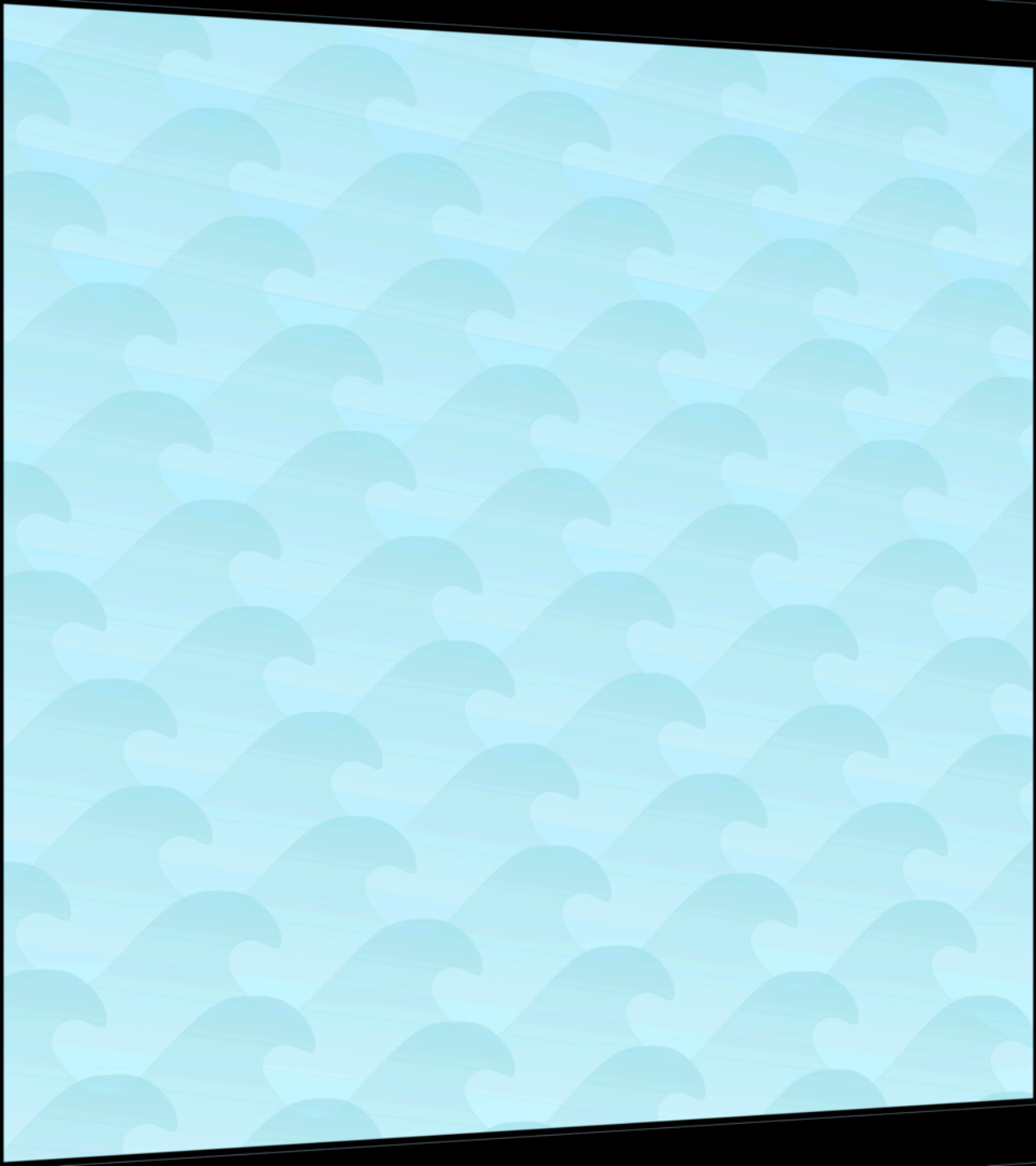


Geometry,
Uncertainty,
&
BATTLESHIPS!!!

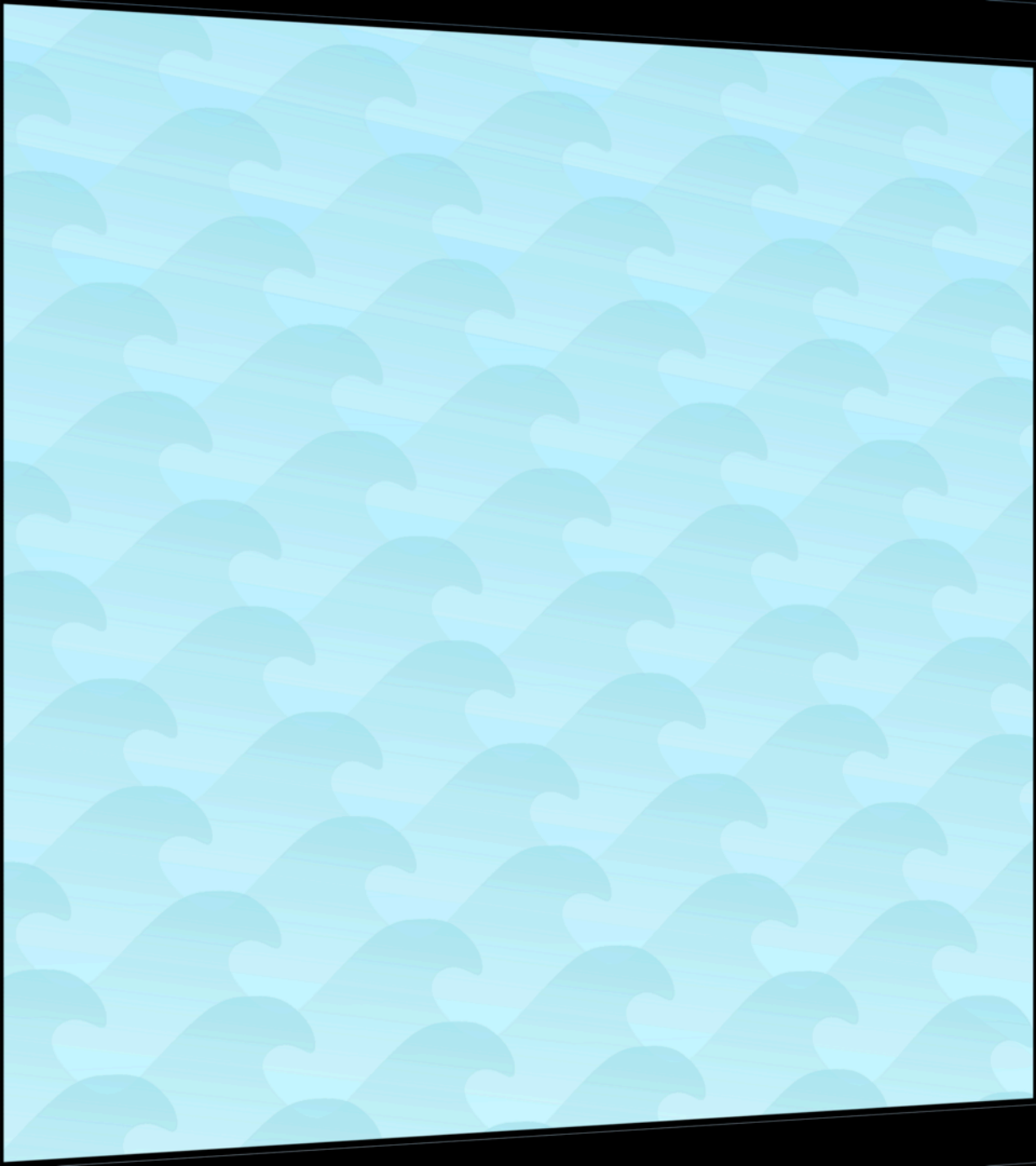
Eva Hainzl
Maarten Löffler
Daniel Perz
Josef Tkadlec
Markus Wallinger



CHAPTER

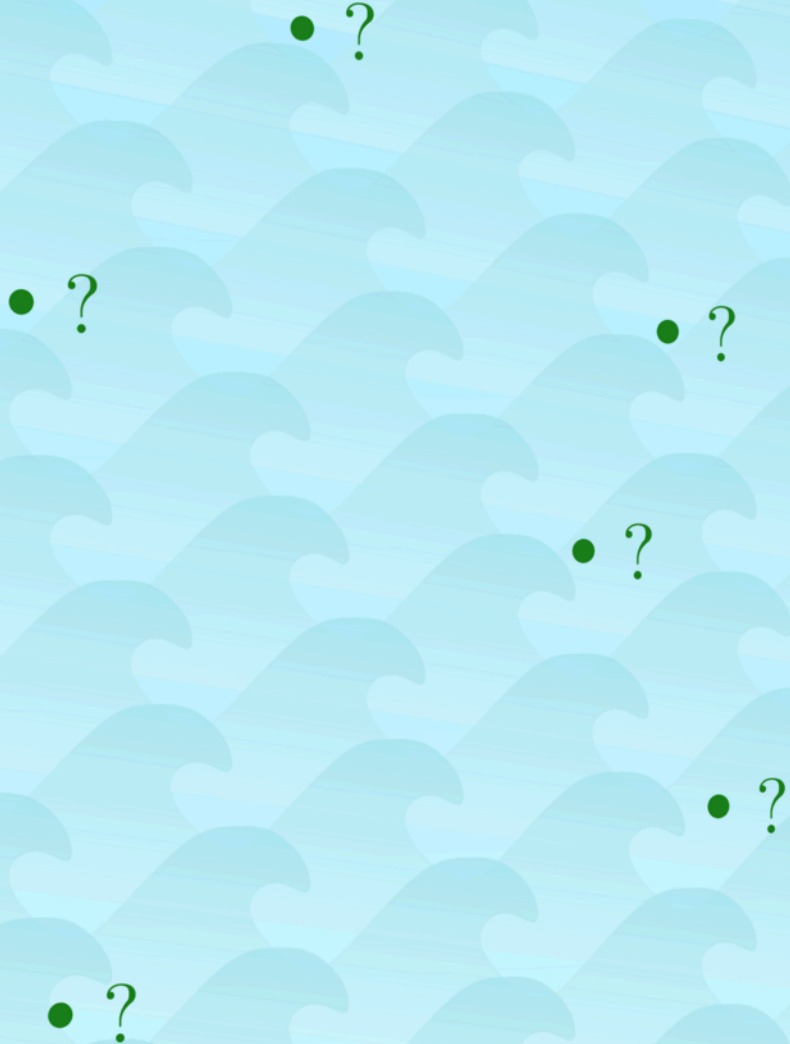
.....
UNCERTAINTY

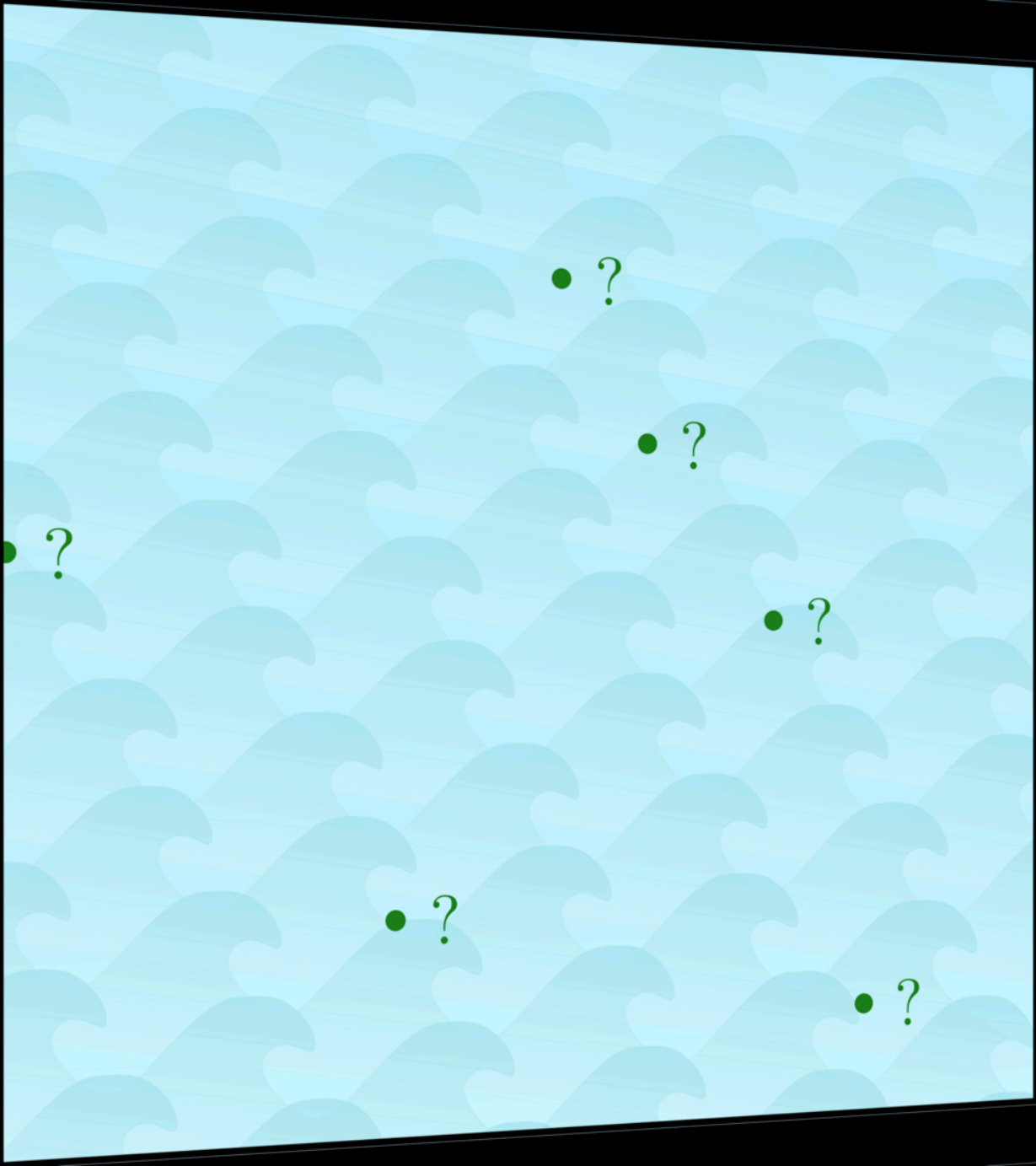
1



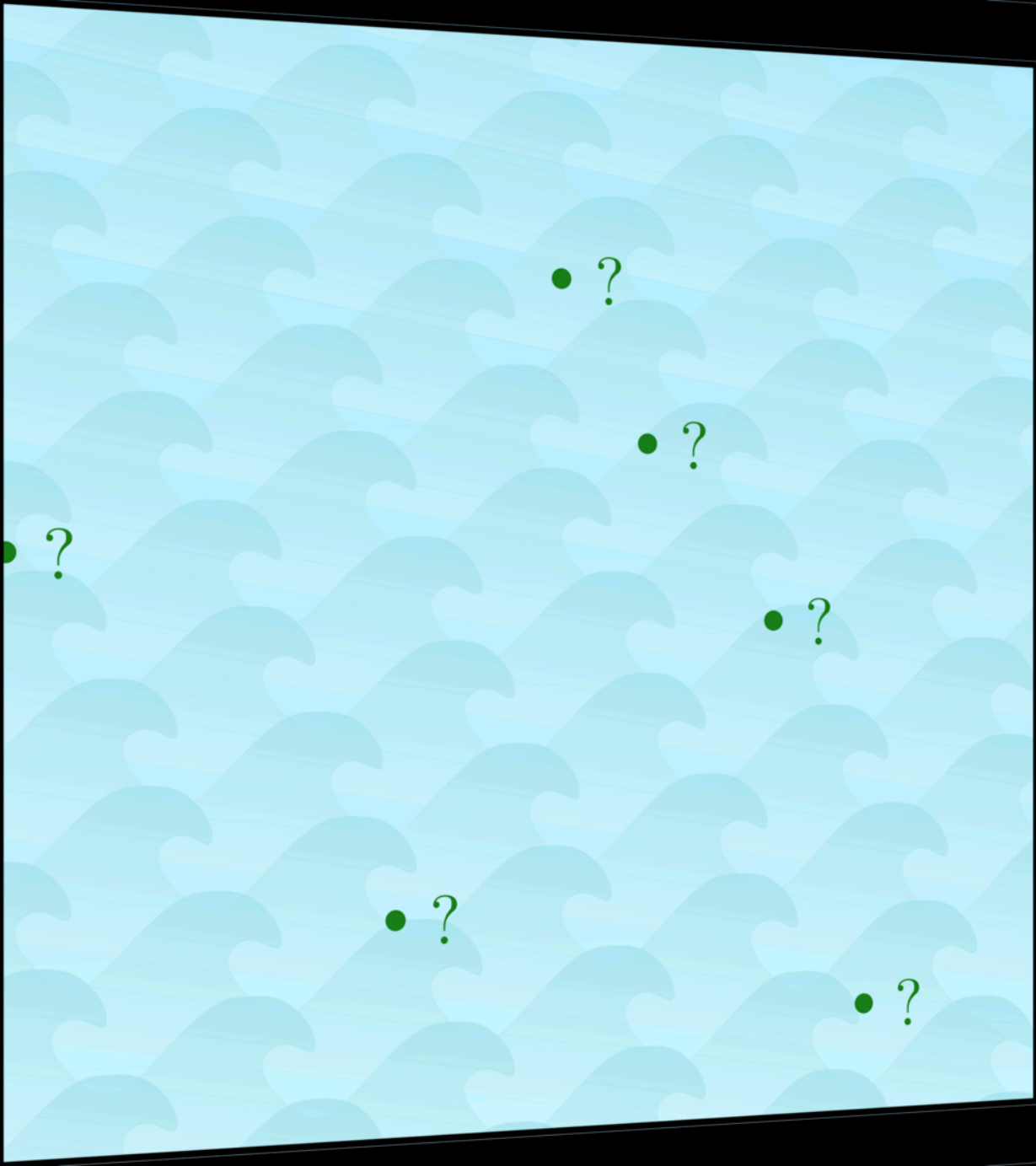
Suppose we have a set of points, but we don't know where they are.

Suppose we have a set of points, but we don't know where they are.





Suppose we have a set of points, but we don't know where they are.



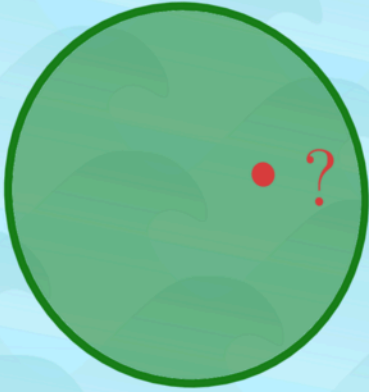
Suppose we have a set of points, but we don't know where they are.

Half an hour ago, I asked:
“*Where* are you”?



Suppose we have a set of points, but we don't know where they are.

Half an hour ago, I asked:
“*Where* are you”?



Suppose we have a set of points, but we don't know where they are.

Half an hour ago, I asked:
“*Where* are you”?



Suppose we have a set of points, but we don't know where they are.

Half an hour ago, I asked:

"Where are you"?

ID → DISK



Suppose we have a set of points, but we don't know where they are.

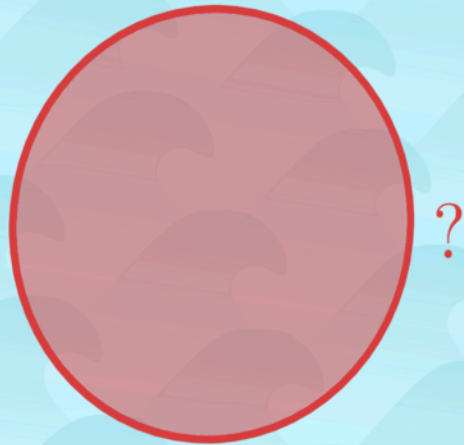
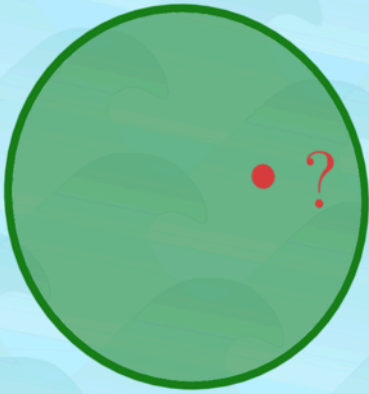
Half an hour ago, I asked:

“*Where* are you”?

ID → DISK

Maybe we can also ask:

“There are *who*”?



Suppose we have a set of points, but we don't know where they are.

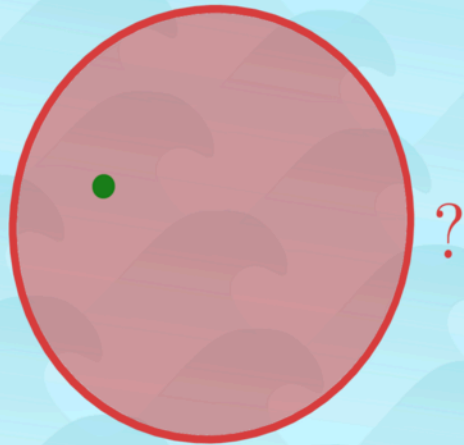
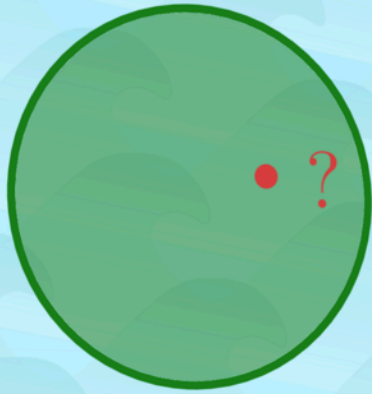
Half an hour ago, I asked:

“*Where* are you”?

ID → DISK

Maybe we can also ask:

“There are *who*”?



Suppose we have a set of points, but we don't know where they are.

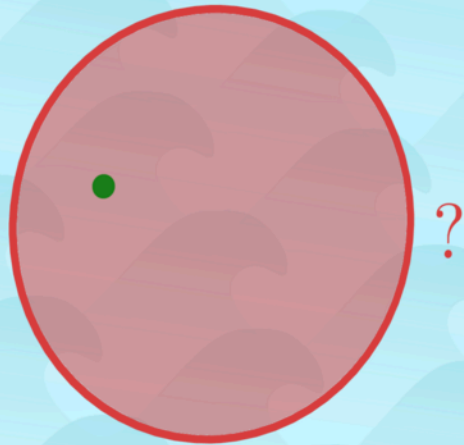
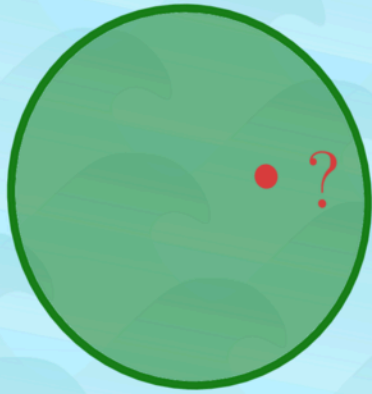
Half an hour ago, I asked:

“*Where* are you”?

ID → DISK

Maybe we can also ask:

“There are *who*”?



Suppose we have a set of points, but we don't know where they are.

Half an hour ago, I asked:

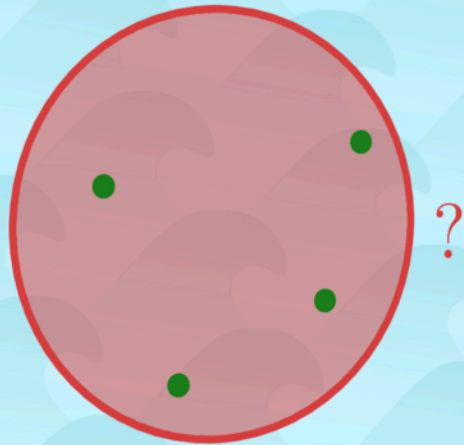
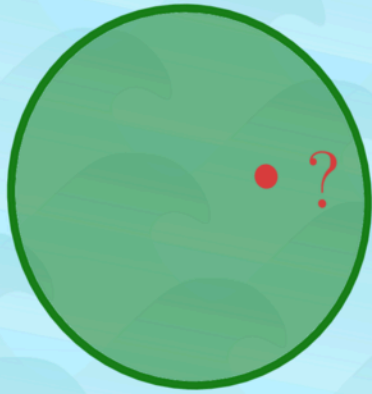
“Where are you”?

ID → DISK

Maybe we can also ask:

“There are *who*”?

DISK → [ID]



Suppose we have a set of points, but we don't know where they are.

Half an hour ago, I asked:

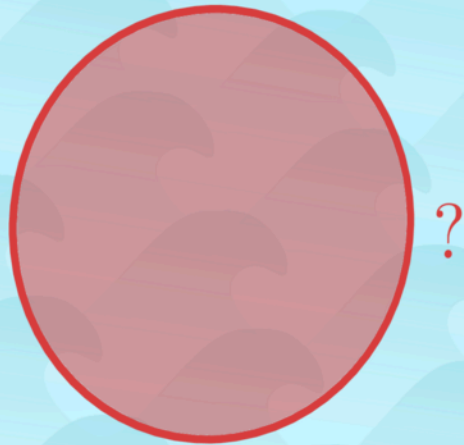
“Where are you”?

ID → DISK

Maybe we can also ask:

“There are *who*”?

DISK → [ID]



Suppose we have a set of points, but we don't know where they are.

Half an hour ago, I asked:

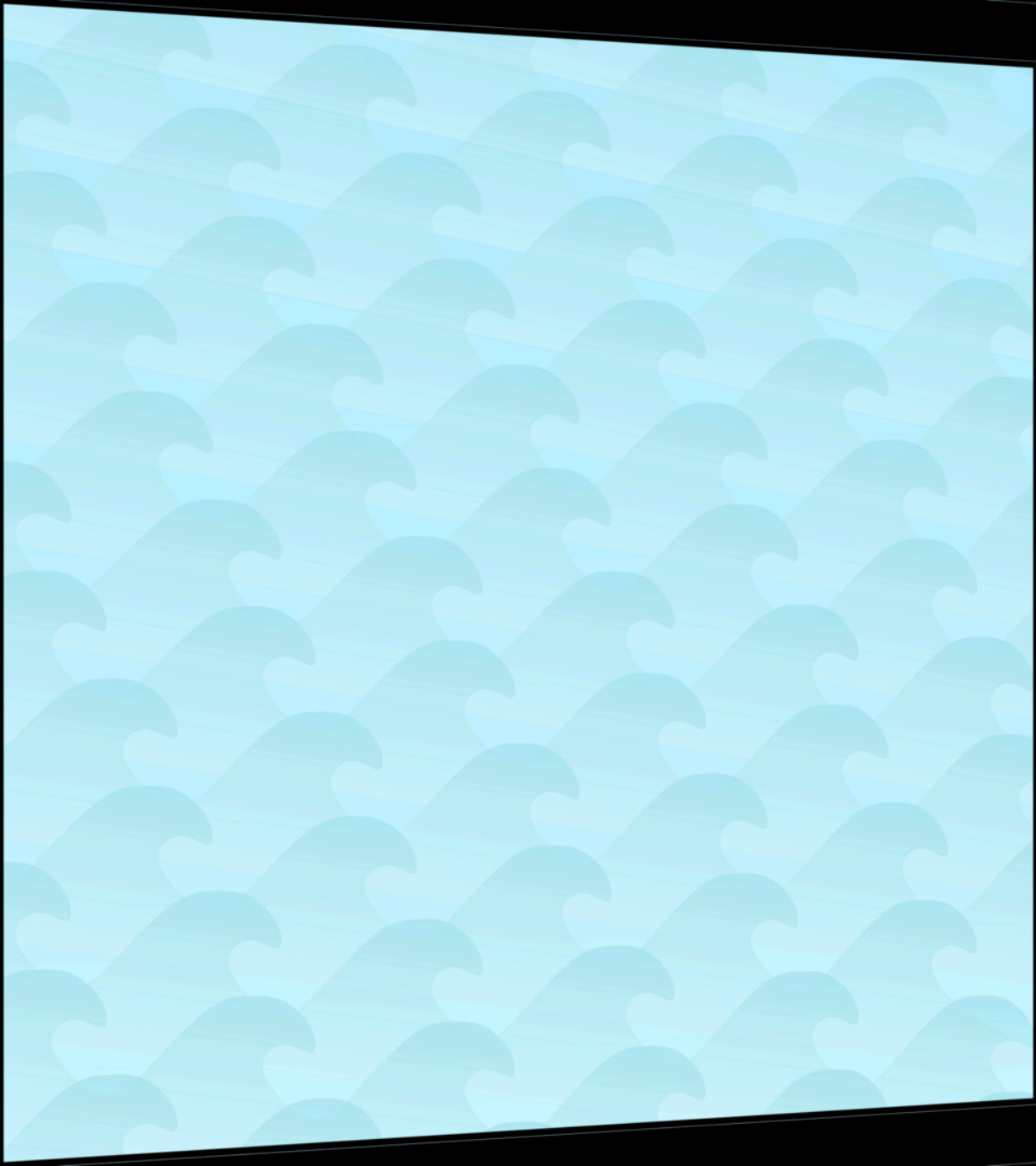
“Where are you”?

ID → DISK

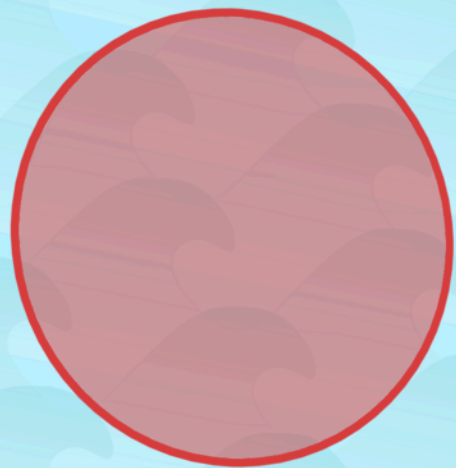
Maybe we can also ask:

“There are *who*”?

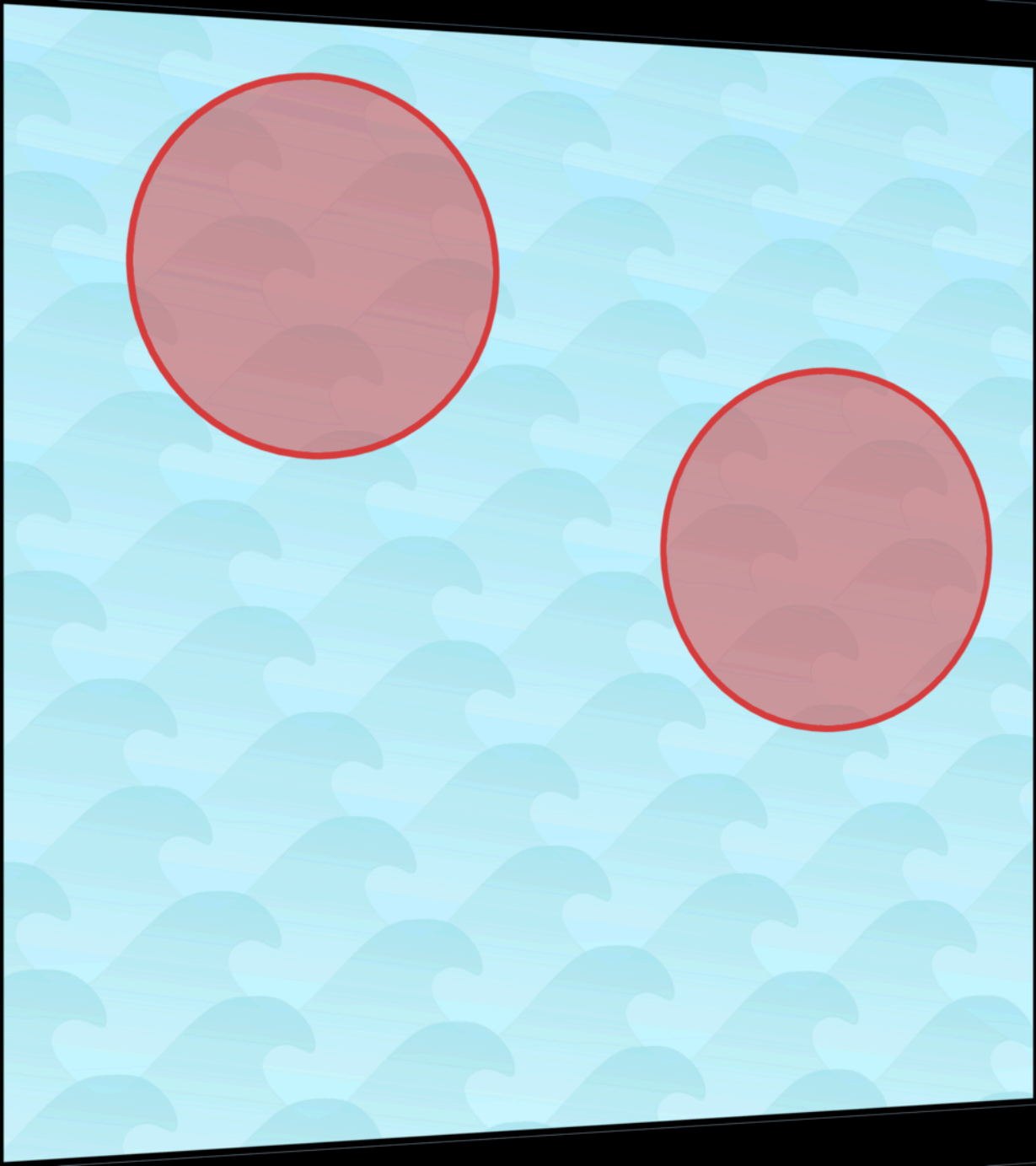
DISK → [ID]



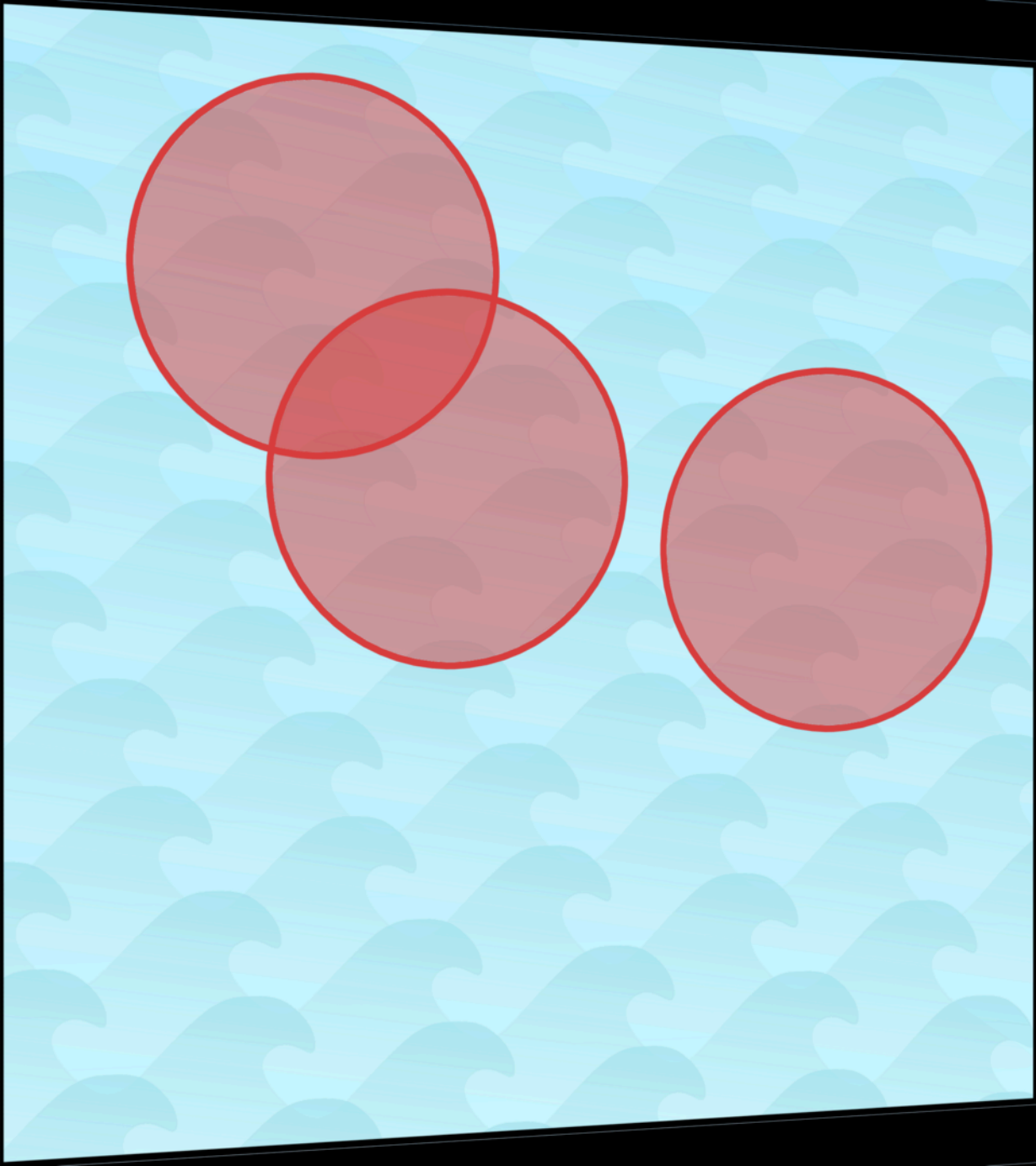
In this model, how should we query to find a specific object?



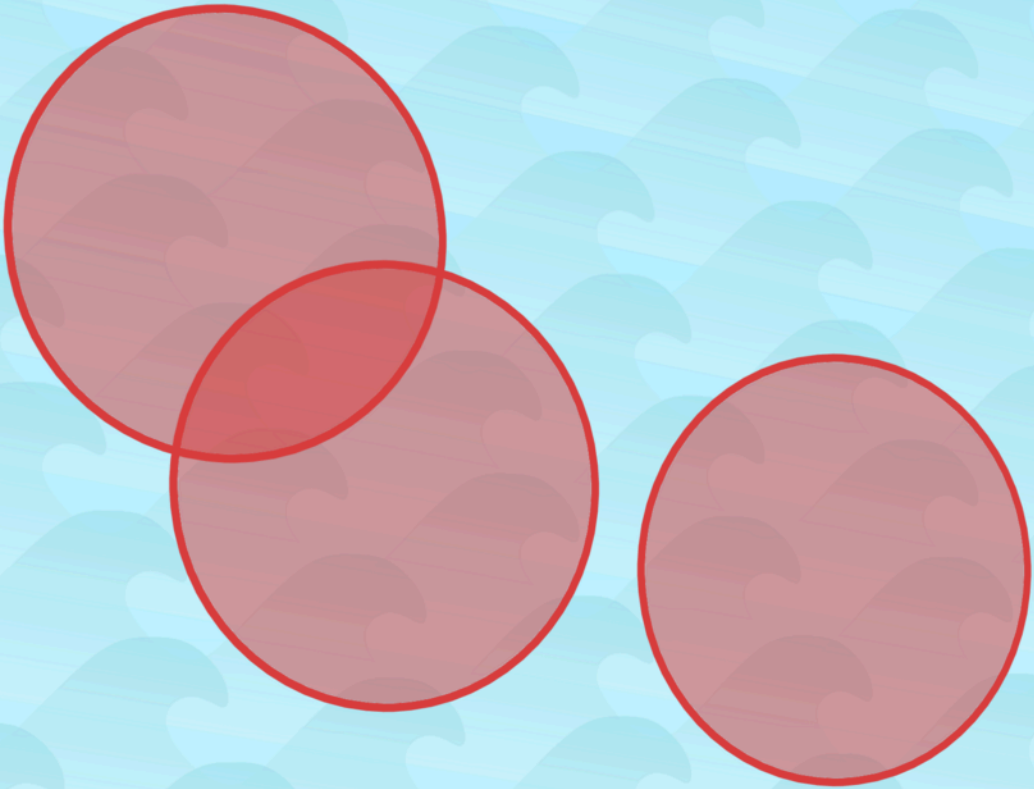
In this model, how should we query to find a specific object?



In this model, how should we query to find a specific object?

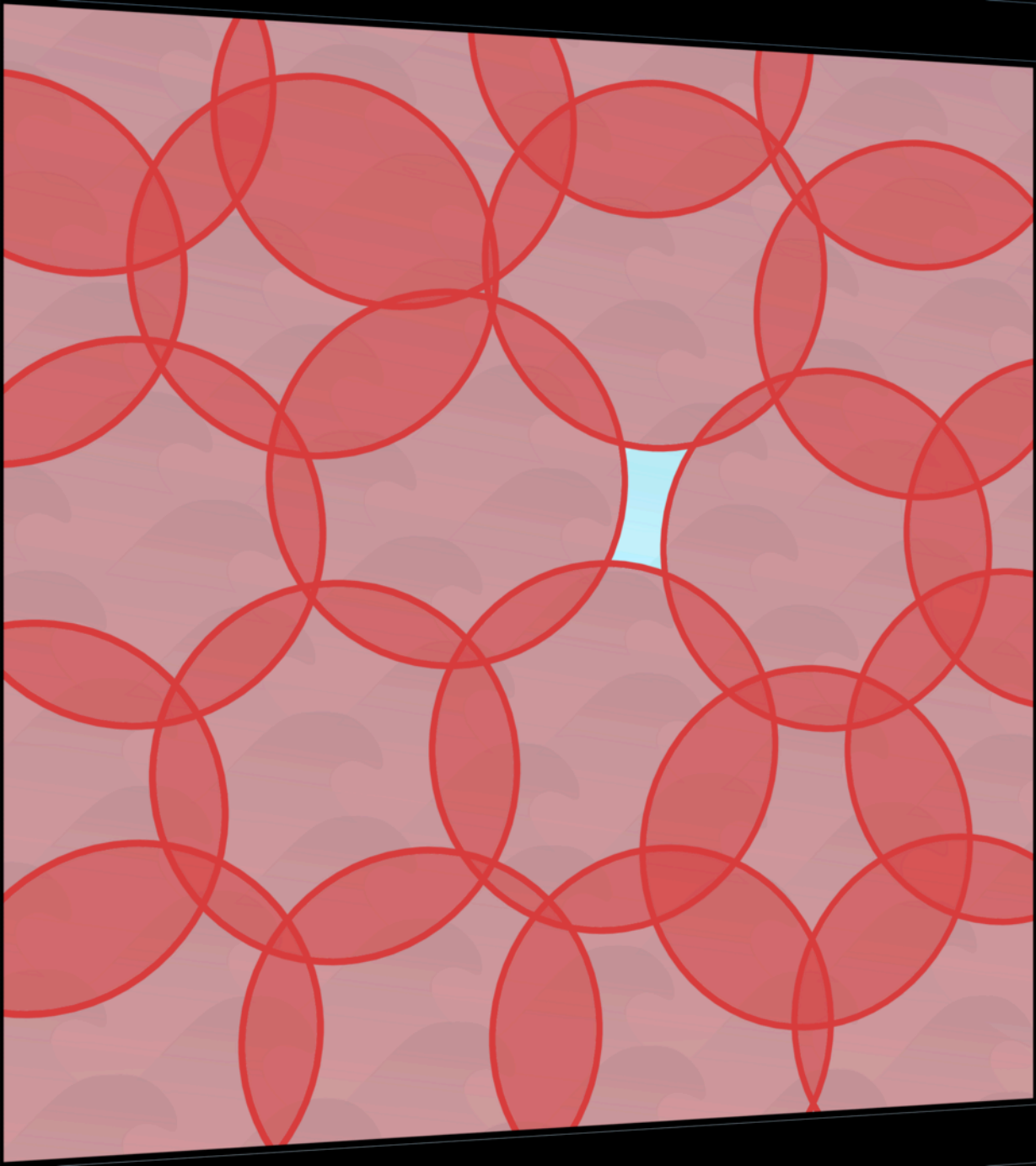


In this model, how should we query to find a specific object?

The image shows three overlapping red circles on a light blue background with a repeating wavy pattern. The circles are semi-transparent, allowing the background pattern to be seen through them. The top-left circle overlaps the middle circle, and the middle circle overlaps the bottom-right circle.

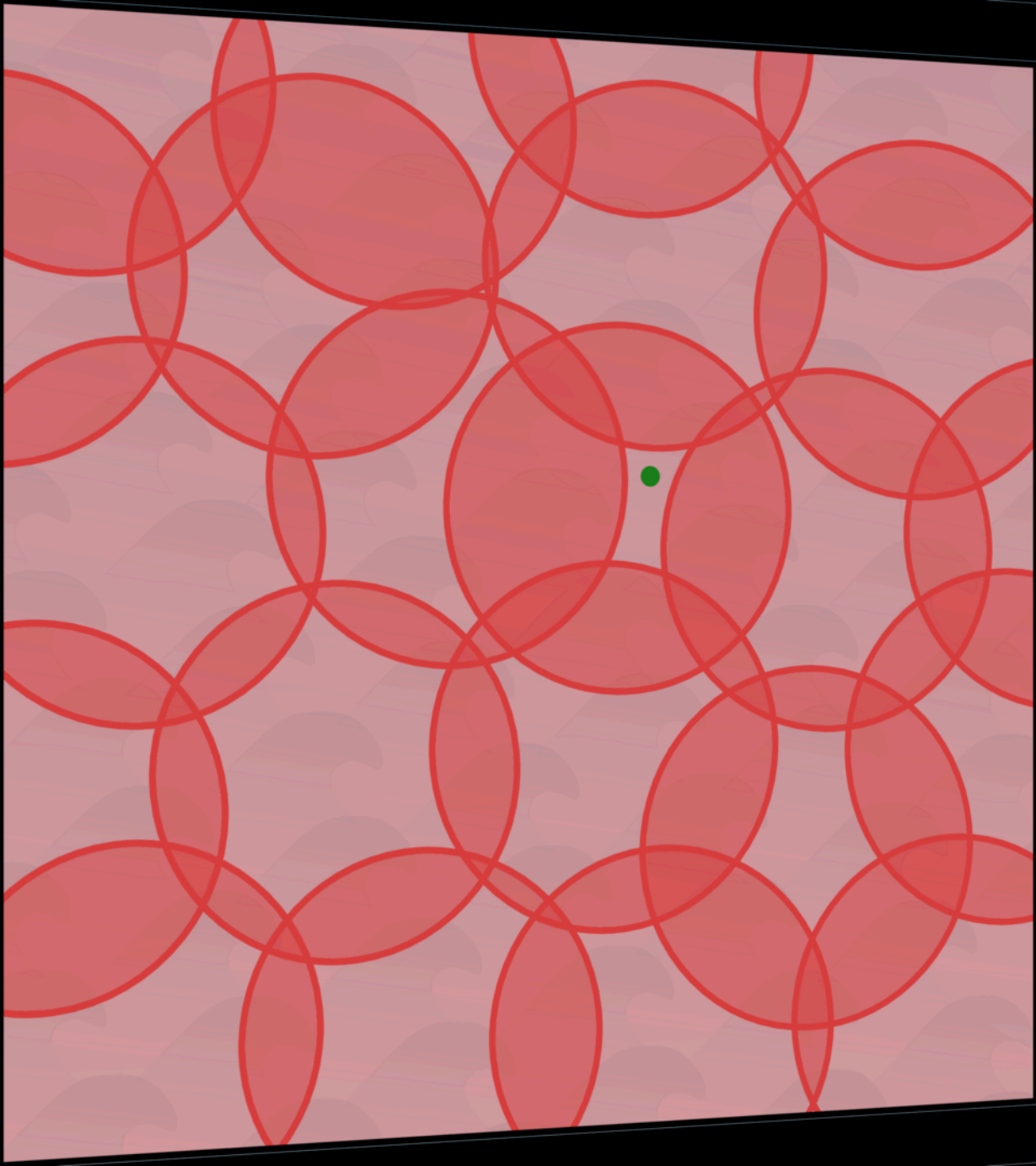
In this model, how should we query to find a specific object?

In the worst case, we find it with the *last* query.



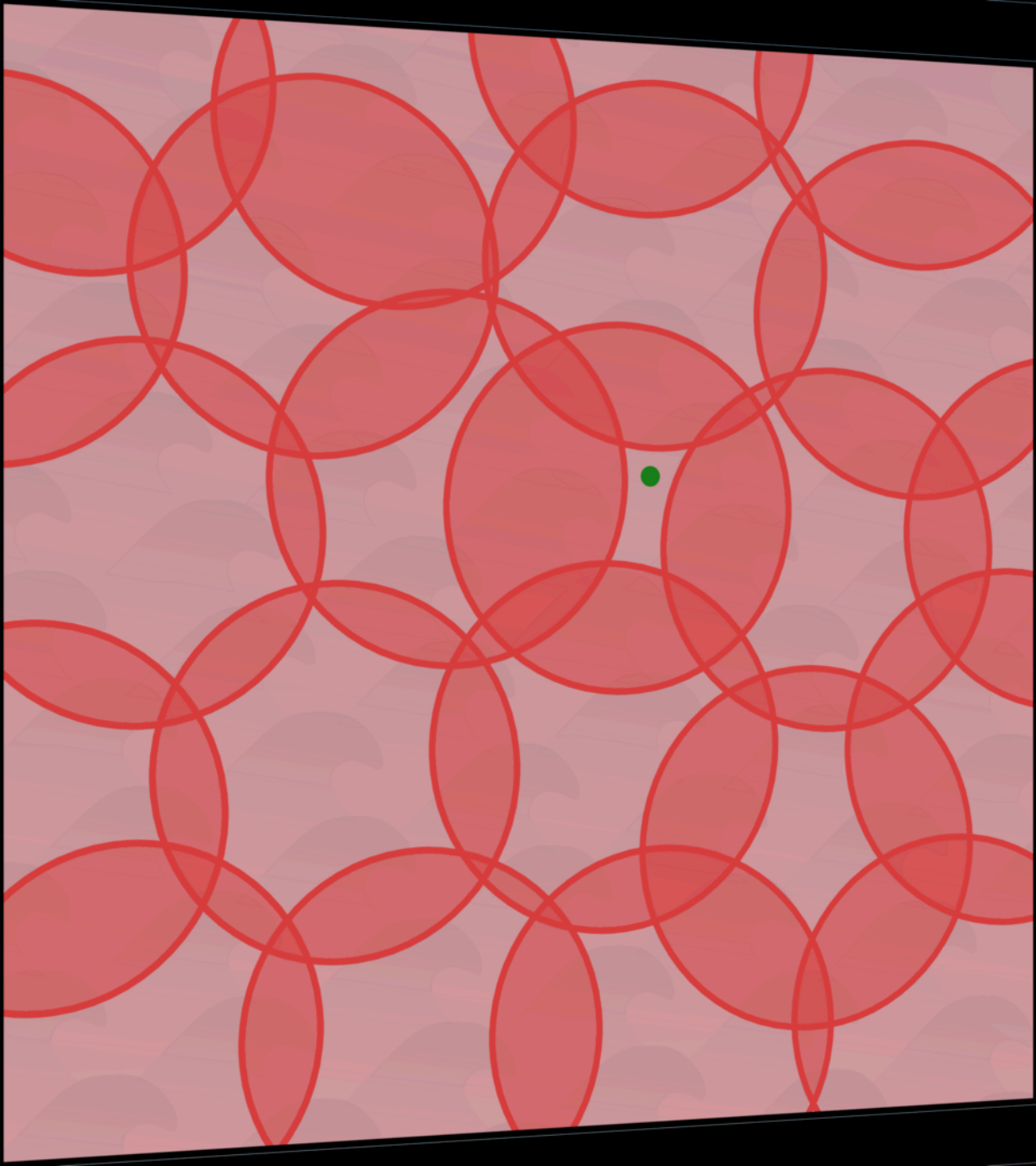
In this model, how should we query to find a specific object?

In the worst case, we find it with the *last* query.



In this model, how should we query to find a specific object?

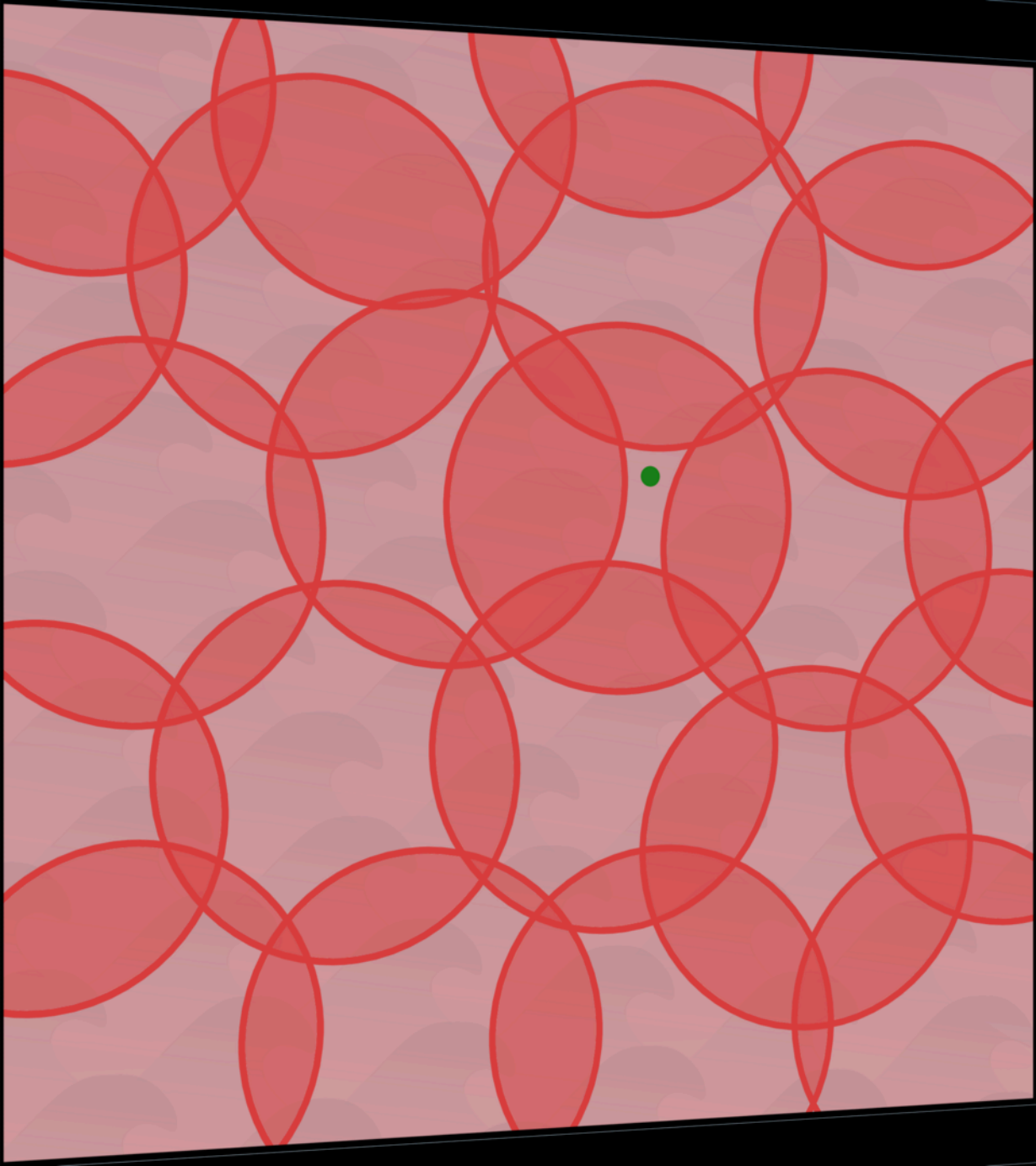
In the worst case, we find it with the *last* query.



In this model, how should we query to find a specific object?

In the worst case, we find it with the *last* query.

How many queries do we need at least?

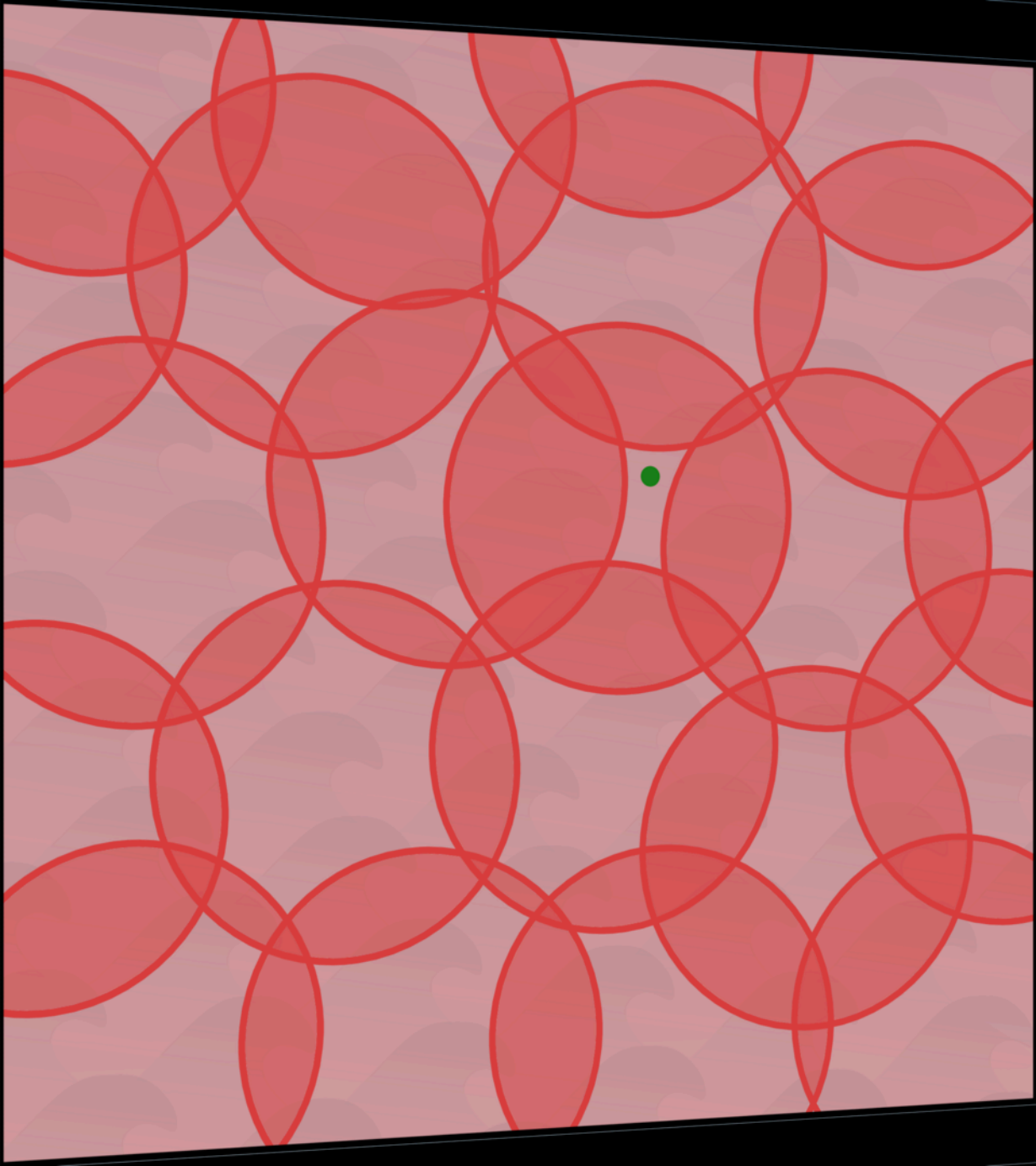


In this model, how should we query to find a specific object?

In the worst case, we find it with the *last* query.

How many queries do we need at least?

How many disks do we need to *cover* a region?



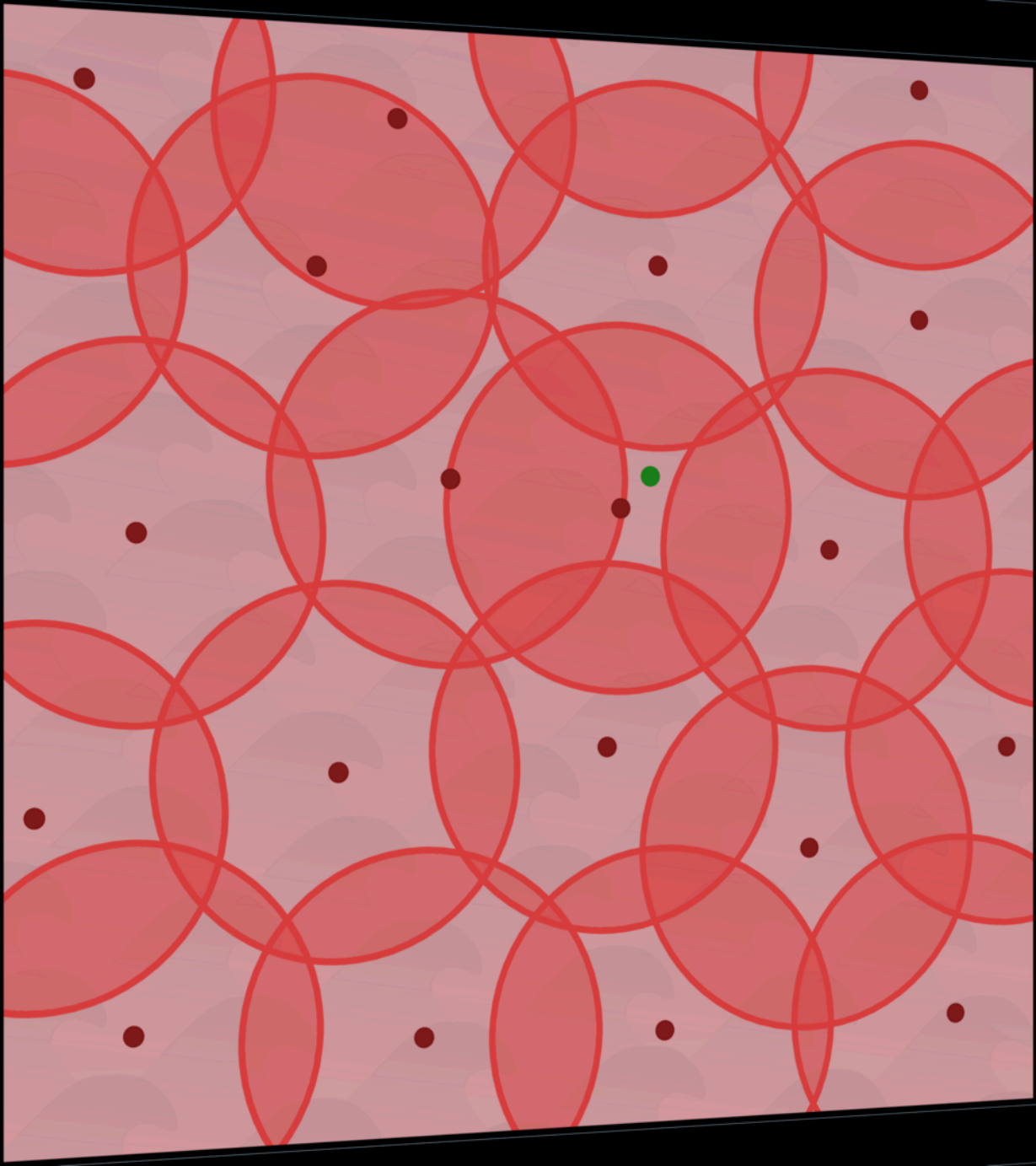
In this model, how should we query to find a specific object?

In the worst case, we find it with the *last* query.

How many queries do we need at least?

How many disks do we need to *cover* a region?

Dually, how many points do we need to *hit* a disk?



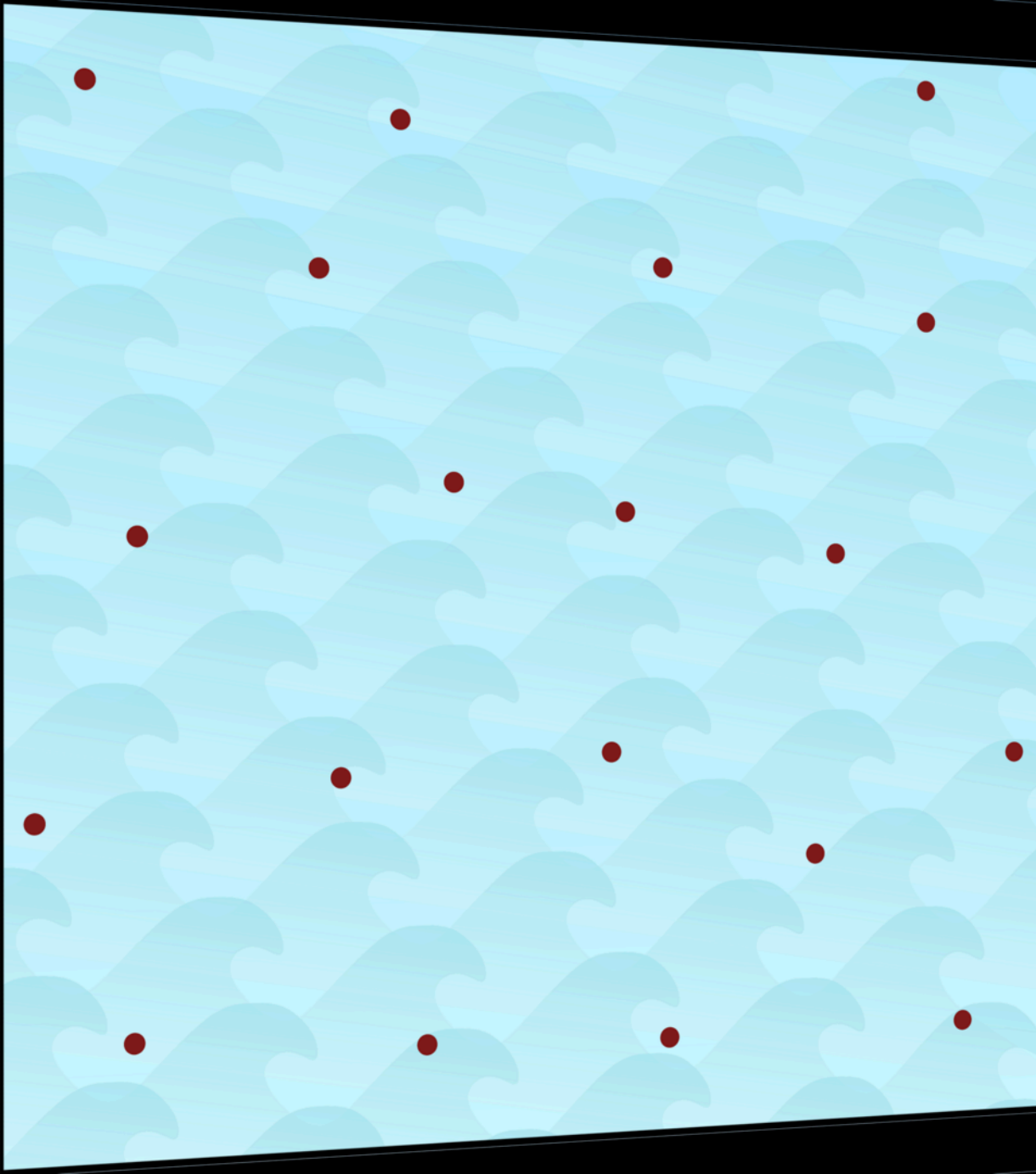
In this model, how should we query to find a specific object?

In the worst case, we find it with the *last* query.

How many queries do we need at least?

How many disks do we need to *cover* a region?

Dually, how many points do we need to *hit* a disk?



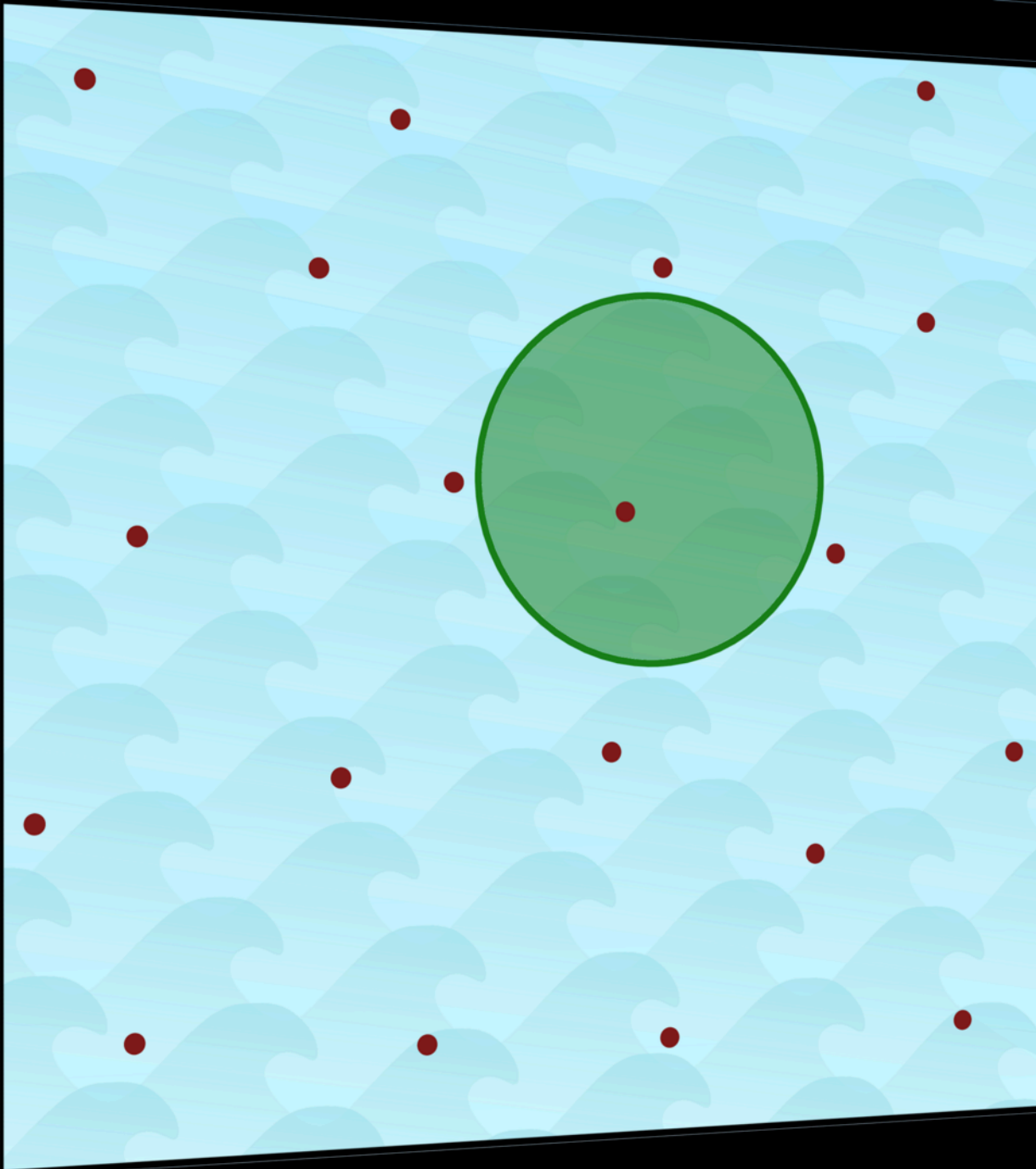
In this model, how should we query to find a specific object?

In the worst case, we find it with the *last* query.

How many queries do we need at least?

How many disks do we need to *cover* a region?

Dually, how many points do we need to *hit* a disk?



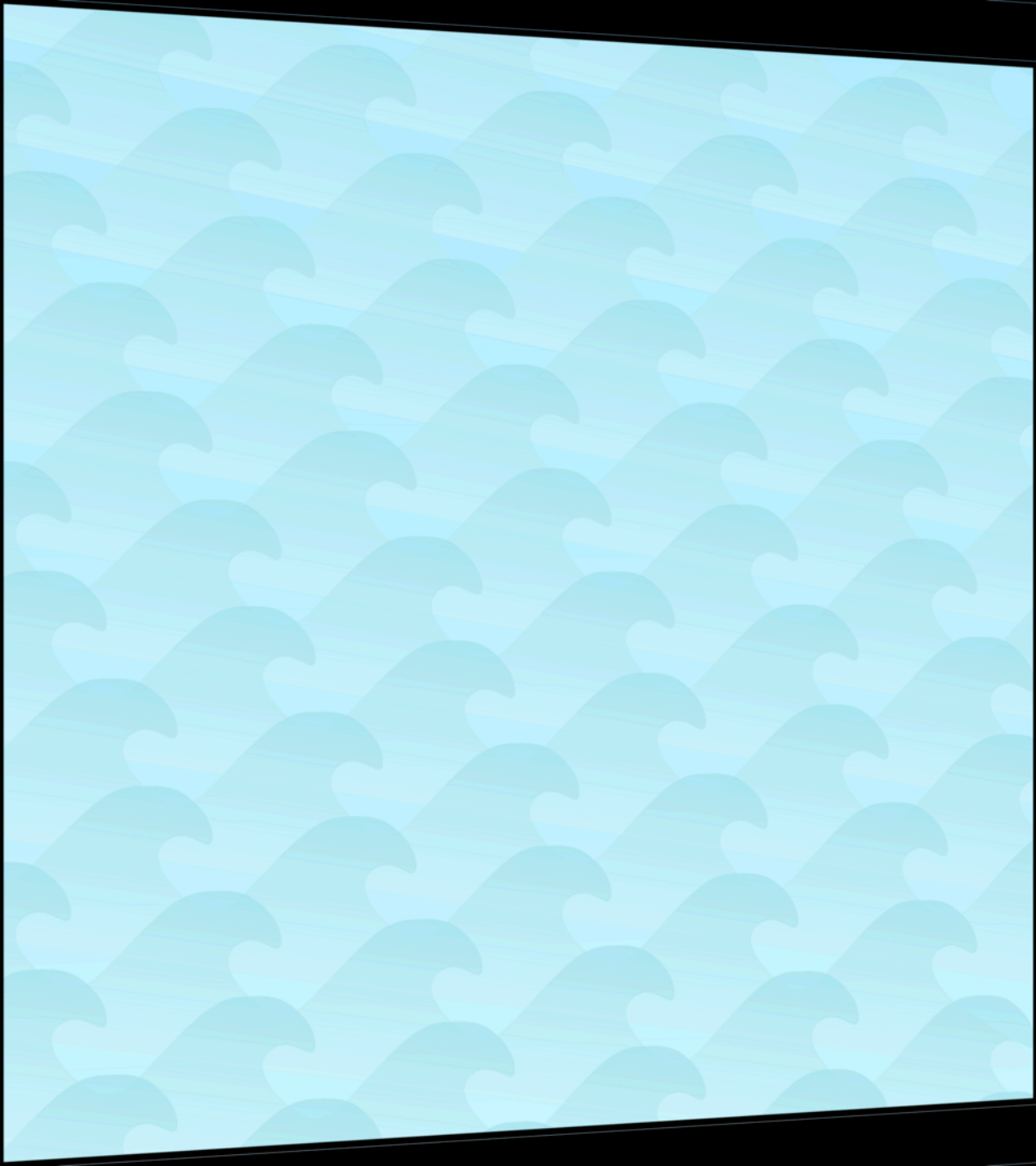
In this model, how should we query to find a specific object?

In the worst case, we find it with the *last* query.

How many queries do we need at least?

How many disks do we need to *cover* a region?

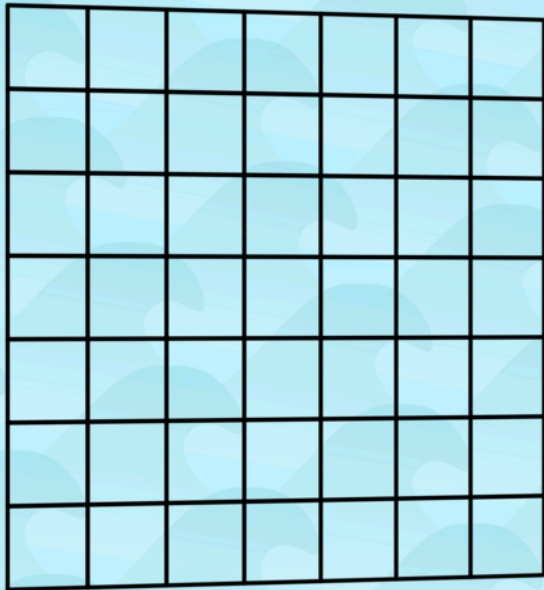
Dually, how many points do we need to *hit* a disk?



2

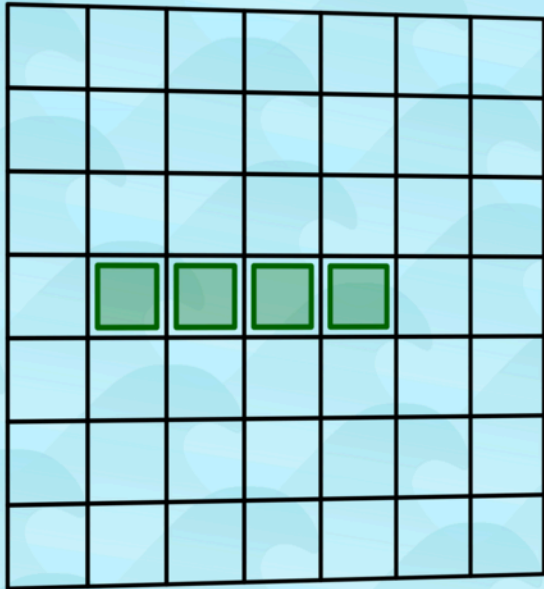
CHAPTER

BATTLESHIPS

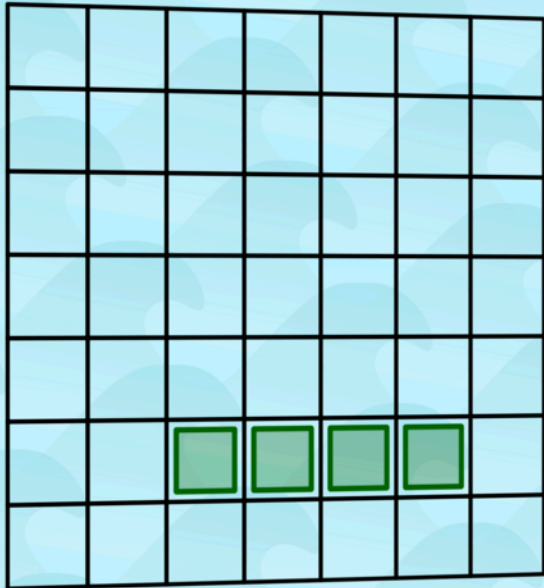


Somewhere on a grid is a battleship of known *shape* but unknown *location*.

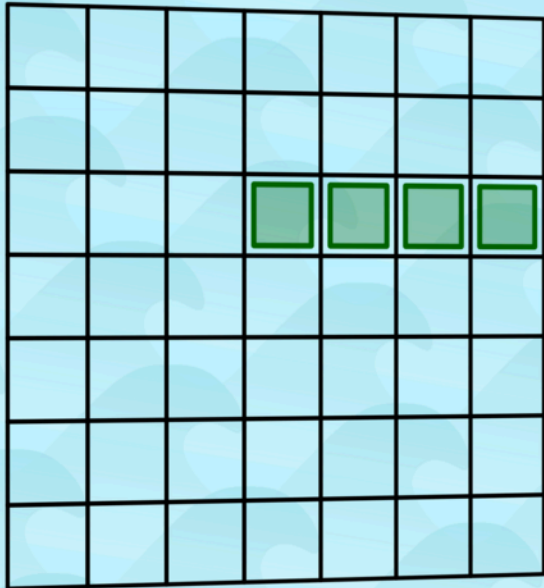
Somewhere on a grid is a battleship of known *shape* but unknown *location*.

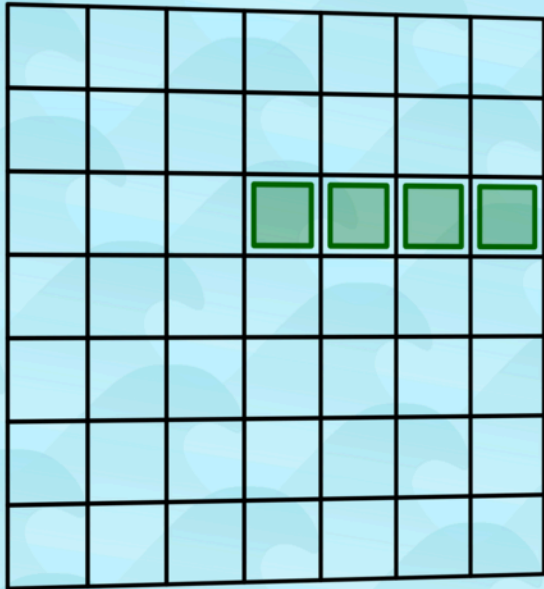


Somewhere on a grid is a battleship of known *shape* but unknown *location*.



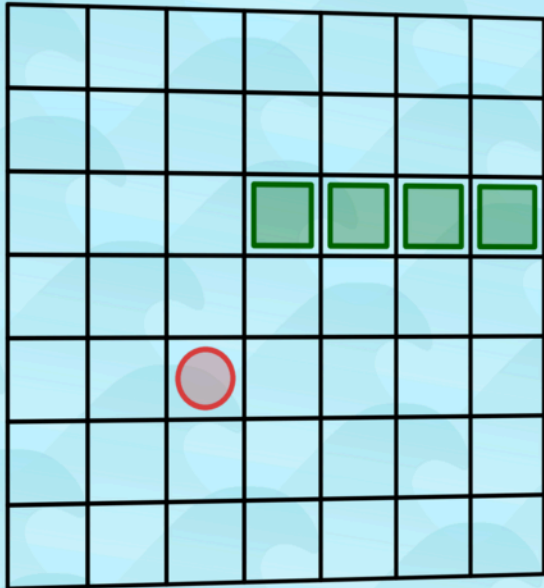
Somewhere on a grid is a battleship of known *shape* but unknown *location*.





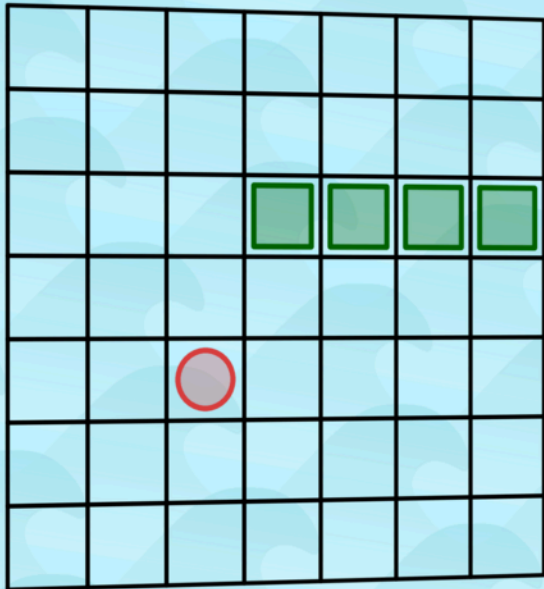
Somewhere on a grid is a battleship of known *shape* but unknown *location*.

We can query a *cell* and ask: is anything there?



Somewhere on a grid is a battleship of known *shape* but unknown *location*.

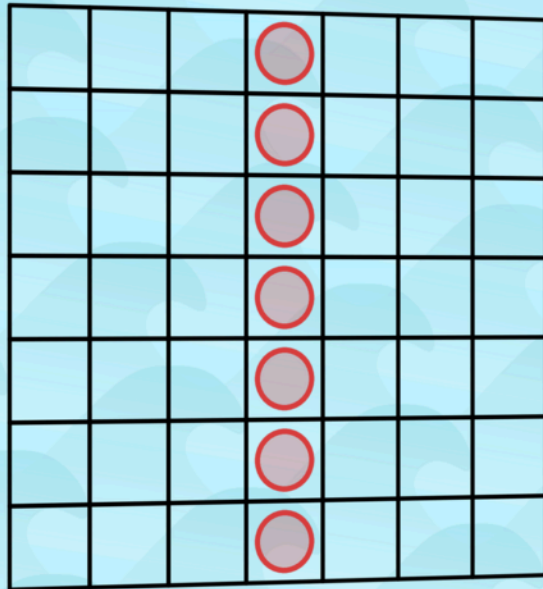
We can query a *cell* and ask: is anything there?



Somewhere on a grid is a battleship of known *shape* but unknown *location*.

We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?

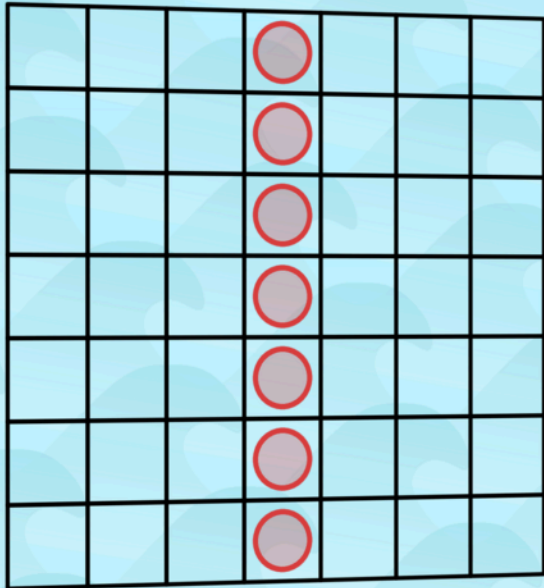


7

Somewhere on a grid is a battleship of known *shape* but unknown *location*.

We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?



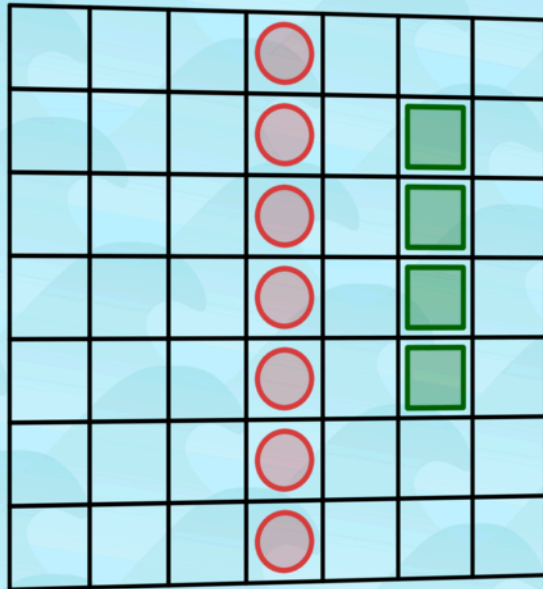
7

Somewhere on a grid is a battleship of known *shape* but unknown *location*.

We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?

In the game, ships may also be *rotated*.



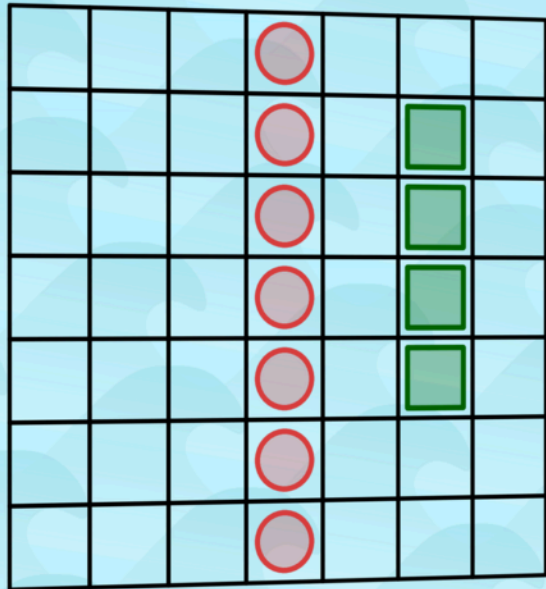
7

Somewhere on a grid is a battleship of known *shape* but unknown *location*.

We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?

In the game, ships may also be *rotated*.



7

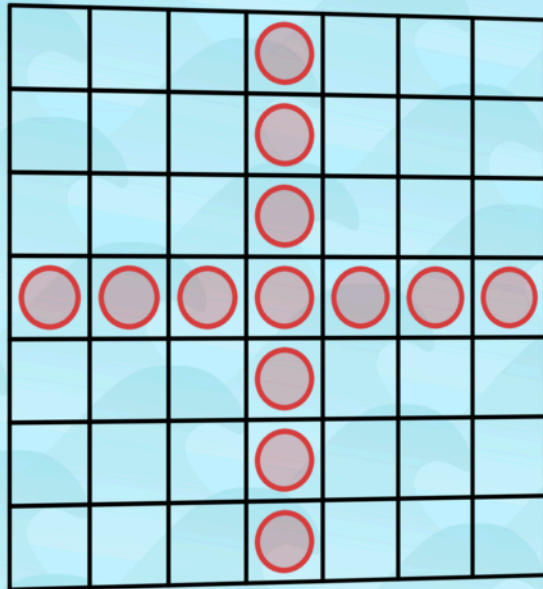
Somewhere on a grid is a battleship of known *shape* but unknown *location*.

We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?

In the game, ships may also be *rotated*.

If so, we say the ship has *unknown shape*: $S = \left\{ \begin{array}{c} \blacksquare \\ \blacksquare \\ \blacksquare \end{array} , \begin{array}{ccc} \blacksquare & \blacksquare & \blacksquare \end{array} \right\}$



13

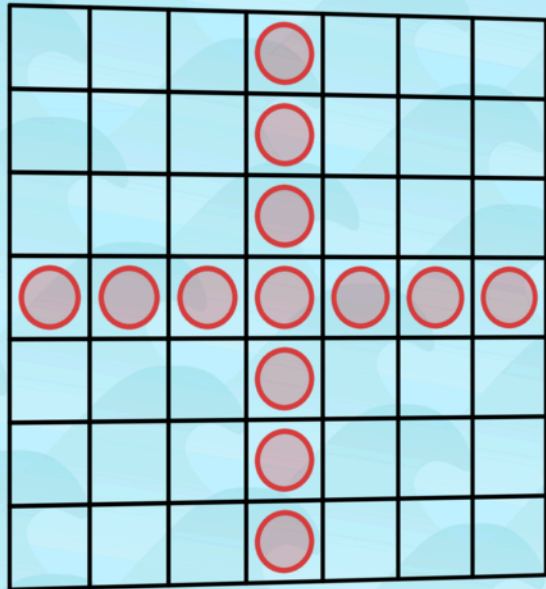
Somewhere on a grid is a battleship of known *shape* but unknown *location*.

We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?

In the game, ships may also be *rotated*.

If so, we say the ship has *unknown shape*: $S = \left\{ \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \end{array} \right\}$



13

Somewhere on a grid is a battleship of known *shape* but unknown *location*.

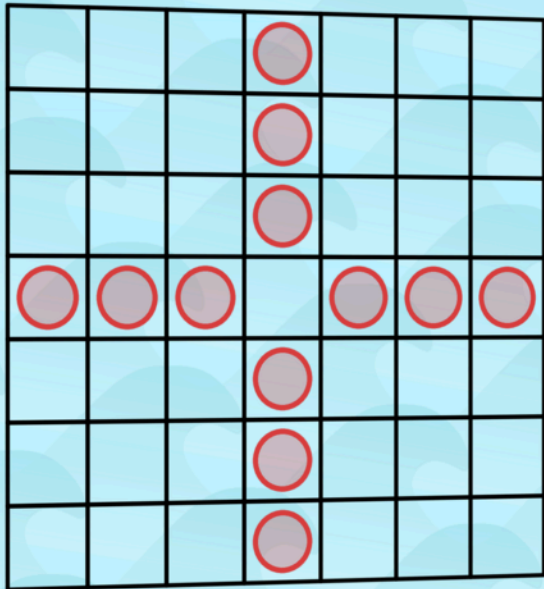
We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?

In the game, ships may also be *rotated*.

If so, we say the ship has *unknown shape*: $S = \left\{ \begin{array}{|c|} \hline \square \\ \square \\ \square \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \end{array} \right\}$

What is the best pattern?



12

Somewhere on a grid is a battleship of known *shape* but unknown *location*.

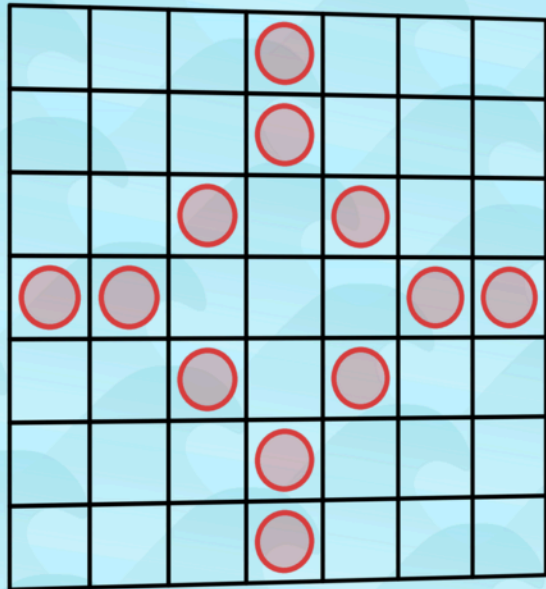
We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?

In the game, ships may also be *rotated*.

If so, we say the ship has *unknown shape*: $S = \left\{ \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \end{array} \right\}$

What is the best pattern?



12

Somewhere on a grid is a battleship of known *shape* but unknown *location*.

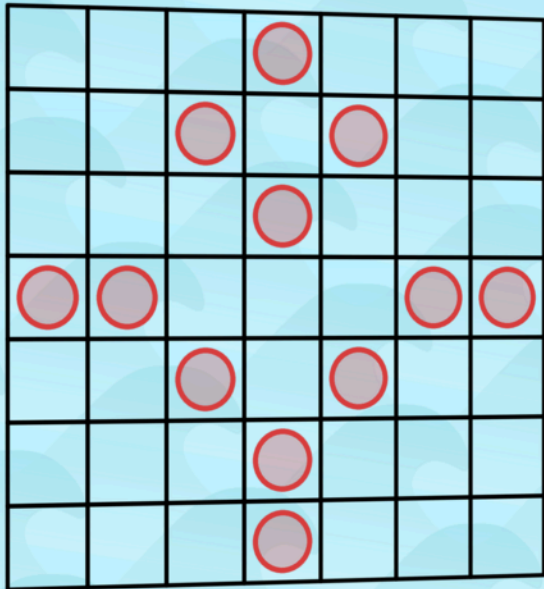
We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?

In the game, ships may also be *rotated*.

If so, we say the ship has *unknown shape*: $S = \left\{ \begin{array}{c} \blacksquare \\ \blacksquare \\ \blacksquare \end{array} , \begin{array}{ccc} \blacksquare & \blacksquare & \blacksquare \end{array} \right\}$

What is the best pattern?



12

Somewhere on a grid is a battleship of known *shape* but unknown *location*.

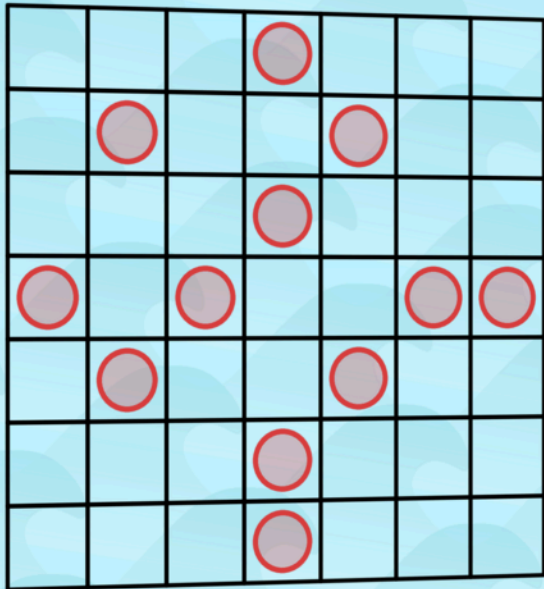
We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?

In the game, ships may also be *rotated*.

If so, we say the ship has *unknown shape*: $S = \left\{ \begin{array}{|c|} \hline \square \\ \square \\ \square \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \right\}$

What is the best pattern?



12

Somewhere on a grid is a battleship of known *shape* but unknown *location*.

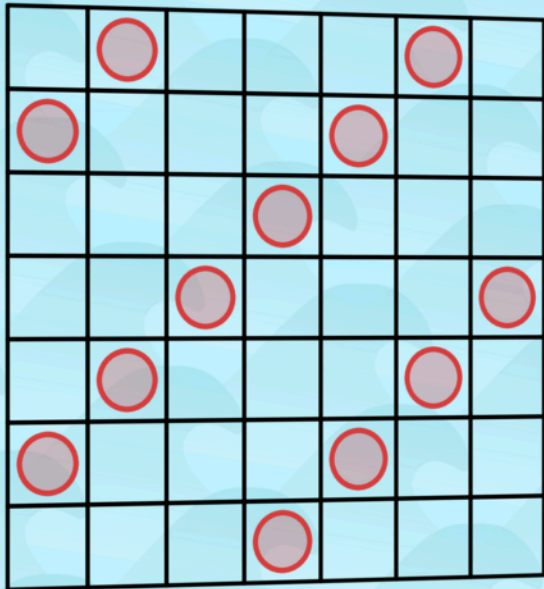
We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?

In the game, ships may also be *rotated*.

If so, we say the ship has *unknown shape*: $S = \left\{ \begin{array}{c} \blacksquare \\ \blacksquare \\ \blacksquare \end{array} , \begin{array}{ccc} \blacksquare & \blacksquare & \blacksquare \end{array} \right\}$

What is the best pattern?



12

Somewhere on a grid is a battleship of known *shape* but unknown *location*.

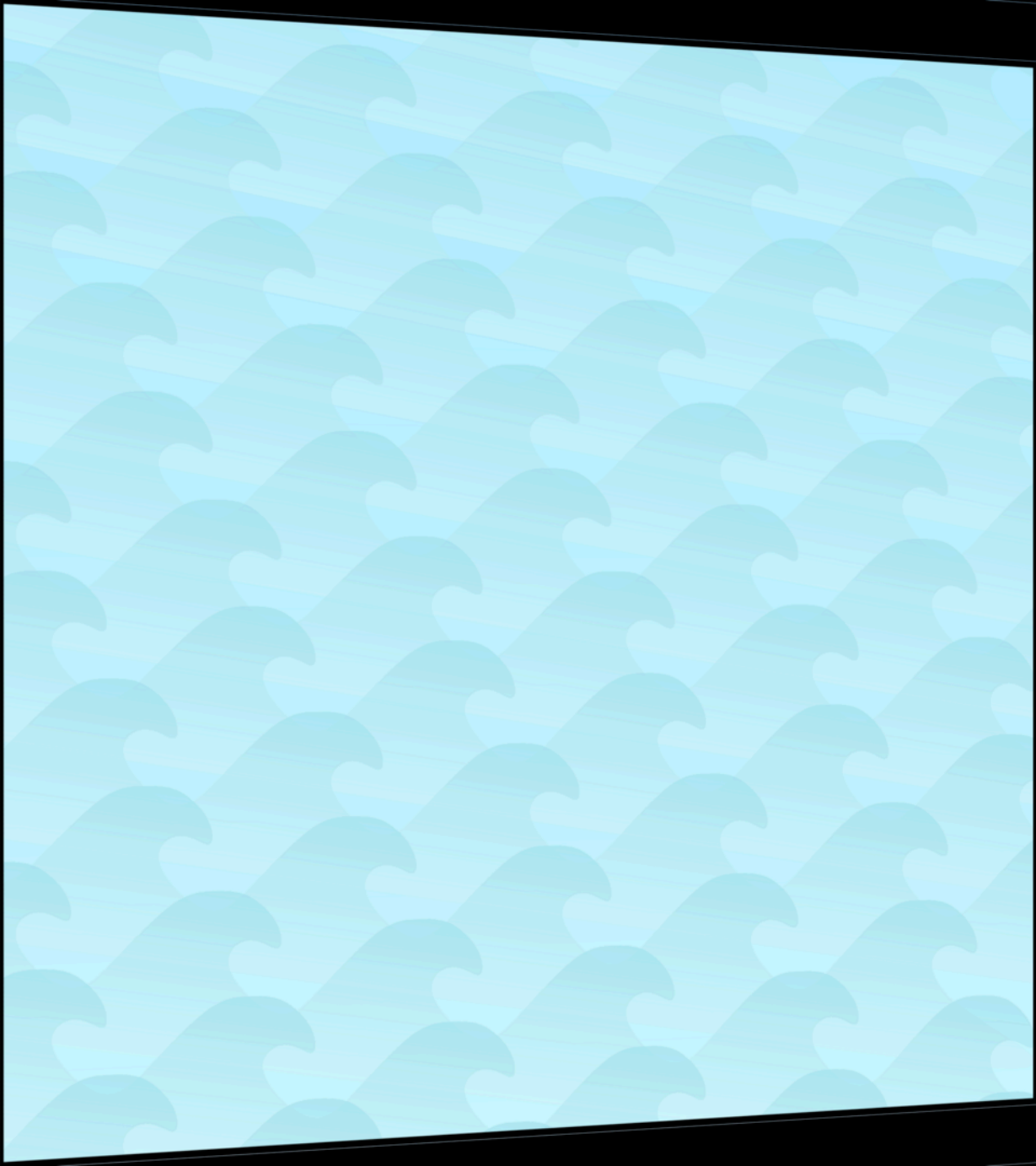
We can query a *cell* and ask: is anything there?

How many cells do we need to query to find the ship?

In the game, ships may also be *rotated*.

If so, we say the ship has *unknown shape*: $S = \left\{ \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \end{array} \right\}$

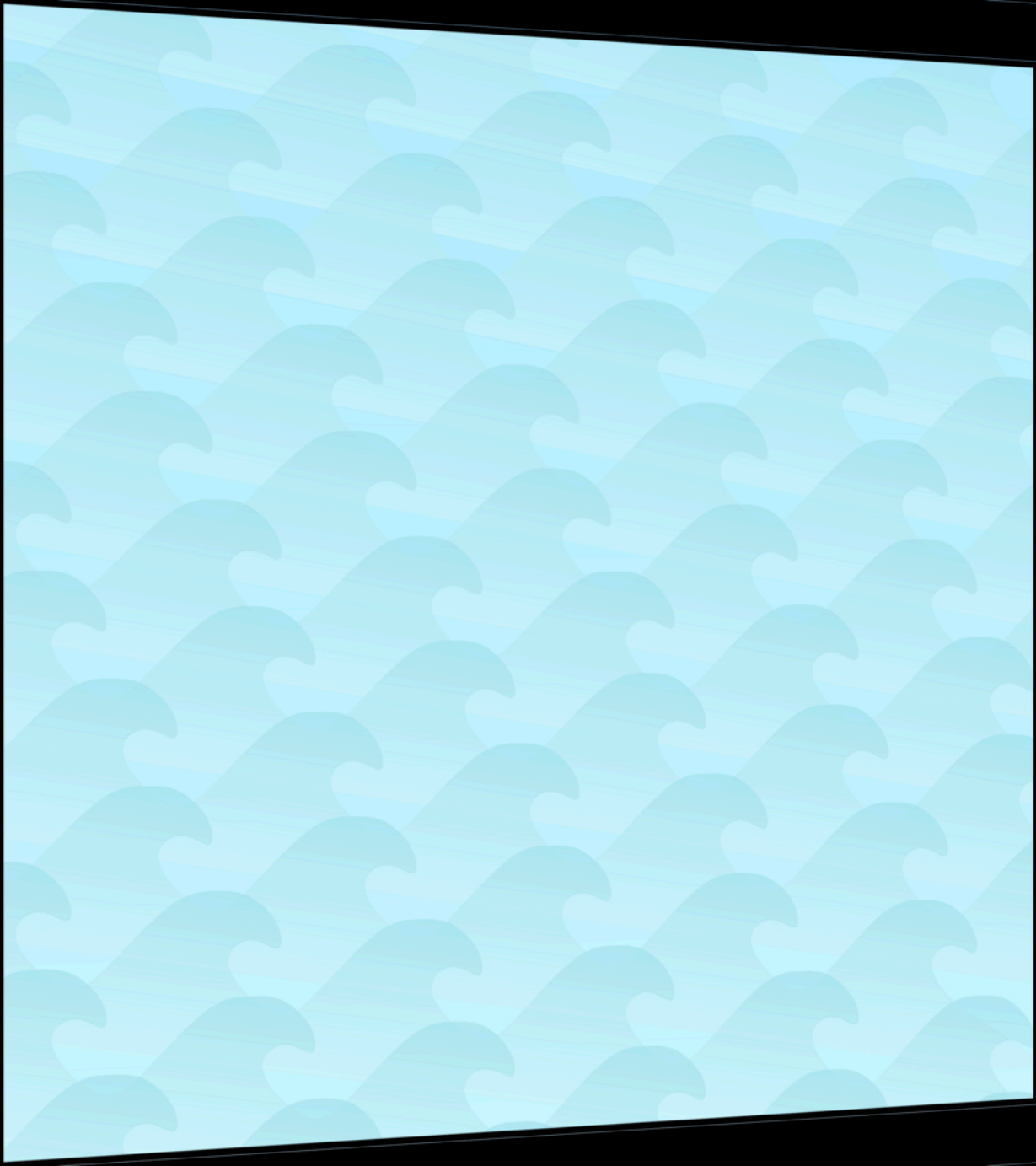
What is the best pattern?



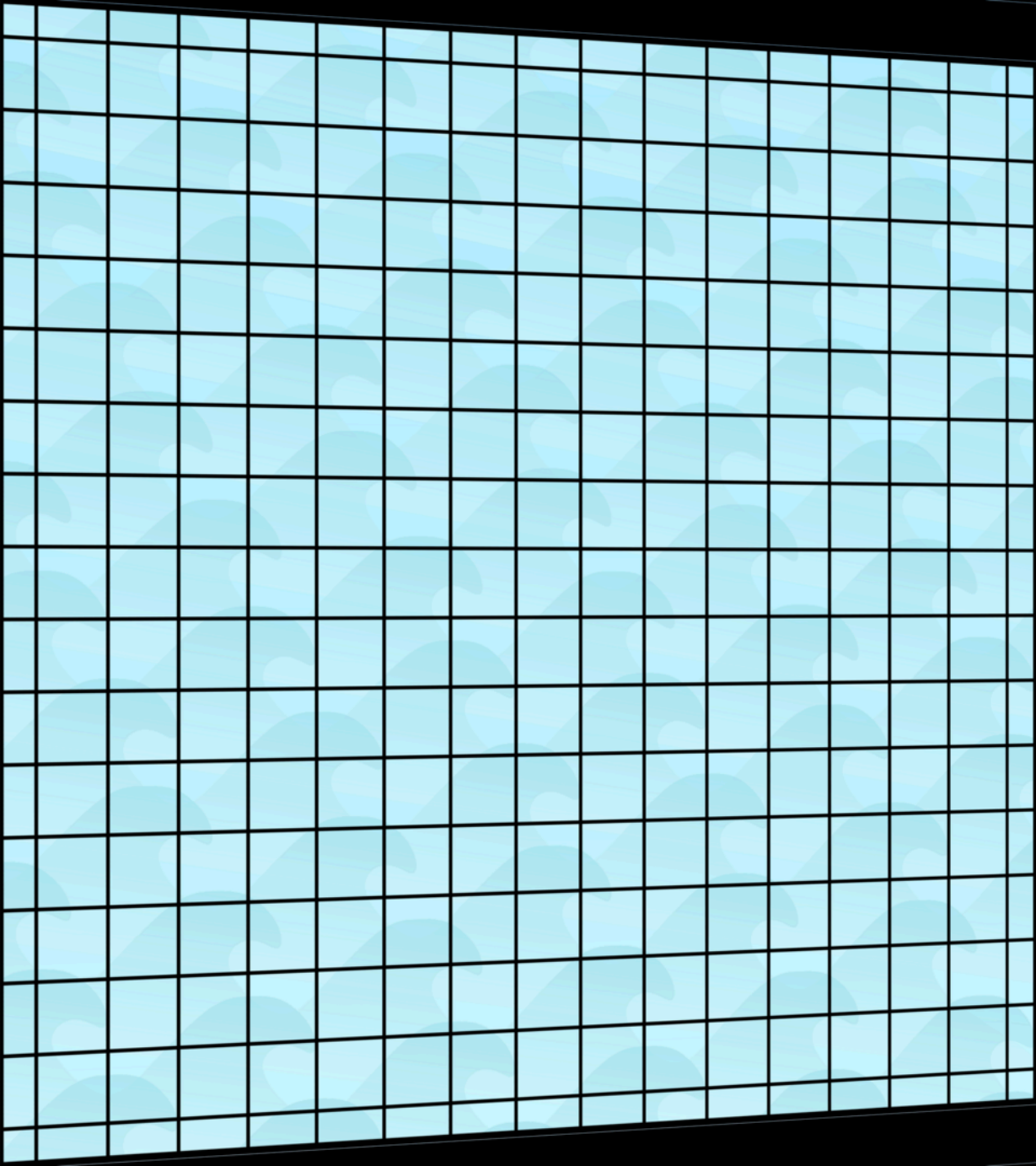
3

CHAPTER

.....
DENSITY



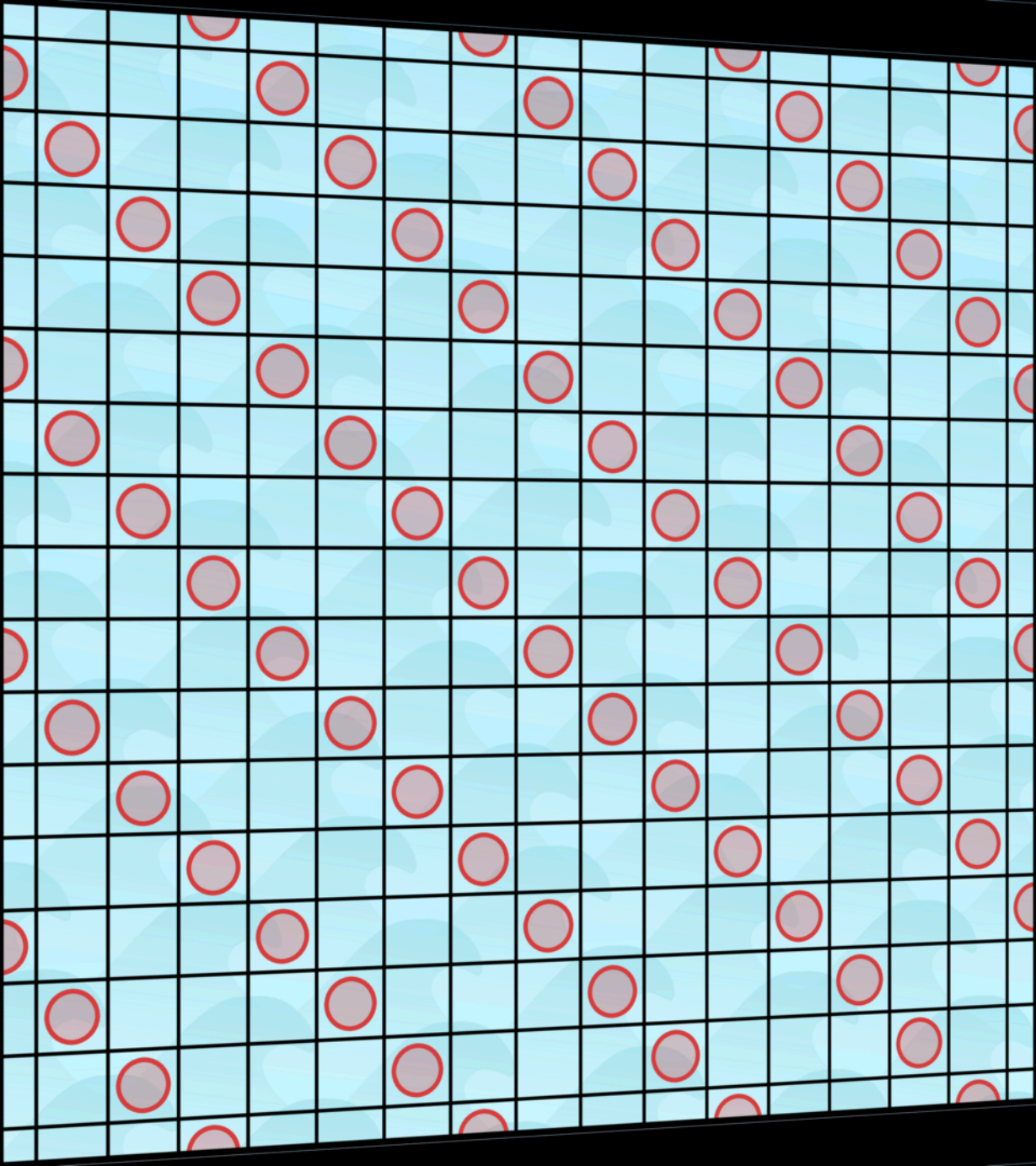
Let's consider battleships
on an *infinite* board.



Let's consider battleships
on an *infinite* board.

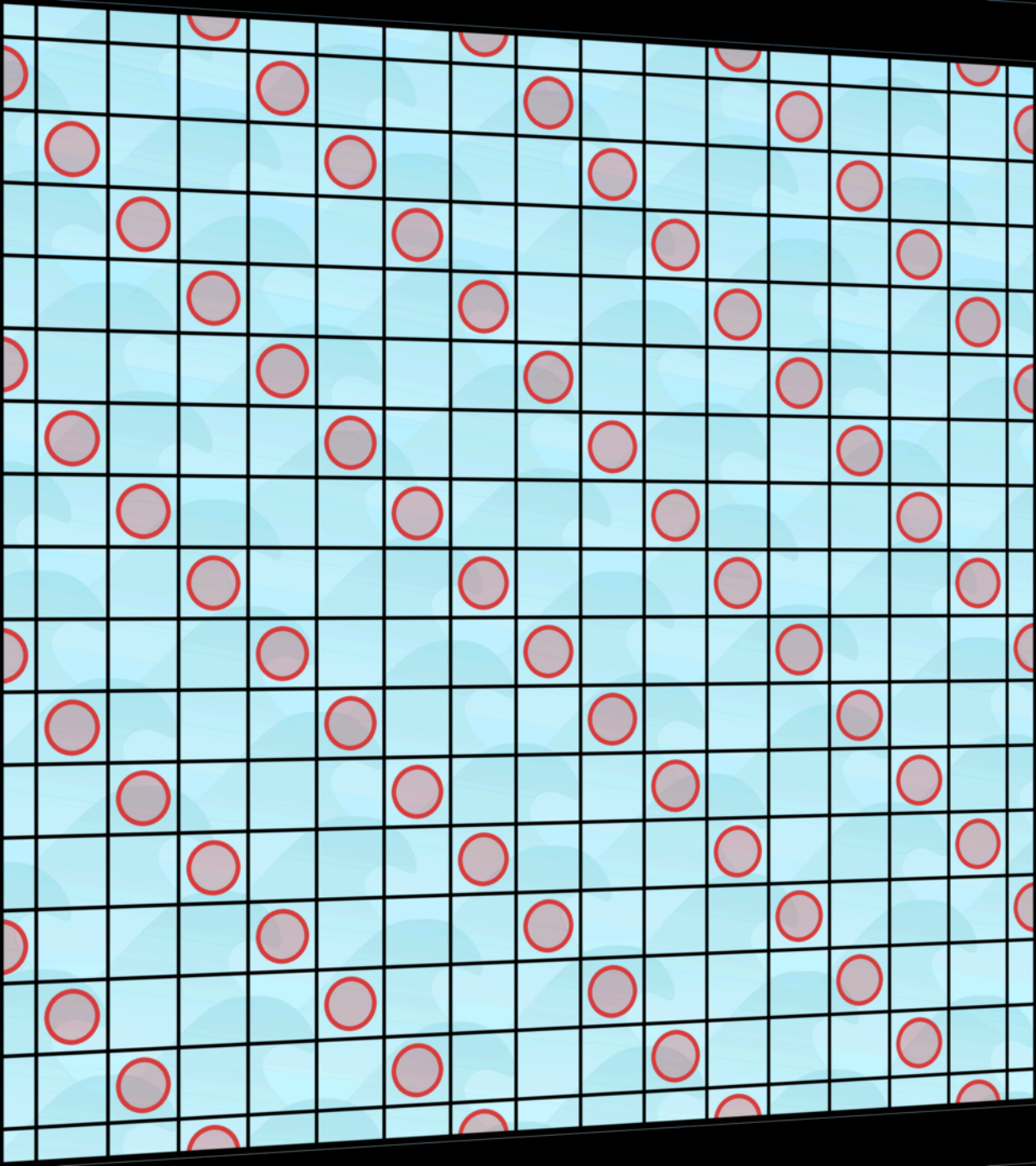
Let's consider battleships
on an *infinite* board.

What is the best pattern
to find $S = \left\{ \begin{array}{|c|} \hline \square \\ \square \\ \square \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \right\}$?



Let's consider battleships
on an *infinite* board.

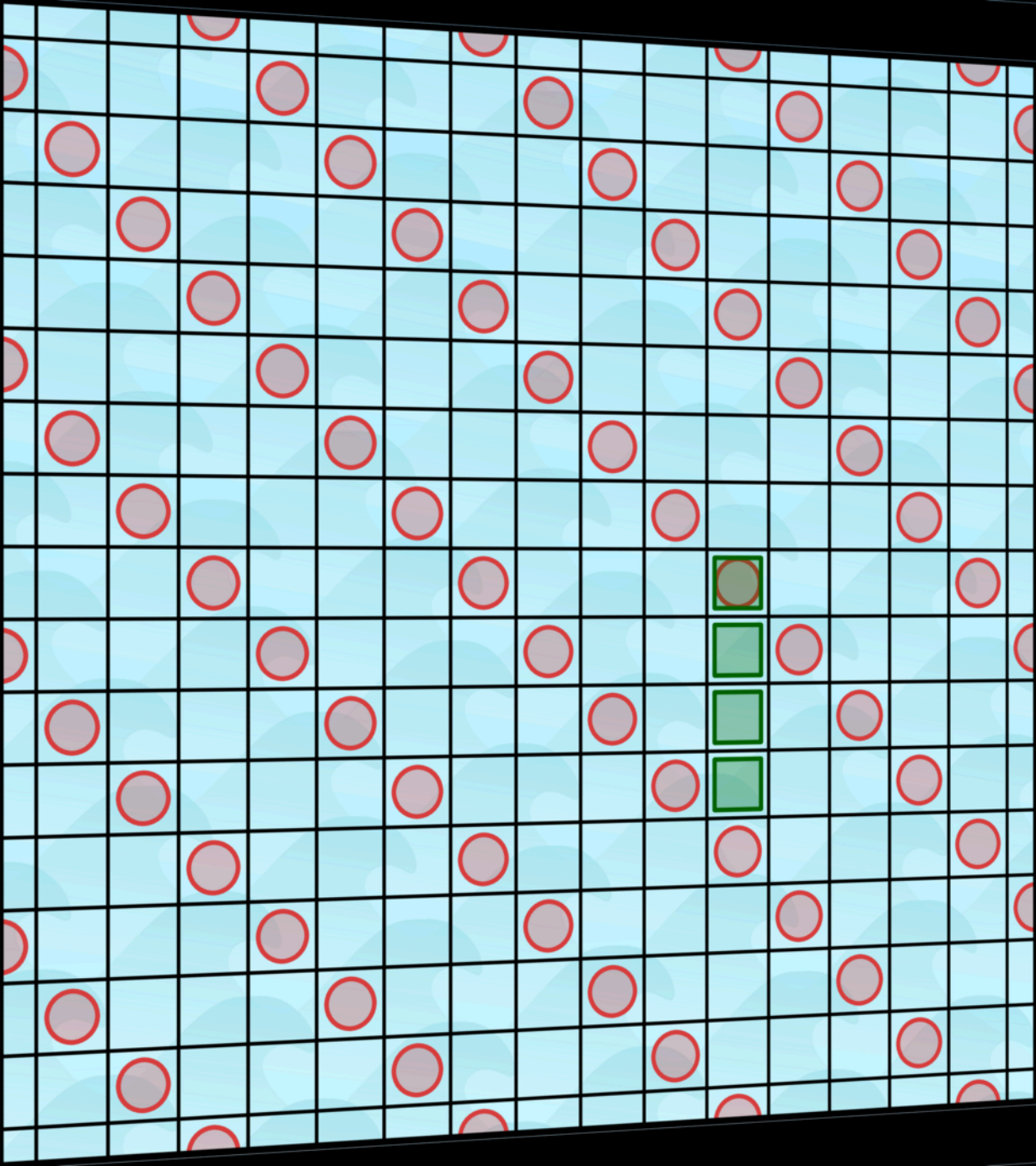
What is the best pattern
to find $S = \{ \begin{array}{|c|} \hline \square \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \}$?



Let's consider battleships on an *infinite* board.

What is the best pattern to find $S = \{ \begin{array}{|c|} \hline \square \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \}$?

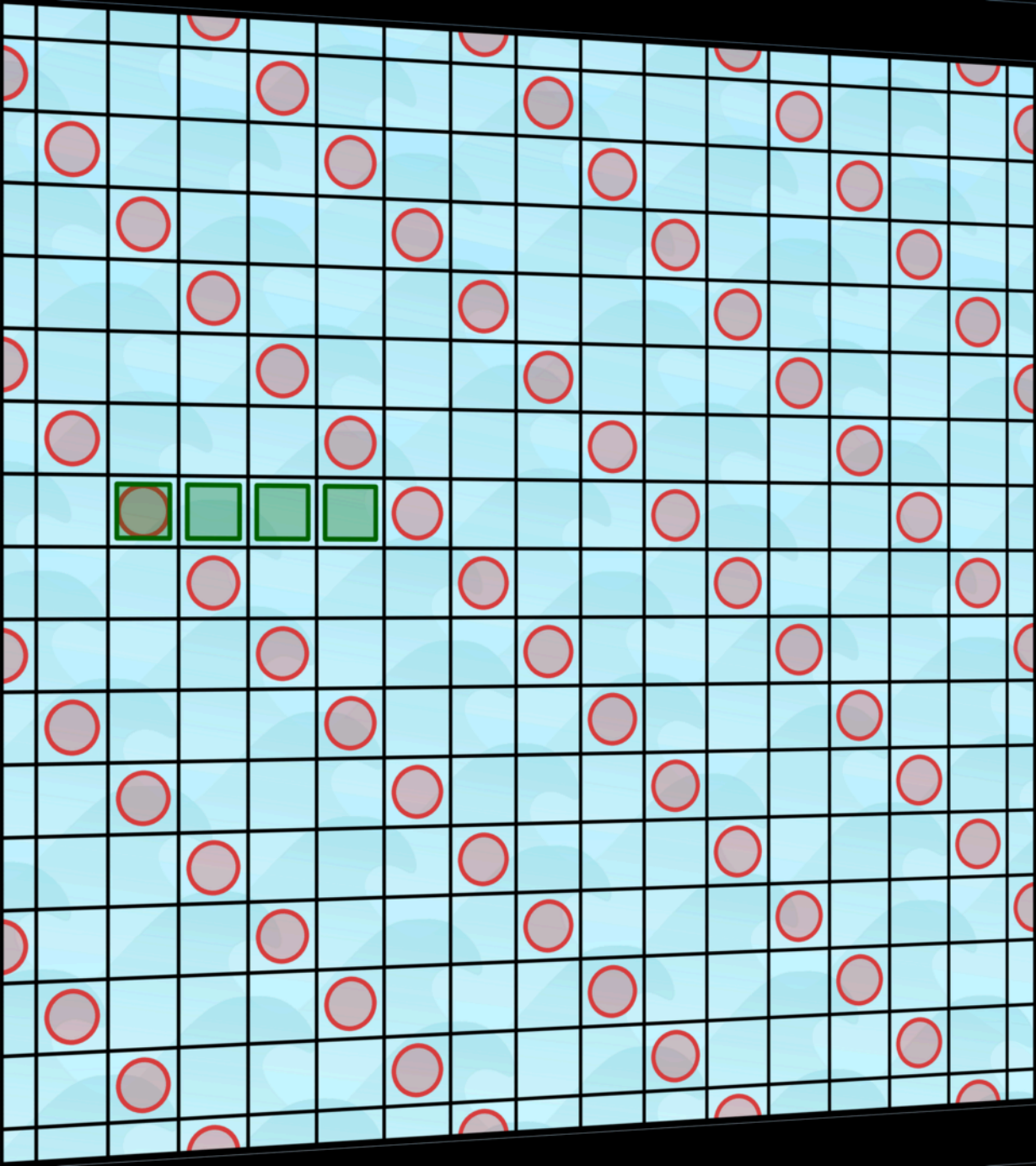
Now, it is easy to see that this pattern is optimal.



Let's consider battleships on an *infinite* board.

What is the best pattern to find $S = \{ \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array} \}$?

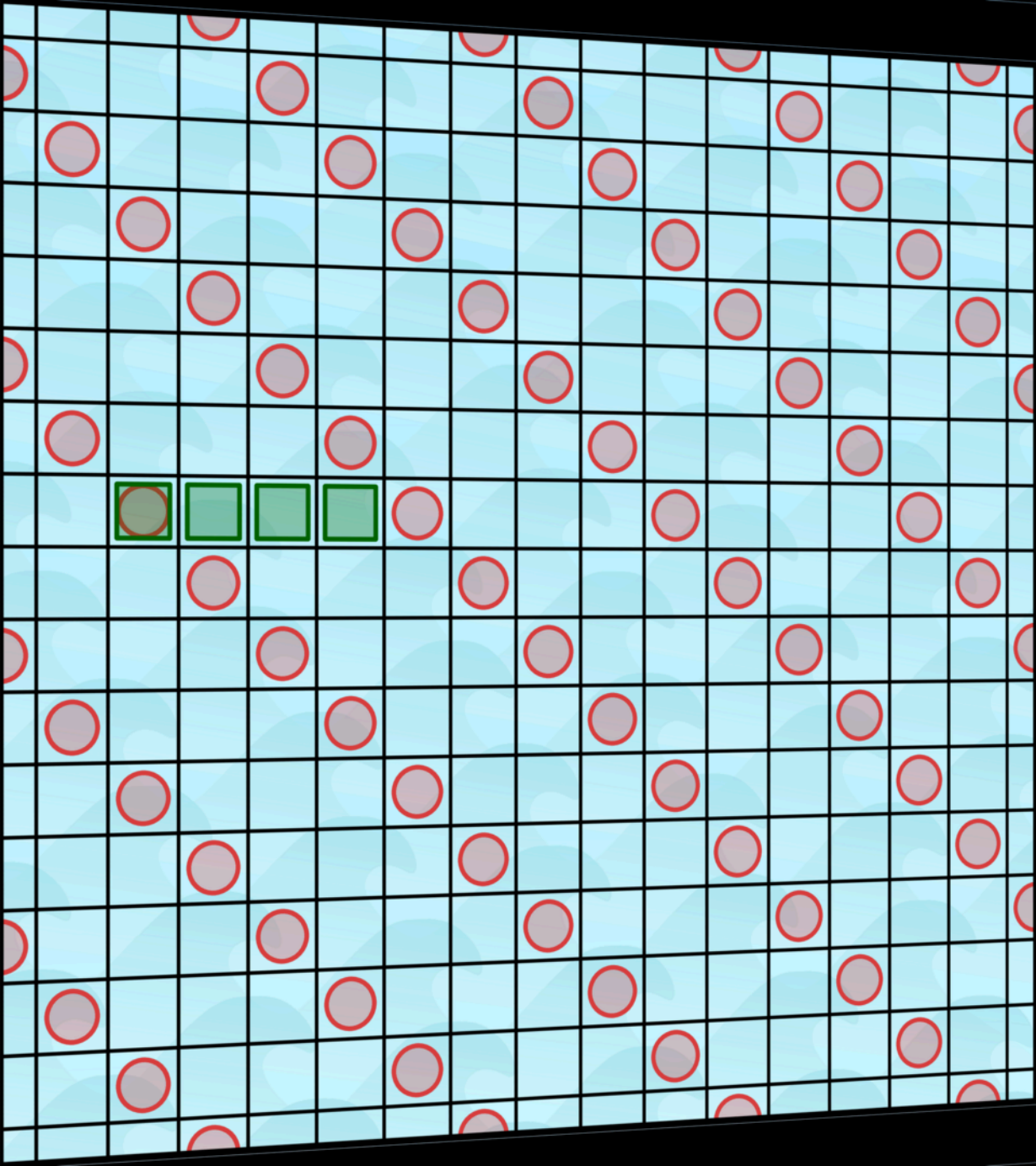
Now, it is easy to see that this pattern is optimal.



Let's consider battleships on an *infinite* board.

What is the best pattern to find $S = \{ \begin{array}{|c|} \hline \square \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \}$?

Now, it is easy to see that this pattern is optimal.

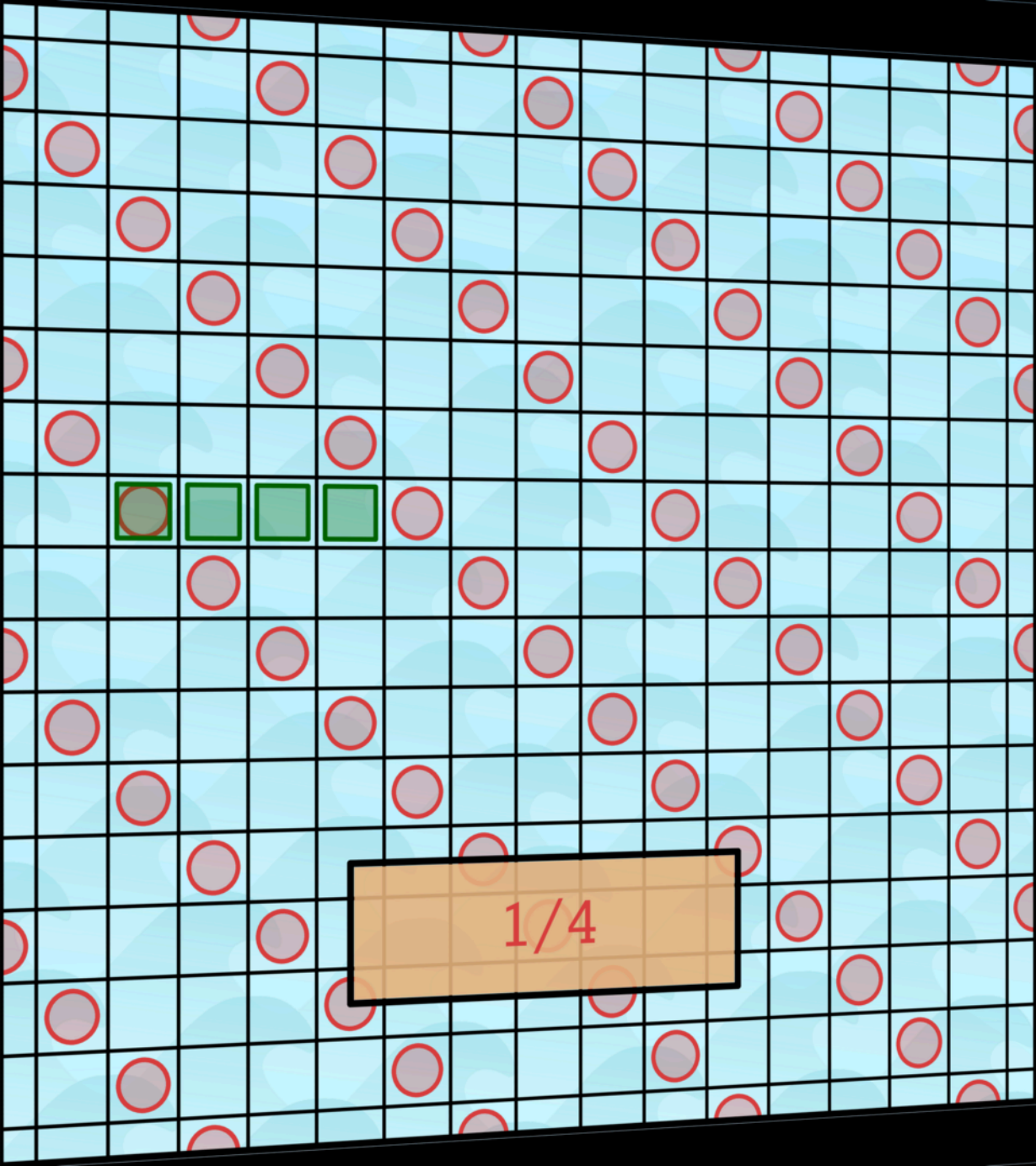


Let's consider battleships on an *infinite* board.

What is the best pattern to find $S = \{ \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array} \}$?

Now, it is easy to see that this pattern is optimal.

The *density* of a solution is the fraction of filled cells.

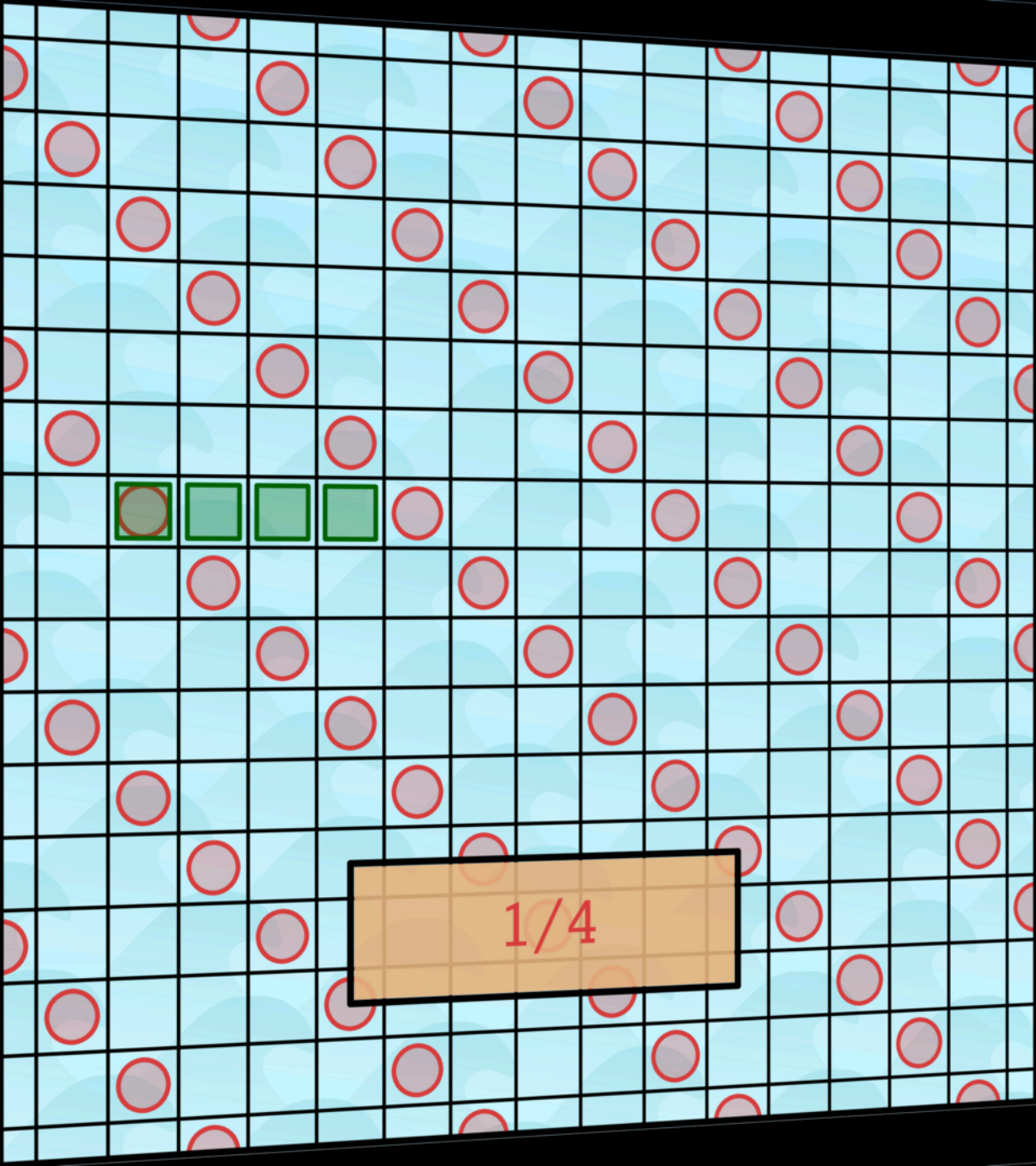


Let's consider battleships on an *infinite* board.

What is the best pattern to find $S = \left\{ \begin{array}{c} \blacksquare \\ \blacksquare \\ \blacksquare \end{array} \right\}, \begin{array}{ccc} \blacksquare & \blacksquare & \blacksquare \end{array} \right\}$?

Now, it is easy to see that this pattern is optimal.

The *density* of a solution is the fraction of filled cells.



1/4

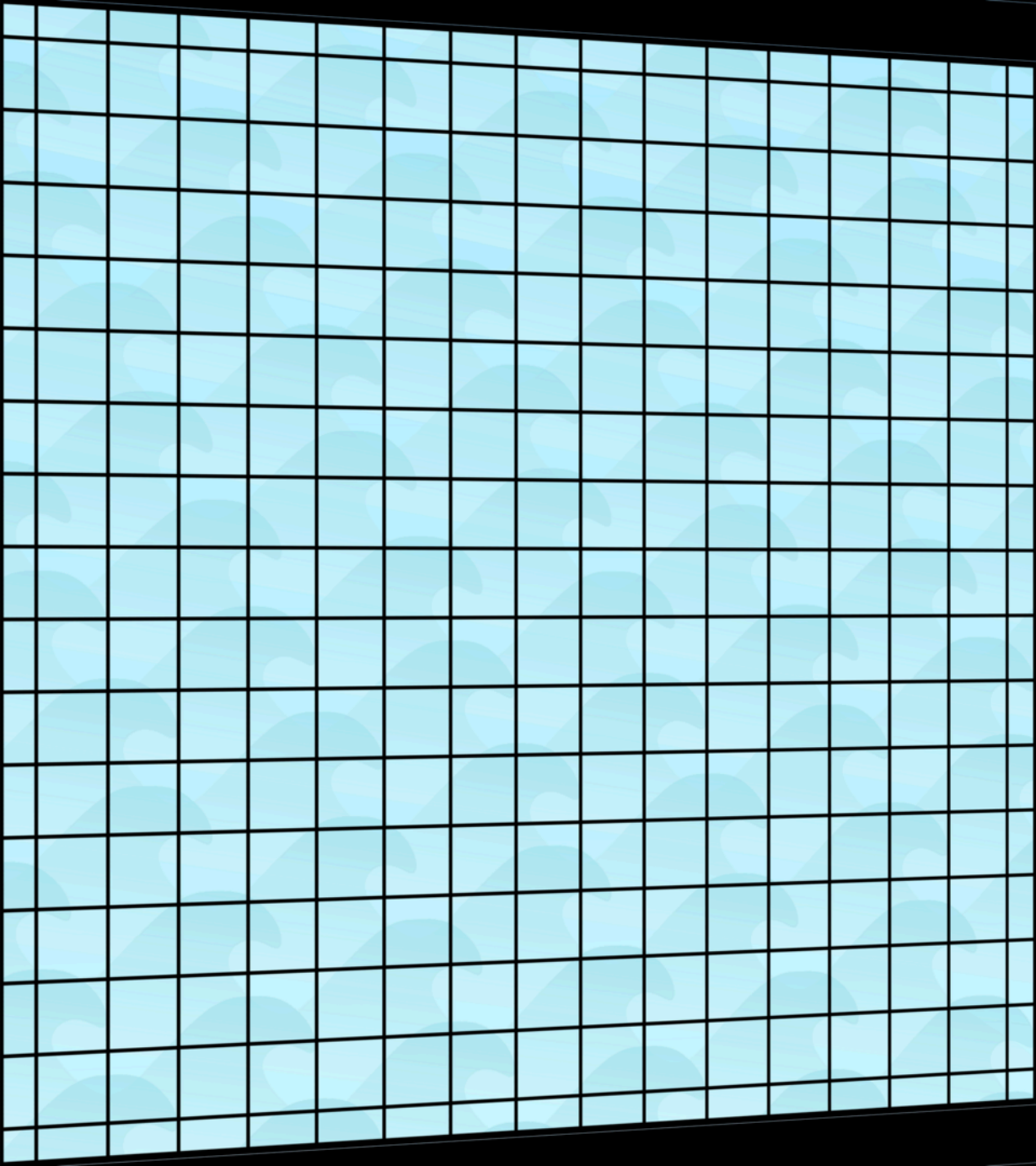
Let's consider battleships on an *infinite* board.

What is the best pattern to find $S = \{ \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array} \}$?

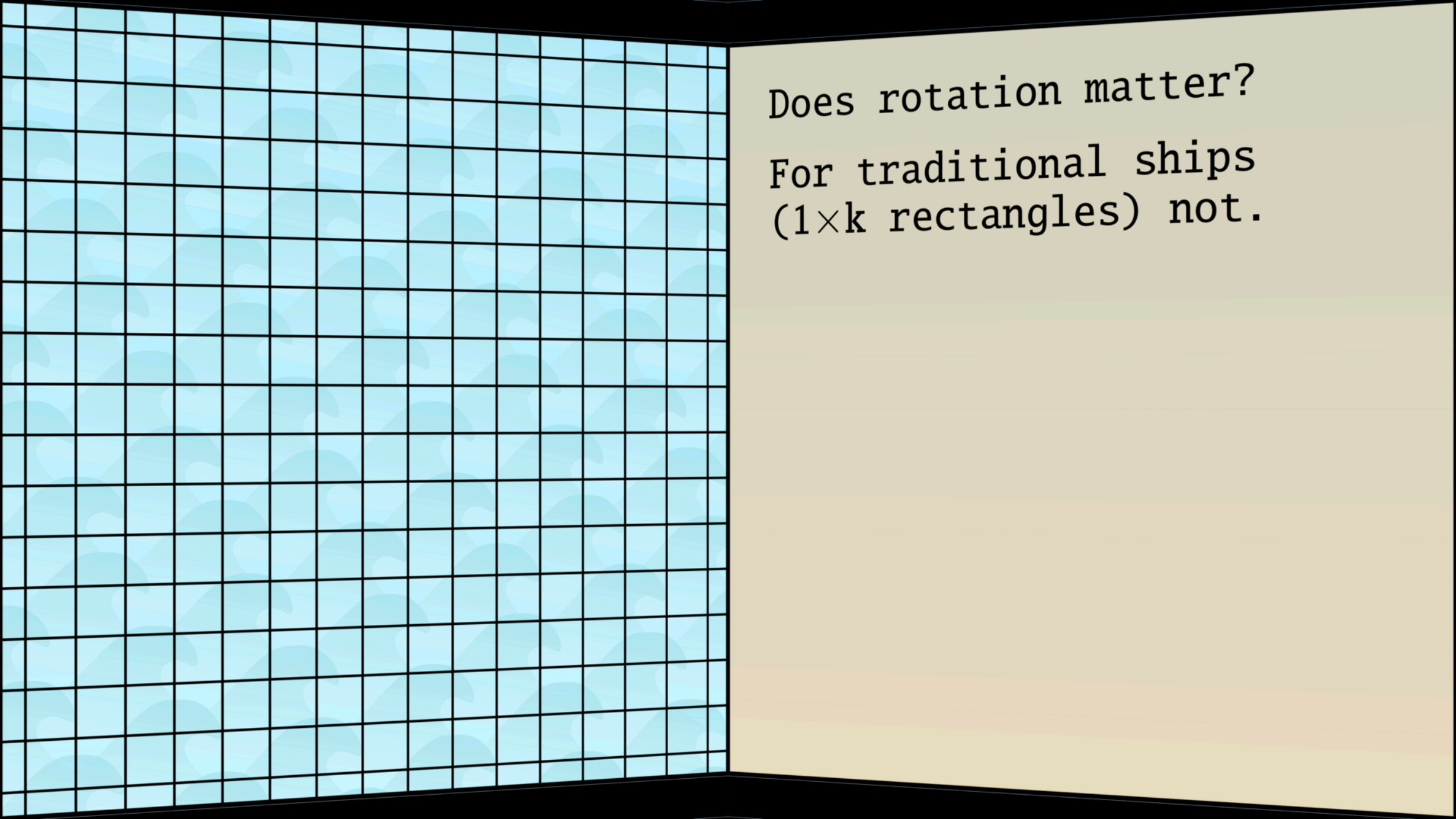
Now, it is easy to see that this pattern is optimal.

The *density* of a solution is the fraction of filled cells.

Note that the *same* pattern works for $S = \{ \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array} \}$!

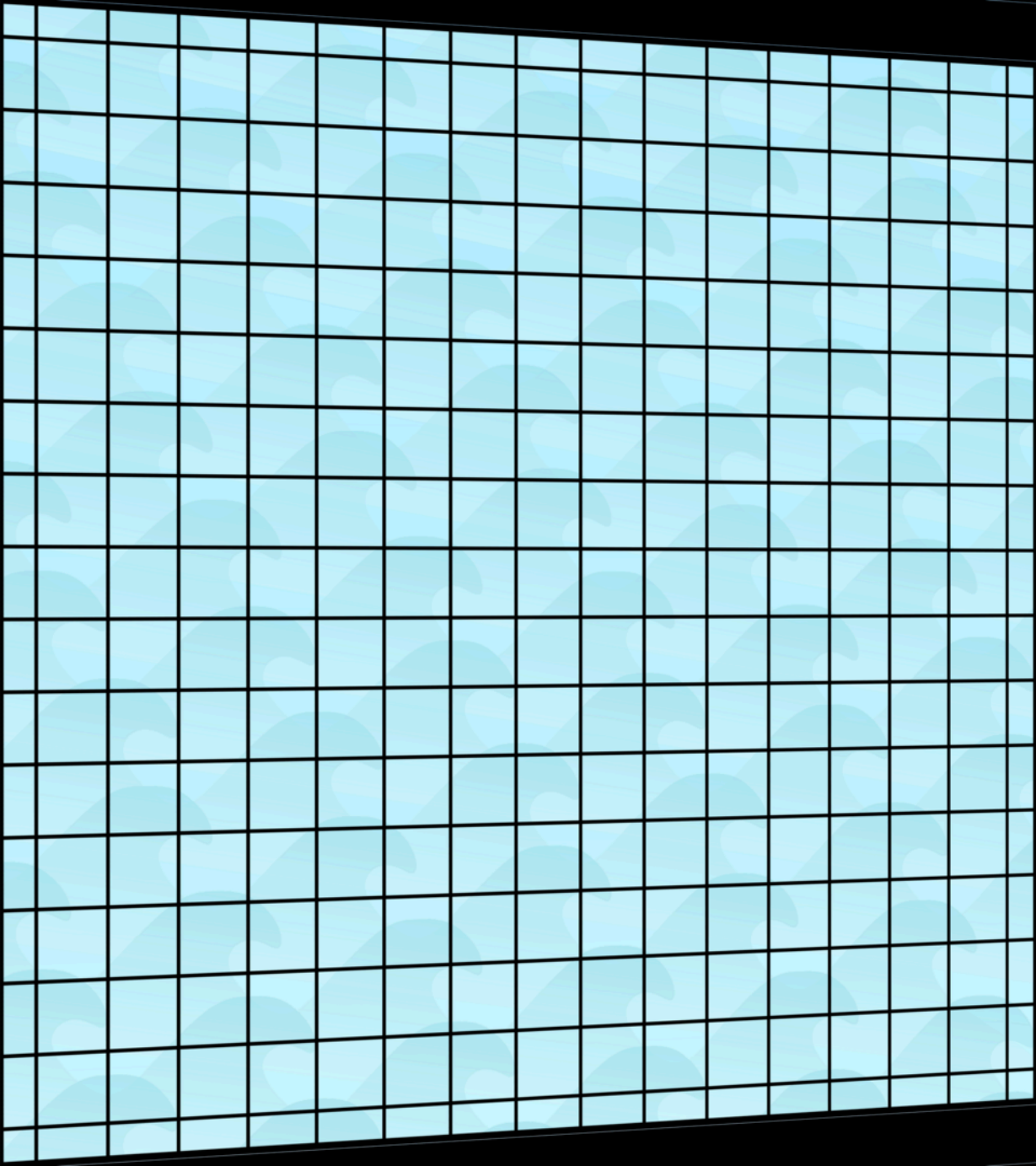


Does rotation matter?



Does rotation matter?

For traditional ships
($1 \times k$ rectangles) not.



Does rotation matter?

For traditional ships
($1 \times k$ rectangles) not.

What about other shapes?

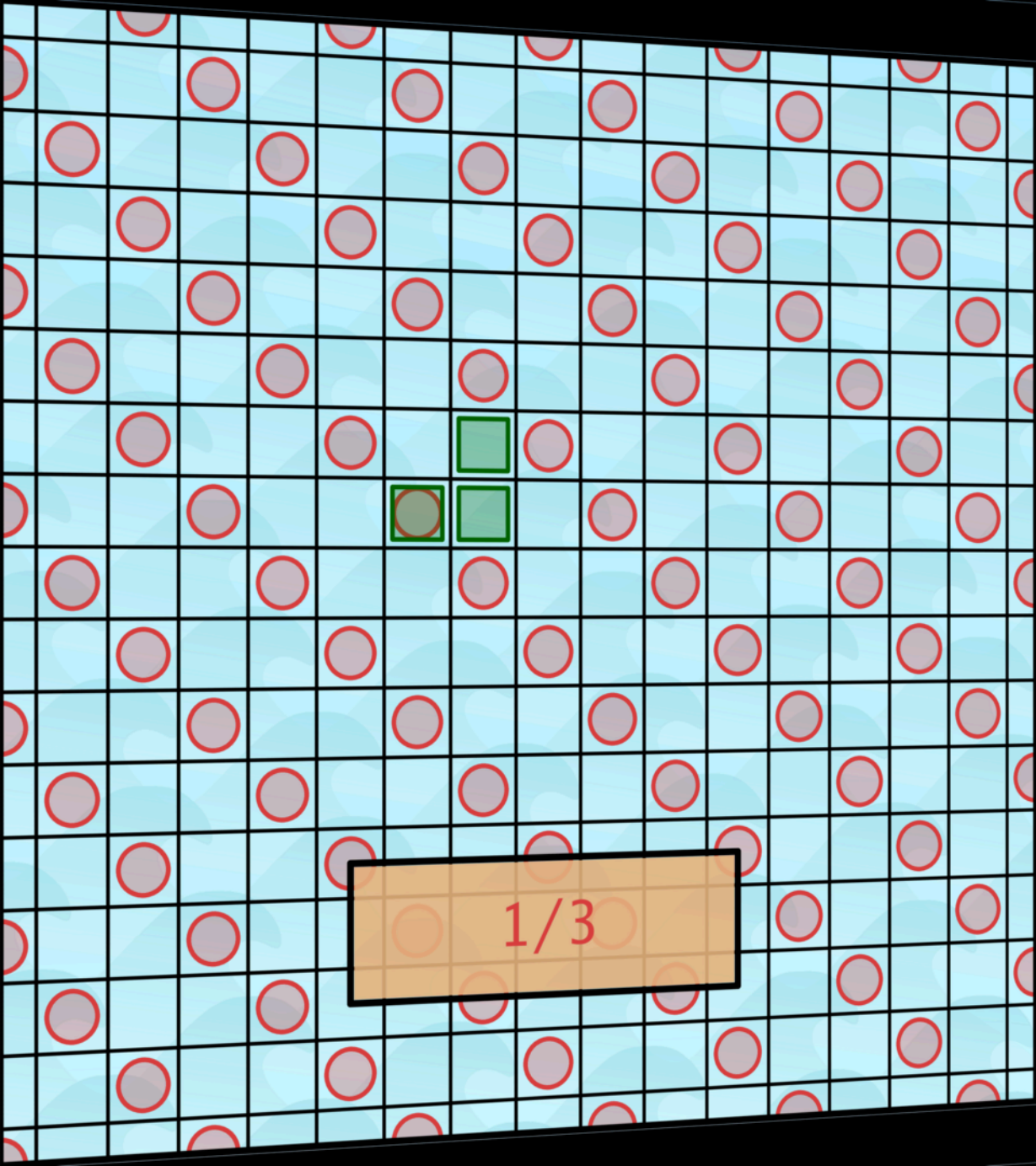
Does rotation matter?

For traditional ships
($1 \times k$ rectangles) not.

What about other shapes?

Say, the L-triomino $S = \{ \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \blacksquare & \blacksquare \\ \hline \end{array} \}$.



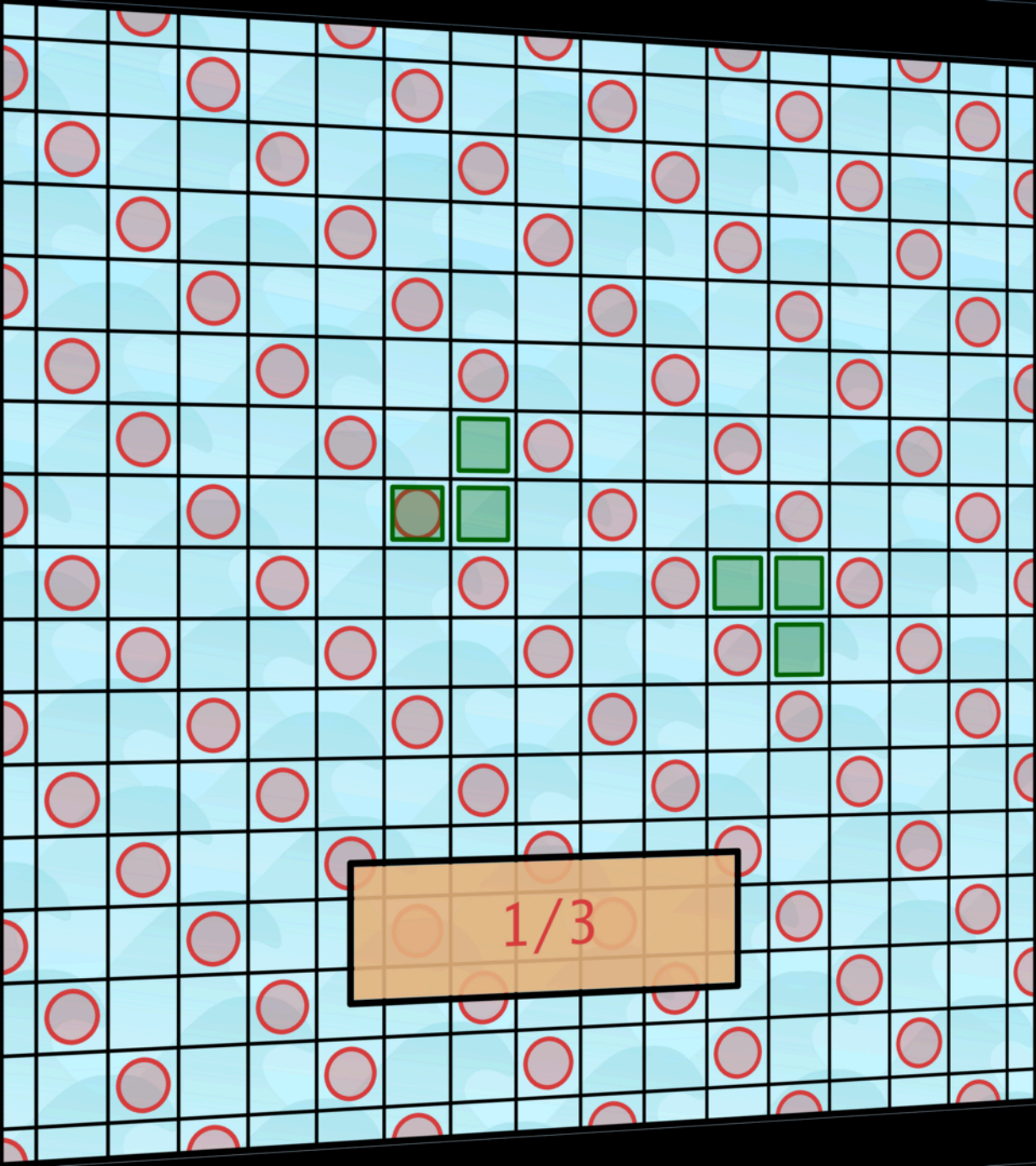


Does rotation matter?

For traditional ships
($1 \times k$ rectangles) not.

What about other shapes?

Say, the L-triomino $S = \left\{ \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \blacksquare & \blacksquare \\ \hline \end{array} \right\}$.

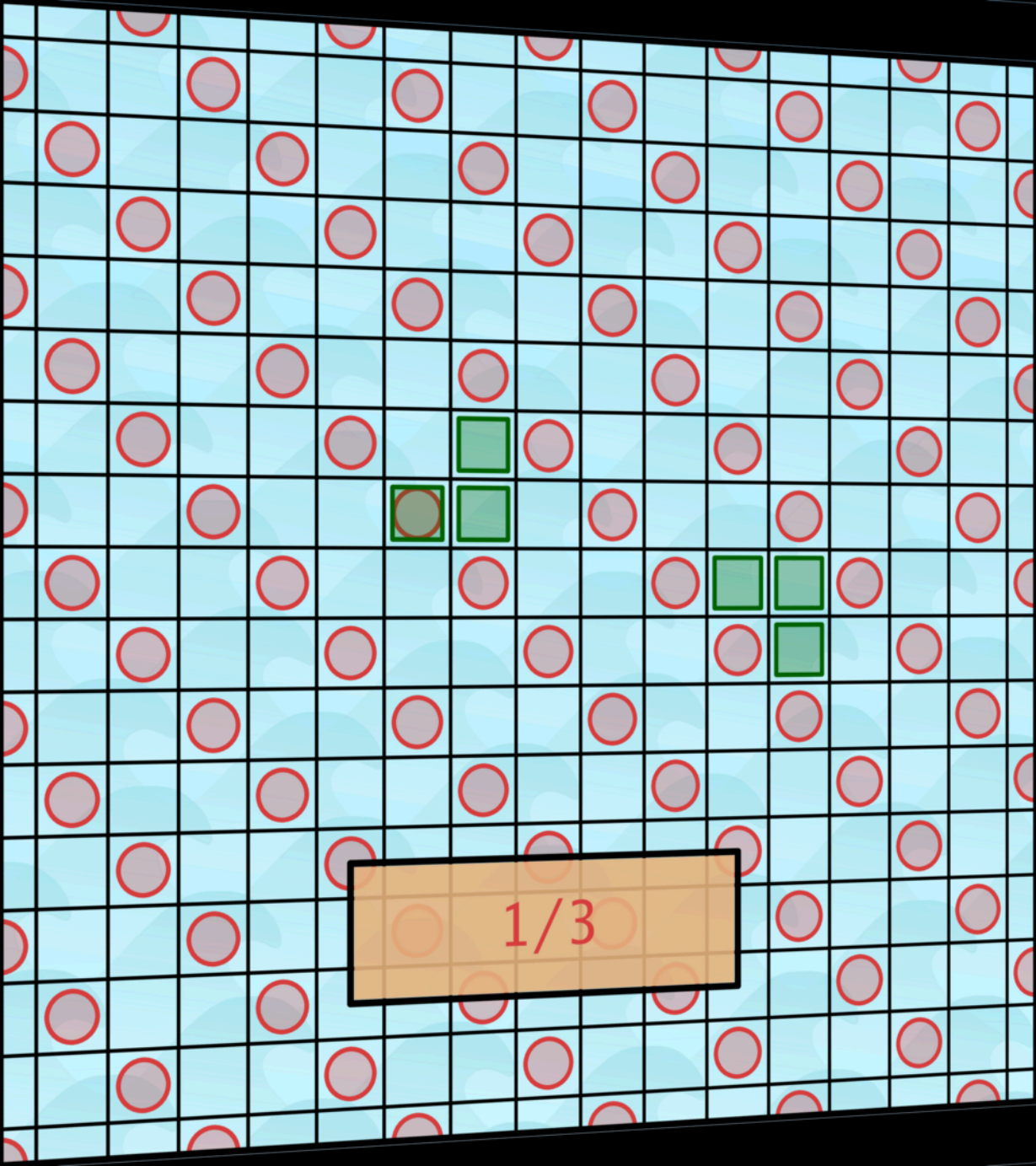


Does rotation matter?

For traditional ships
($1 \times k$ rectangles) not.

What about other shapes?

Say, the L-triomino $S = \left\{ \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \blacksquare & \\ \hline \end{array} \right\}$.



1/3

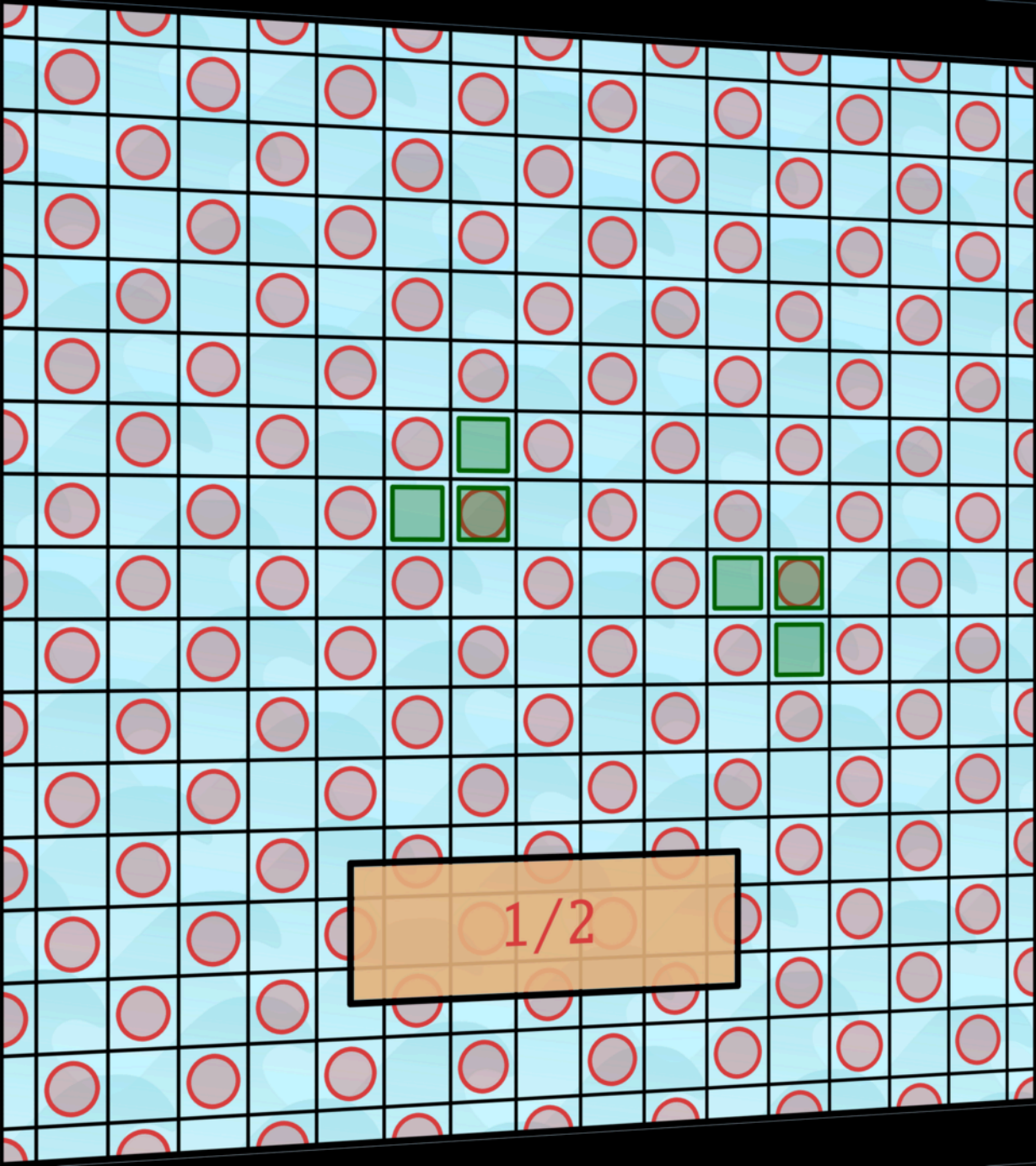
Does rotation matter?

For traditional ships
($1 \times k$ rectangles) not.

What about other shapes?

Say, the L-triomino $S = \{\text{L-shape}\}$.

Not the same as $S = \{\text{L-shape}, \text{rotated L-shape}\}$!



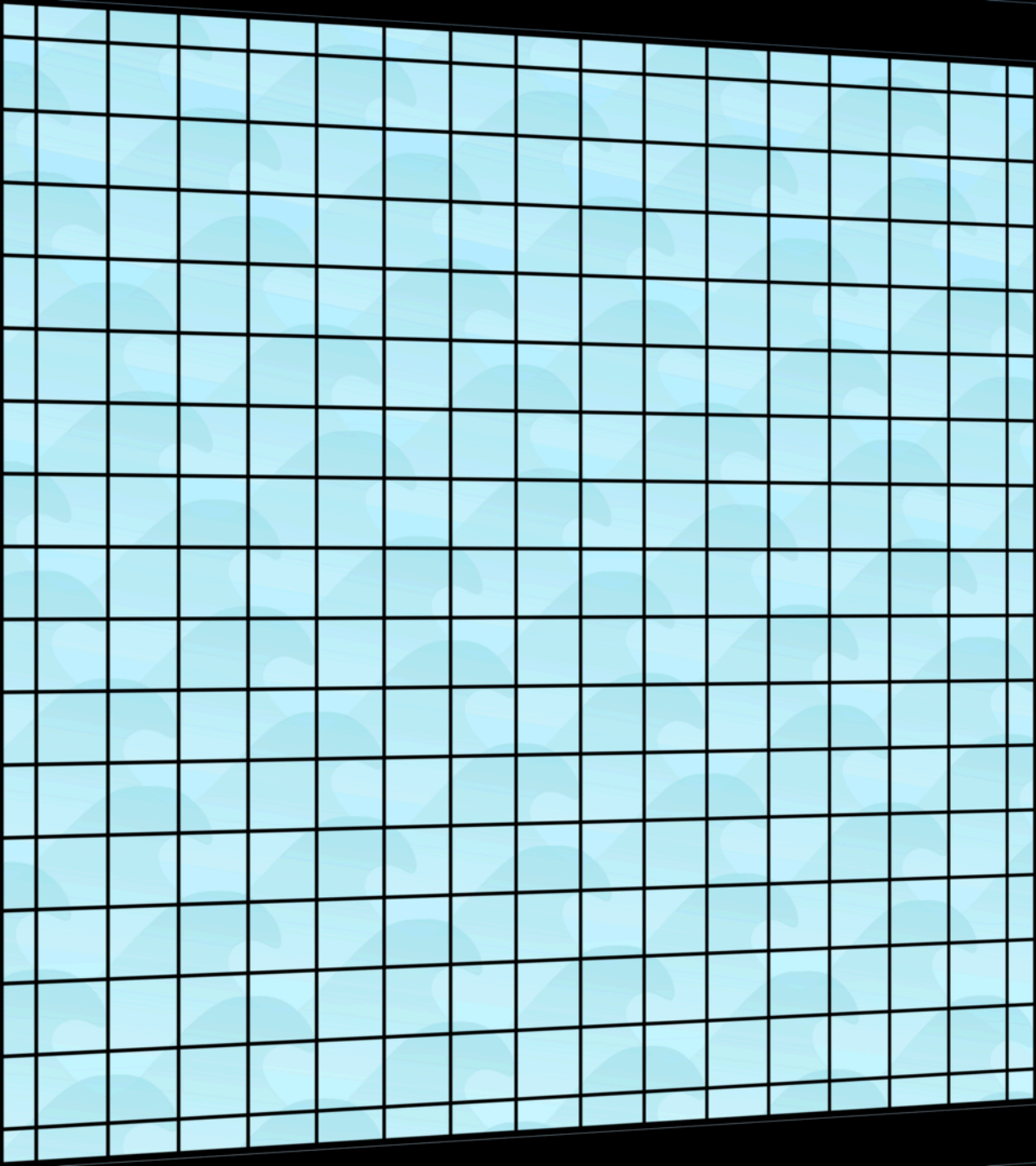
Does rotation matter?

For traditional ships
($1 \times k$ rectangles) not.

What about other shapes?

Say, the L-triomino $S = \{\text{L-shape}\}$.

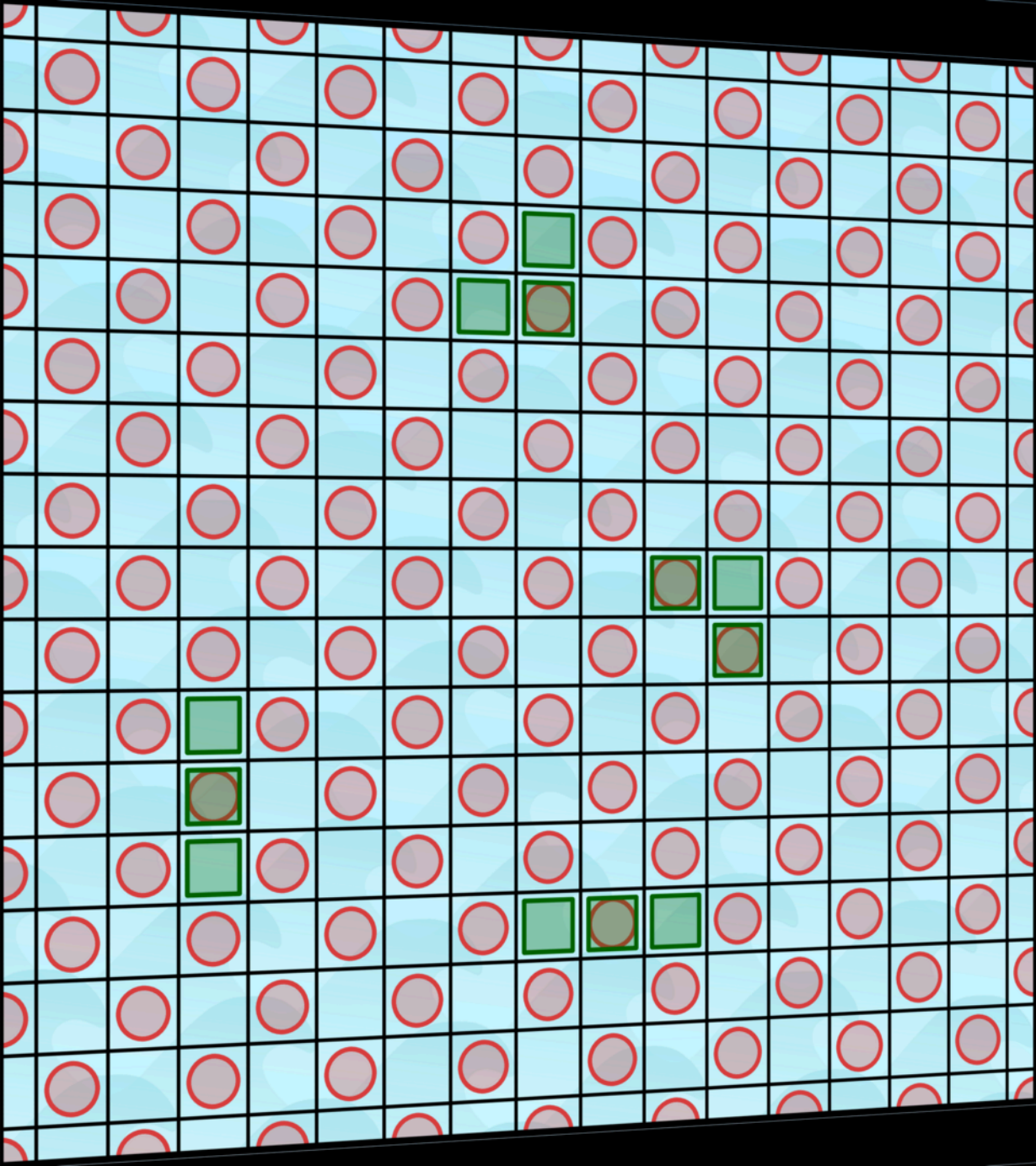
Not the same as $S = \{\text{L-shape}, \text{rotated L-shape}\}$!



Suppose that all we know
is the *size* of the ship.

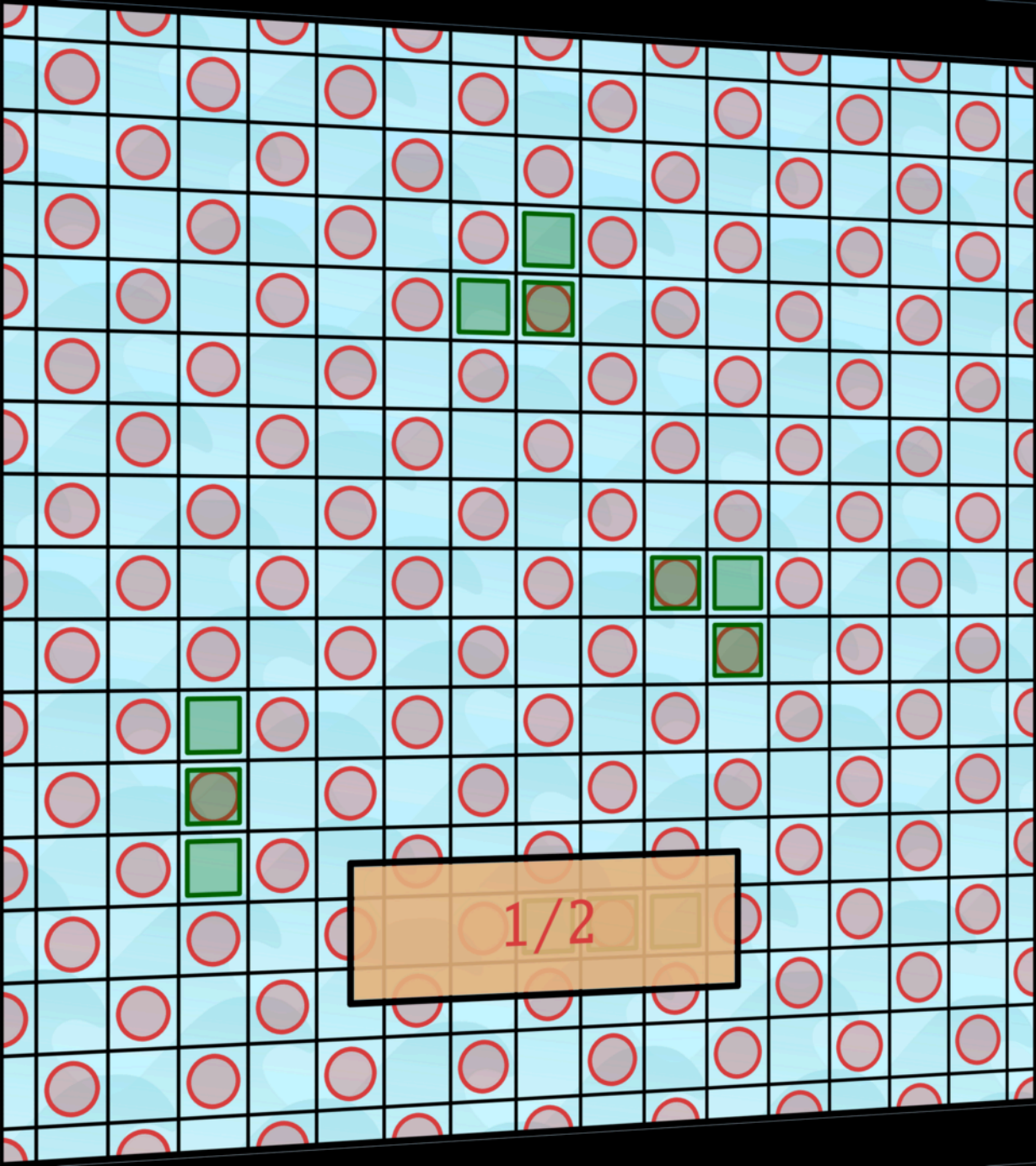
Suppose that all we know
is the *size* of the ship.

$S = \{ \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|} \hline \blacksquare \\ \hline \blacksquare \\ \hline \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array} \}.$



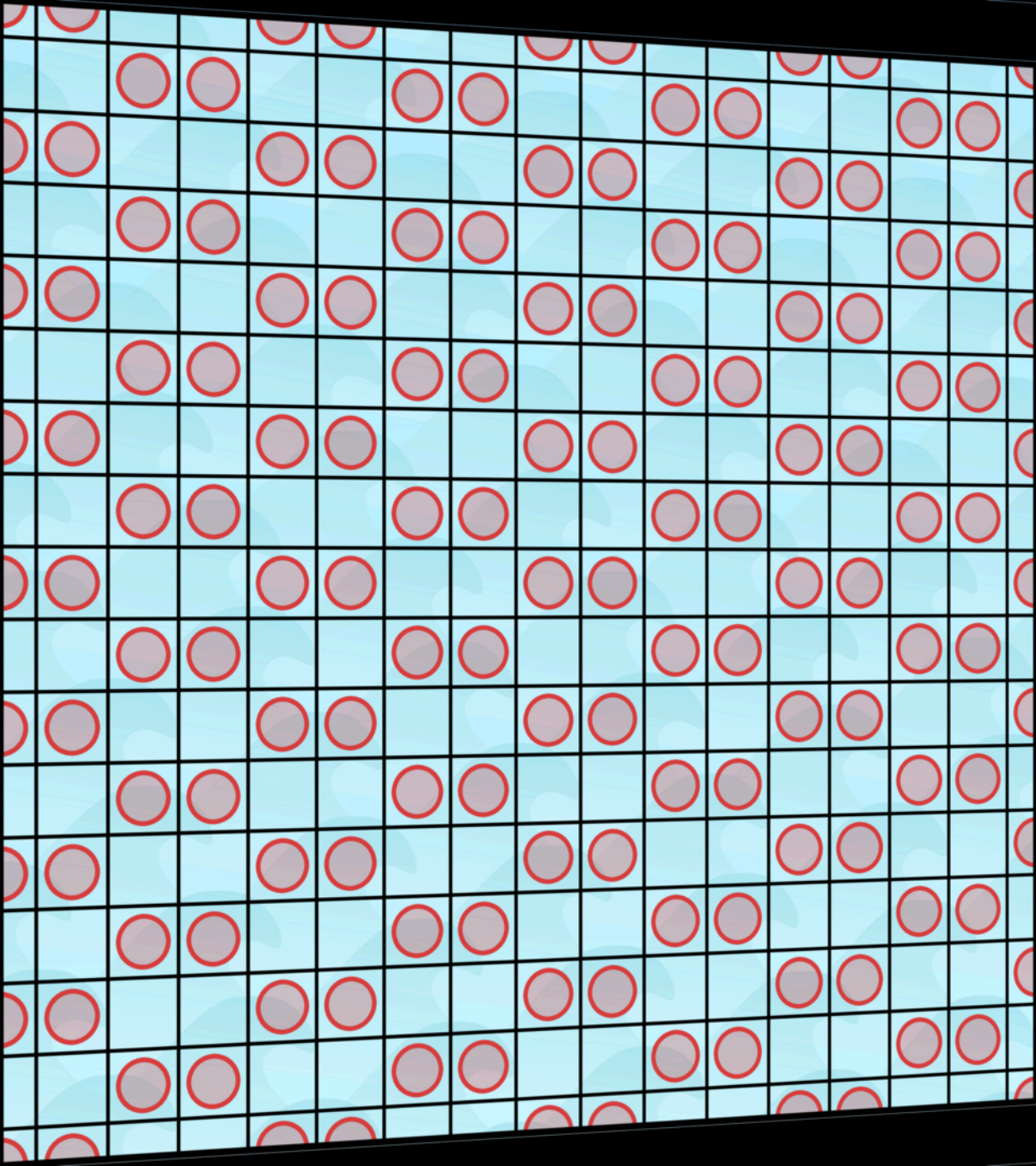
Suppose that all we know
is the *size* of the ship.

$S = \{ \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}, \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \end{array} \}.$



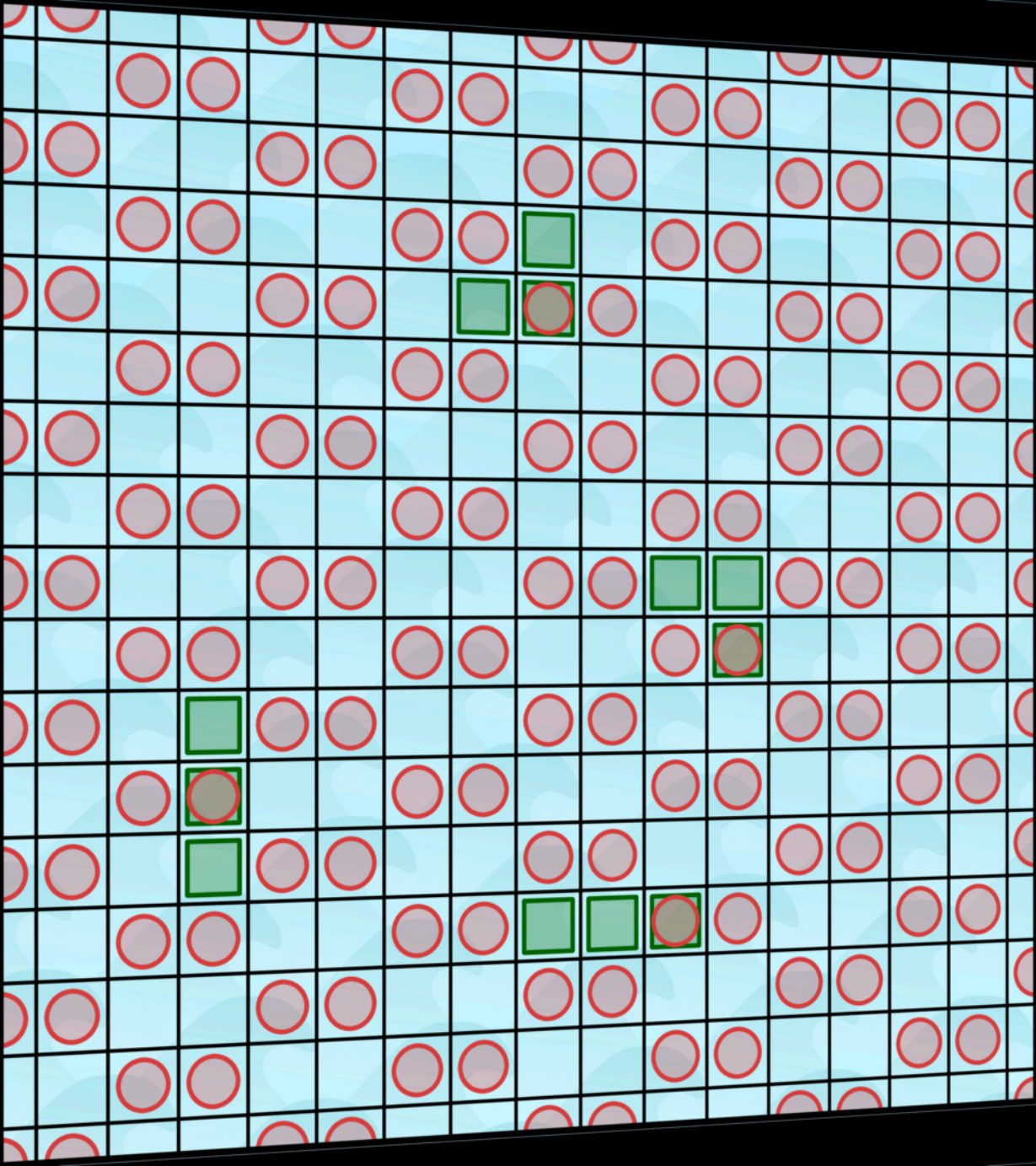
Suppose that all we know is the *size* of the ship.

$$S = \{ \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|} \hline \blacksquare \\ \hline \blacksquare \\ \hline \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array} \}.$$



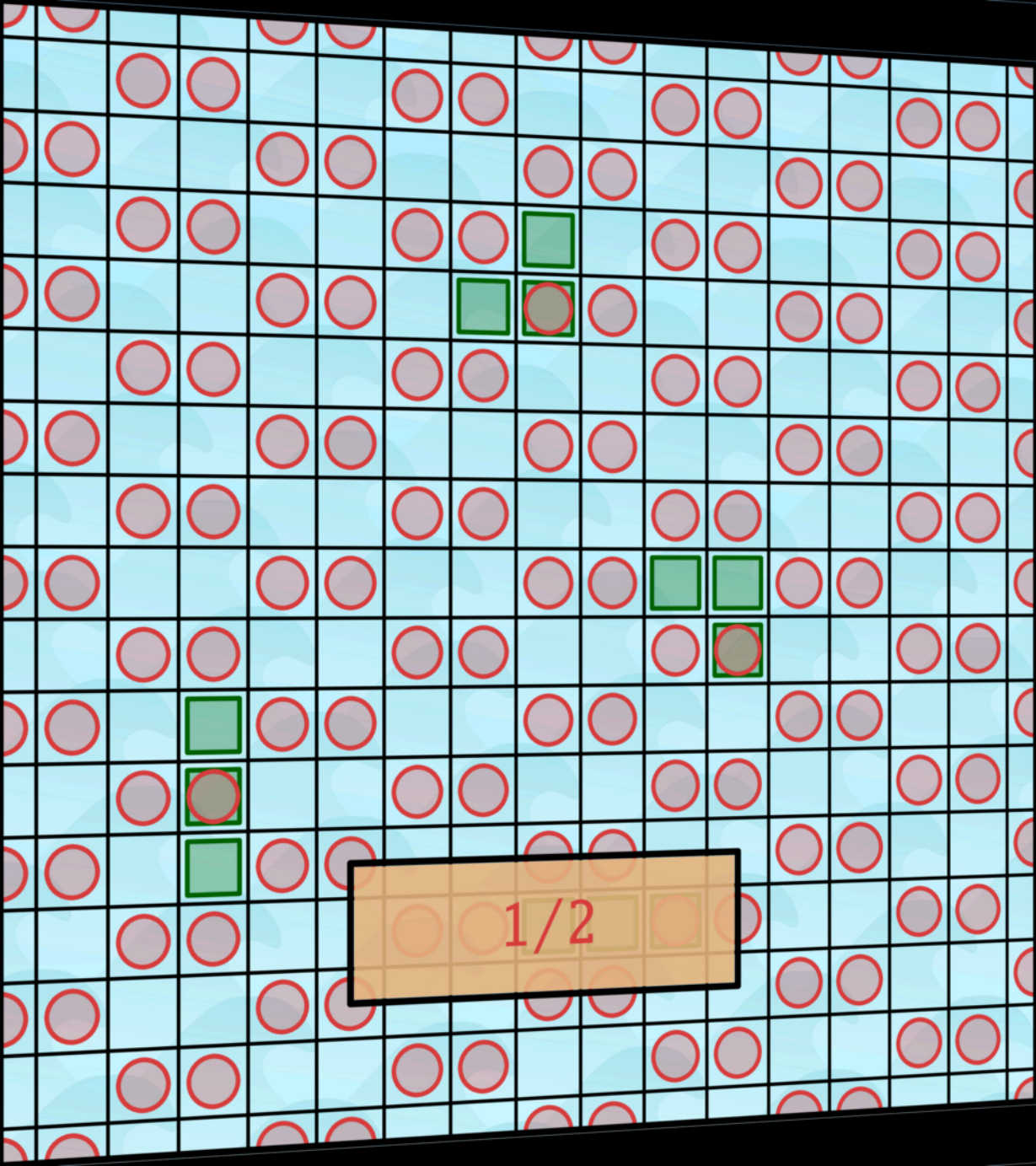
Suppose that all we know
is the *size* of the ship.

$$S = \{ \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|} \hline \blacksquare \\ \hline \blacksquare \\ \hline \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array} \}.$$



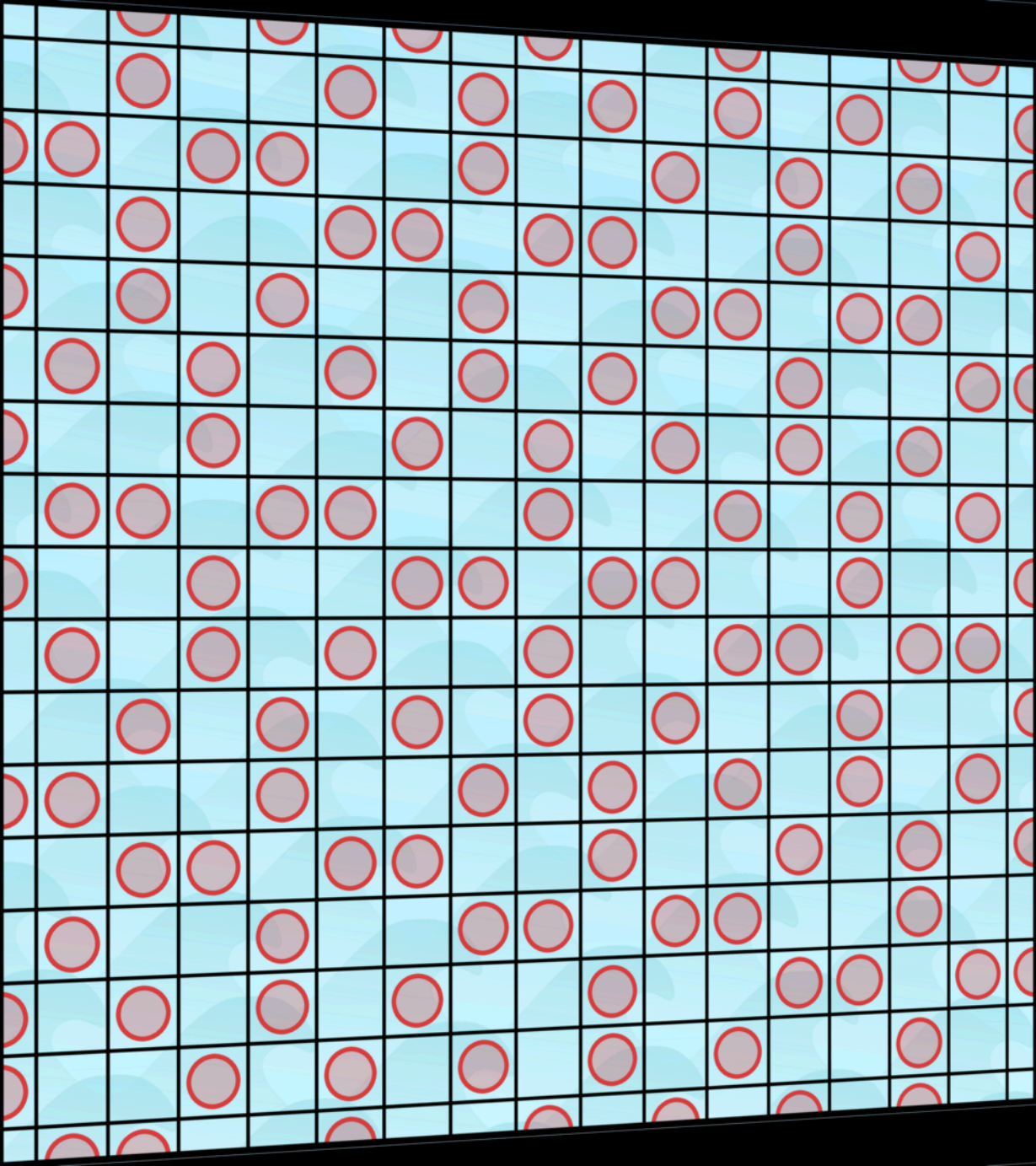
Suppose that all we know
is the *size* of the ship.

$S = \{ \begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array}, \begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array}, \begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array}, \begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \}.$



Suppose that all we know is the *size* of the ship.

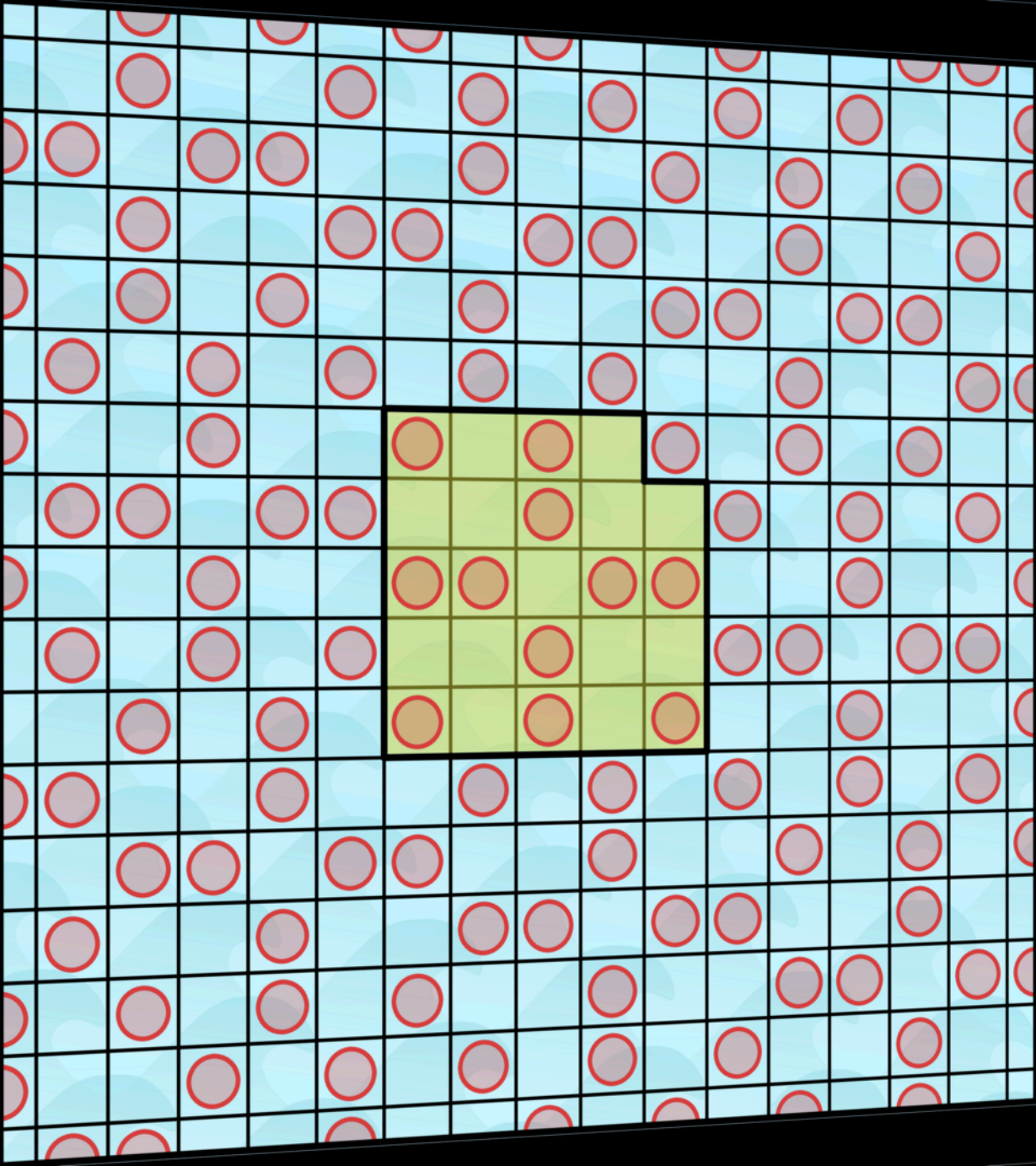
$$S = \{ \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array}, \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array}, \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array}, \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array} \}.$$



Suppose that all we know is the *size* of the ship.

$S = \{ \text{2x1}, \text{1x2}, \text{2x2}, \text{1x3}, \text{3x1}, \text{1x4} \}.$

$S = \{ \text{2x2}, \text{2x3}, \text{3x2}, \text{3x3}, \text{3x4}, \text{4x3}, \text{4x4}, \text{4x5}, \text{5x4}, \text{5x5}, \text{5x6}, \text{6x5}, \text{6x6}, \text{6x7}, \text{7x6}, \text{7x7}, \text{7x8}, \text{8x7}, \text{8x8}, \text{8x9}, \text{9x8}, \text{9x9} \}.$

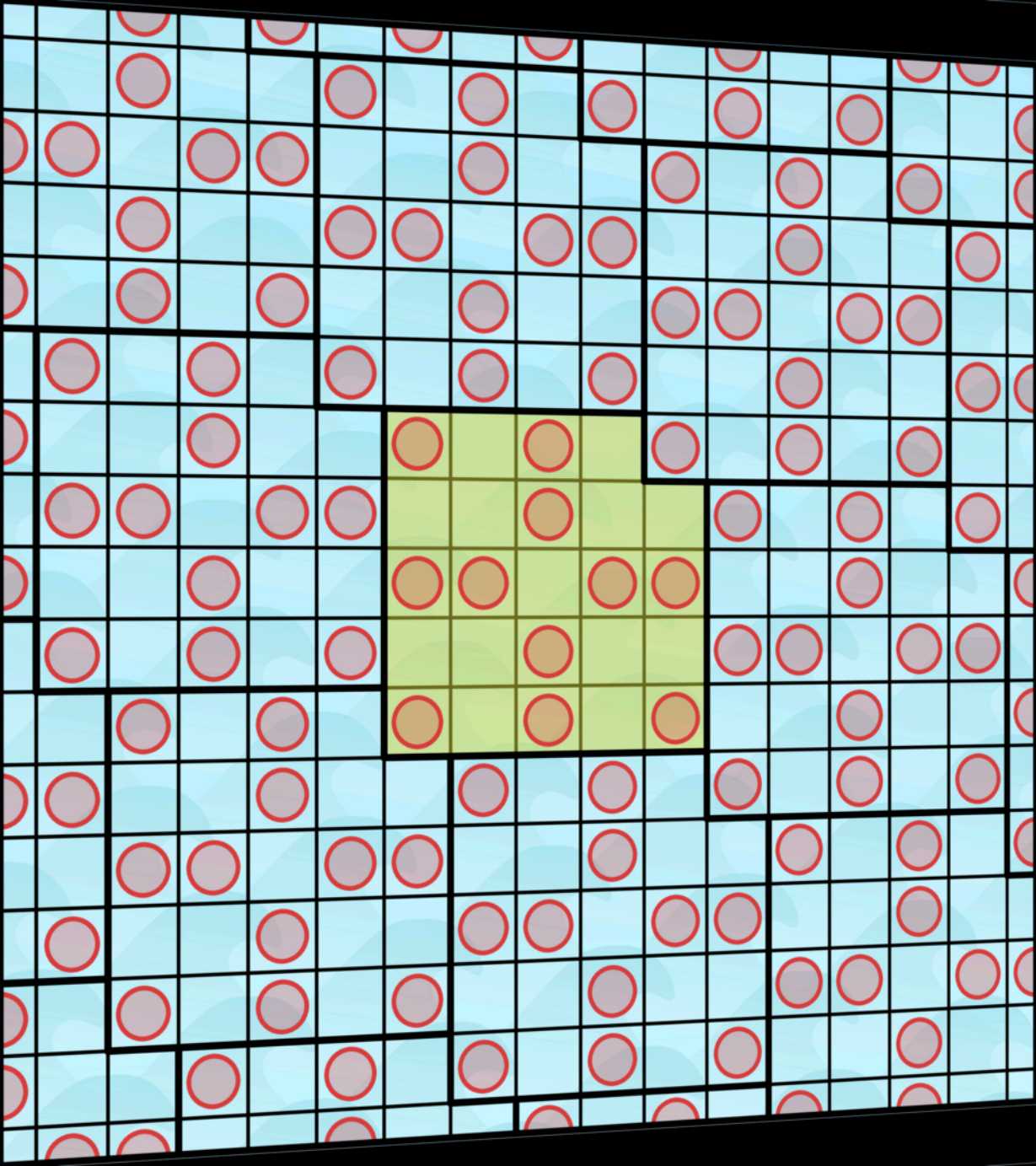


Suppose that all we know is the *size* of the ship.

$$S = \{ \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array} \}.$$

$$S = \{ \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array}, \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array} \}.$$

What is the density of this pattern?

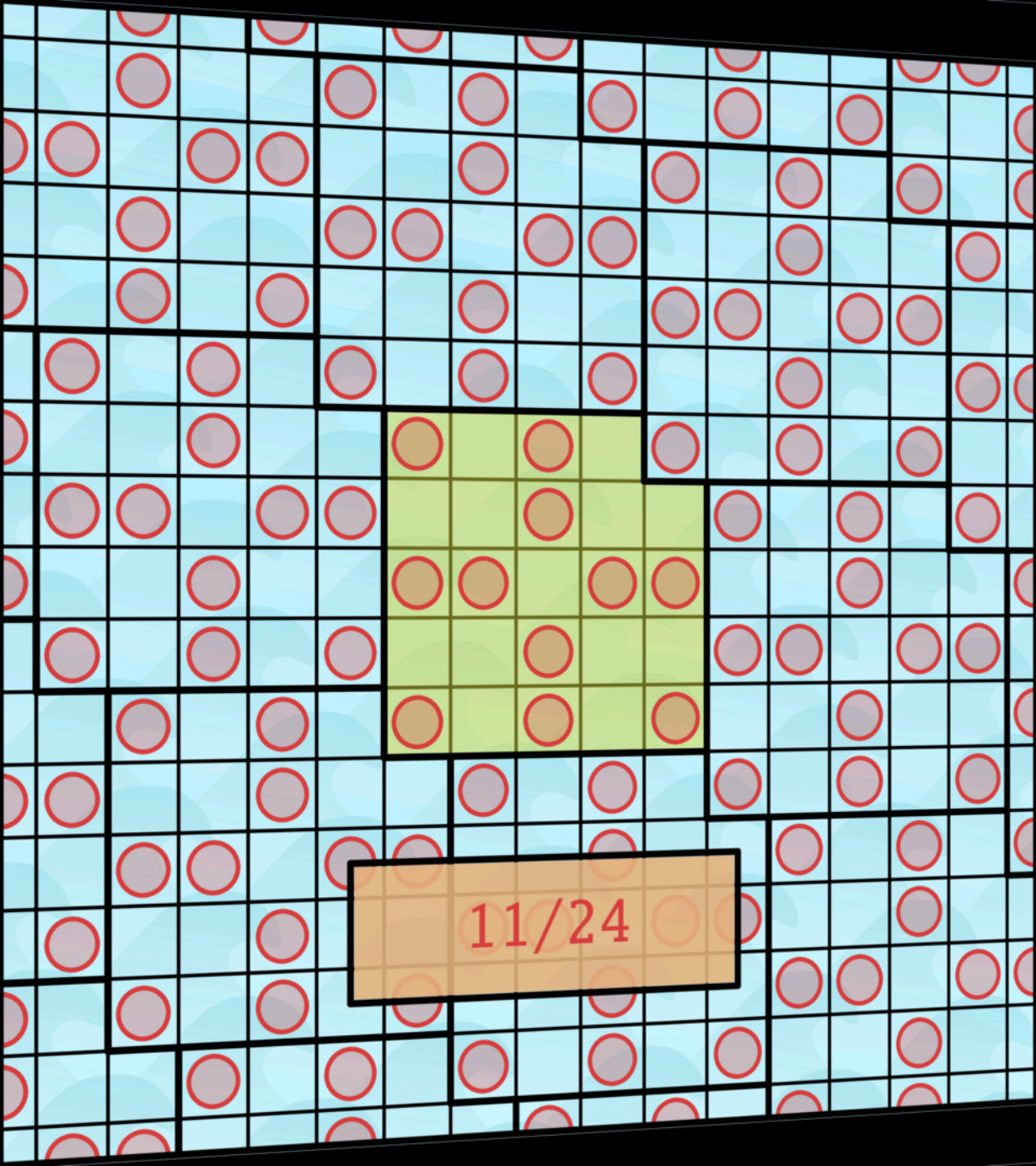


Suppose that all we know is the *size* of the ship.

$S = \{ \text{[2x2 square]}, \text{[3x1 rectangle]}, \text{[1x3 rectangle]}, \text{[2x2 square]}, \text{[3x1 rectangle]}, \text{[1x3 rectangle]} \}$.

$S = \{ \text{[2x2 square]}, \text{[3x1 rectangle]}, \text{[1x3 rectangle]}, \text{[2x2 square]}, \text{[3x1 rectangle]}, \text{[1x3 rectangle]}, \text{[2x2 square]}, \text{[3x1 rectangle]}, \text{[1x3 rectangle]}, \text{[2x2 square]}, \text{[3x1 rectangle]}, \text{[1x3 rectangle]}, \text{[2x2 square]}, \text{[3x1 rectangle]}, \text{[1x3 rectangle]} \}$.

What is the density of this pattern?



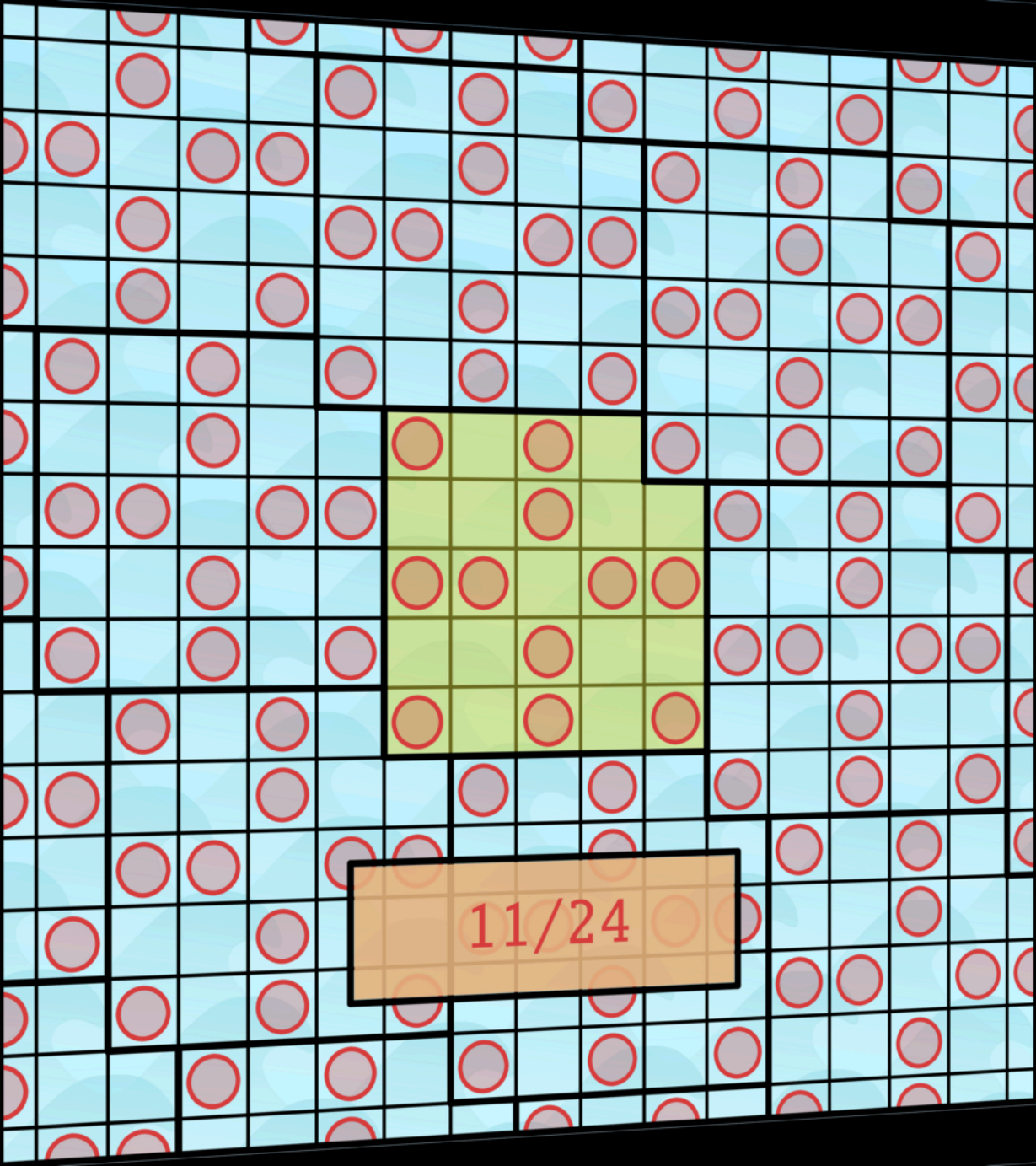
11/24

Suppose that all we know is the *size* of the ship.

$$S = \{ \text{L-shape}, \text{2x2 square}, \text{3x1 horizontal}, \text{3x1 vertical}, \text{1x3 horizontal}, \text{1x3 vertical} \}.$$

$$S = \{ \text{2x2 square}, \text{3x2 horizontal}, \text{3x2 vertical}, \text{4x1 horizontal}, \text{4x1 vertical}, \text{1x4 horizontal}, \text{1x4 vertical}, \text{2x3 horizontal}, \text{2x3 vertical}, \text{3x3 square}, \text{1x5 horizontal}, \text{1x5 vertical} \}.$$

What is the density of this pattern?



11/24

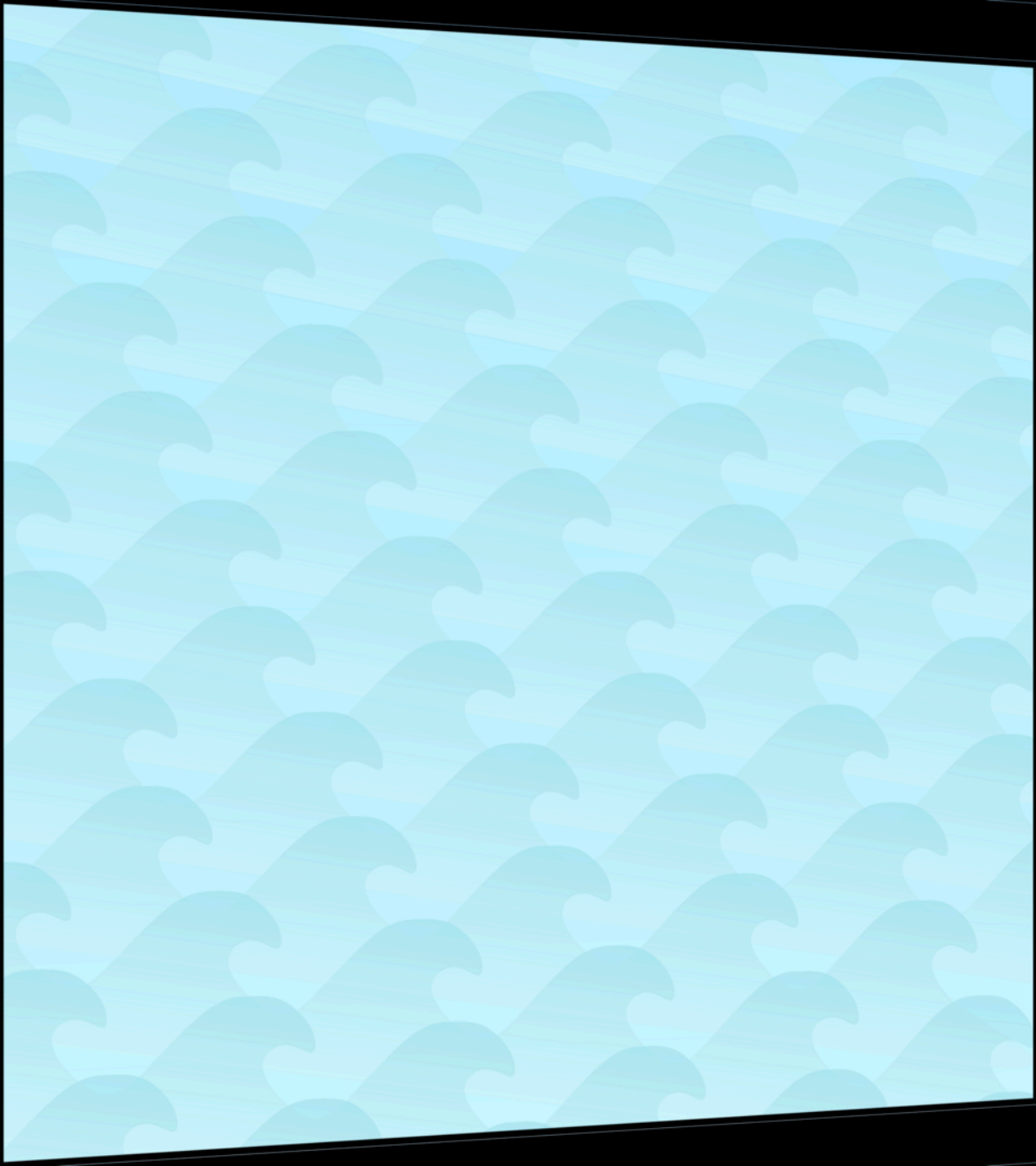
Suppose that all we know is the *size* of the ship.

$$S = \{ \text{2x2}, \text{2x3}, \text{3x2}, \text{3x3}, \text{1x4}, \text{4x1} \}.$$

$$S = \{ \text{2x2}, \text{2x3}, \text{3x2}, \text{3x3}, \text{3x4}, \text{4x3}, \text{4x4}, \text{1x5}, \text{5x1}, \text{2x4}, \text{4x2}, \text{3x5}, \text{5x3}, \text{4x5}, \text{5x4} \}.$$

What is the density of this pattern?

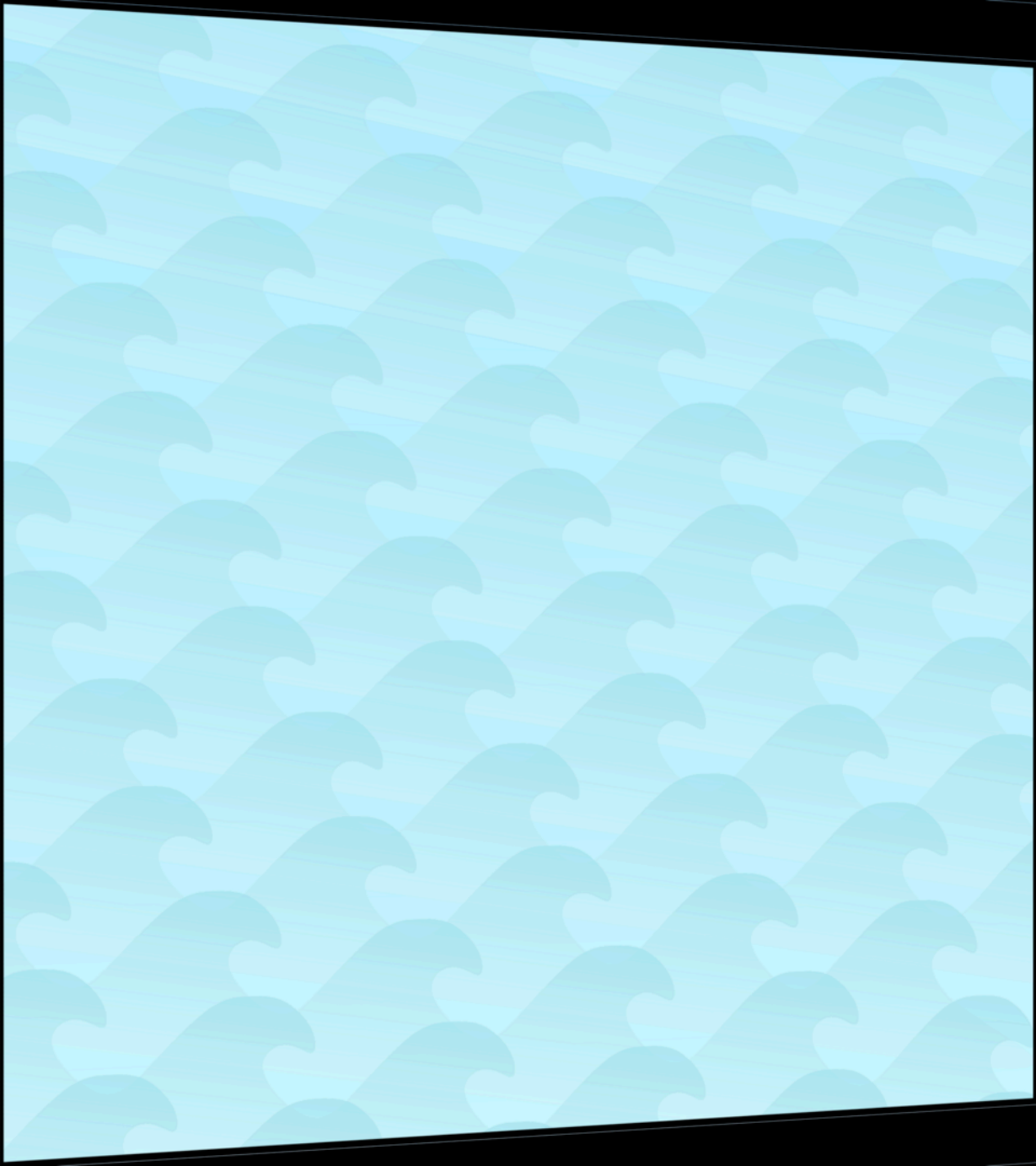
Is this optimal?



4

CHAPTER

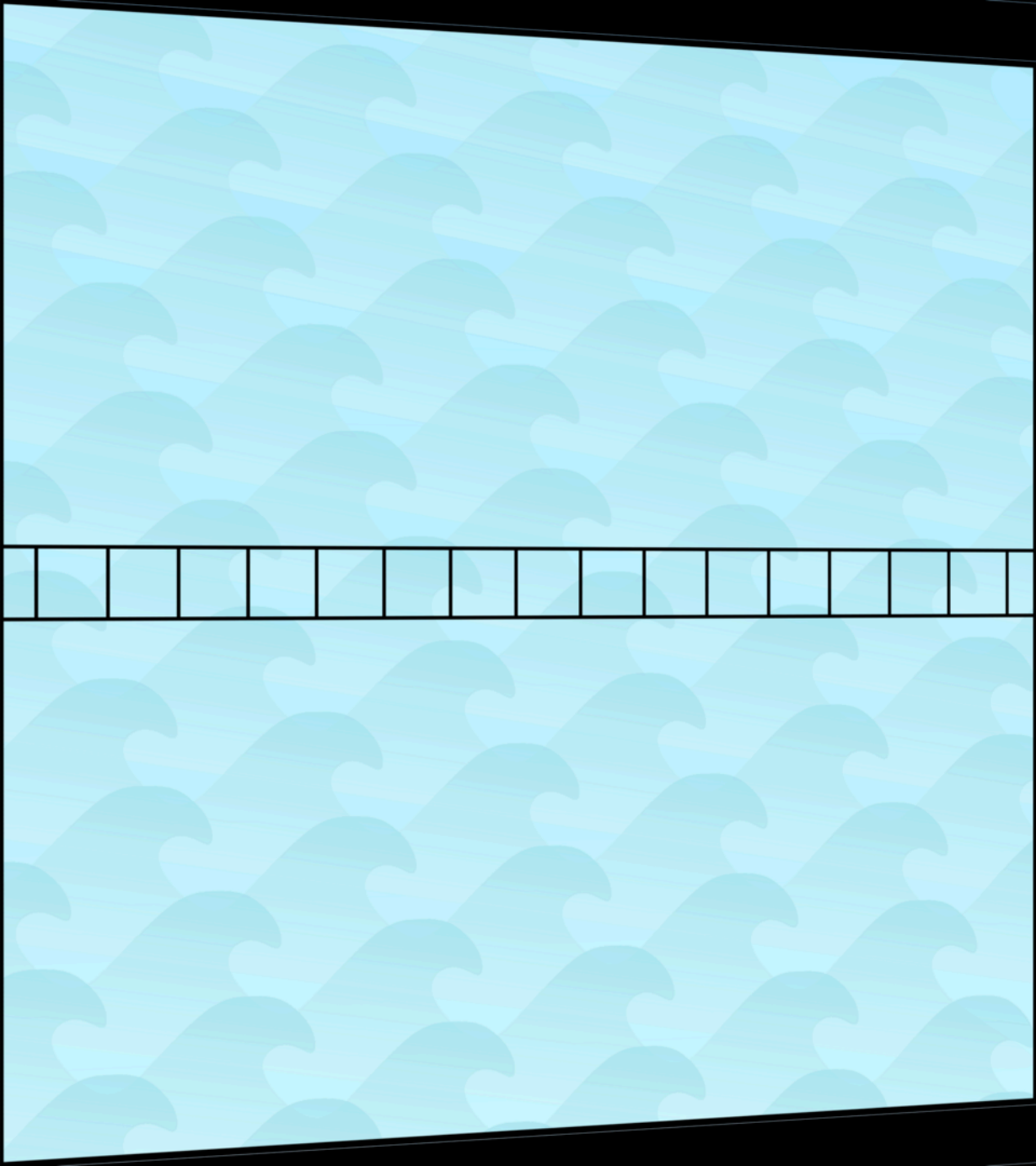
1-DIMENSIONAL SHIPS



Consider an infinite
1-dimensional grid.

Consider an infinite
1-dimensional grid.





Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k\dots$

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...



Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...



$$1/5$$

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.



$1/5$

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.



$1/5$

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k\dots$

\dots if they are *connected*.

Disconnected ships behave
in less predictable ways.



Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Consider an infinite
1-dimensional grid.

Ships of length k can be
hit with density $1/k$...

...if they are *connected*.

Disconnected ships behave
in less predictable ways.

2/5

Ships $[0, a, b]$ with a and b
coprime.

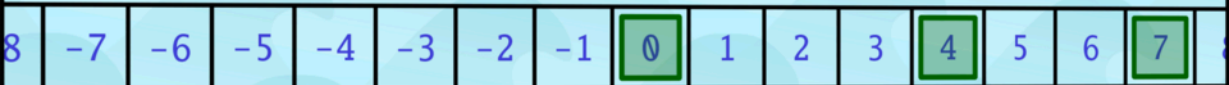
8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8
---	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---

Ships $[0, a, b]$ with a and b
coprime.

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

Ships $[0, a, b]$ with a and b coprime.

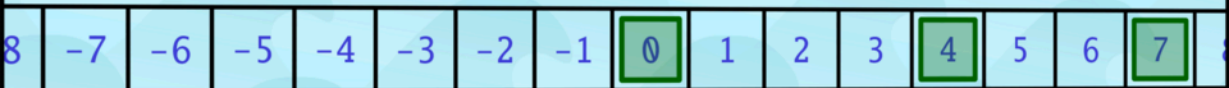
We can reuse our understanding about 2-dimensional ships!



Ships $[0, a, b]$ with a and b coprime.

We can reuse our understanding about 2-dimensional ships!

Consider the map
 $(x, y) \mapsto ay + bx$.



81	-77	-73	-69	-65	-61	-57	-53	-49	-45	-41	-37	-33	-29	-25	-21	-17
74	-70	-66	-62	-58	-54	-50	-46	-42	-38	-34	-30	-26	-22	-18	-14	-10
67	-63	-59	-55	-51	-47	-43	-39	-35	-31	-27	-23	-19	-15	-11	-7	-3
60	-56	-52	-48	-44	-40	-36	-32	-28	-24	-20	-16	-12	-8	-4	0	4
53	-49	-45	-41	-37	-33	-29	-25	-21	-17	-13	-9	-5	-1	3	7	11
46	-42	-38	-34	-30	-26	-22	-18	-14	-10	-6	-2	2	6	10	14	18
39	-35	-31	-27	-23	-19	-15	-11	-7	-3	1	5	9	13	17	21	25
32	-28	-24	-20	-16	-12	-8	-4	0	4	8	12	16	20	24	28	32
25	-21	-17	-13	-9	-5	-1	3	7	11	15	19	23	27	31	35	39
18	-14	-10	-6	-2	2	6	10	14	18	22	26	30	34	38	42	46
11	-7	-3	1	5	9	13	17	21	25	29	33	37	41	45	49	53
4	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
-3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67
-10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70	74
-17	21	25	29	33	37	41	45	49	53	57	61	65	69	73	77	81
-24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88

Ships $[0, a, b]$ with a and b coprime.

We can reuse our understanding about 2-dimensional ships!

Consider the map $(x, y) \mapsto ay + bx$.

81	-77	-73	-69	-65	-61	-57	-53	-49	-45	-41	-37	-33	-29	-25	-21	-17	-13	-9	-5	-1	3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71	75	79	83	87	91	95	99																																																																																																																																																																																																																														
74	-70	-66	-62	-58	-54	-50	-46	-42	-38	-34	-30	-26	-22	-18	-14	-10	-6	-2	2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70	74	78	82	86	90	94	98	102	106	110	114	118	122																																																																																																																																																																																																																										
67	-63	-59	-55	-51	-47	-43	-39	-35	-31	-27	-23	-19	-15	-11	-7	-3	1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69	73	77	81	85	89	93	97	101	105	109	113	117	121	125	129	133	137	141	145	149	153	157	161	165	169	173	177	181	185	189	193	197	201	205	209	213	217	221	225	229	233	237	241	245	249	253	257	261	265	269	273	277	281	285	289	293	297	301	305	309	313	317	321	325	329	333	337	341	345	349	353	357	361	365	369	373	377	381	385	389	393	397	401	405	409	413	417	421	425	429	433	437	441	445	449	453	457	461	465	469	473	477	481	485	489	493	497	501	505	509	513	517	521	525	529	533	537	541	545	549	553	557	561	565	569	573	577	581	585	589	593	597	601	605	609	613	617	621	625	629	633	637	641	645	649	653	657	661	665	669	673	677	681	685	689	693	697	701	705	709	713	717	721	725	729	733	737	741	745	749	753	757	761	765	769	773	777	781	785	789	793	797	801	805	809	813	817	821	825	829	833	837	841	845	849	853	857	861	865	869	873	877	881	885	889	893	897	901	905	909	913	917	921	925	929	933	937	941	945	949	953	957	961	965	969	973	977	981	985	989	993	997	1001

Ships $[0, a, b]$ with a and b coprime.

We can reuse our understanding about 2-dimensional ships!

Consider the map $(x, y) \mapsto ay + bx$.

Since a and b are coprime, every strip of width b contains all integers exactly once!

58	61	66	73	82	93	106
-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

-68	-61	-53	-49	-45	-41	-37
-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

-68	-61	-53	-49	-45	-41	-37
-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

-68	-61	-53	-49	-45	-41	-37
-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

-68	-61	-53	-49	-45	-41	-37
-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

-68	-61	-53	-49	-45	-41	-37
-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

...except on the border.

-68	-61	-53	-49	-45	-41	-37
-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

...except on the border.

-68	-61	-53	-49	-45	-41	-37
-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

...except on the border.

We know how to optimally hit this ship!

-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

...except on the border.

We know how to optimally hit this ship!

-58	-57	-56	-55	-54	-53	-52	-51
-61	-57	-53	-49	-45	-41	-37	
-54	-50	-46	-42	-38	-34	-30	
-47	-43	-39	-35	-31	-27	-23	
-40	-36	-32	-28	-24	-20	-16	
-33	-29	-25	-21	-17	-13	-9	
-26	-22	-18	-14	-10	-6	-2	
-19	-15	-11	-7	-3	1	5	
-12	-8	-4	0	4	8	12	
-5	-1	3	7	11	15	19	
2	6	10	14	18	22	26	
9	13	17	21	25	29	33	
16	20	24	28	32	36	40	
23	27	31	35	39	43	47	
30	34	38	42	46	50	54	
37	41	45	49	53	57	61	
44	48	52	56	60	64	68	

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

...except on the border.

We know how to optimally hit this ship!

-58	-57	-56	-55	-54	-53	-52	-51
-61	-57	-53	-49	-45	-41	-37	
-54	-50	-46	-42	-38	-34	-30	
-47	-43	-39	-35	-31	-27	-23	
-40	-36	-32	-28	-24	-20	-16	
-33	-29	-25	-21	-17	-13	-9	
-26	-22	-18	-14	-10	-6	-2	
-19	-15	-11	-7	-3	1	5	
-12	-8	-4	0	4	8	12	
-5	-1	3	7	11	15	19	
2	6	10	14	18	22	26	
9	13	17	21	25	29	33	
16	20	24	28	32	36	40	
23	27	31	35	39	43	47	
30	34	38	42	46	50	54	
37	41	45	49	53	57	61	
44	48	52	56	60	64	68	

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

...except on the border.

We know how to optimally hit this ship!

-58	-57	-56	-55	-54	-53	-52	-51
-61	-57	-53	-49	-45	-41	-37	
-54	-50	-46	-42	-38	-34	-30	
-47	-43	-39	-35	-31	-27	-23	
-40	-36	-32	-28	-24	-20	-16	
-33	-29	-25	-21	-17	-13	-9	
-26	-22	-18	-14	-10	-6	-2	
-19	-15	-11	-7	-3	1	5	
-12	-8	-4	0	4	8	12	
-5	-1	3	7	11	15	19	
2	6	10	14	18	22	26	
9	13	17	21	25	29	33	
16	20	24	28	32	36	40	
23	27	31	35	39	43	47	
30	34	38	42	46	50	54	
37	41	45	49	53	57	61	
44	48	52	56	60	64	68	

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

...except on the border.

We know how to optimally hit this ship!

We just need a few extra.

-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

...except on the border.

We know how to optimally hit this ship!

We just need a few extra.

-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

Ships $[0, a, b]$ with a and b coprime.

A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

...except on the border.

We know how to optimally hit this ship!

We just need a few extra.

This pattern works. What is the density?

-61	-57	-53	-49	-45	-41	-37
-54	-50	-46	-42	-38	-34	-30
-47	-43	-39	-35	-31	-27	-23
-40	-36	-32	-28	-24	-20	-16
-33	-29	-25	-21	-17	-13	-9
-26	-22	-18	-14	-10	-6	-2
-19	-15	-11	-7	-3	1	5
-12	-8	-4	0	4	8	12
-5	-1	3	7	11	15	19
2	6	10	14	18	22	26
9	13	17	21	25	29	33
16	20	24	28	32	36	40
23	27	31	35	39	43	47
30	34	38	42	46	50	54
37	41	45	49	53	57	61
44	48	52	56	60	64	68

8/21

Ships $[0, a, b]$ with a and b coprime.

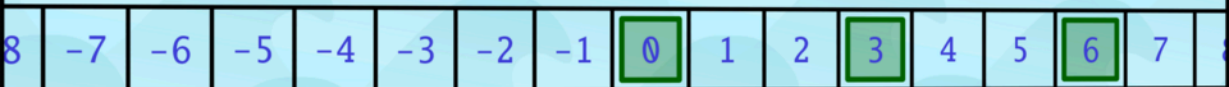
A disconnected 1-dimensional ship becomes a connected 2-dimensional ship...

$$\text{density} = \frac{b-1}{b} \times \frac{1}{3} + \frac{1}{b} \times \frac{2}{3}$$

density $\rightarrow 1/3$ as $b \rightarrow \infty$

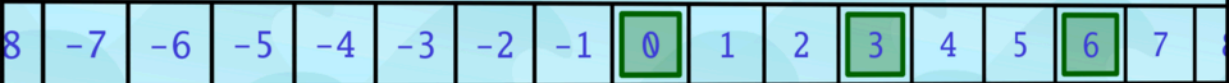
Small disconnected ships are harder to hit!

Ships $[0, a, b]$ with a and b
not coprime



Ships $[0, a, b]$ with a and b
not coprime

$$a = k \cdot a' \quad b = k \cdot b'$$



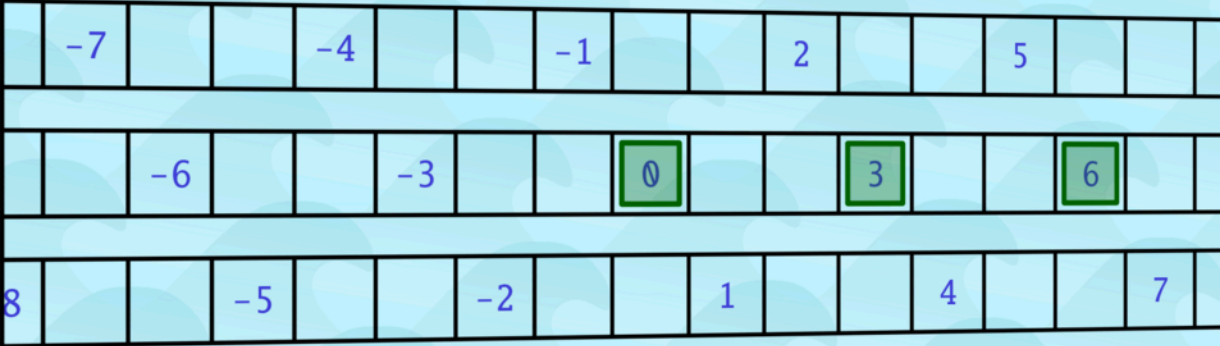
Ships $[0, a, b]$ with a and b
not coprime

$$a = k \cdot a' \quad b = k \cdot b'$$

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

Ships $[0, a, b]$ with a and b
not coprime

$$a = k \cdot a' \quad b = k \cdot b'$$



Ships $[0, a, b]$ with a and b
not coprime

$$a = k \cdot a' \quad b = k \cdot b'$$

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

Ships $[0, a, b]$ with a and b
not coprime

$$a = k \cdot a' \quad b = k \cdot b'$$

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

Ships $[0, a, b]$ with a and b
not coprime

$$a = k \cdot a' \quad b = k \cdot b'$$

25	-22	-19	-16	-13	-10	-7	-4	-1	2	5	8	11	14	17	20	23
24	-21	-18	-15	-12	-9	-6	-3	0	3	6	9	12	15	18	21	24
23	-20	-17	-14	-11	-8	-5	-2	1	4	7	10	13	16	19	22	25

Ships $[0, a, b]$ with a and b
not coprime

$$a = k \cdot a' \quad b = k \cdot b'$$

25	-22	-19	-16	-13	-10	-7	-4	-1	2	5	8	11	14	17	20	23
24	-21	-18	-15	-12	-9	-6	-3	0	3	6	9	12	15	18	21	24
23	-20	-17	-14	-11	-8	-5	-2	1	4	7	10	13	16	19	22	25

Ships $[0, a, b]$ with a and b
not coprime

$$a = k \cdot a' \quad b = k \cdot b'$$

25	-22	-19	-16	-13	-10	-7	-4	-1	2	5	8	11	14	17	20	23
24	-21	-18	-15	-12	-9	-6	-3	0	3	6	9	12	15	18	21	24
23	-20	-17	-14	-11	-8	-5	-2	1	4	7	10	13	16	19	22	25

Ships $[0, a, b]$ with a and b
not coprime

$$a = k \cdot a' \quad b = k \cdot b'$$

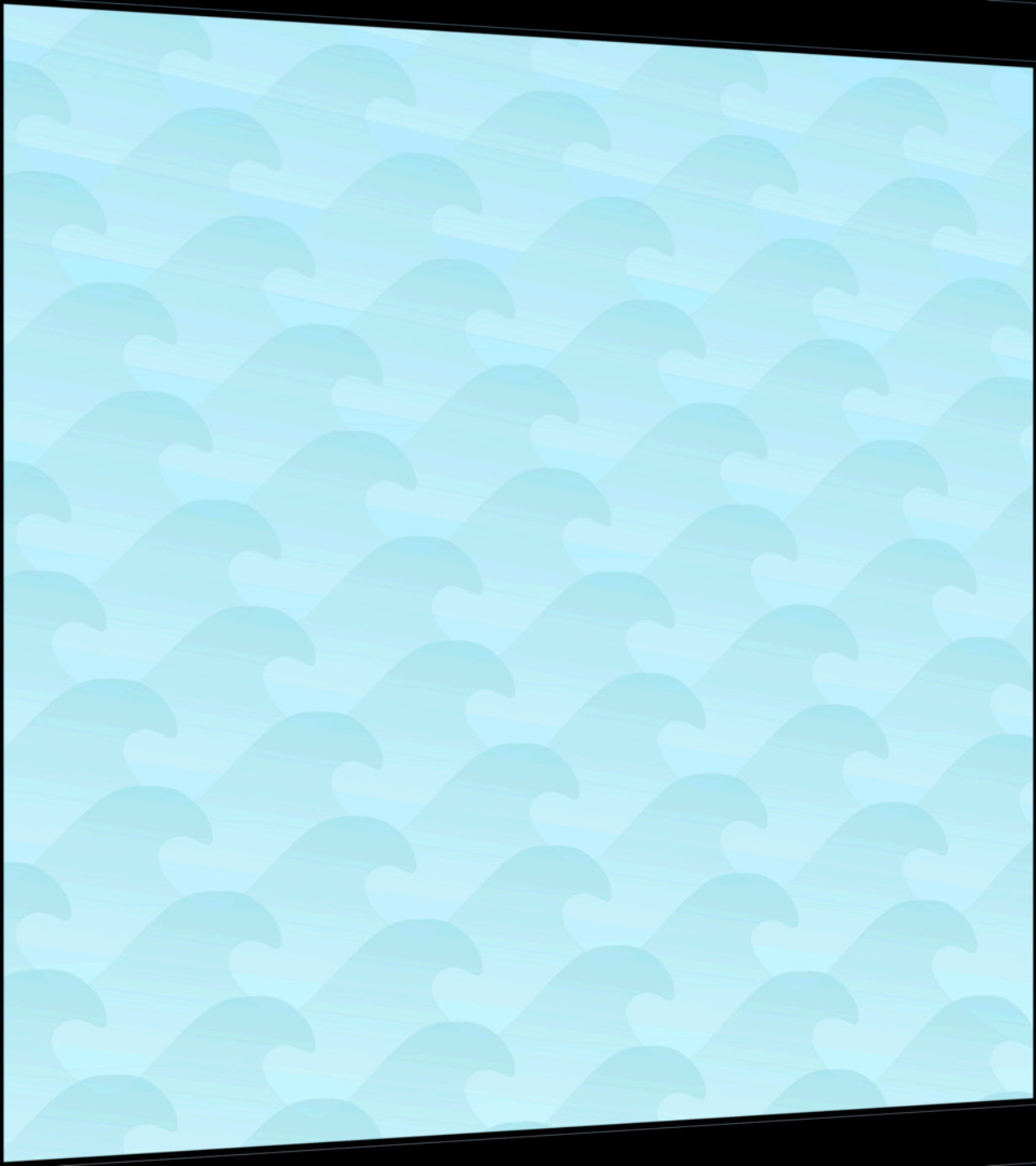
25	-22	-19	-16	-13	-10	-7	-4	-1	2	5	8	11	14	17	20	23
24	-21	-18	-15	-12	-9	-6	-3	0	3	6	9	12	15	18	21	24
23	-20	-17	-14	-11	-8	-5	-2	1	4	7	10	13	16	19	22	25

Ships $[0, a, b]$ with a and b
not coprime

$$a = k \cdot a' \quad b = k \cdot b'$$

Same density as $[0, a', b']!$

25	-22	-19	-16	-13	-10	-7	-4	-1	2	5	8	11	14	17	20	23
24	-21	-18	-15	-12	-9	-6	-3	0	3	6	9	12	15	18	21	24
23	-20	-17	-14	-11	-8	-5	-2	1	4	7	10	13	16	19	22	25



5

CHAPTER

.....
CONTRIBUTION

Theorem 1 Given a ship S with span s , the density of S can be computed in time polynomial in 2^s .

Theorem 2 $S = \{[0, da], [0, db]\}$, a, b coprime. Density is

$$\begin{cases} 1/2 & \text{if } a \text{ and } b \text{ odd,} \\ \frac{a+b+1}{2(a+b)} & \text{otherwise.} \end{cases}$$

Theorem 3 The toughest set of n shapes of size 2 has density $n/(n+1)$.

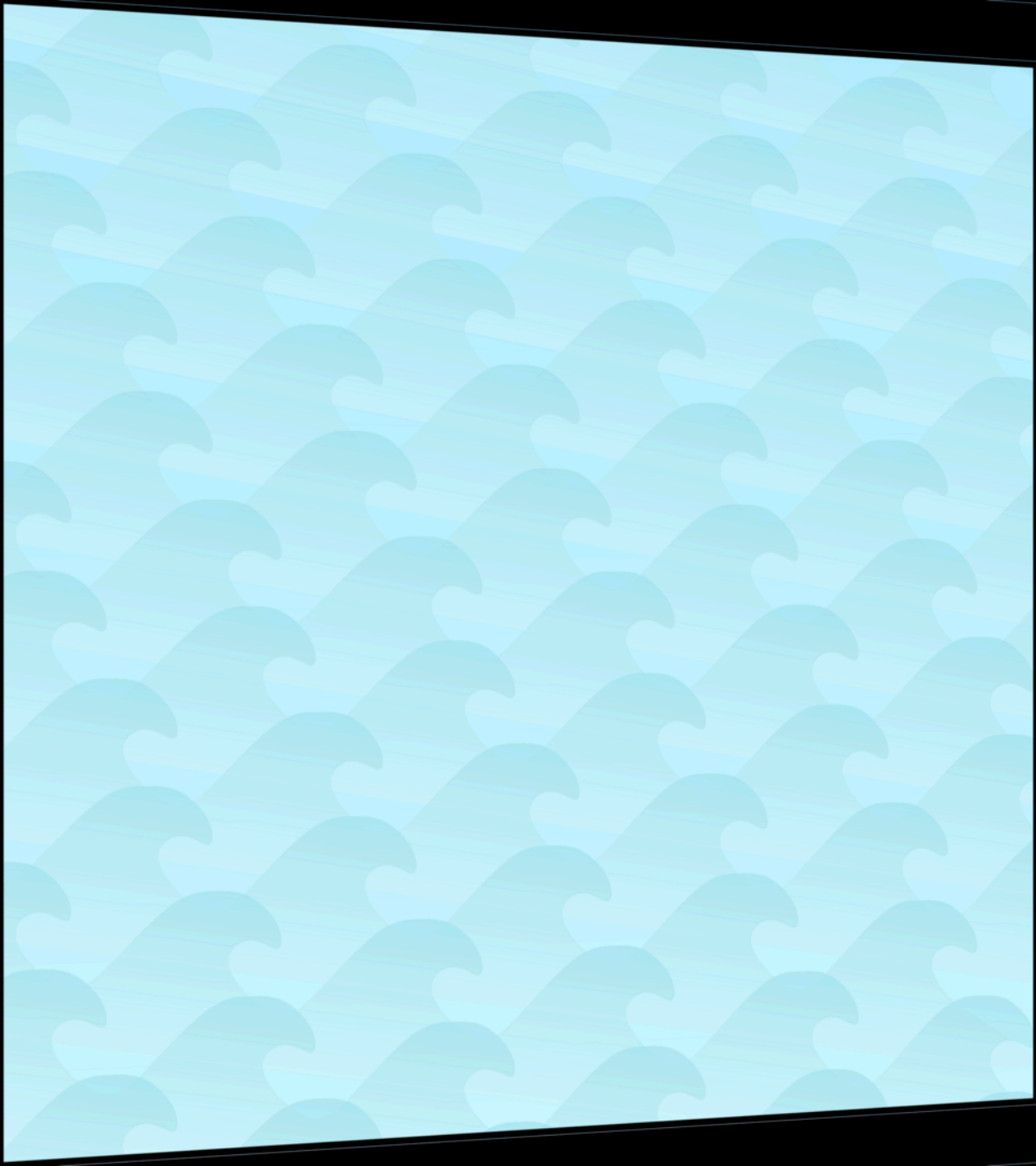
Theorem 4 Let S be a 3-shape and its reflection. Then the density is $\leq \frac{2}{5}$.

Theorem 5 The simplest set of n shapes of size k has density $\frac{1}{k}$.

Theorem 6 The toughest set of n shapes of size k has density d where

$$1 - \frac{k-1}{\sqrt[k]{nk!}} \leq d \leq \min \left\{ \frac{n}{n+1}, \frac{1 + \log(kn)}{k} \right\}$$

Generalizations to 2D

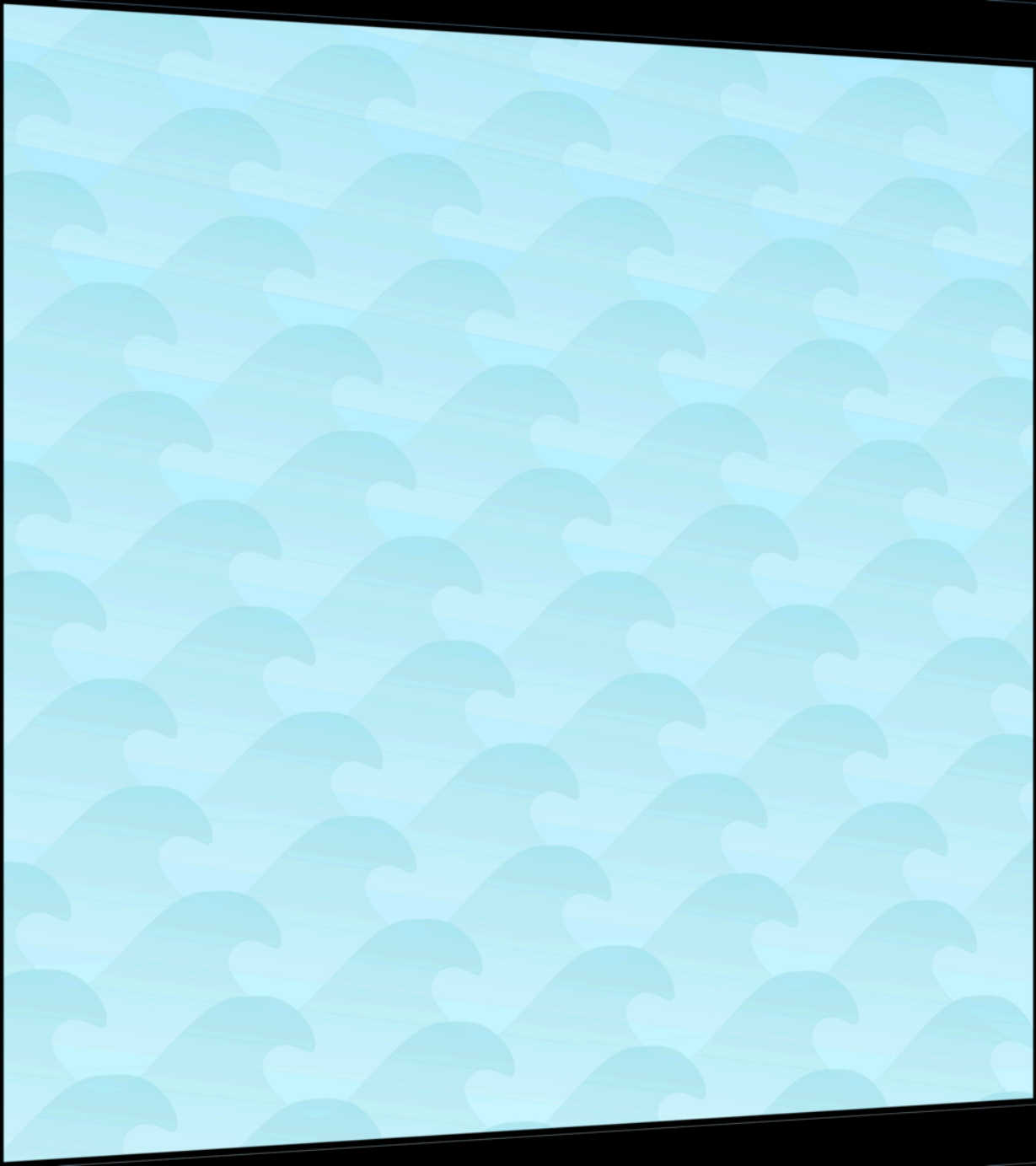


6

CHAPTER



OPEN PROBLEMS



Problem 1.

Is a 1-dimensional optimal solution always periodic?

Problem 1.

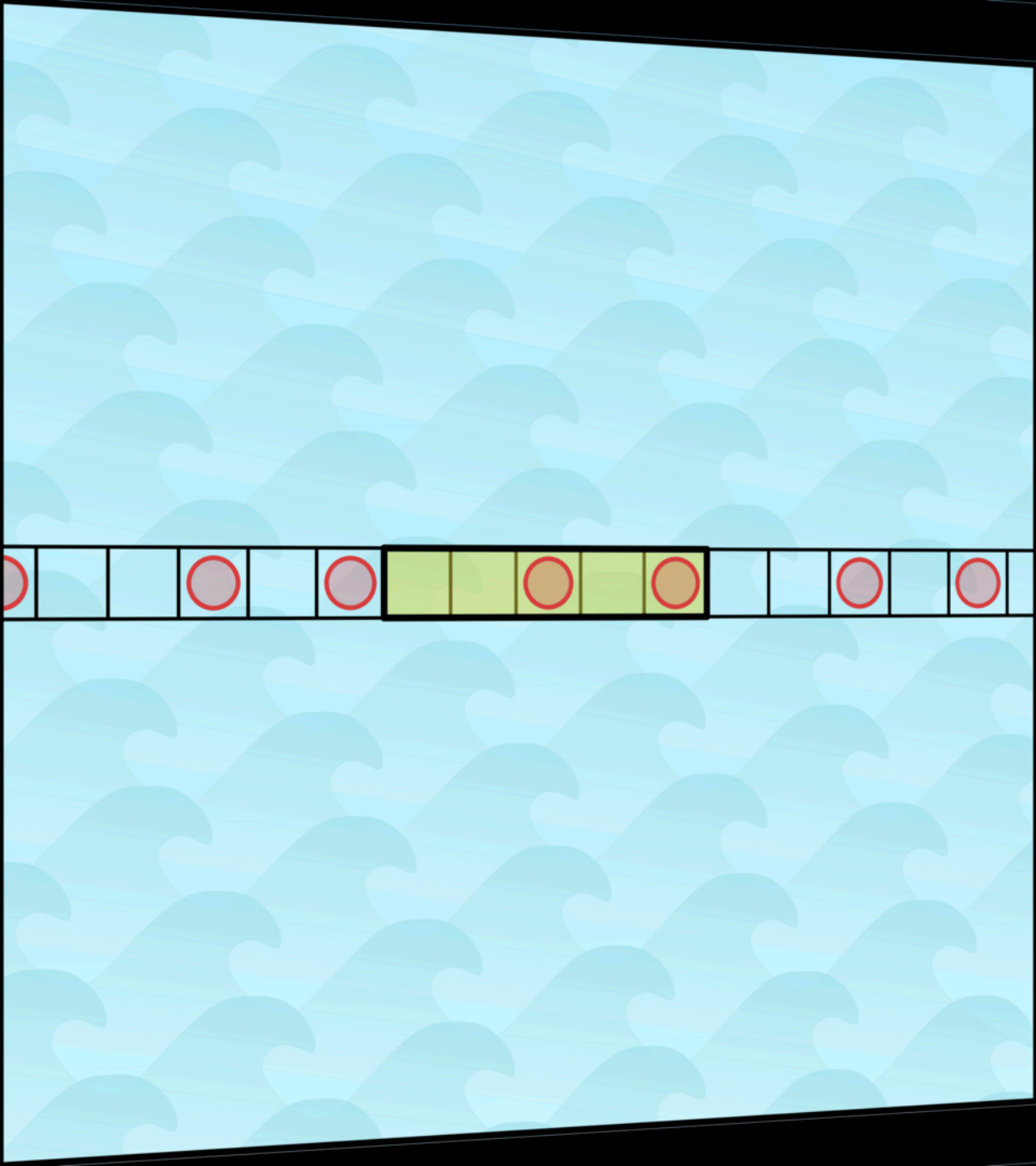
Is a 1-dimensional optimal solution always periodic?



Problem 1.

Is a 1-dimensional optimal solution always periodic?

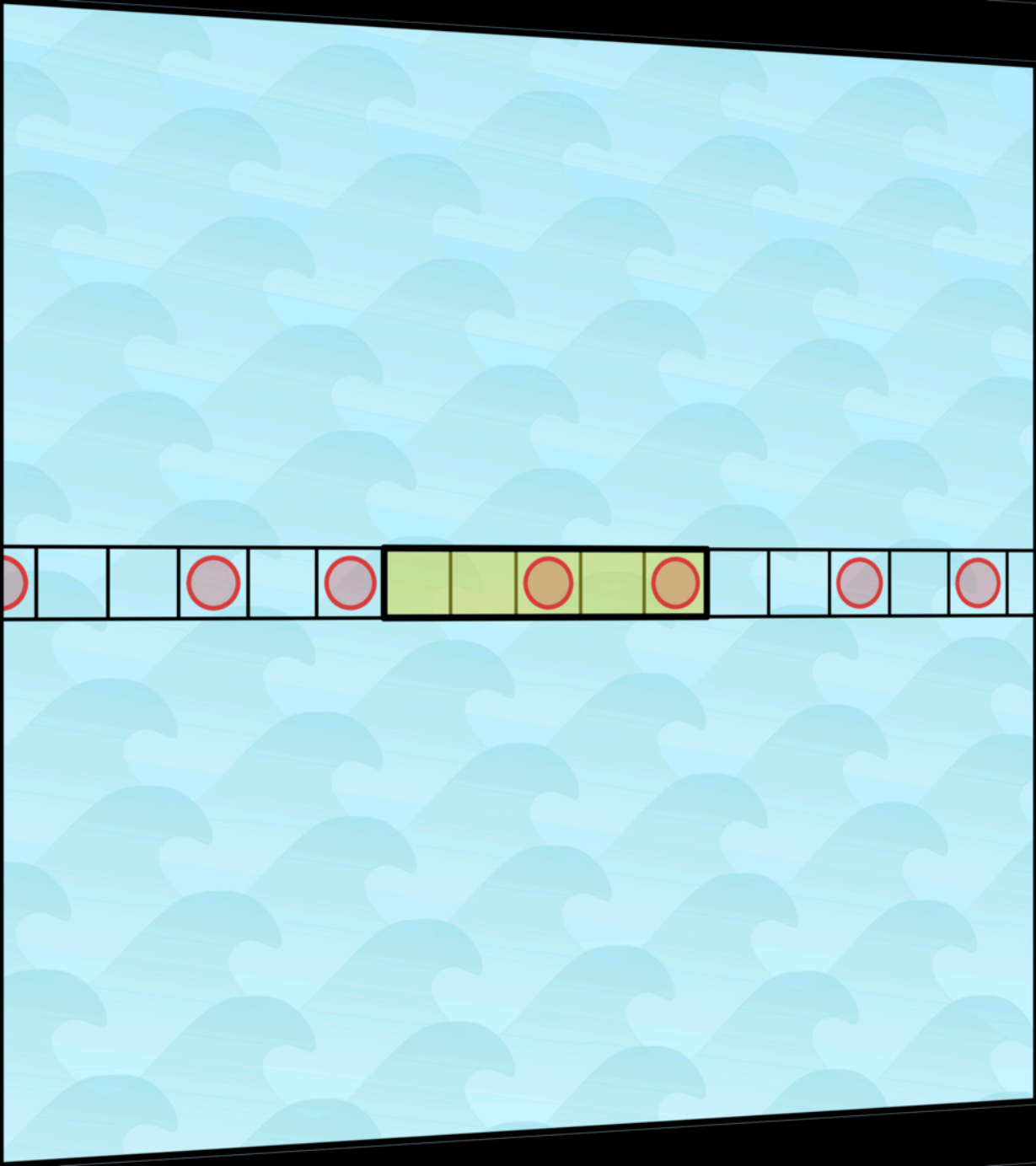




Problem 1.

Is a 1-dimensional optimal solution always periodic?

Short period would imply possible algorithms.

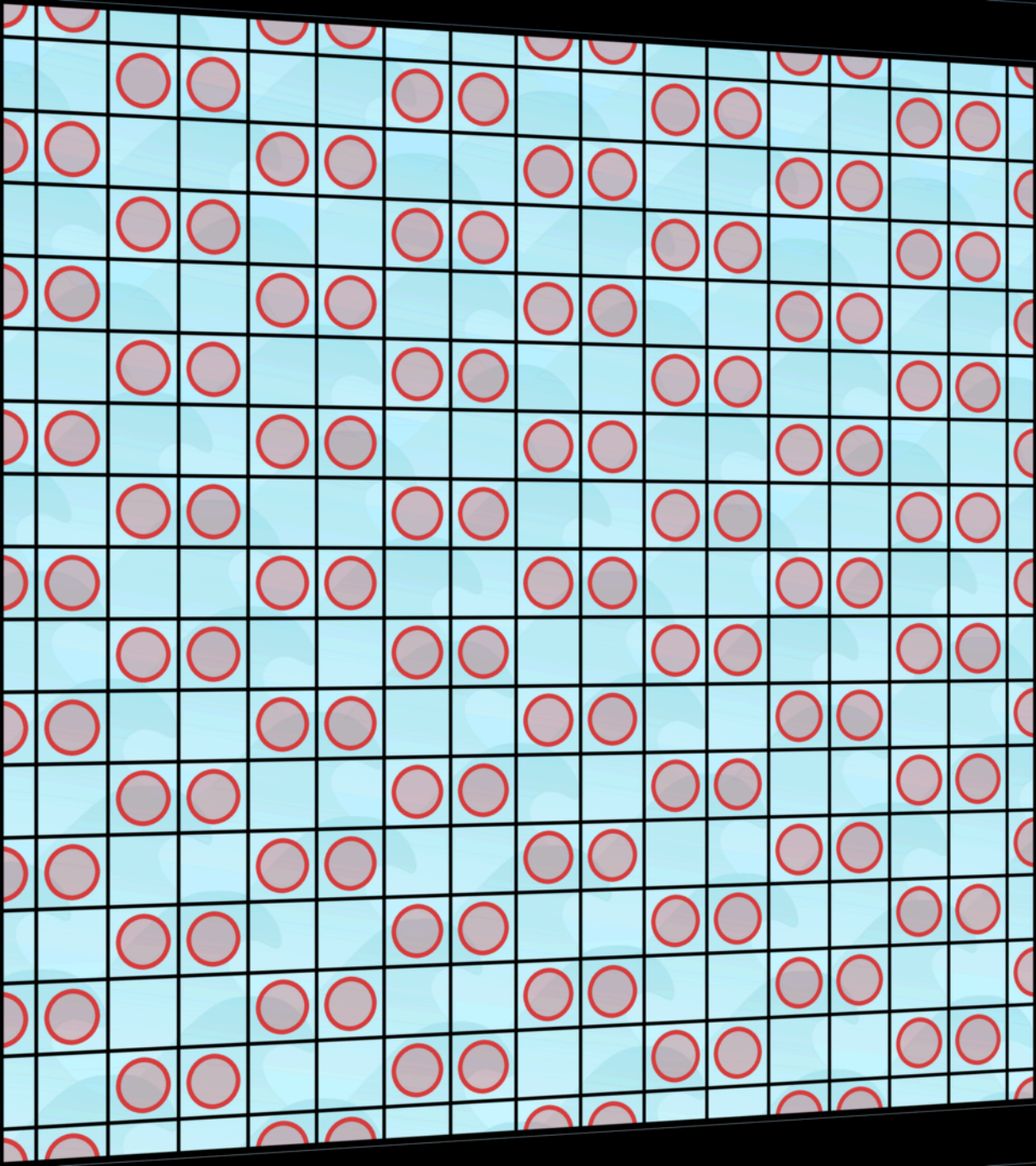


Problem 1.

Is a 1-dimensional optimal solution always periodic?

Short period would imply possible algorithms.

What about 2-dimensional optimal solutions?

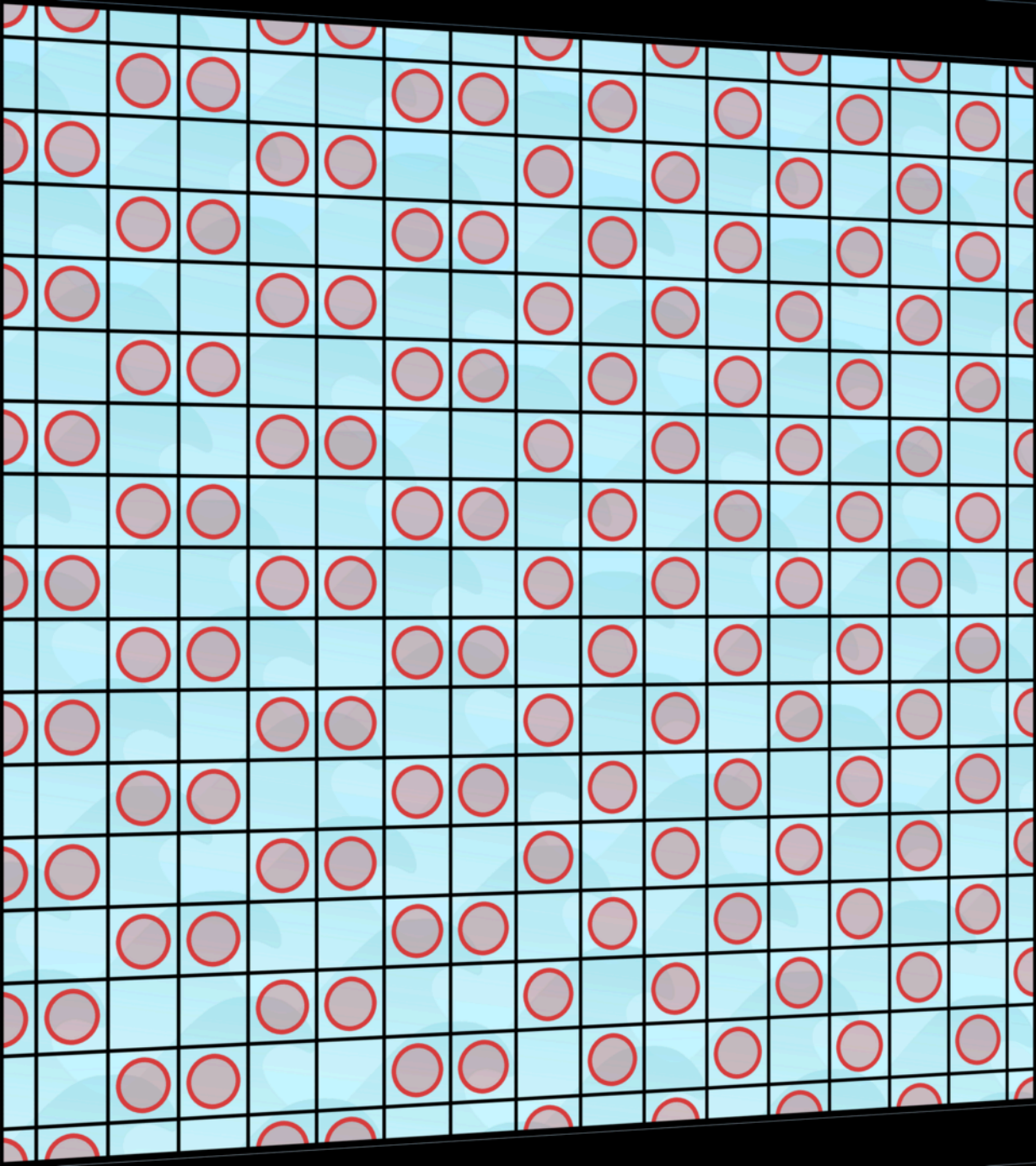


Problem 1.

Is a 1-dimensional optimal solution always periodic?

Short period would imply possible algorithms.

What about 2-dimensional optimal solutions?



Problem 1.

Is a 1-dimensional optimal solution always periodic?

Short period would imply possible algorithms.

What about 2-dimensional optimal solutions?

Problem 2.

How does any of this insight help us with the original (continuous) problem?

Problem 2.

How does any of this insight help us with the original (continuous) problem?

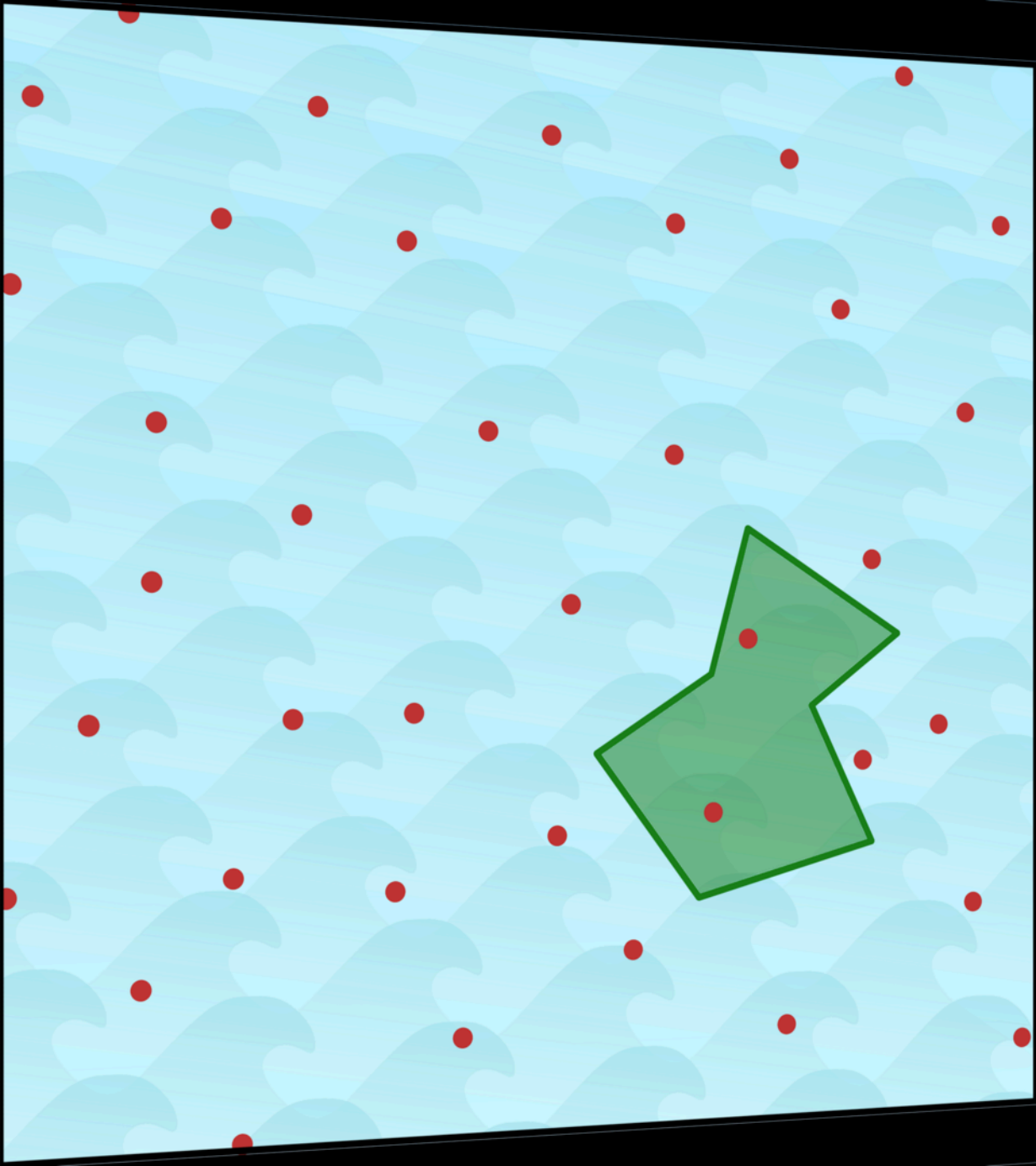
E.g.: given a polygon, what is the best point set to hit it no matter where it is hiding.



Problem 2.

How does any of this insight help us with the original (continuous) problem?

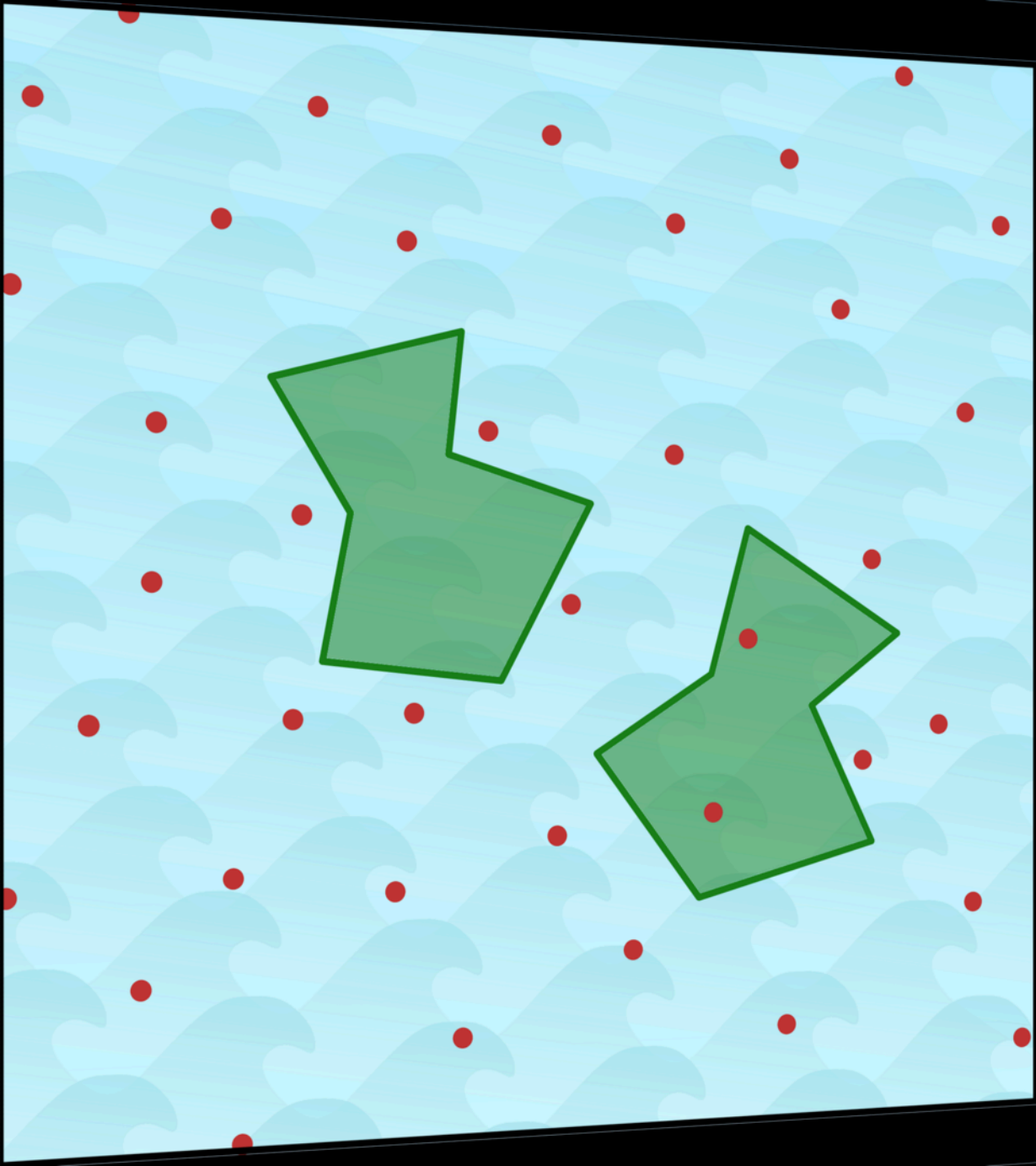
E.g.: given a polygon, what is the best point set to hit it no matter where it is hiding.



Problem 2.

How does any of this insight help us with the original (continuous) problem?

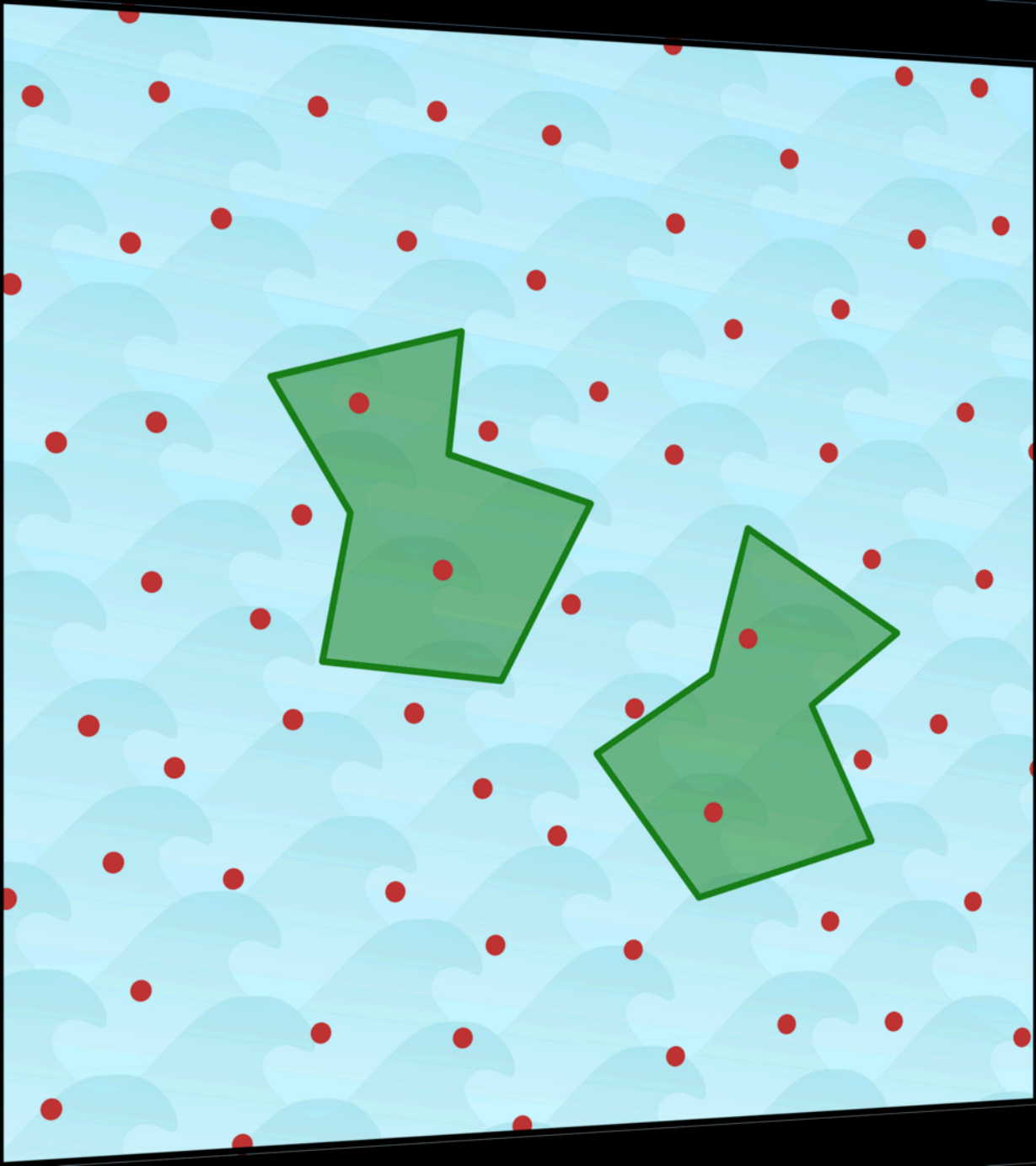
E.g.: given a polygon, what is the best point set to hit it no matter where it is hiding.



Problem 2.

How does any of this insight help us with the original (continuous) problem?

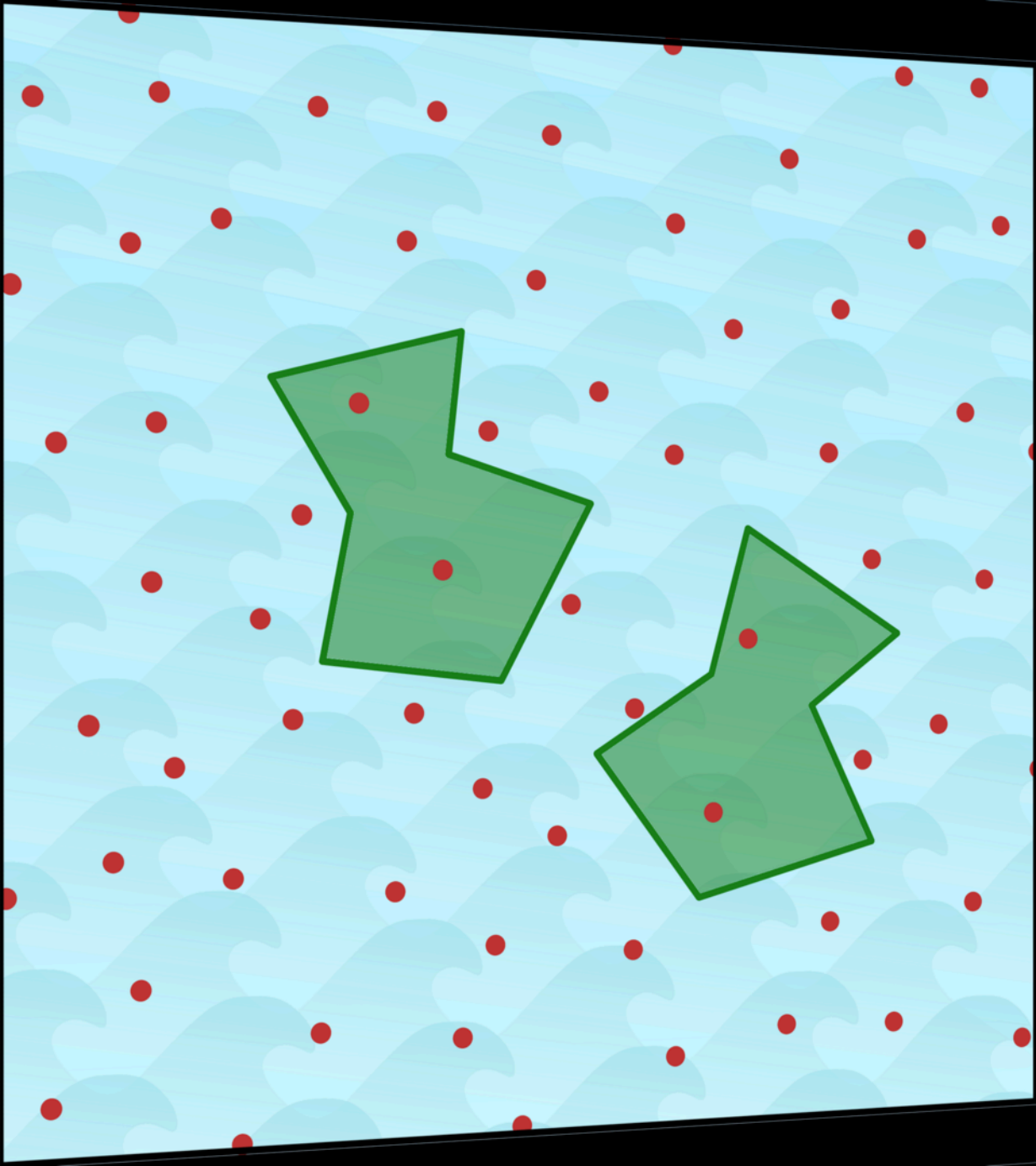
E.g.: given a polygon, what is the best point set to hit it no matter where it is hiding.



Problem 2.

How does any of this insight help us with the original (continuous) problem?

E.g.: given a polygon, what is the best point set to hit it no matter where it is hiding.



Problem 2.

How does any of this insight help us with the original (continuous) problem?

E.g.: given a polygon, what is the best point set to hit it no matter where it is hiding.

Not much is known!

Geometry,
Uncertainty,
&
BATTLESHIPS!!!

Eva Hainzl
Maarten Löffler
Daniel Perz
Josef Tkadlec