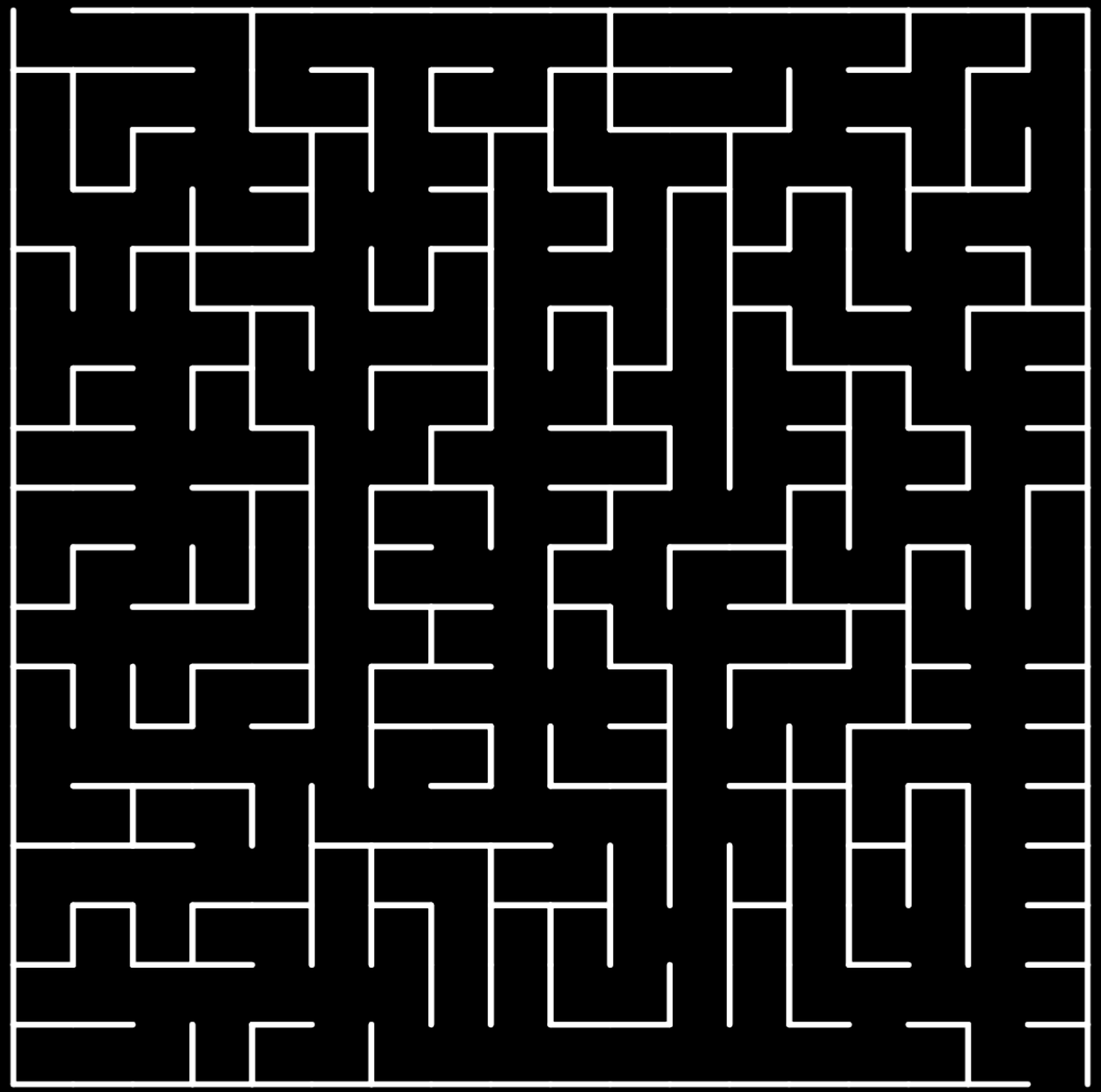
Depth-Limited Minimax

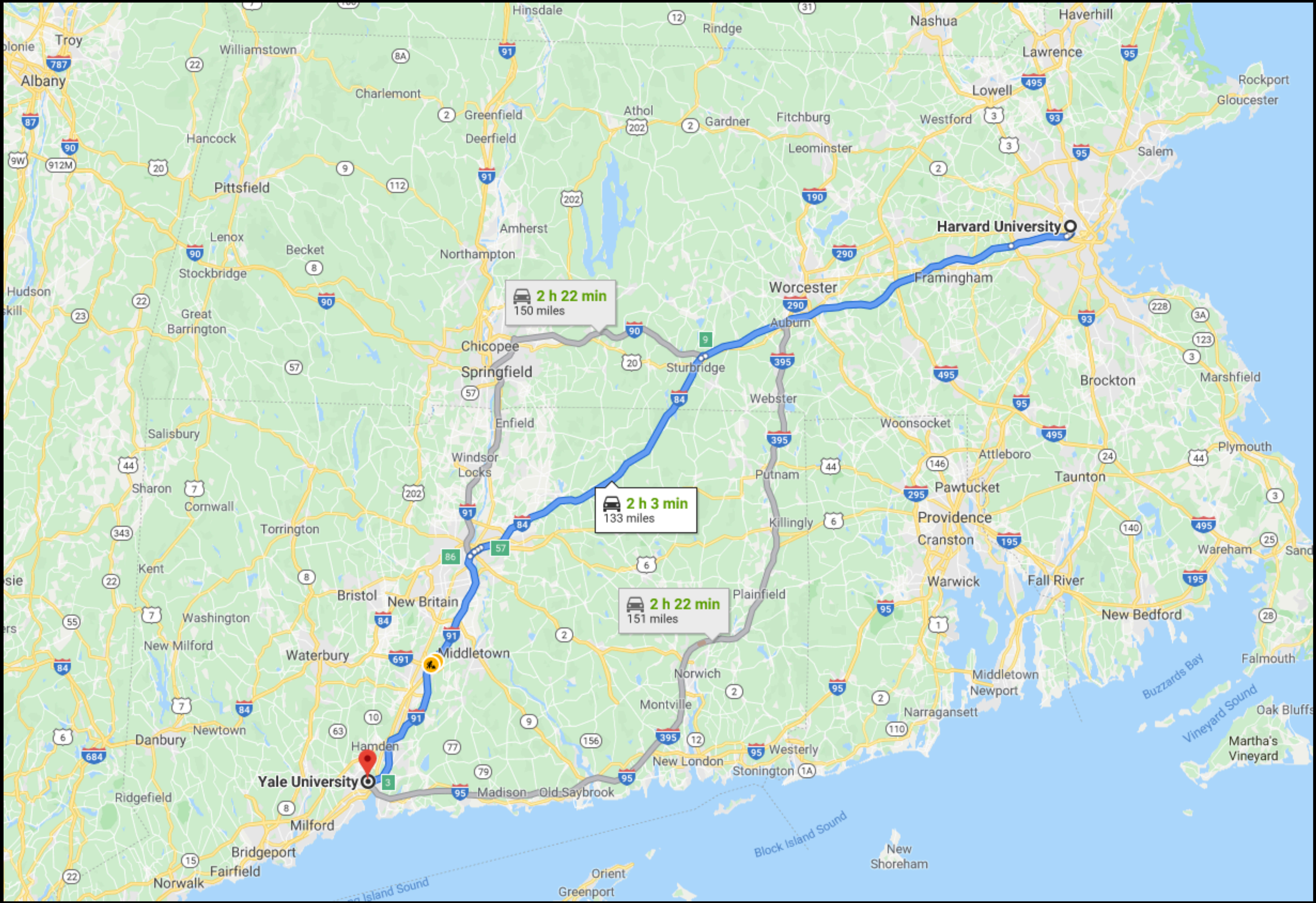
evaluation function

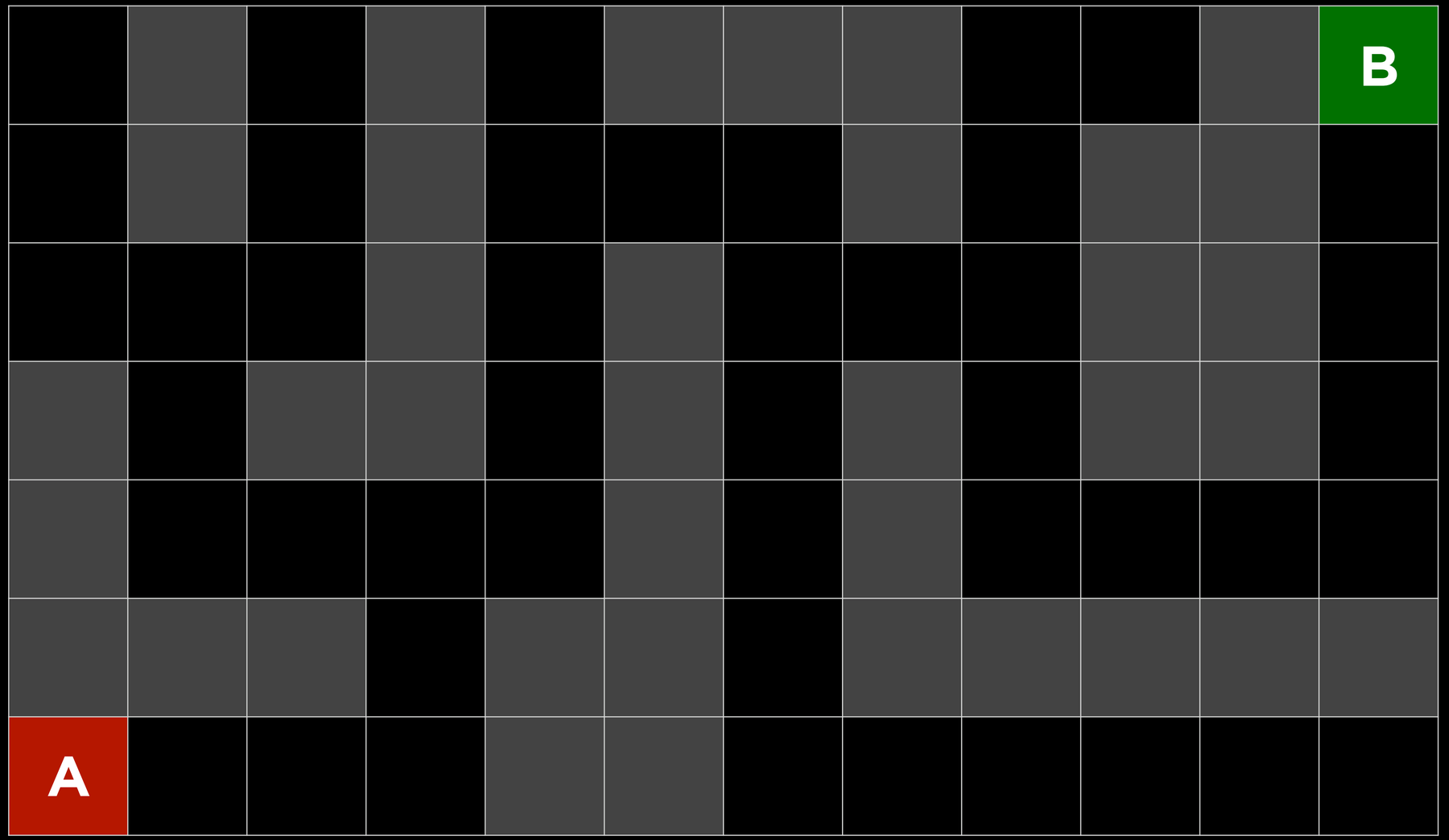
function that estimates the expected utility of the game from a given state

Search


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

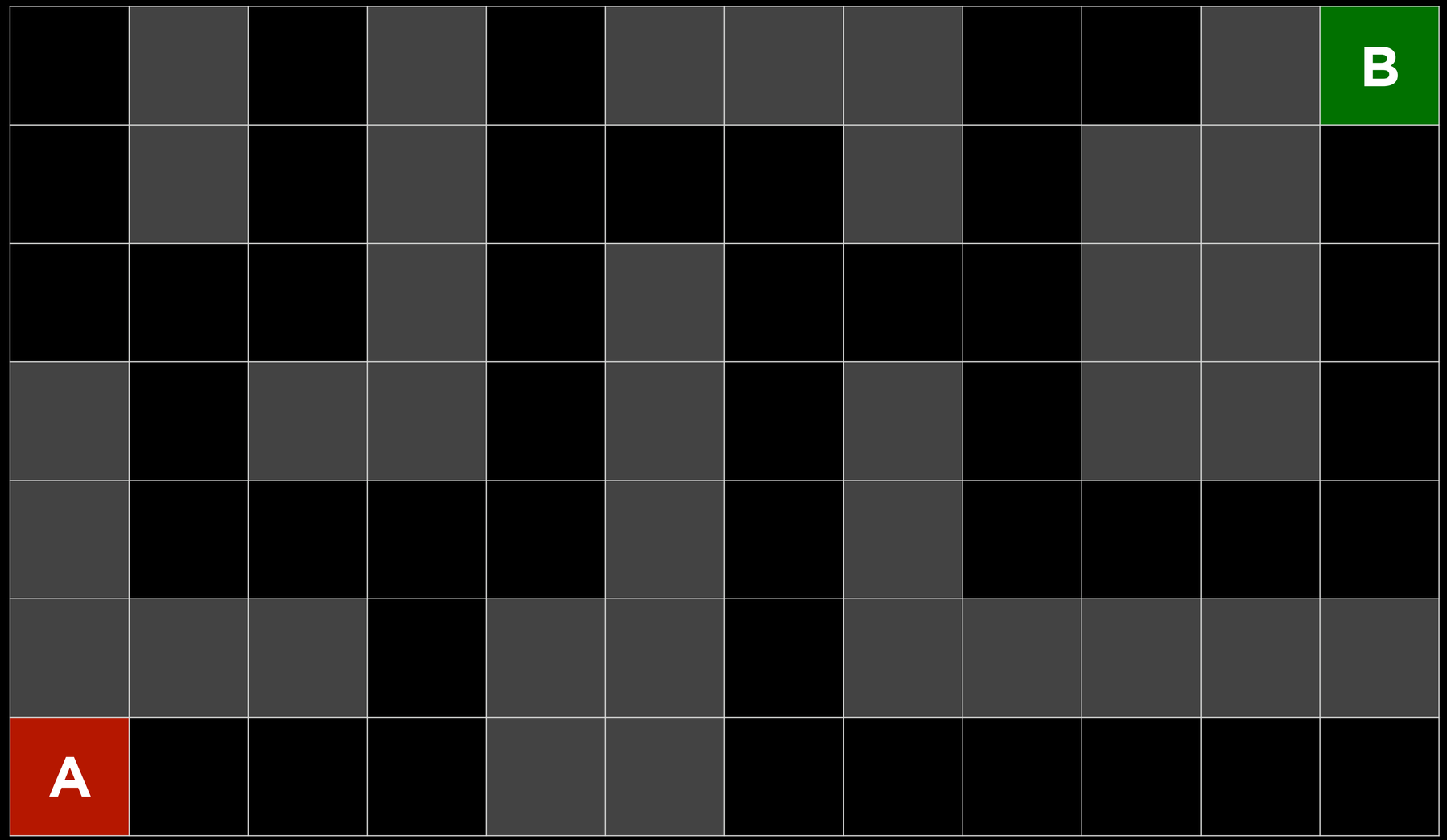



Harvard University
Yale University
2 h 22 min
150 miles
2 h 3 min
133 miles
2 h 22 min
151 miles
Albany
Troy
Williamstown
Hinsdale
Ringe
Nashua
Haverhill
Lawrence
Lowell
Rockport
Gloucester
Charlemont
Greenfield
Deerfield
Athol
Gardner
Fitchburg
Westford
Leominster
Salem
Hancock
Pittsfield
Amherst
Northampton
Lenox
Becket
Stockbridge
Worcester
Framingham
Auburn
Great Barrington
Chicopee
Springfield
Sturbridge
Webster
Brockton
Marshfield
Enfield
Woonsocket
Plymouth
Salisbury
Windsor Locks
Putnam
Attleboro
Taunton
Sharon
Cornwall
Pawtucket
Torrington
Killingly
Providence
Cranston
Kent
Warwick
Fall River
Wareham
Bristol
New Britain
Washington
Plainfield
New Bedford
New Milford
Waterbury
Middletown
Falmouth
Norwich
Middletown
Newport
Narragansett
Montville
Oak Bluffs
Danbury
Newtown
Hamden
Martha's Vineyard
Westerly
New London
Stonington
Madison
Old Saybrook
Ridgefield
Milford
Bridgeport
Fairfield
Norwalk
Orient
Greenport
New Shoreham
Block Island Sound
Buzzards Bay
Vineyard Sound

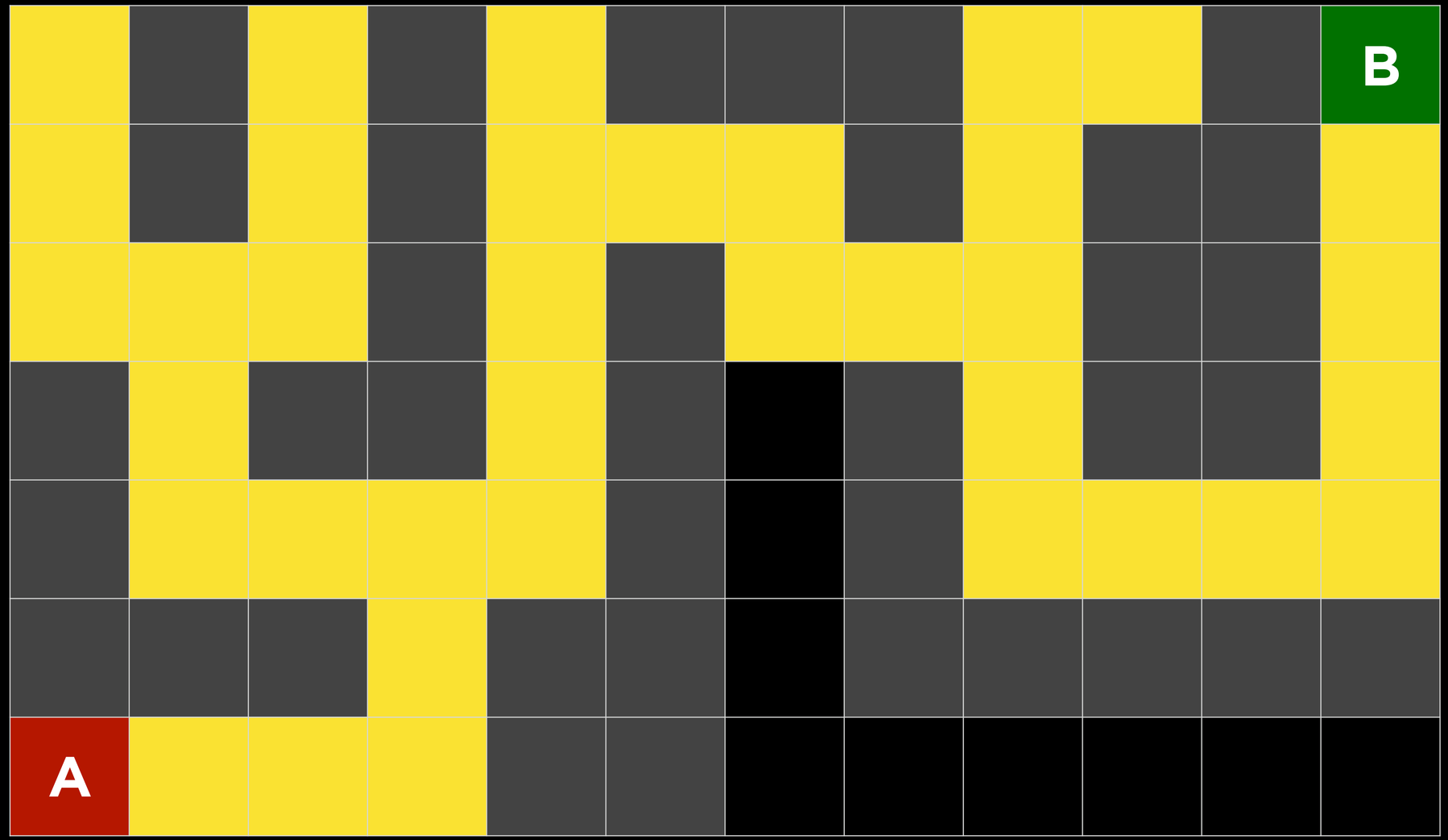

B
A

Depth-First Search

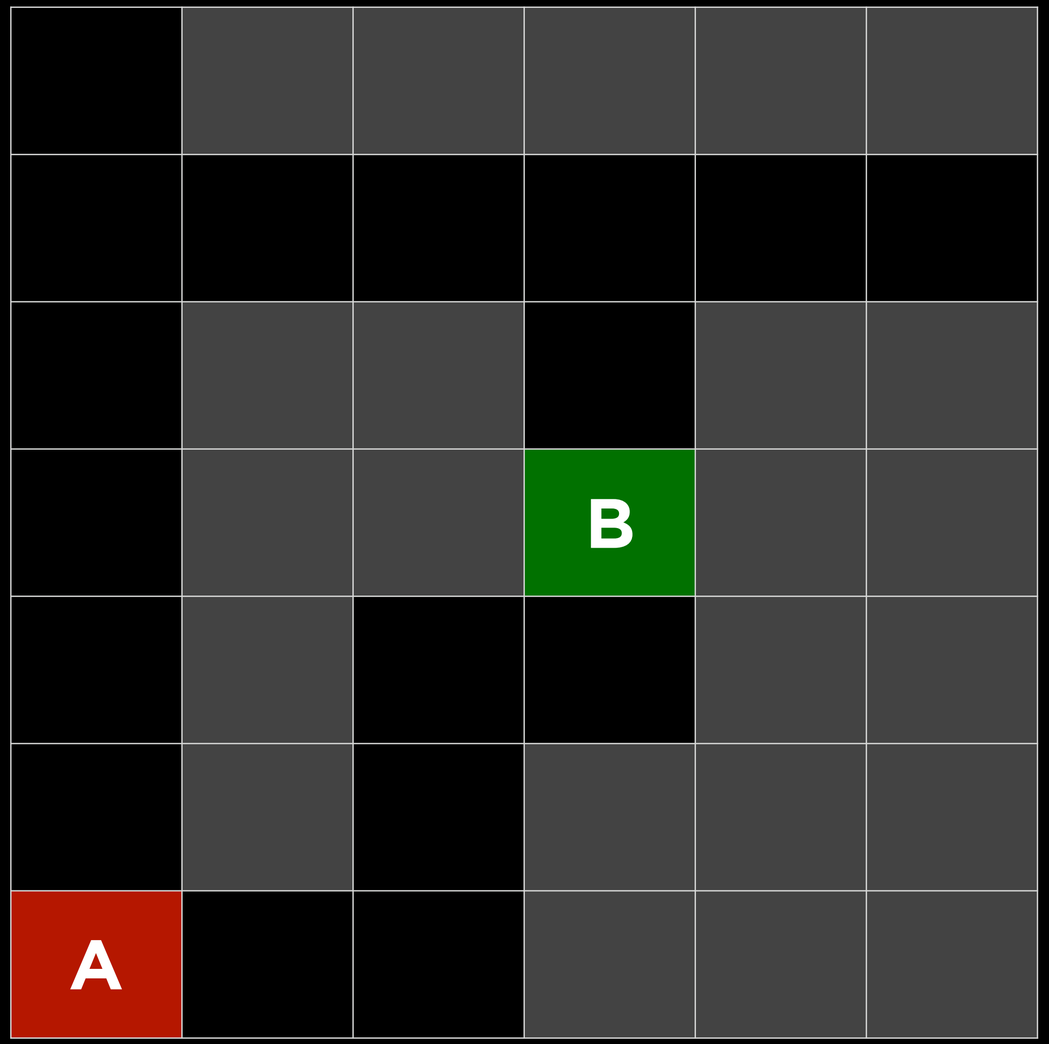
Depth-First Search



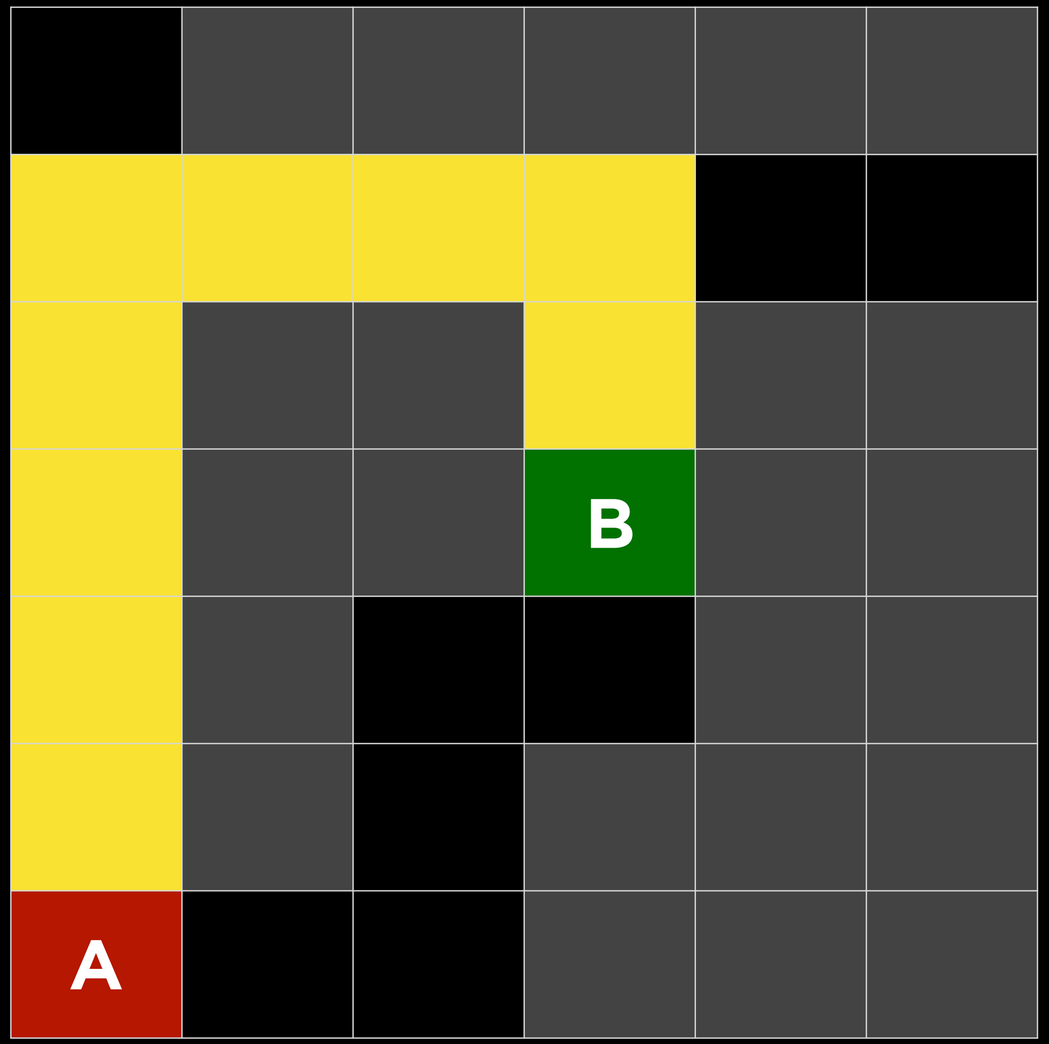
Depth-First Search



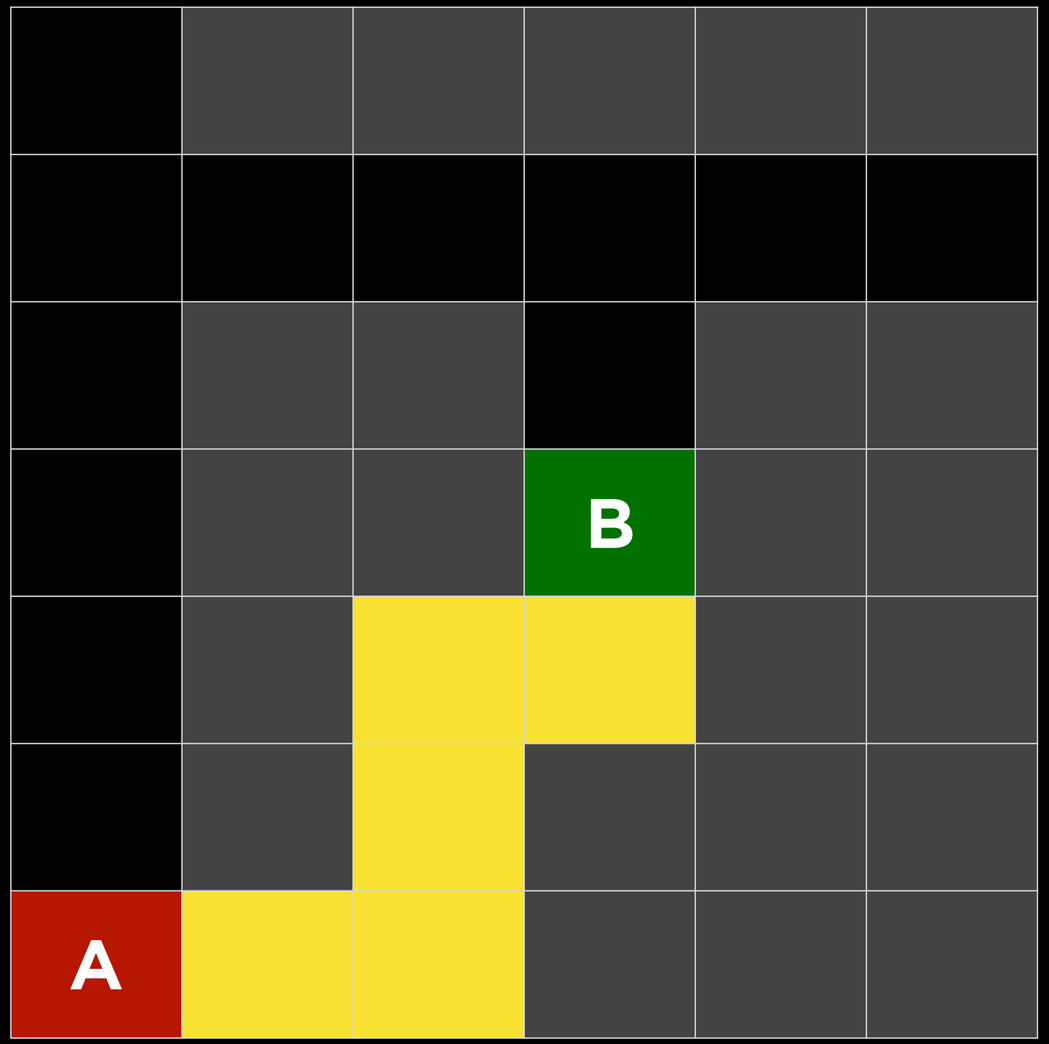
Depth-First Search



Depth-First Search

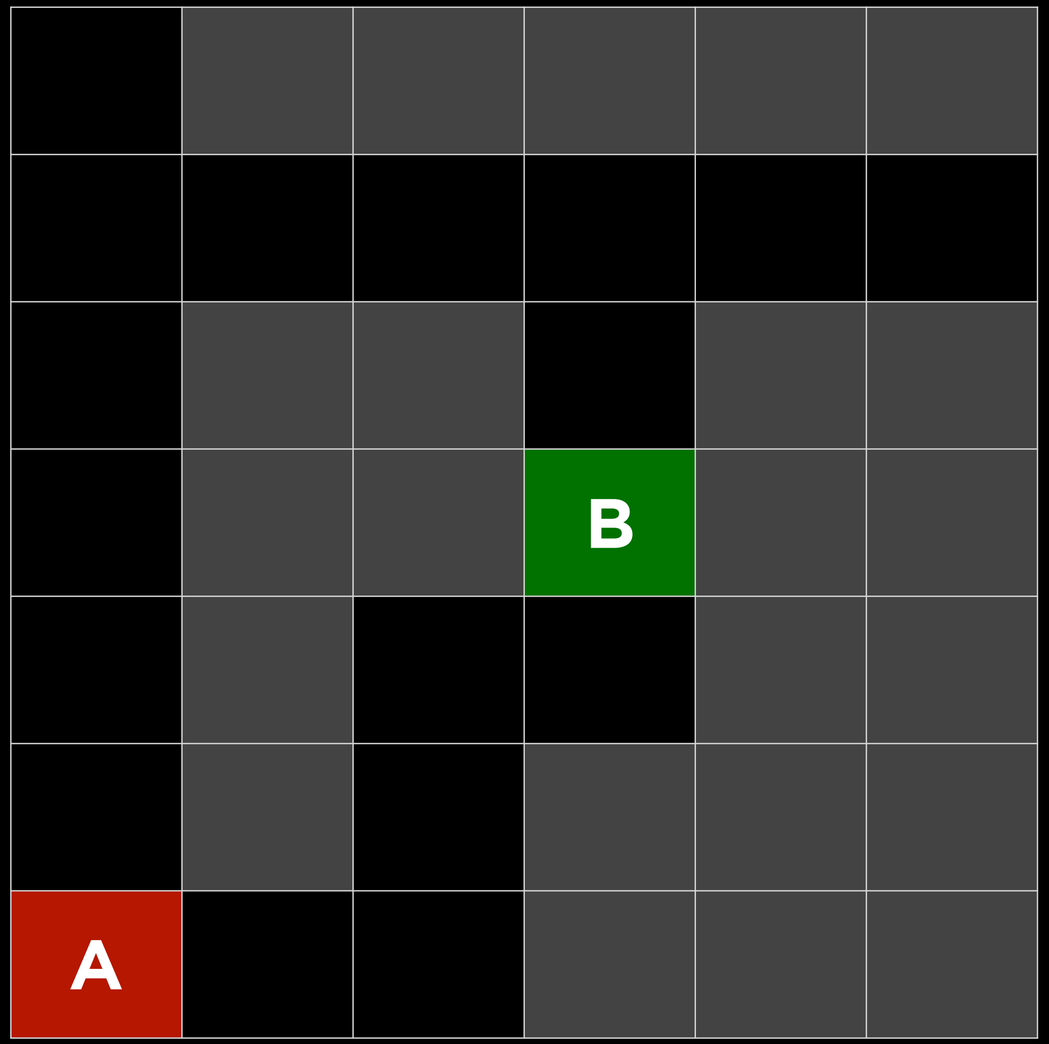


Depth-First Search

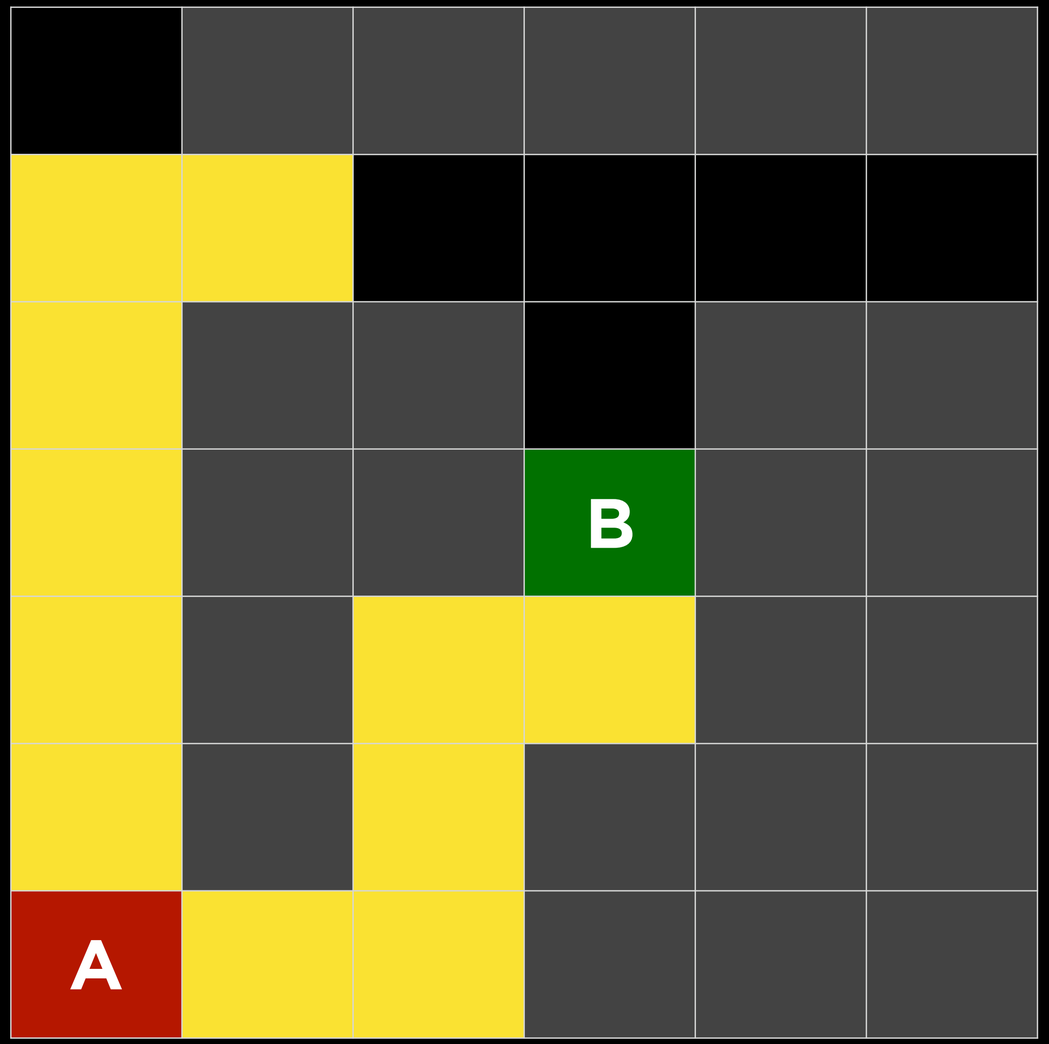


Breadth-First Search

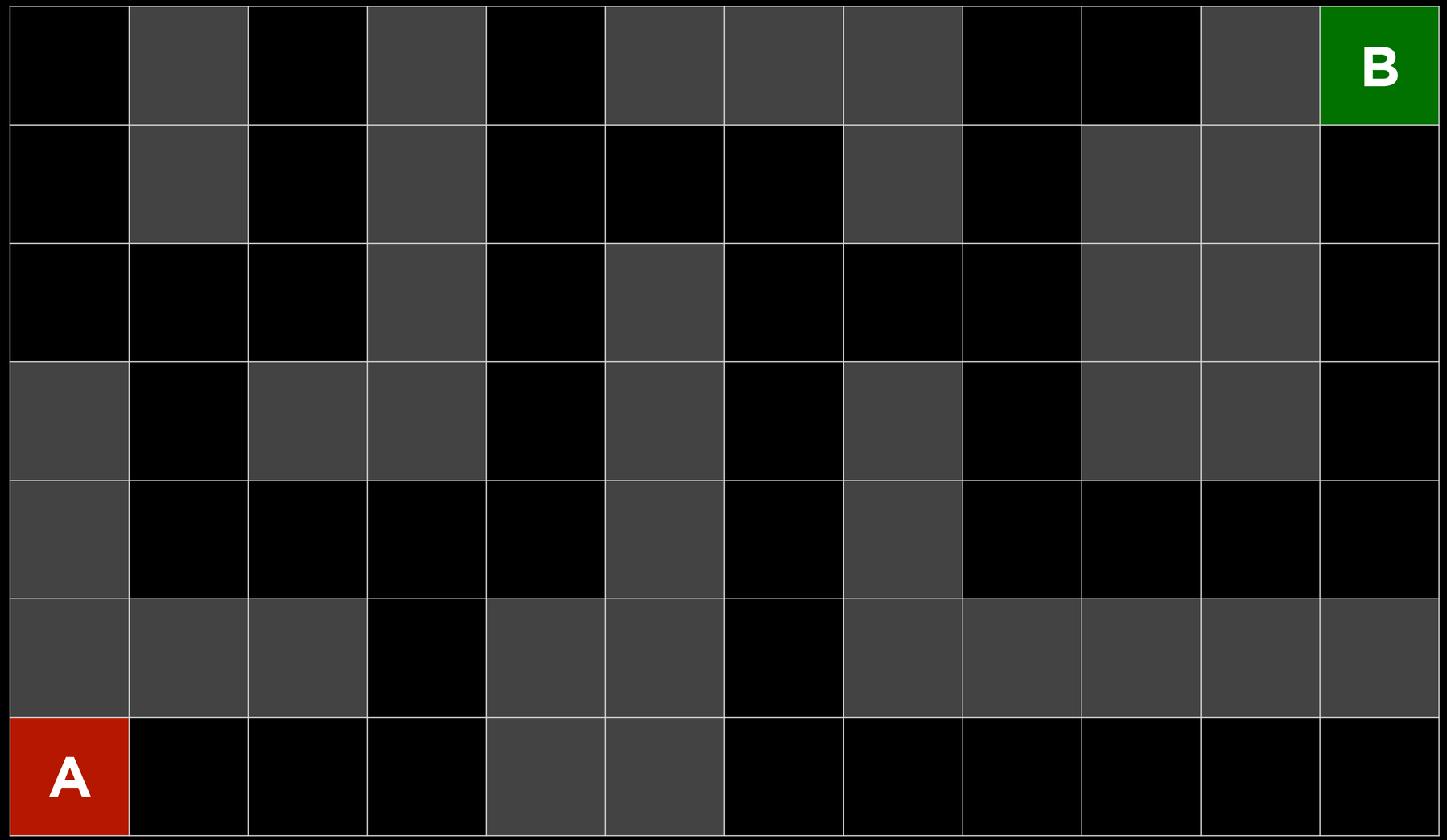
Depth-First Search



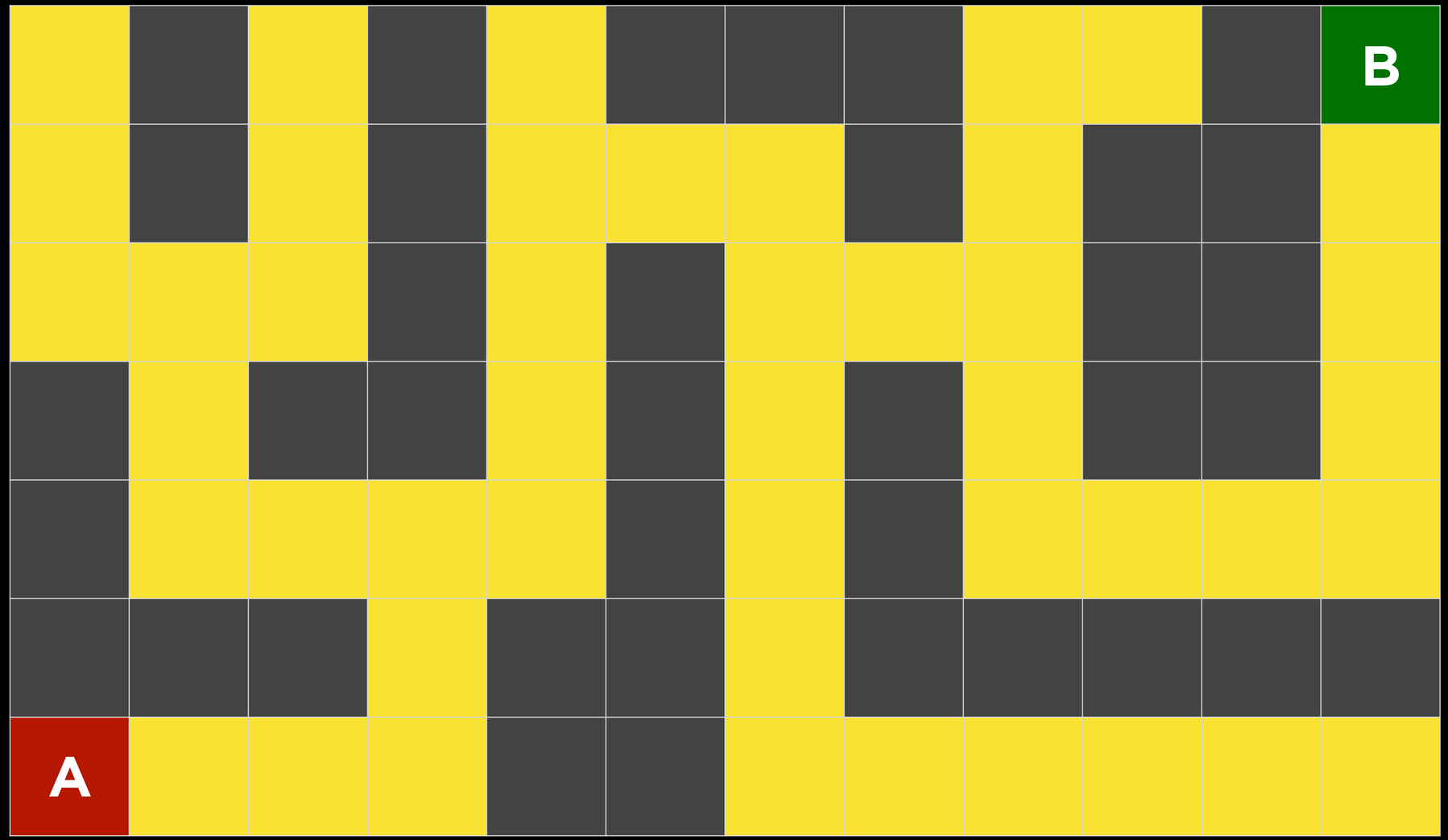
Depth-First Search



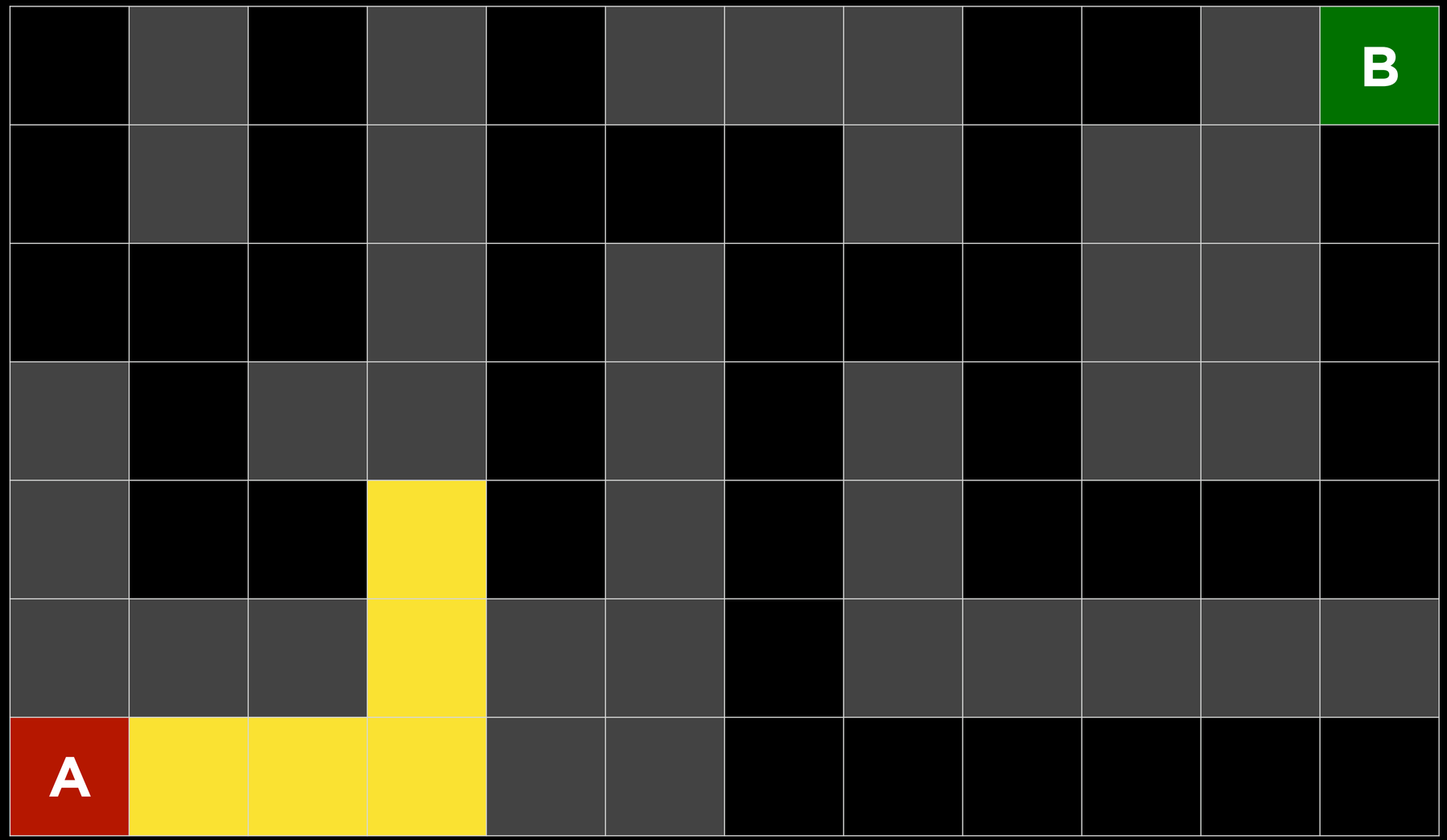
Breadth-First Search



Breadth-First Search



Breadth-First Search



uninformed search

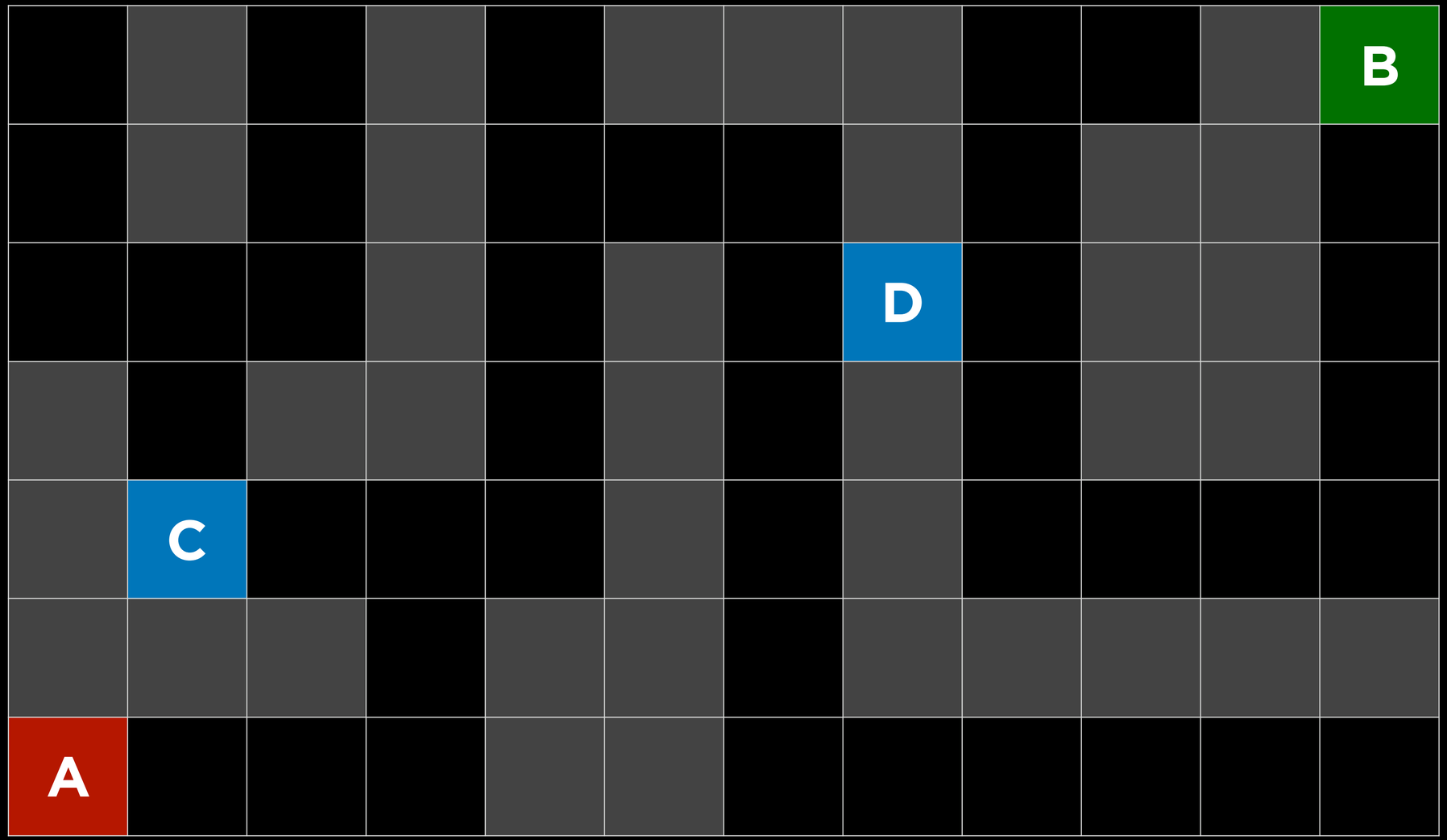
informed search

search strategy that uses problem-specific knowledge to find solutions more efficiently

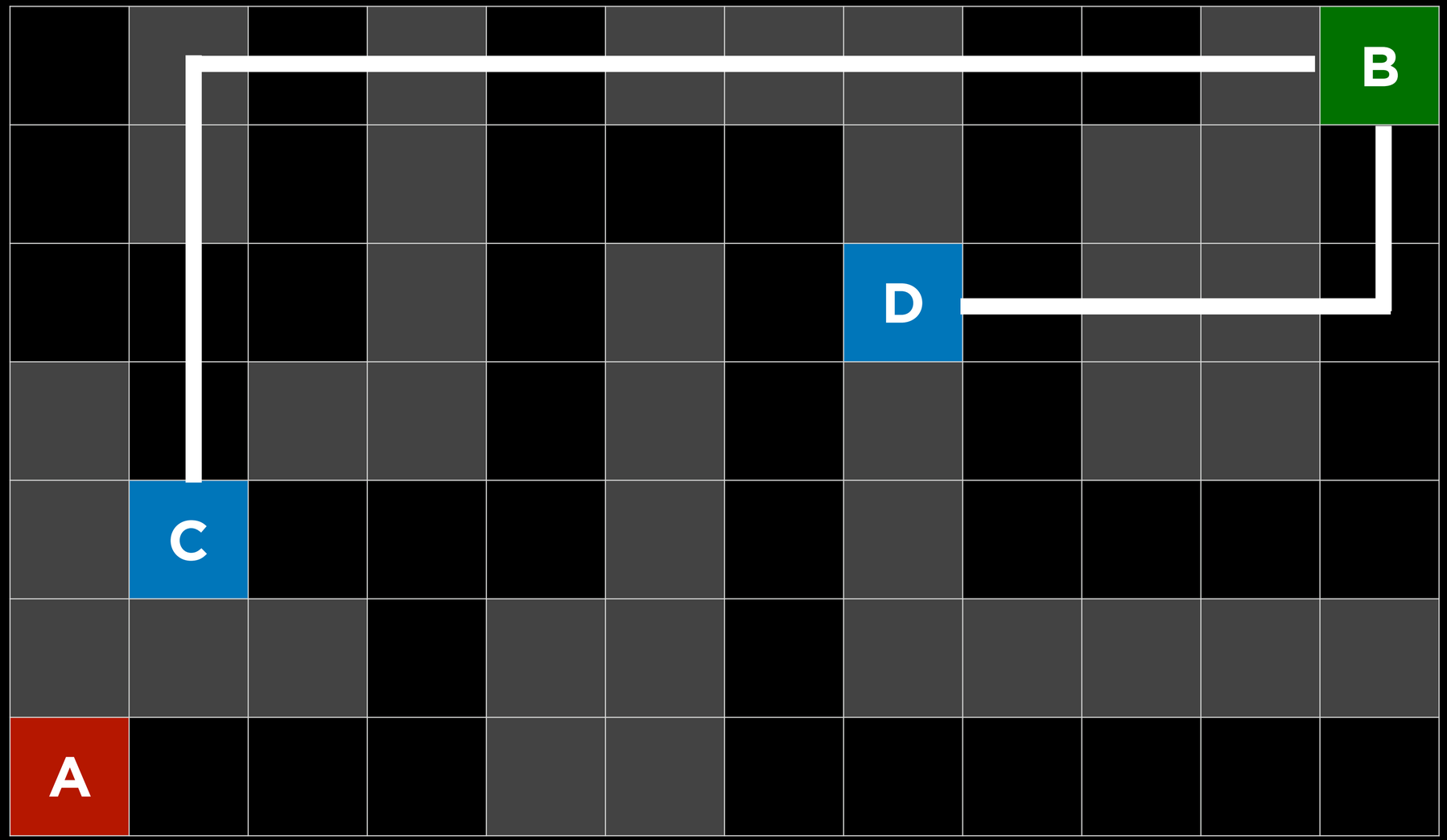
greedy best-first search

search algorithm that chooses what to explore based on a heuristic

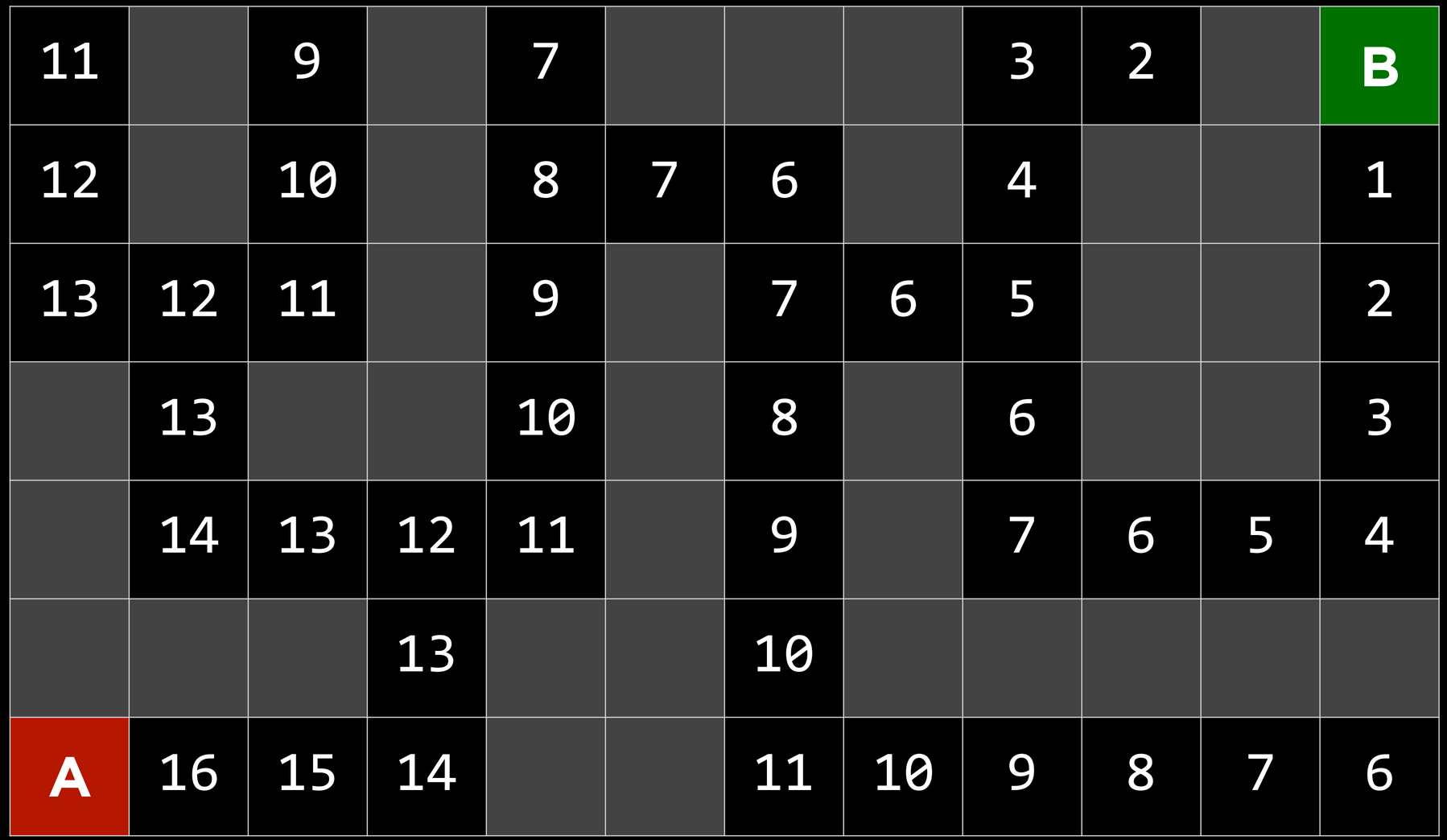
Heuristic function?



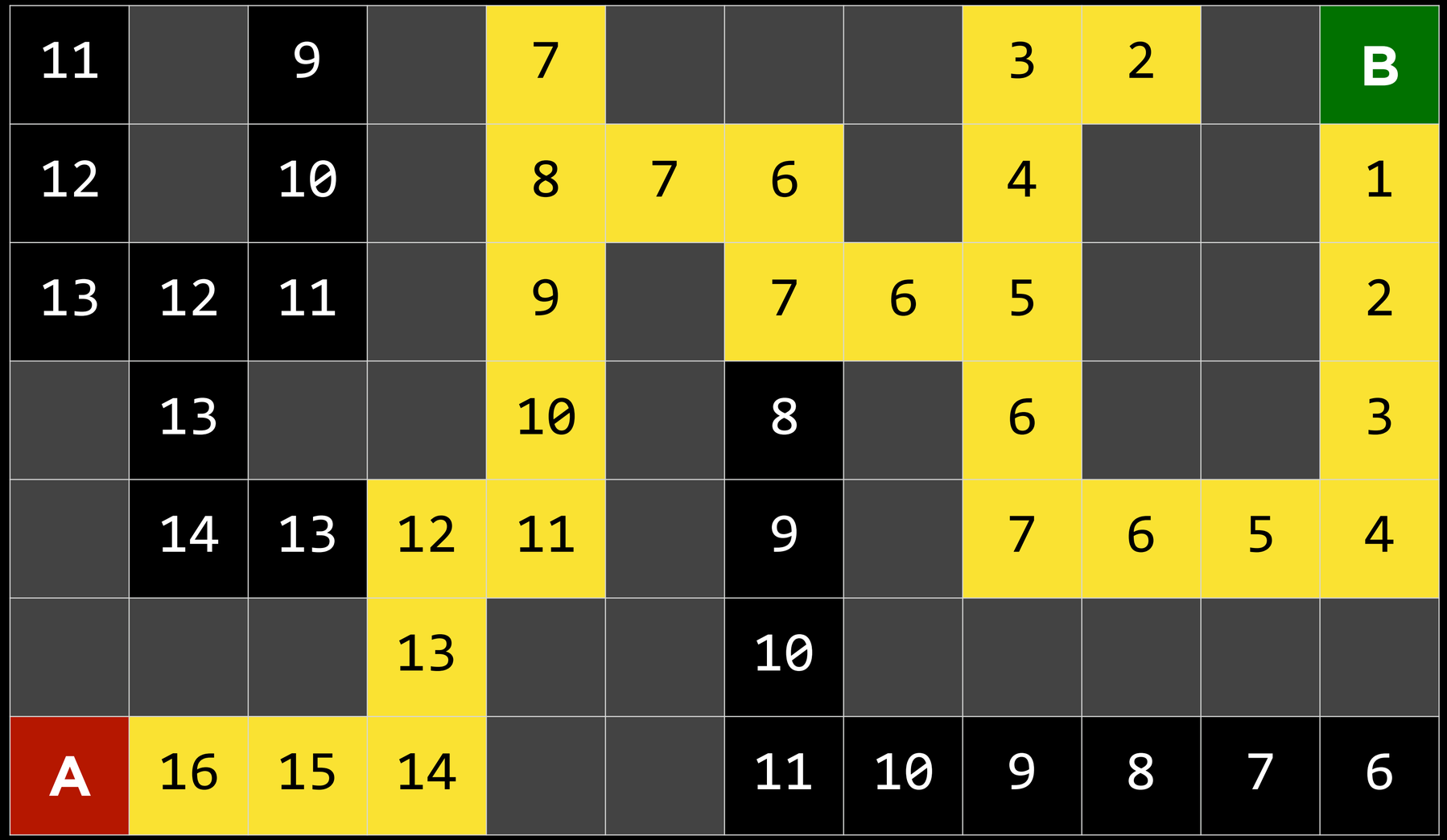
Heuristic function? Manhattan distance.



Greedy Best-First Search



Greedy Best-First Search



Greedy Best-First Search

	10	9	8	7	6	5	4	3	2	1	**B**
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	16	15	14		12	11	10	9	8	7	6

Greedy Best-First Search

	10	9	8	7	6	5	4	3	2	1	**B**
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	16	15	14		12	11	10	9	8	7	6

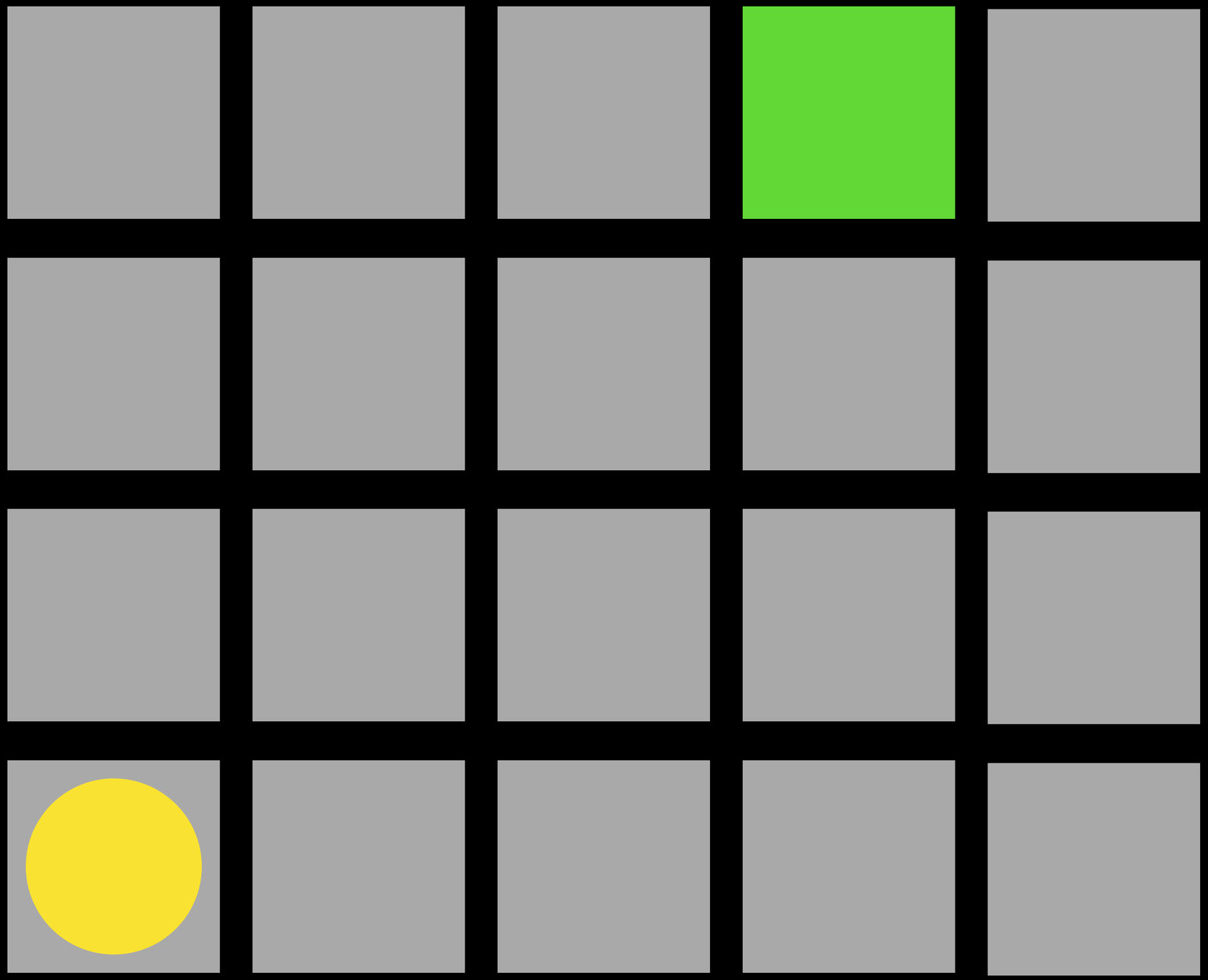
Greedy Best-First Search

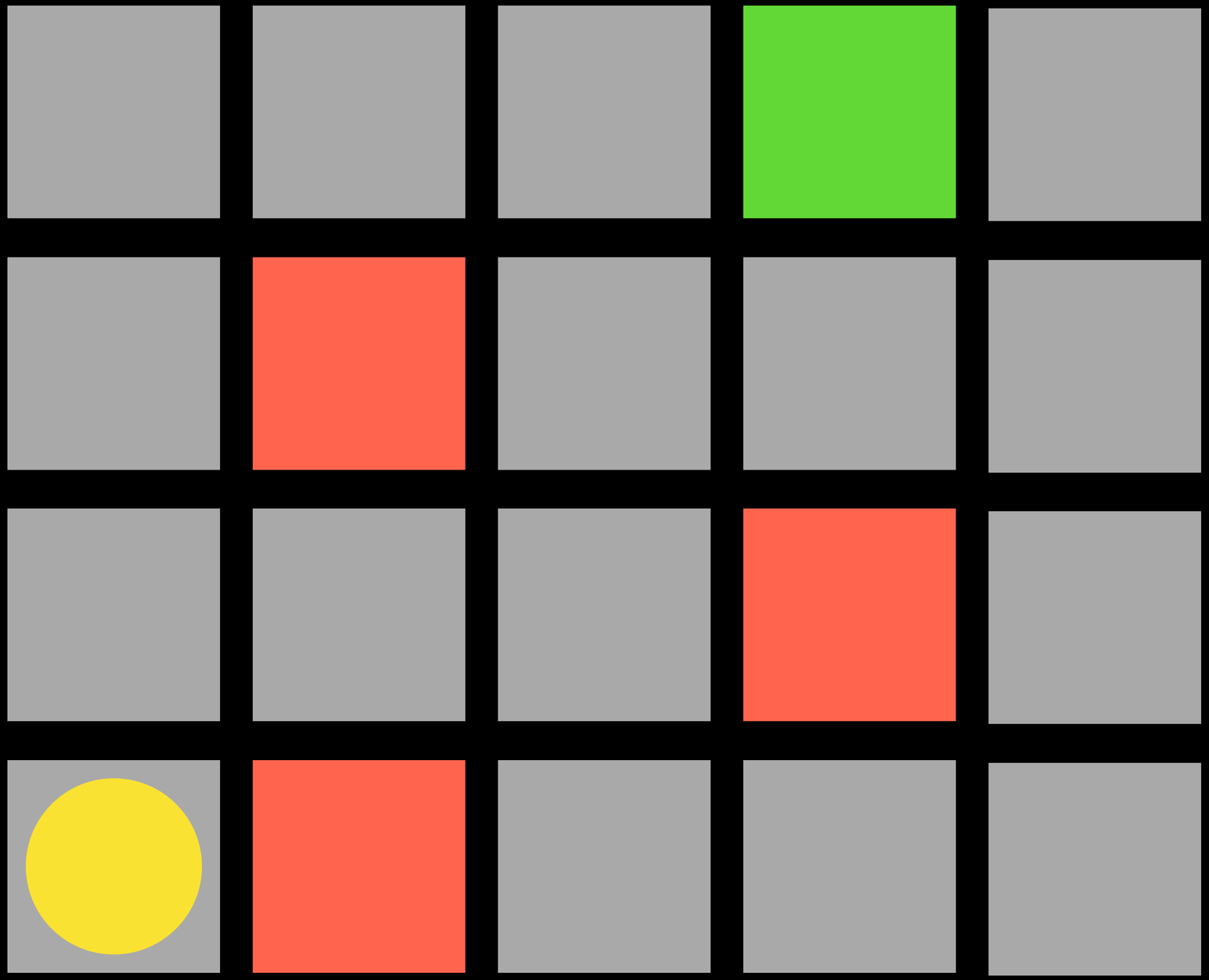
	10	9	8	7	6	5	4	3	2	1	**B**
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	16	15	14		12	11	10	9	8	7	6

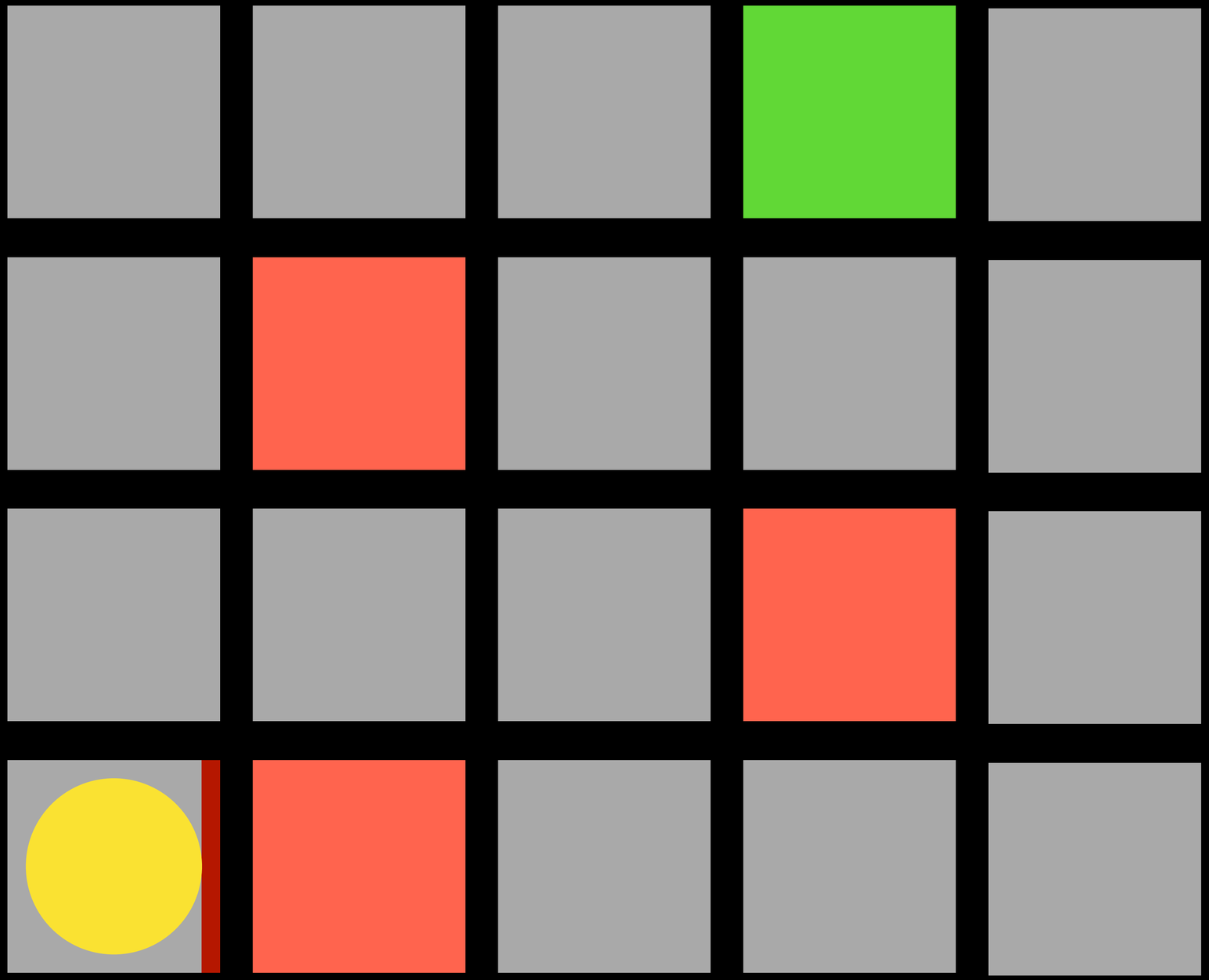
A* search

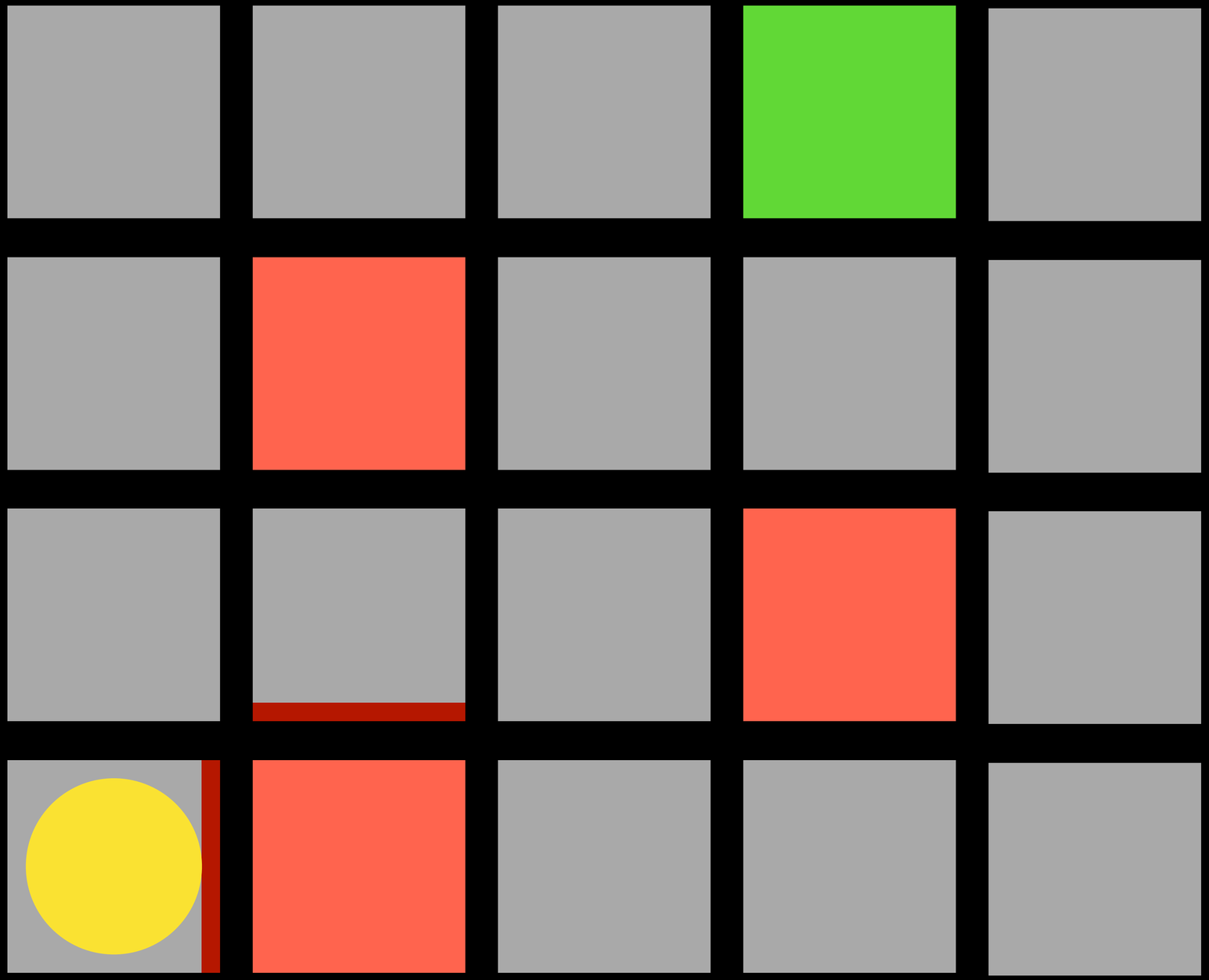
search algorithm that chooses what to explore based both on cost so far and a heuristic

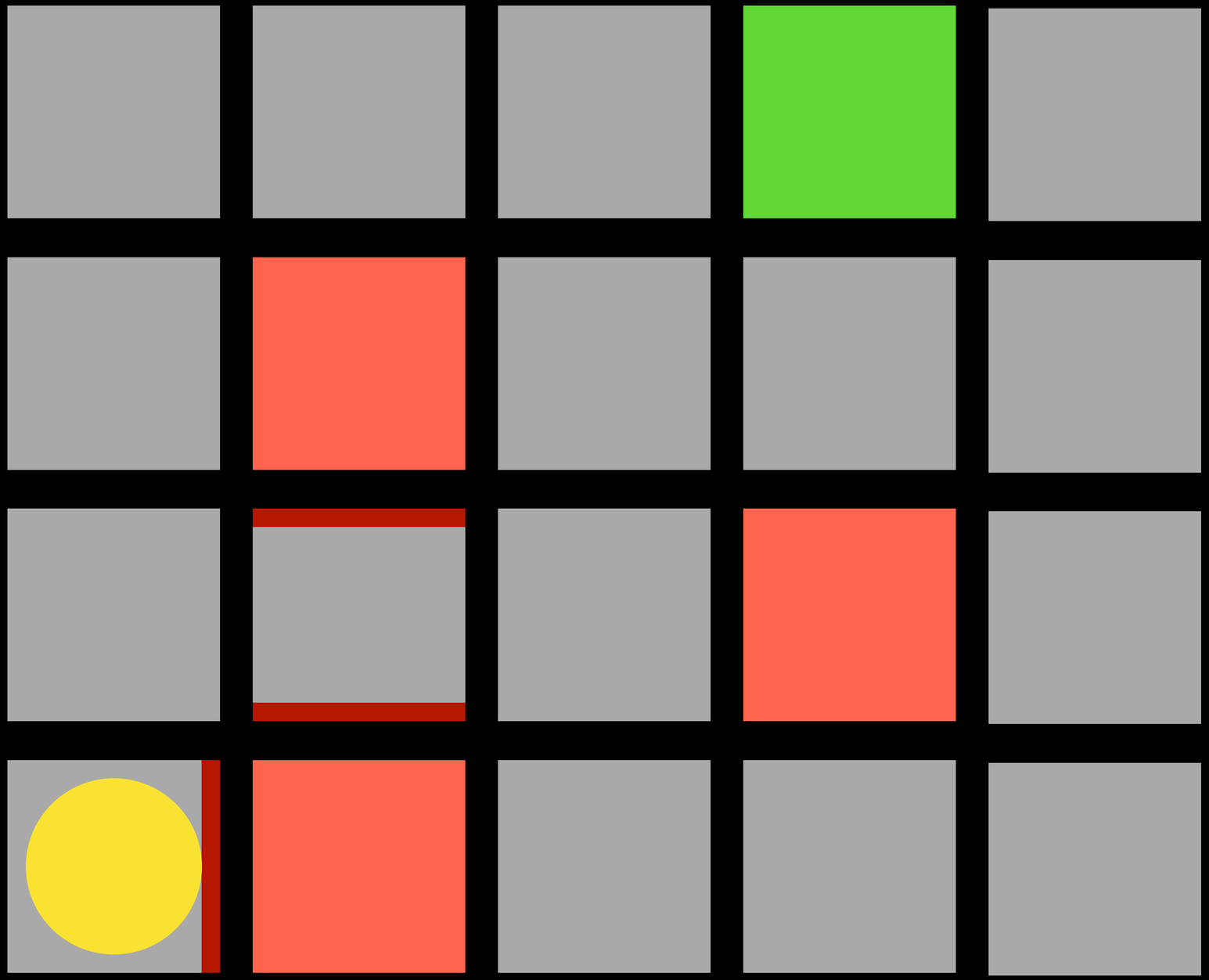
Reinforcement Learning

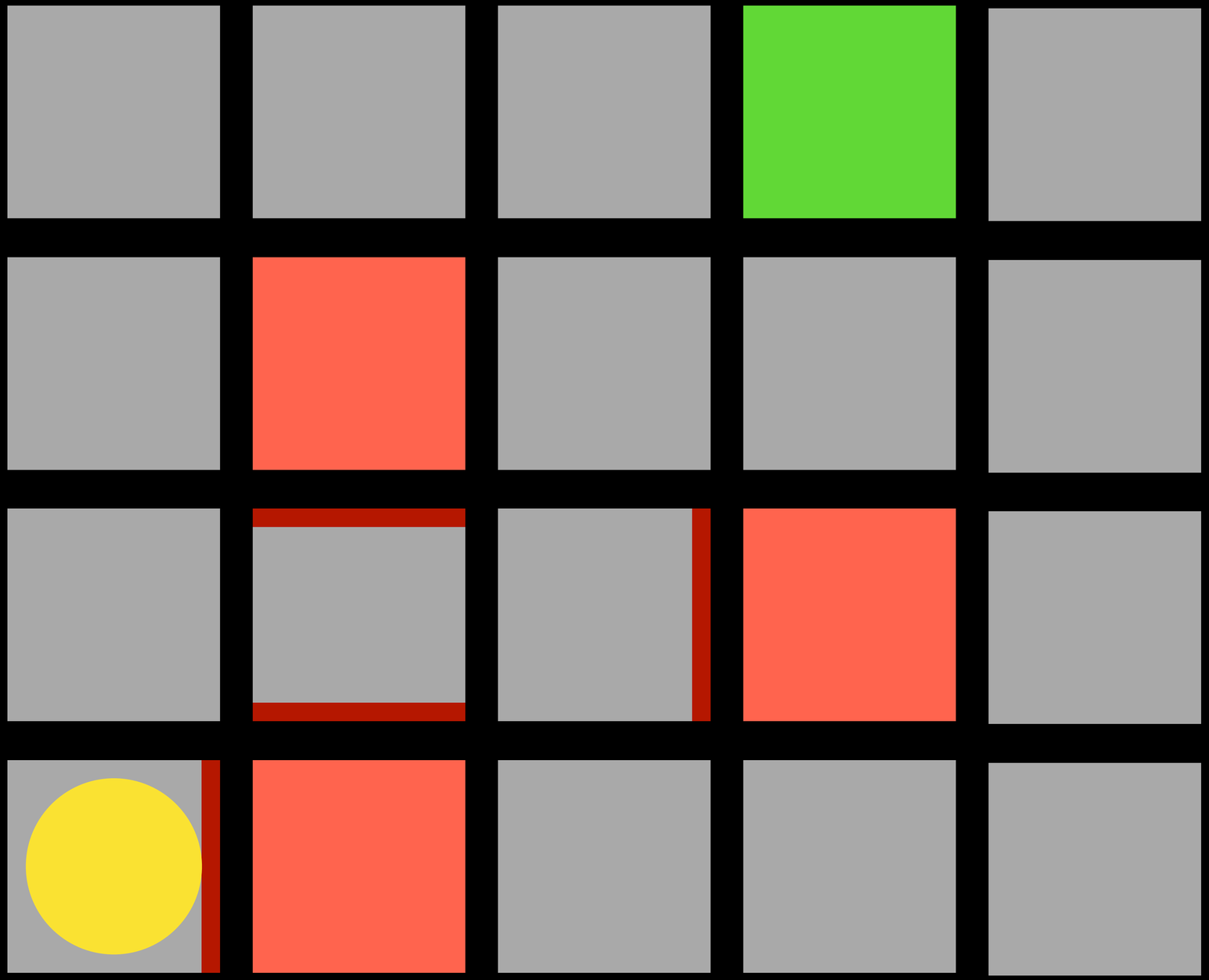


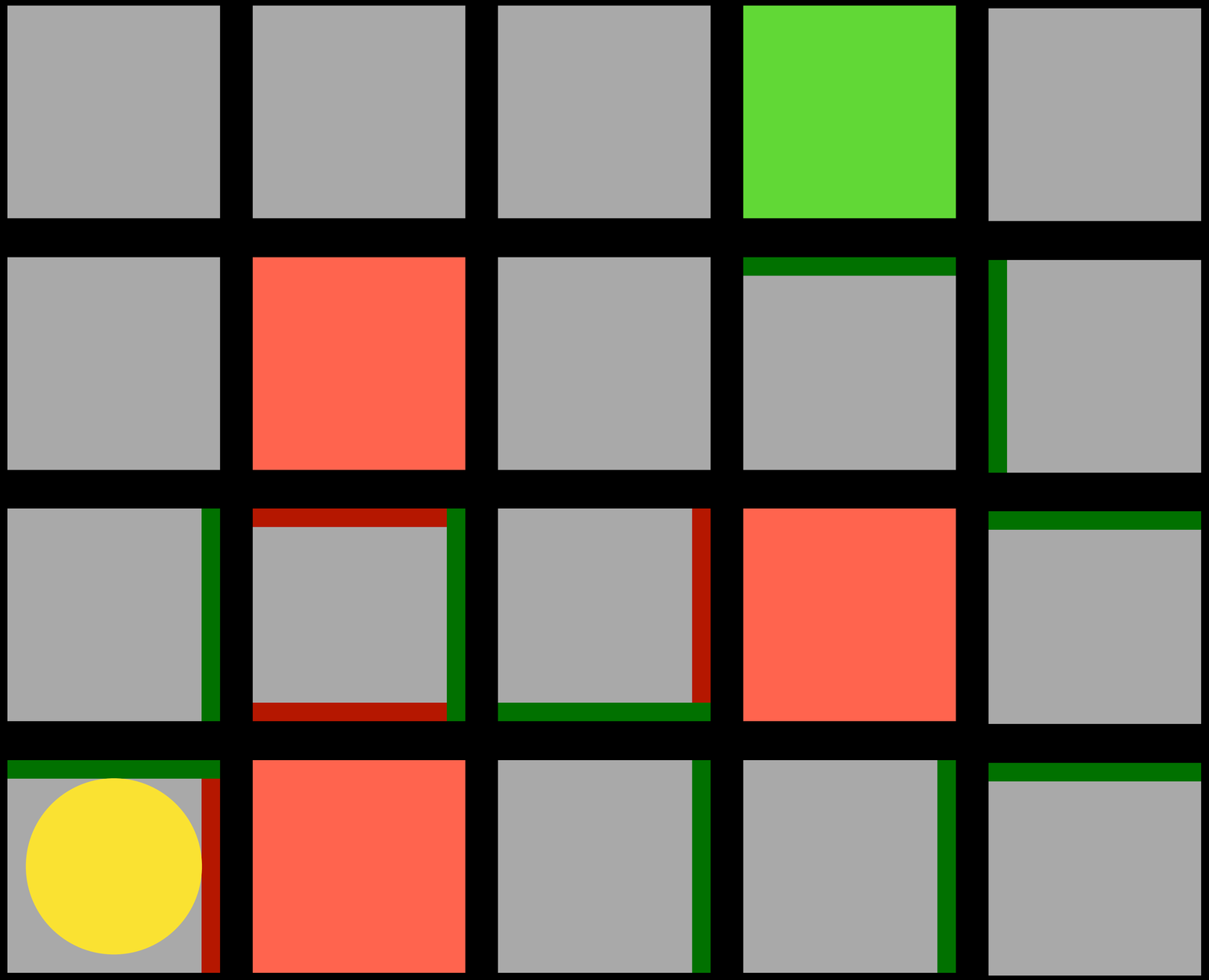








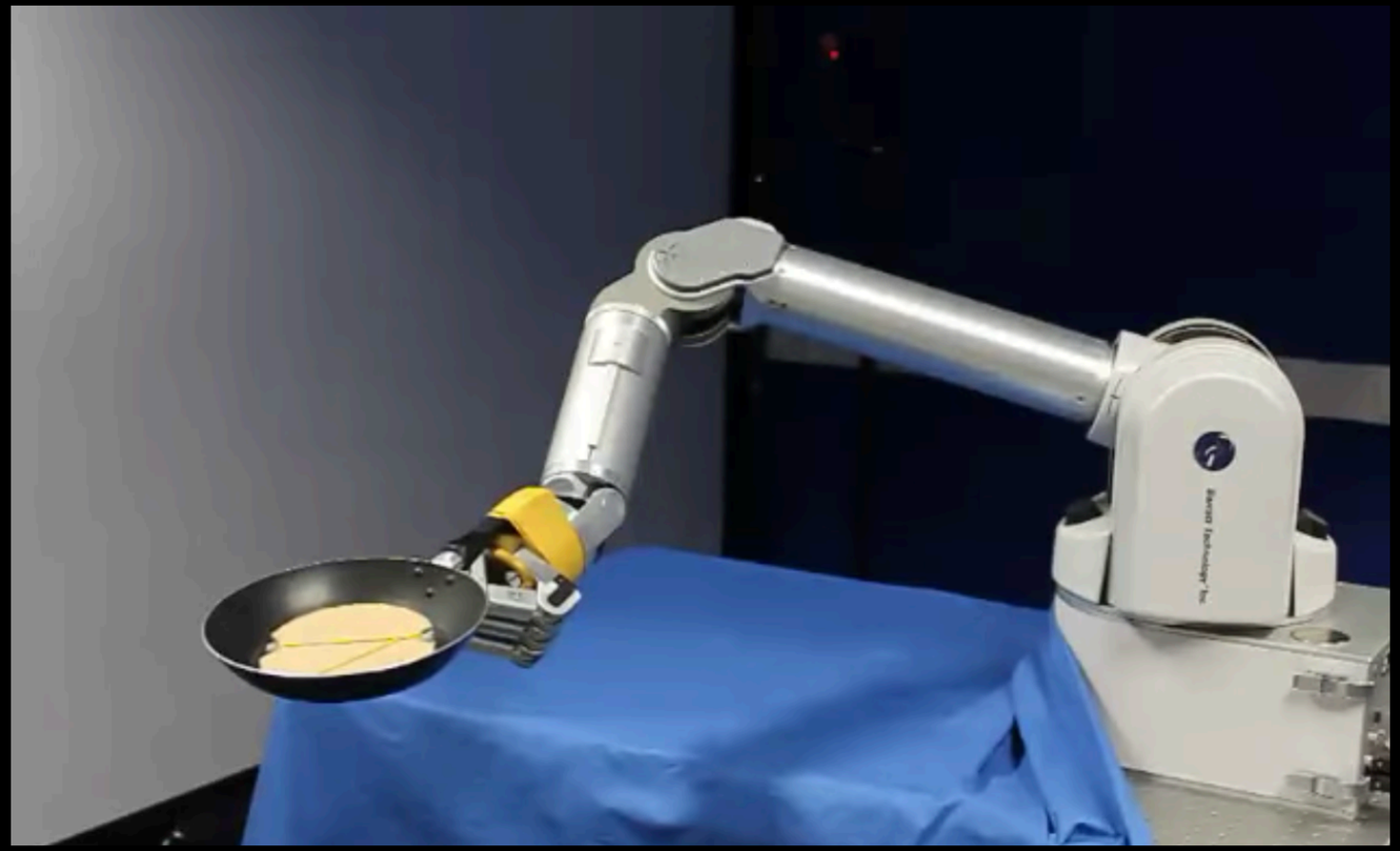




Explore vs. Exploit

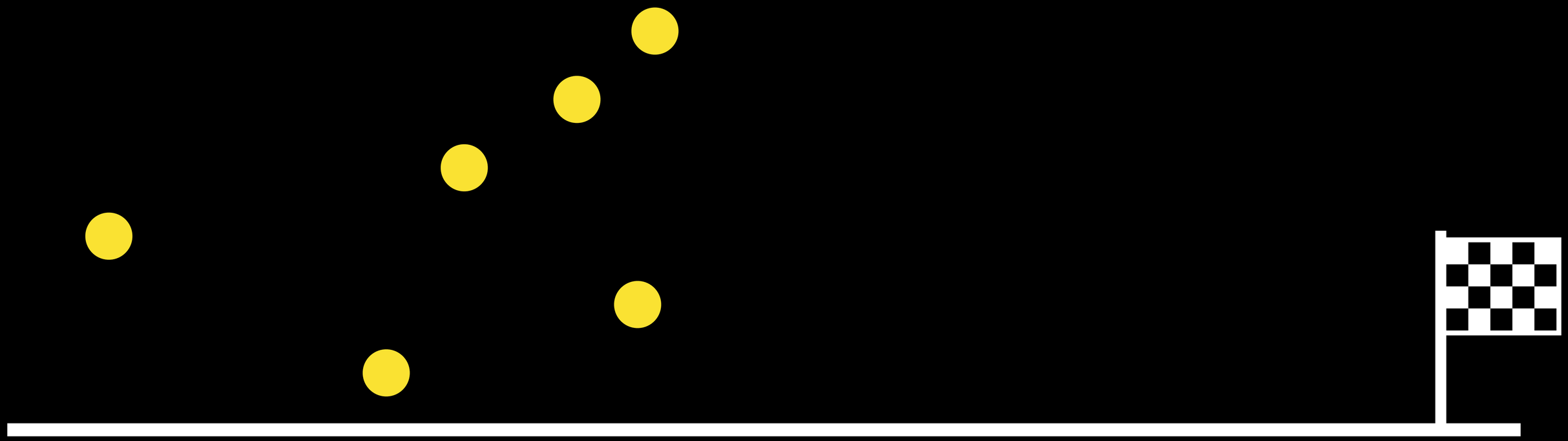
Explore vs. Exploit Strategy

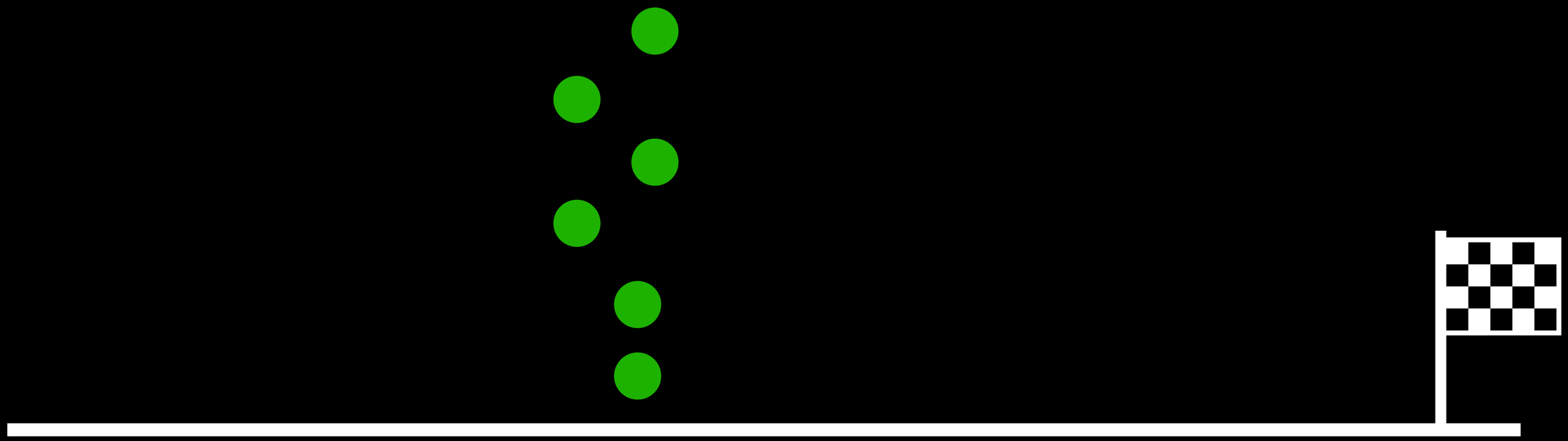
```
epsilon = 0.10

if random() < epsilon:
    make a random move
else:
    make the move with the highest value
```



https://www.youtube.com/watch?v=W_gxLKSsSIE

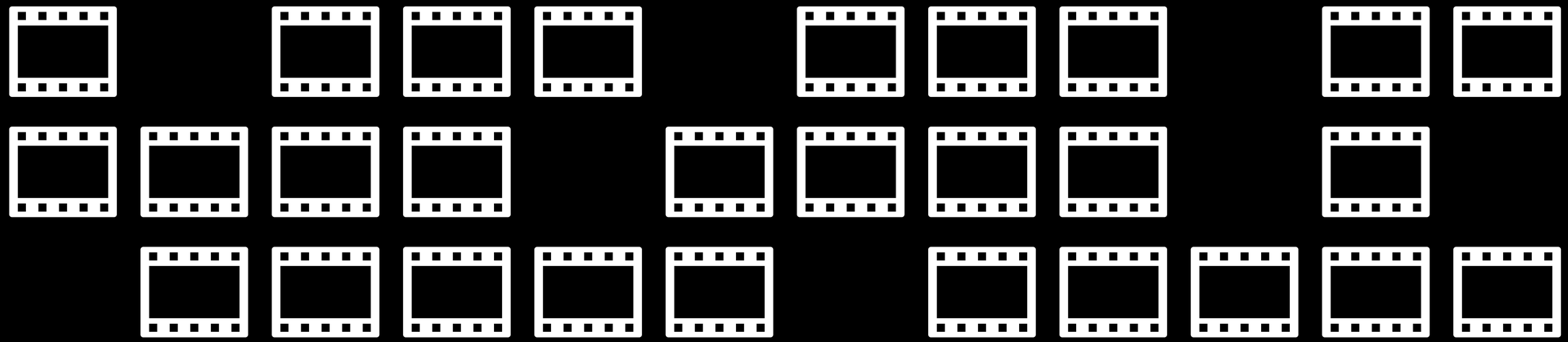
Genetic Algorithms

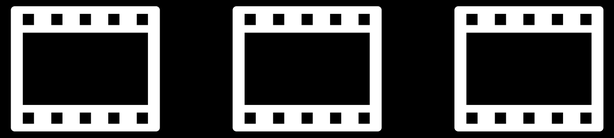




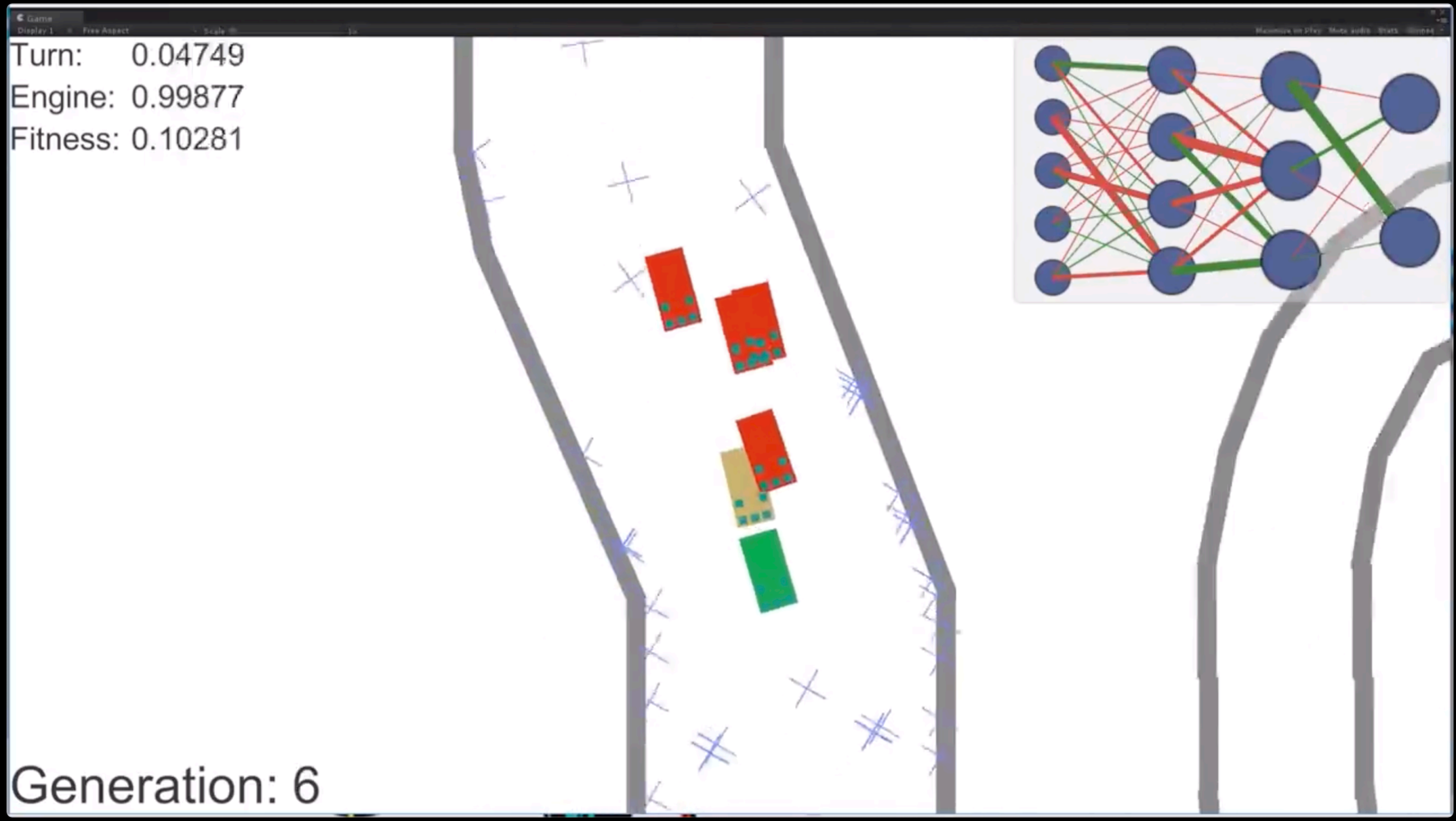
Genetic Algorithm

```
make initial generation of candidates randomly
repeat until successful:
    for each candidate:
        calculate candidate's fitness
    remove least fit candidates
    make new generation from remaining candidates
```



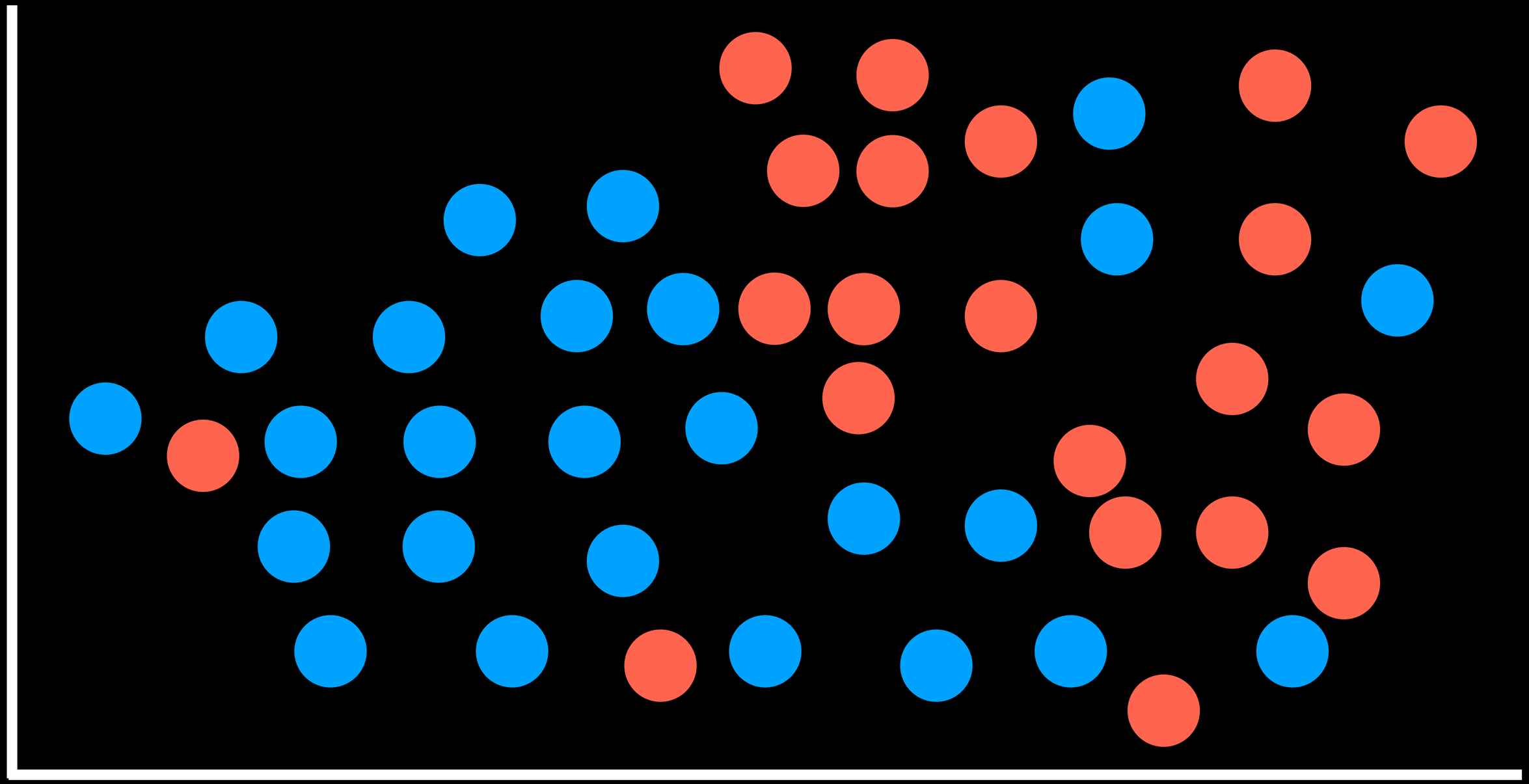
Watch History 

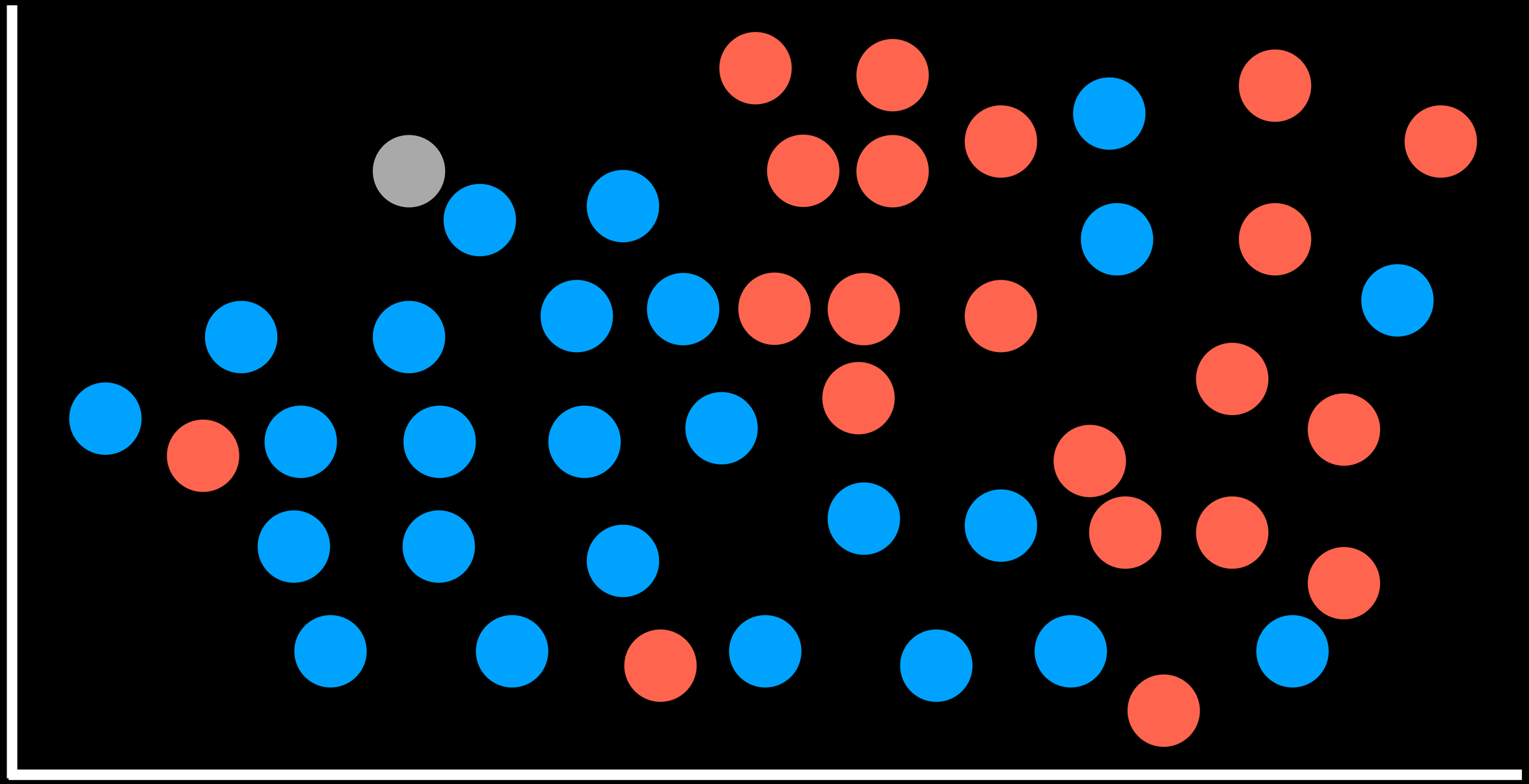
Recommended 

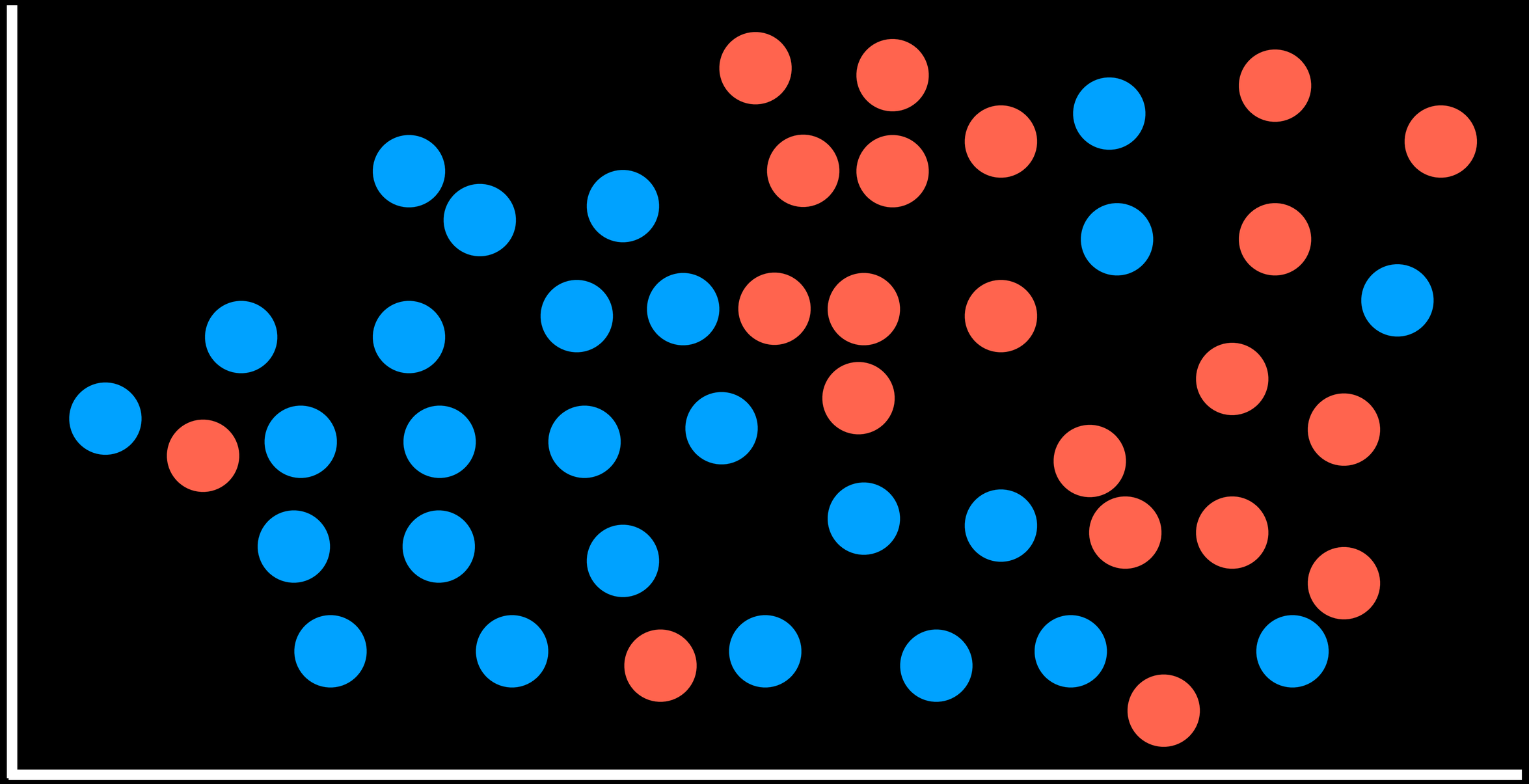

Turn: 0.04749
Engine: 0.99877
Fitness: 0.10281
Generation: 6

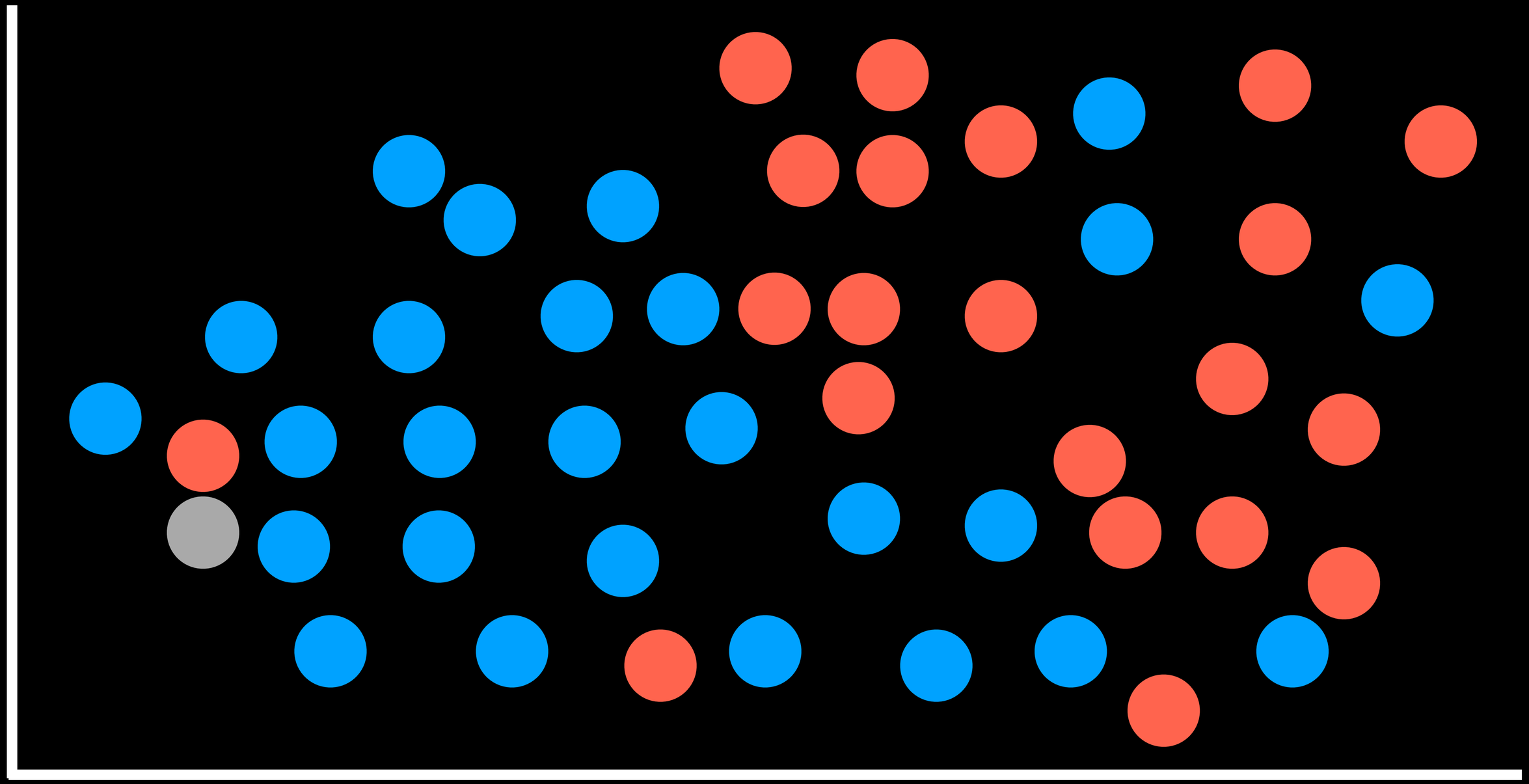
Classification

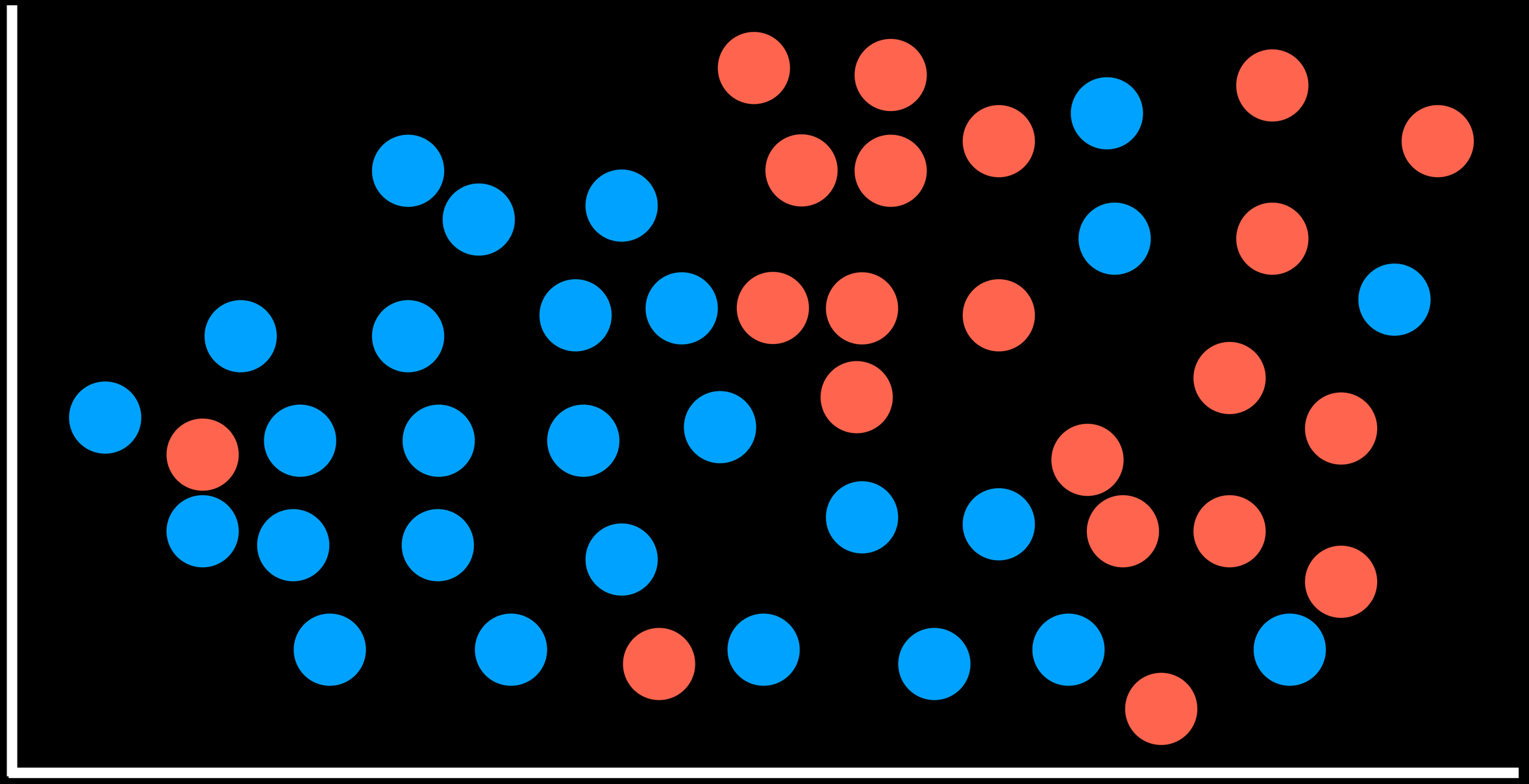
Nearest-Neighbor Classification



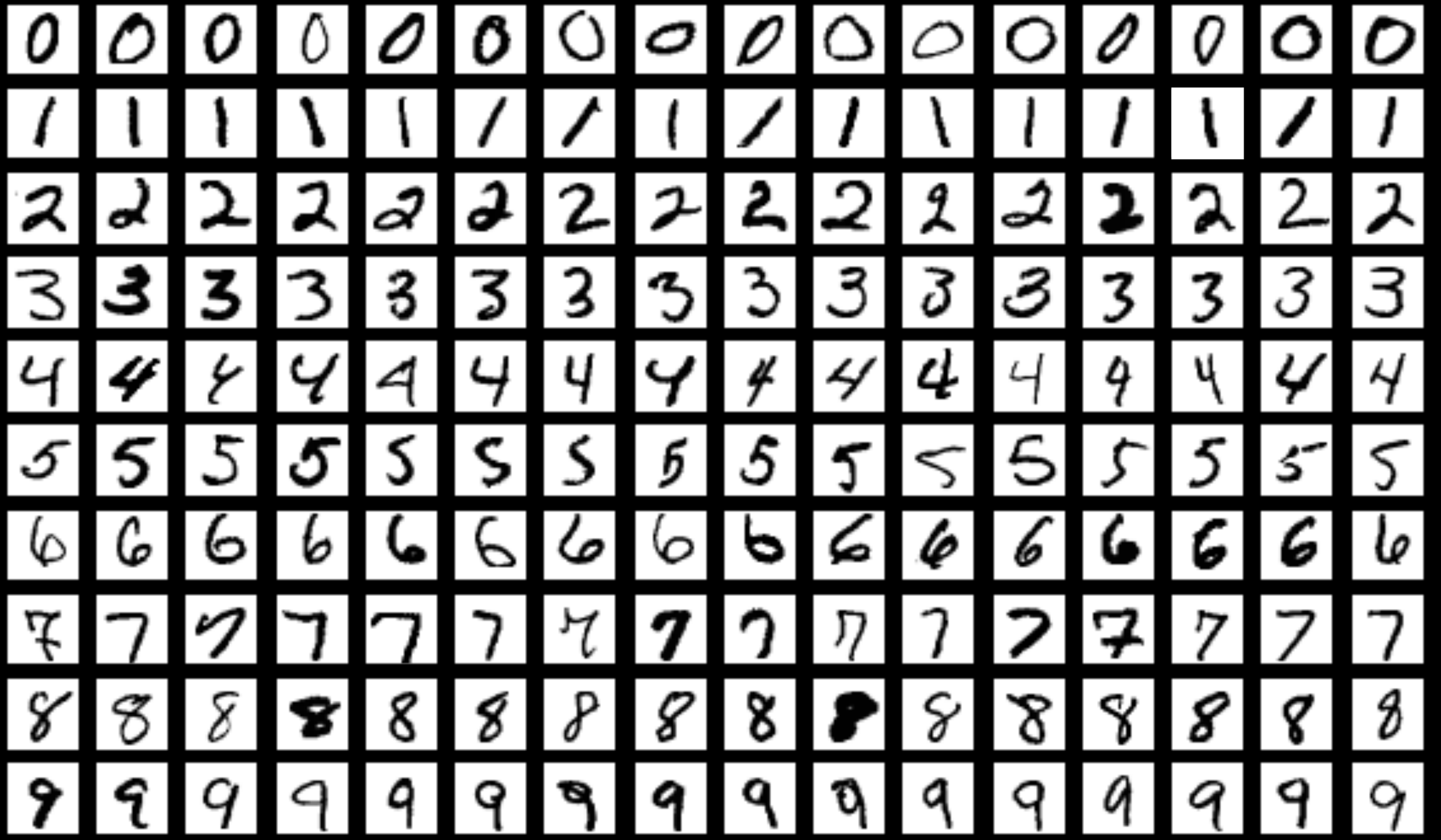




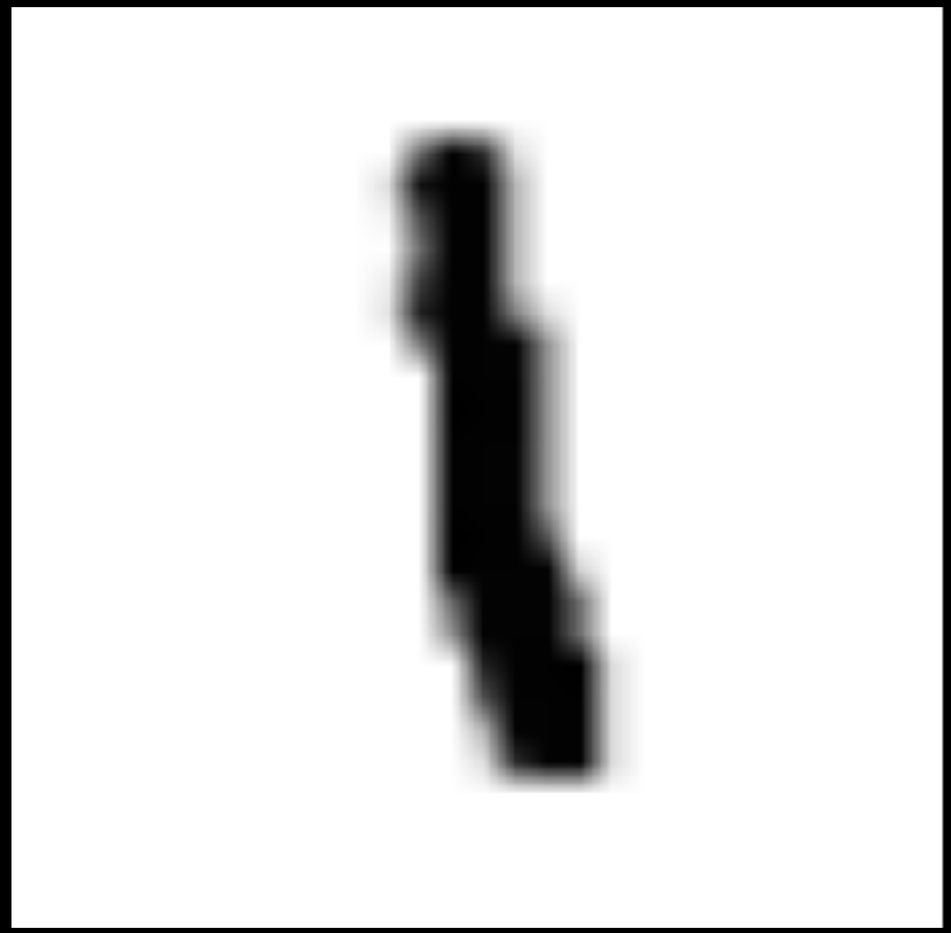




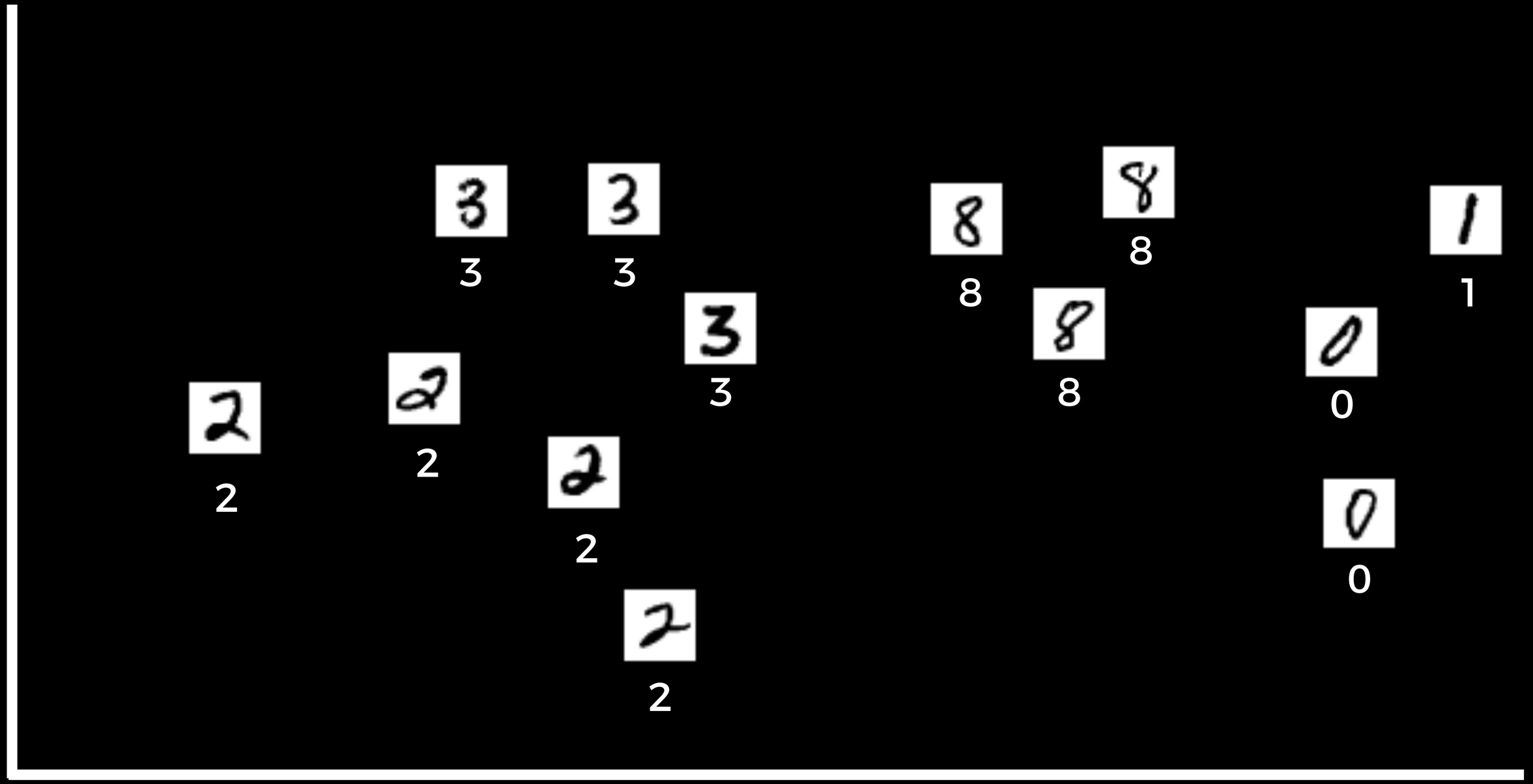
k-Nearest-Neighbor Classification

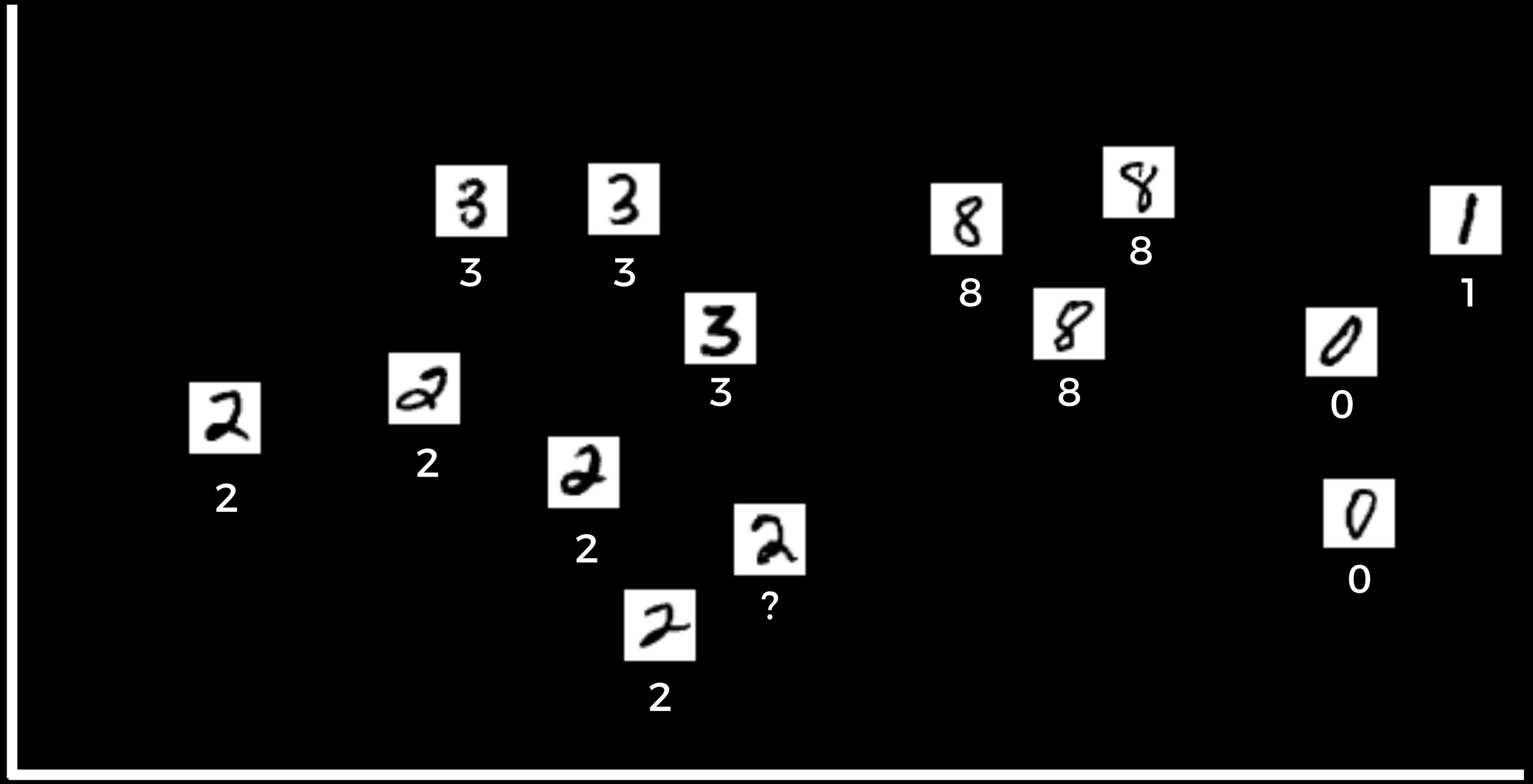


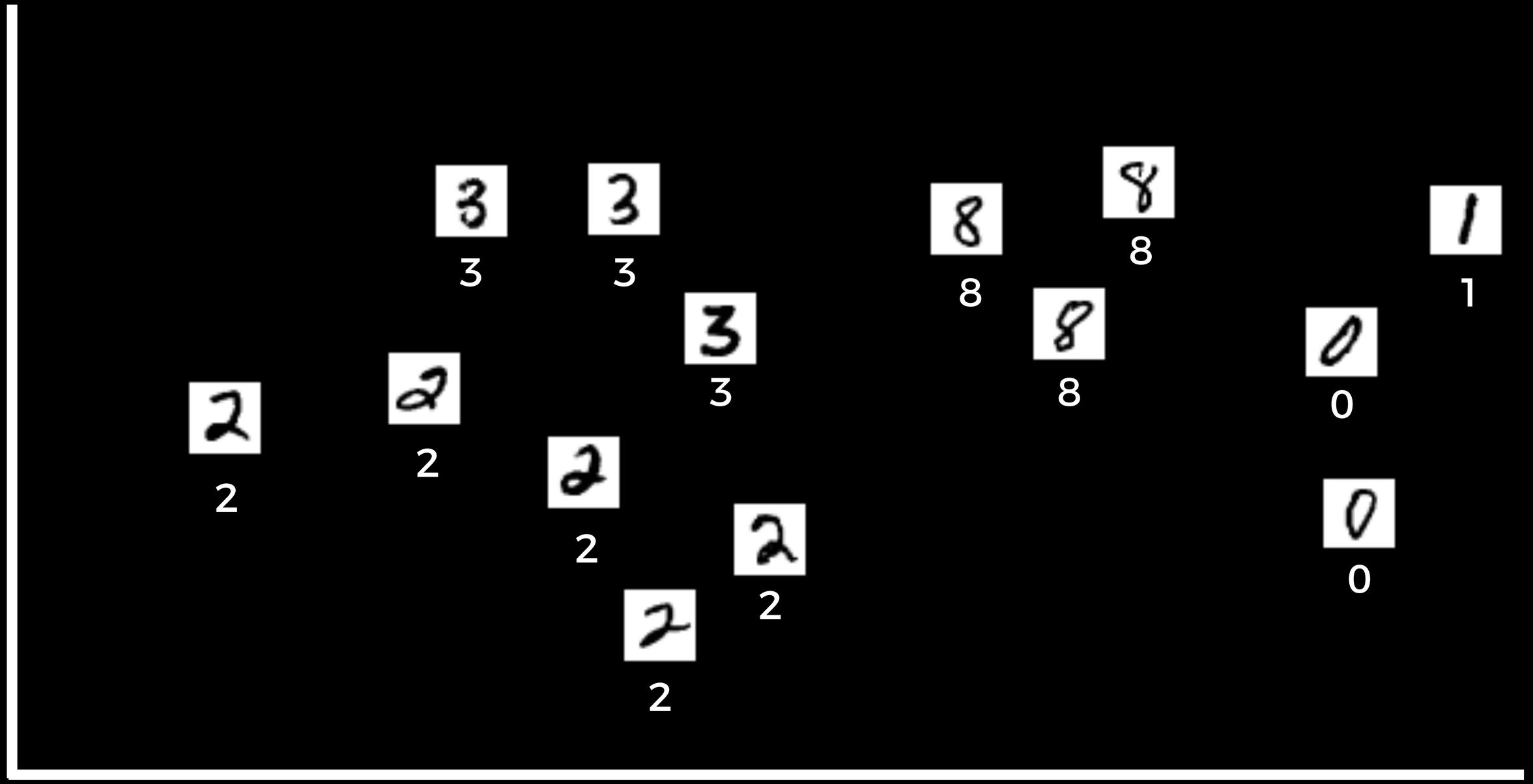
0	0	0	0	0	0	0	0	0
0	0	0	0.6	0.8	0	0	0	0
0	0	0	0.8	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0	0.8	0.6	0	0	0
0	0	0	0	0	0	0	0	0

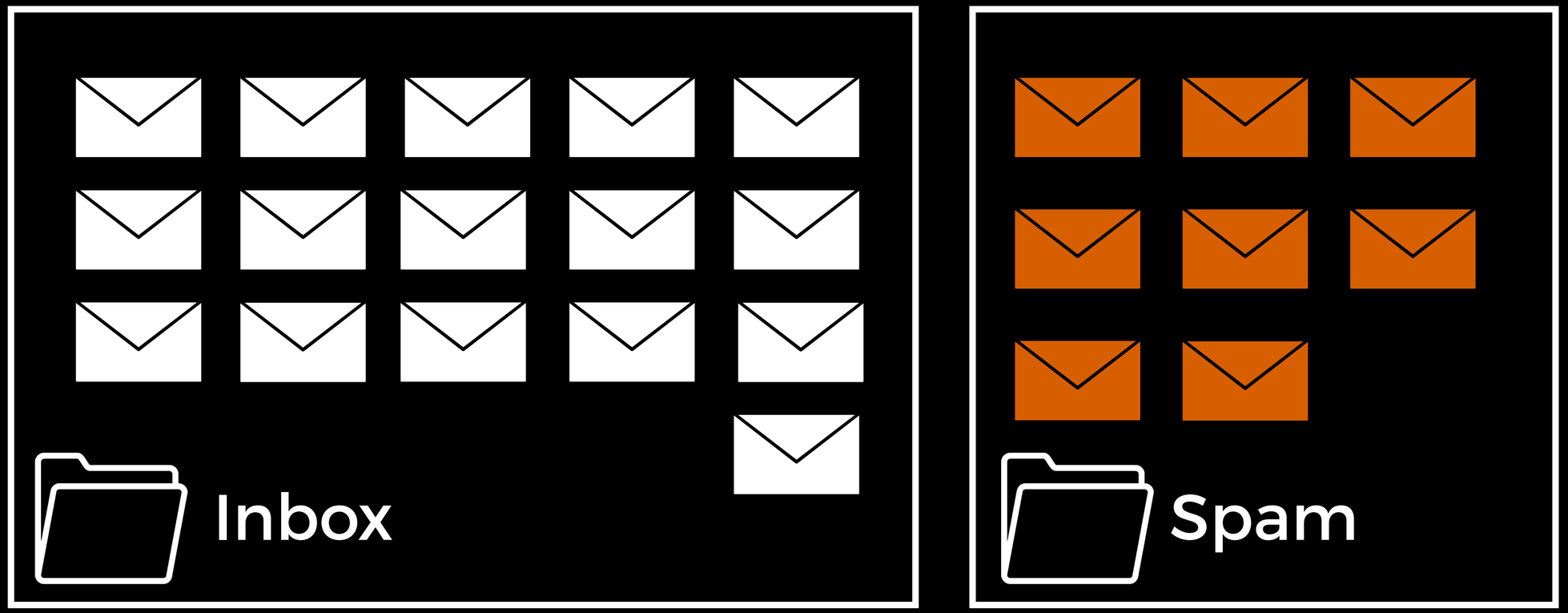


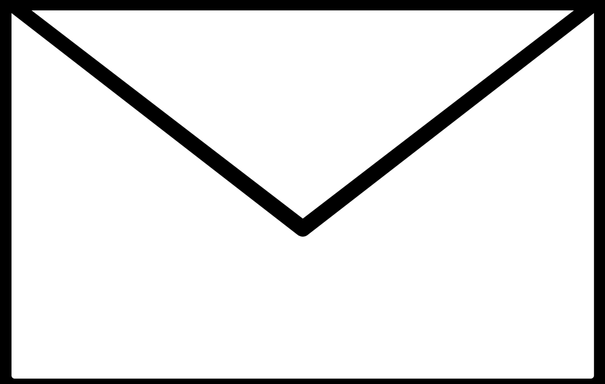
0	0	0	0	0	0	0	0	0
0	0	0	0.6	0.8	0	0	0	0
0	0	0	0.8	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0.9	0.9	0.9	0	0	0
0	0	0	0	0.8	0.6	0	0	0
0	0	0	0	0	0	0	0	0


3
3
3
3
8
8
8
8
1
1
2
2
2
2
2
2
0
0
0
0

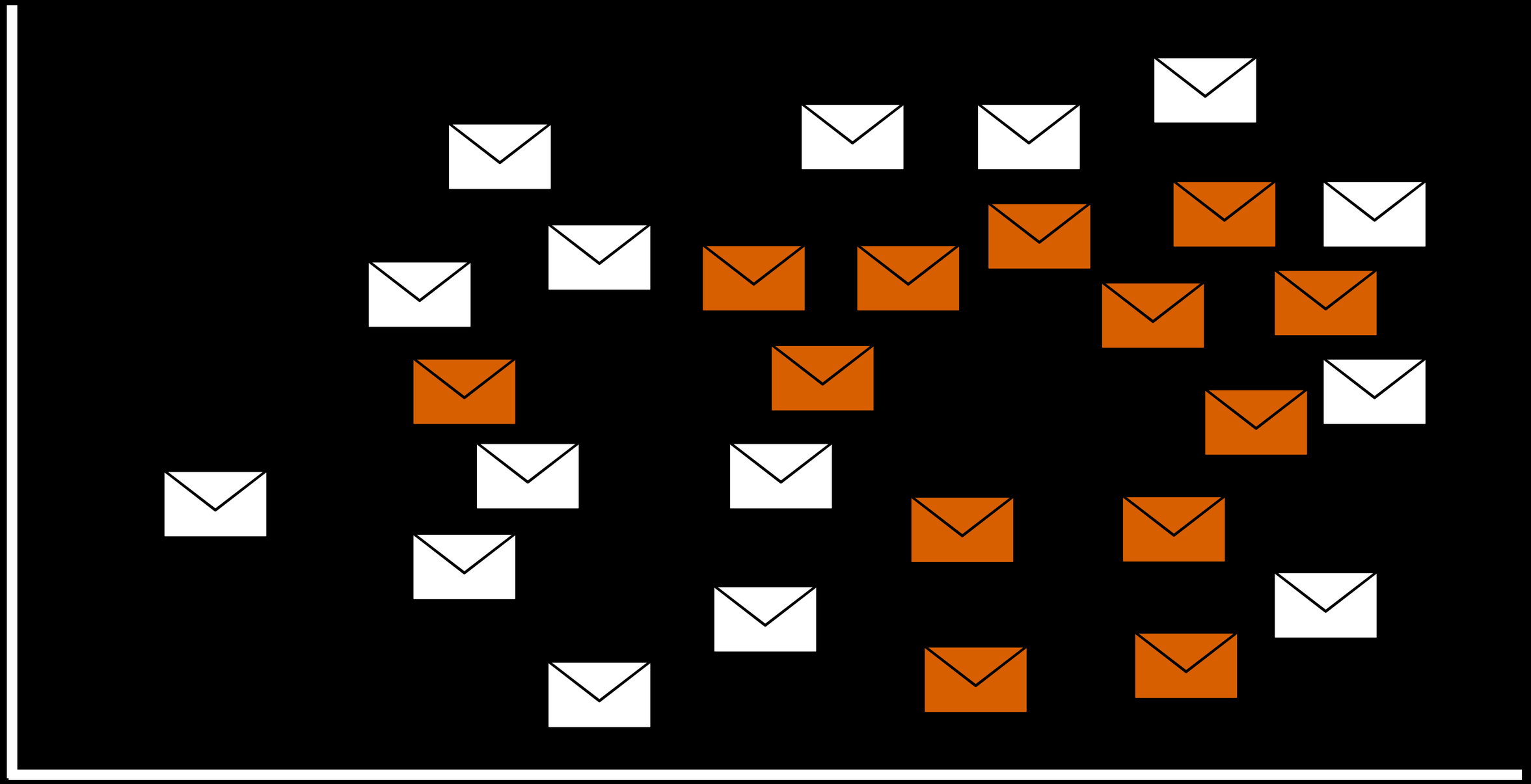

3
3
3
3
8
8
8
8
1
1
2
2
2
2
2
2
2
?
0
0
0
0

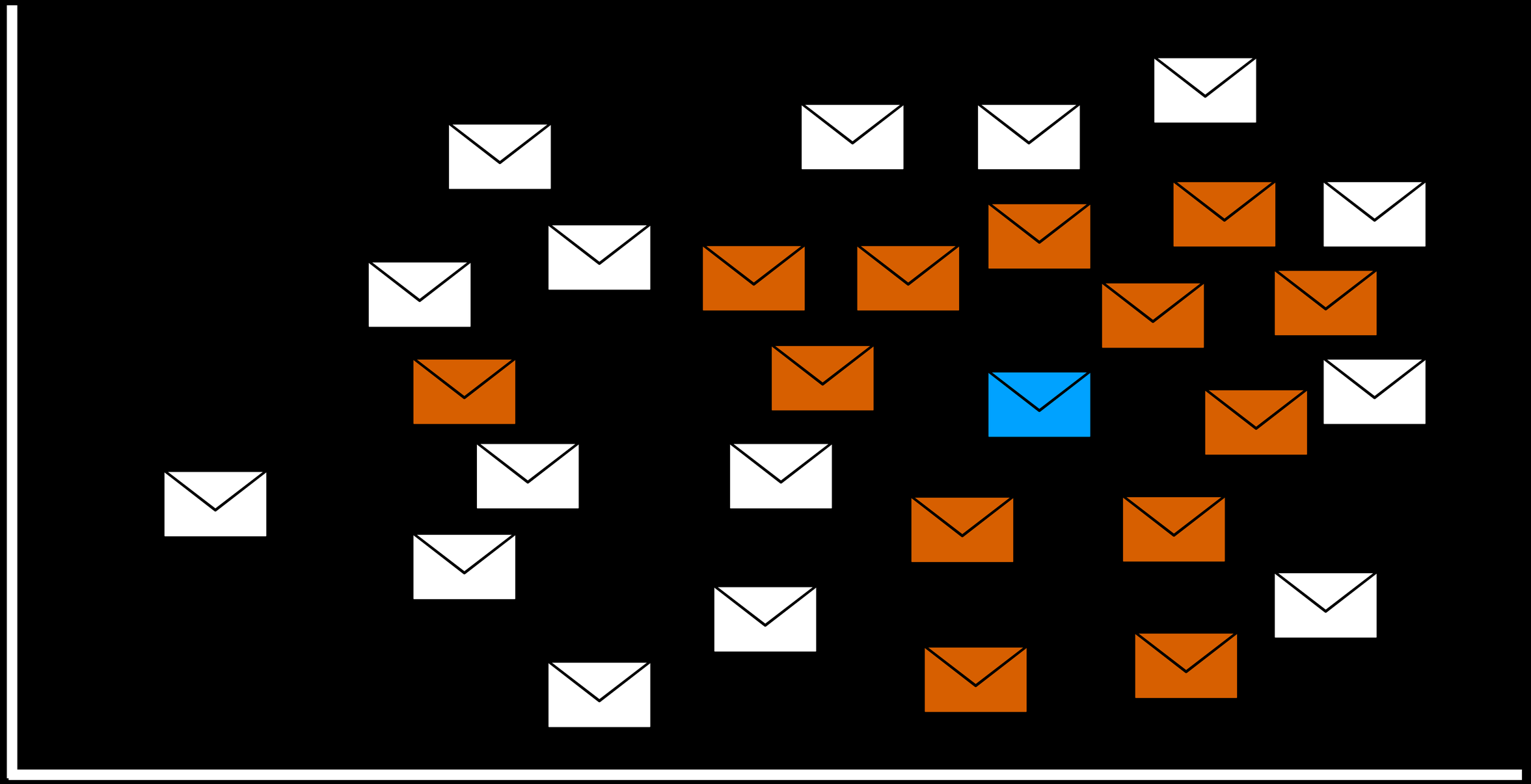

3
3
3
3
8
8
8
8
1
1
2
2
2
2
2
2
2
2
0
0
0
0

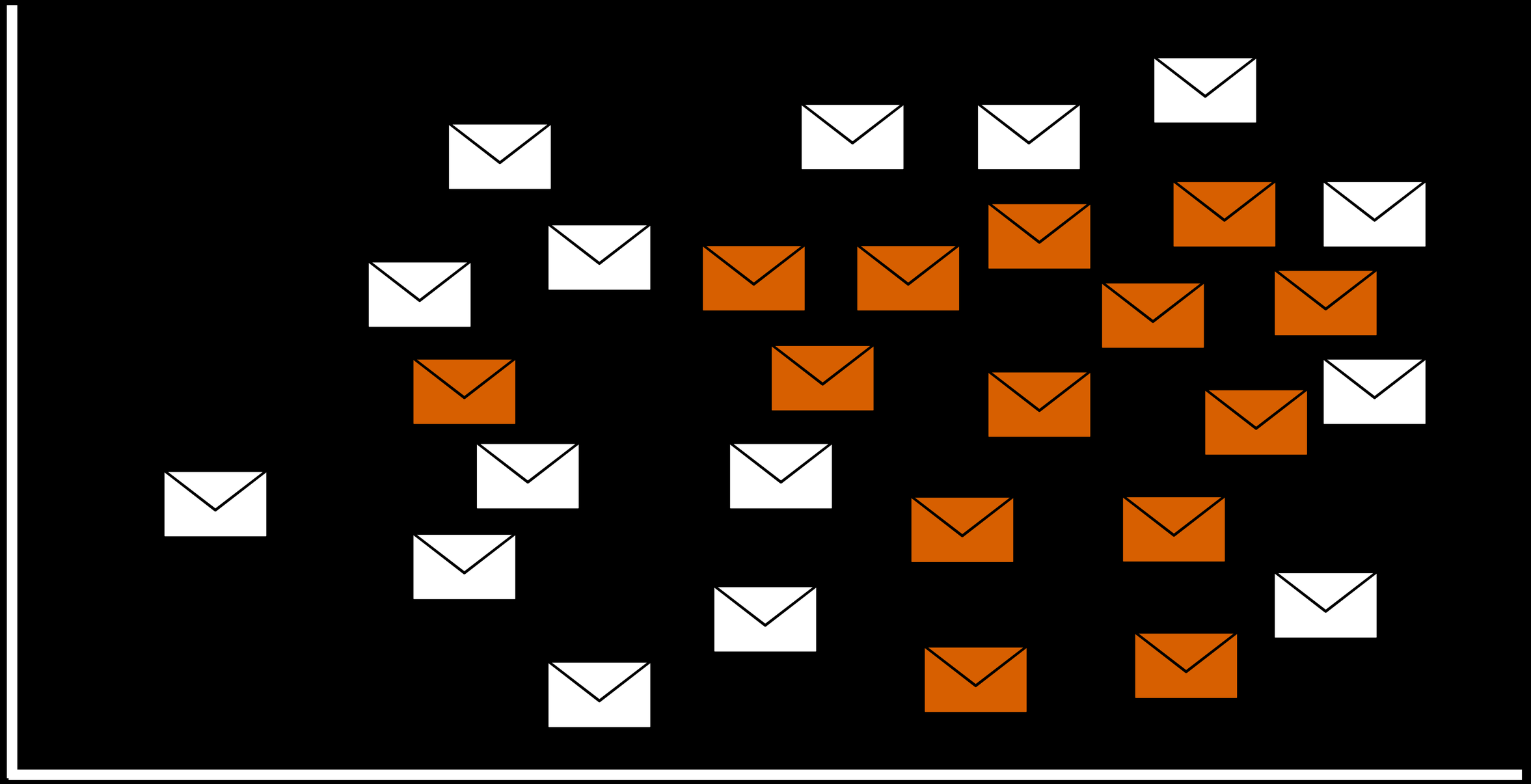

Inbox
Spam



[1, 2, 5, 2, 3, 1, 2, 8, 1, 3]







Artificial Intelligence