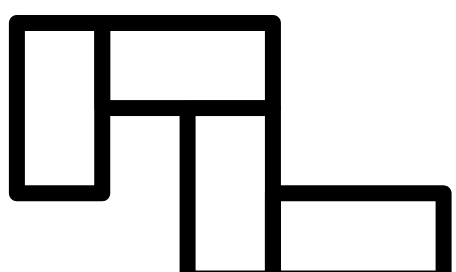
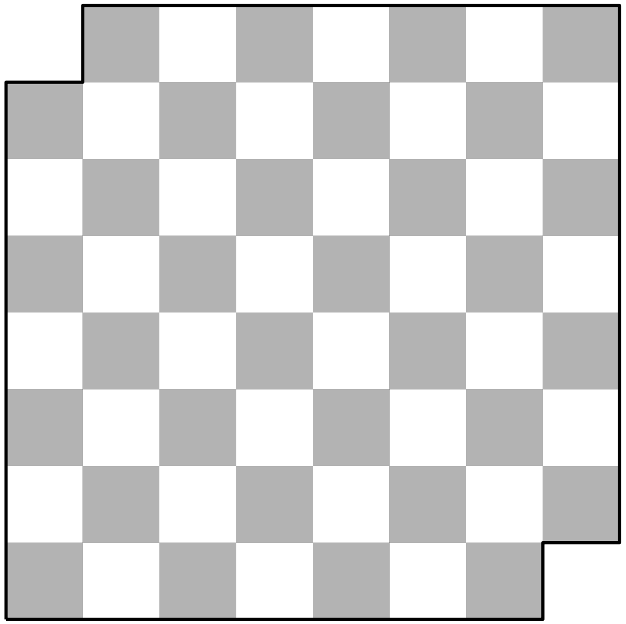
Tiling with Squares and Packing Dominos in Polynomial Time

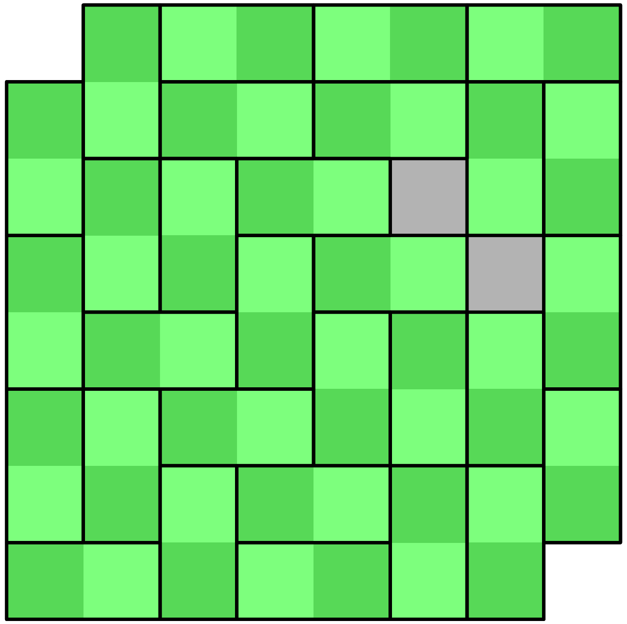
Anders Aamand, <u>Mikkel Abrahamsen</u>, Thomas D. Ahle, Peter M. R. Rasmussen



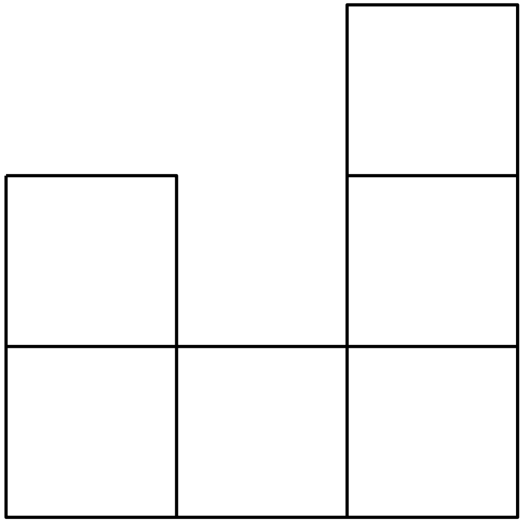
Max Black, 1946: Two diagonally opposite corners have been removed from a chessboard. Can 31 1×2 dominos be placed to cover the remaining squares?



Max Black, 1946: Two diagonally opposite corners have been removed from a chessboard. Can 31 1×2 dominos be placed to cover the remaining squares?



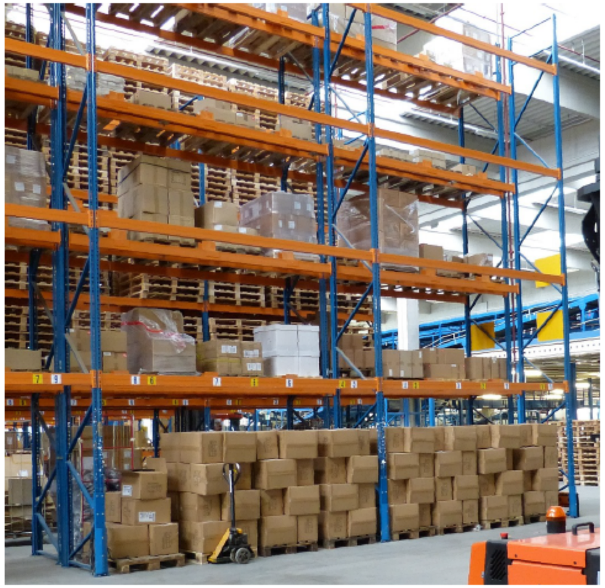
International Mathematical Olympiad 2004:
For which m and n can an $m \times n$ rectangle be tiled with 'hooks' of the following type:

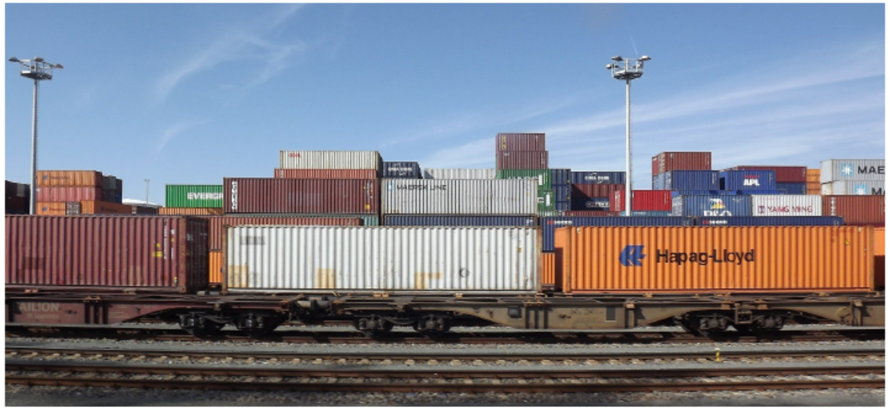


Motivation

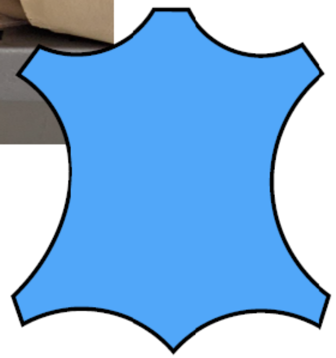
Motivation

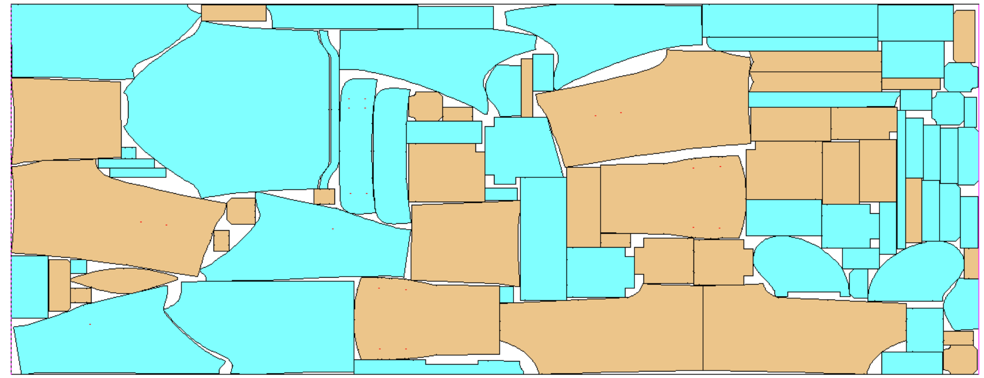
Motivation

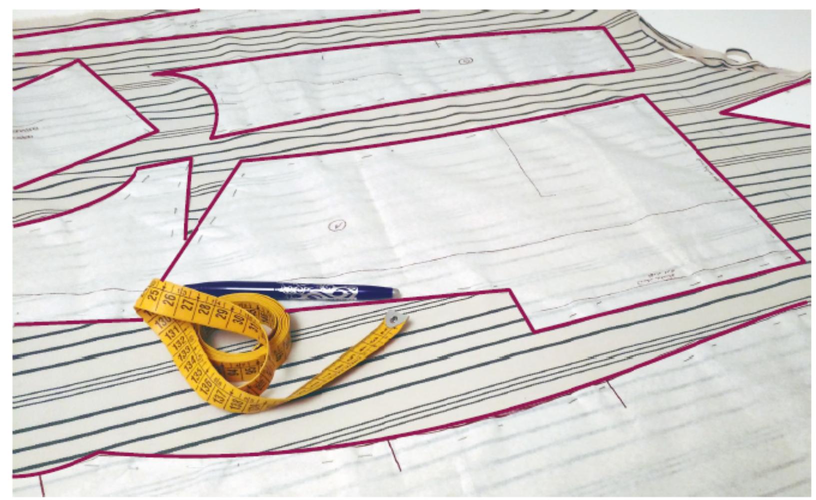




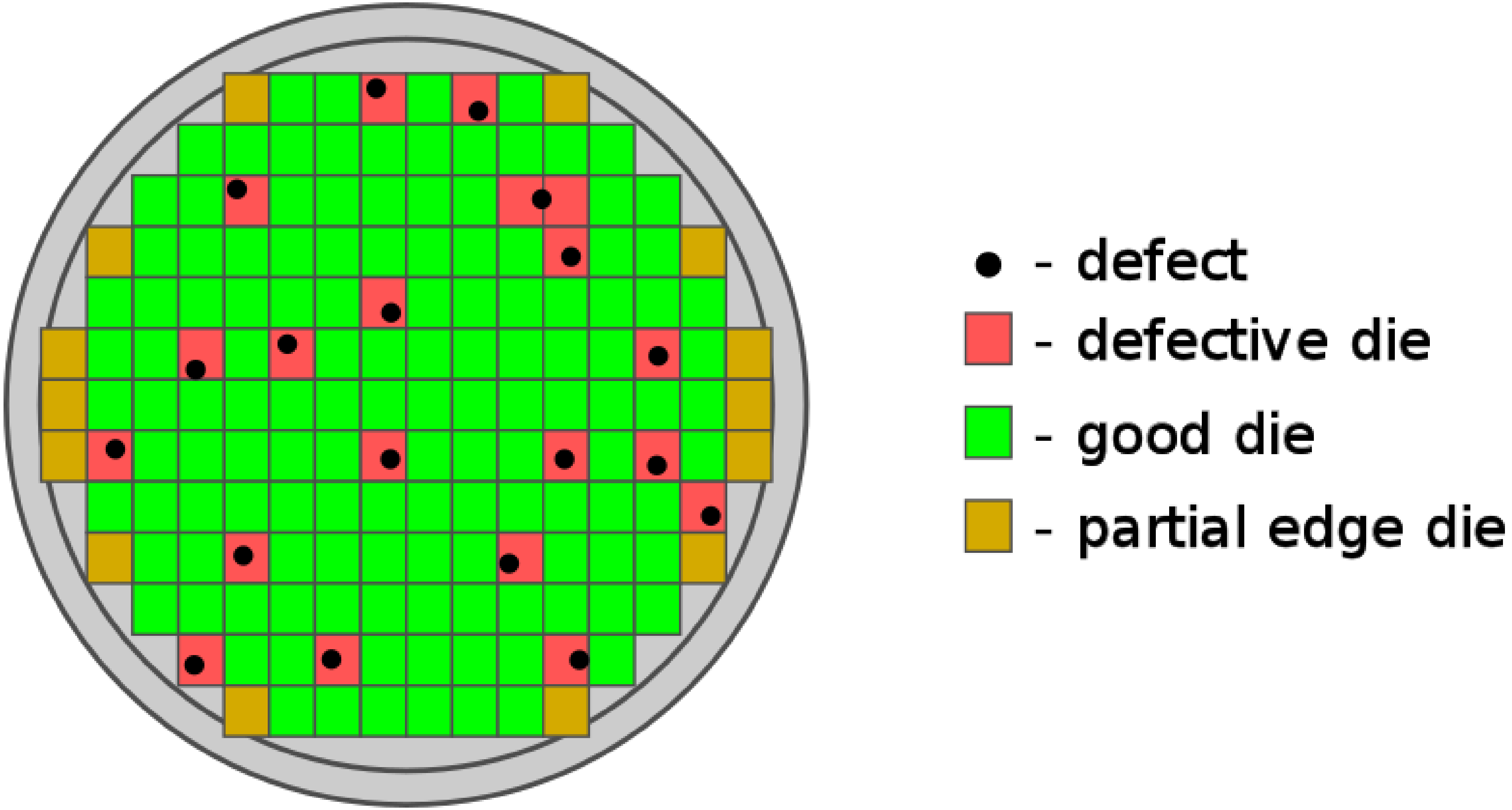
Motivation



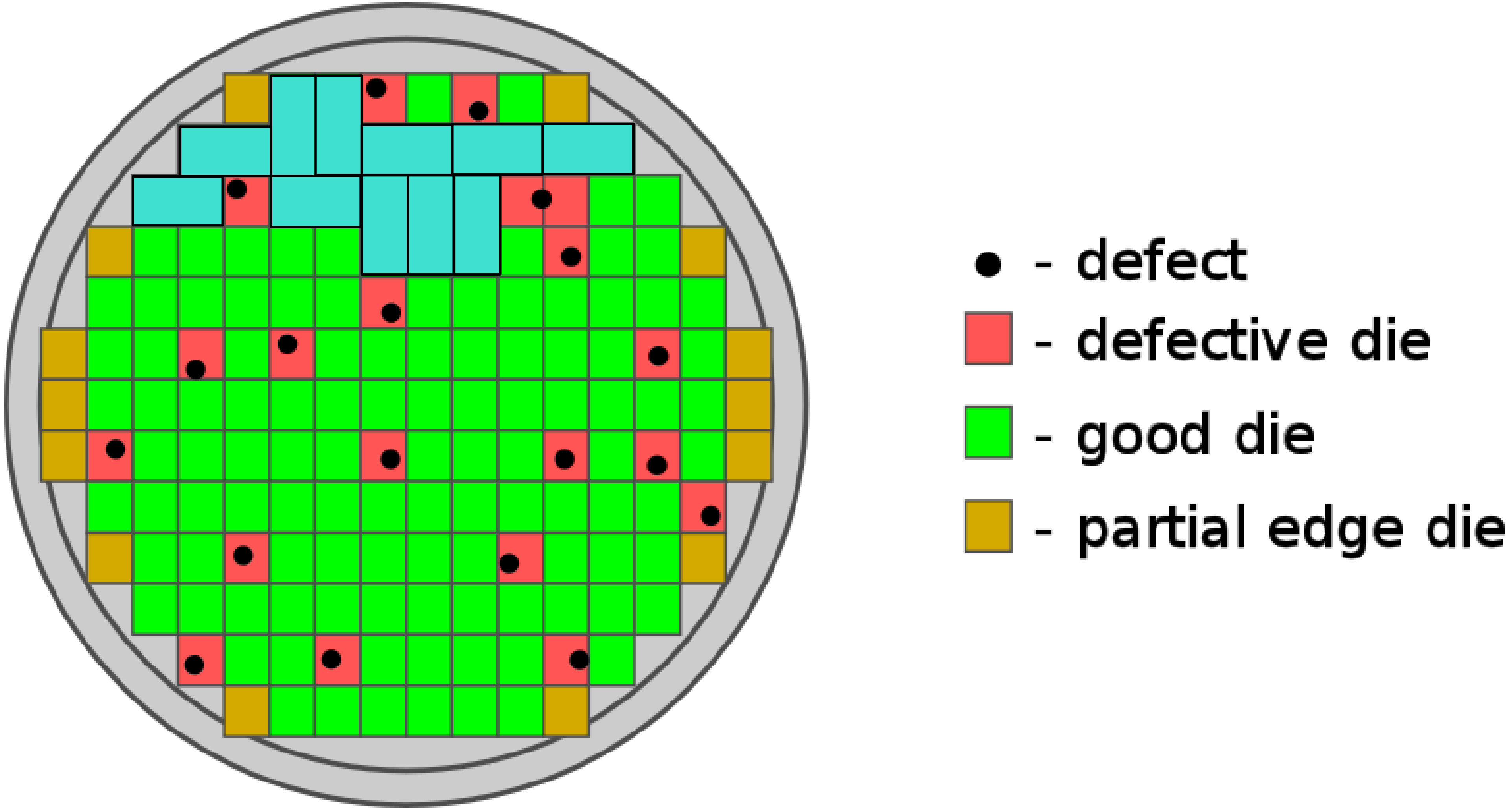




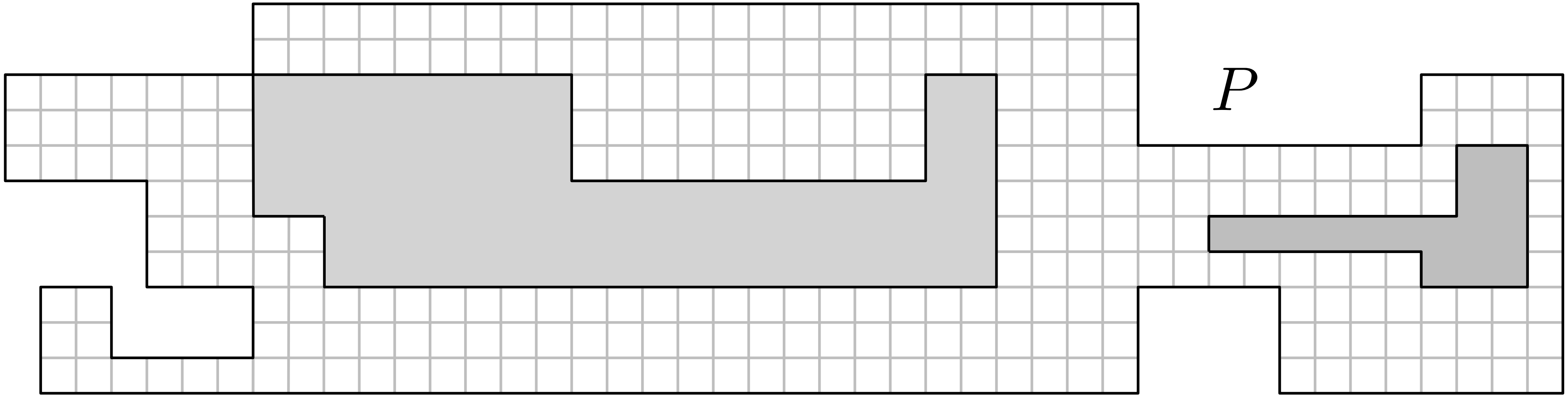
Motivation of domino packing



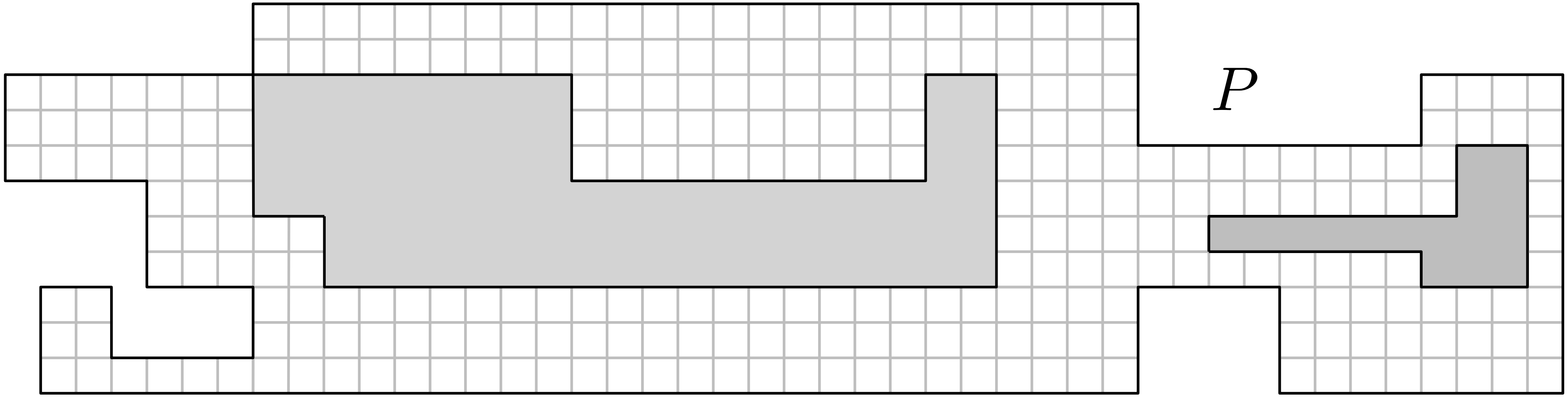
Motivation of domino packing



Polyomino: A polygonal region in the plane with axis-parallel edges and corners of integral coordinates.

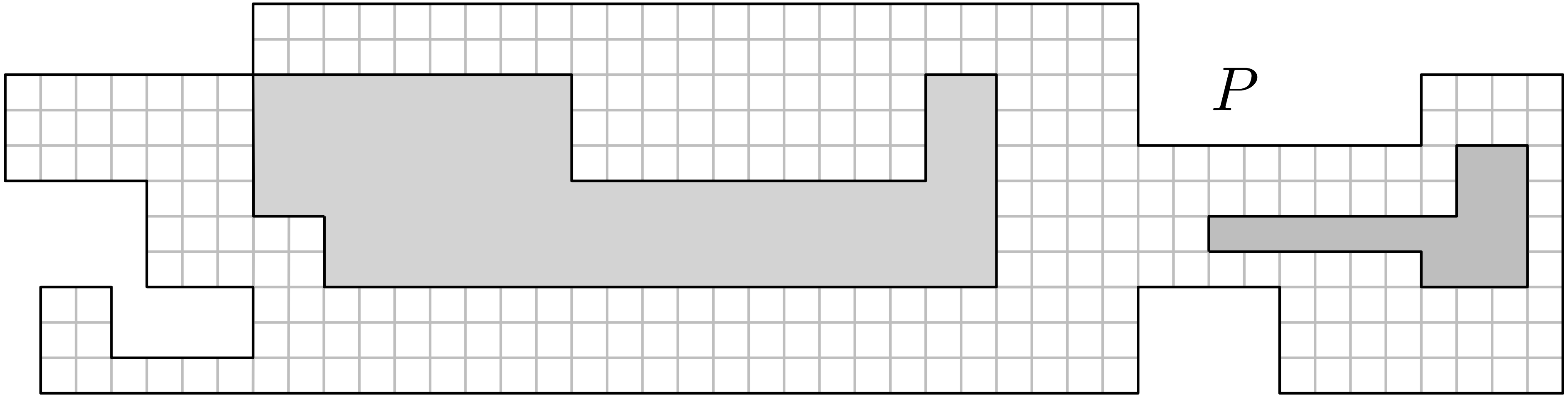


Polyomino: A polygonal region in the plane with axis-parallel edges and corners of integral coordinates.



Tiling: Can a given large polyomino P be tiled with copies of a given small polyomino Q?

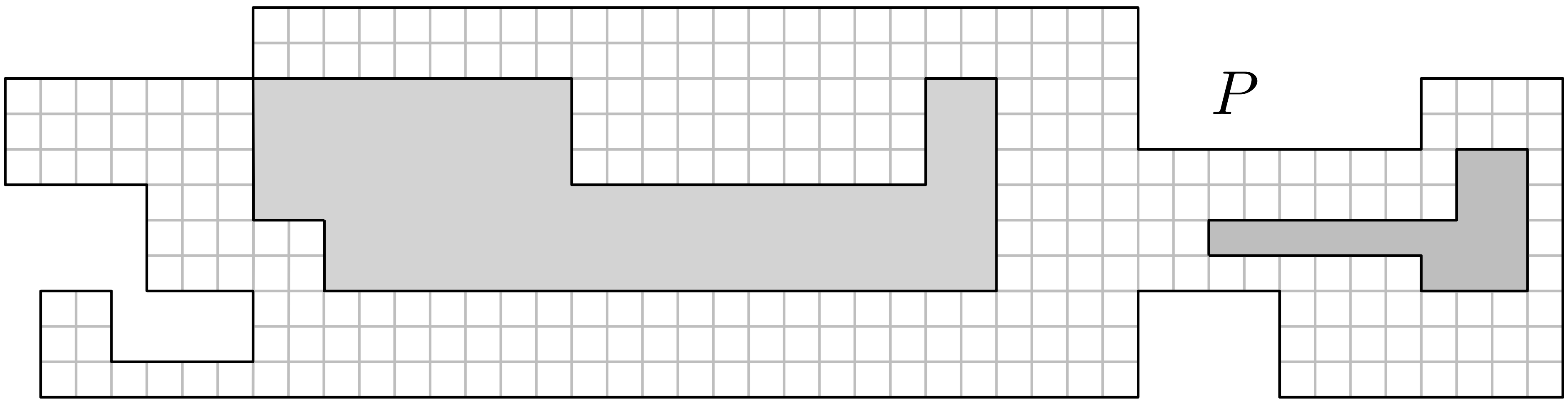
Polyomino: A polygonal region in the plane with axis-parallel edges and corners of integral coordinates.



Tiling: Can a given large polyomino P be tiled with copies of a given small polyomino Q?

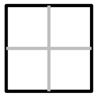
Packing: How many non-overlapping copies of Q can be fit inside P?

Polyomino: A polygonal region in the plane with axis-parallel edges and corners of integral coordinates.

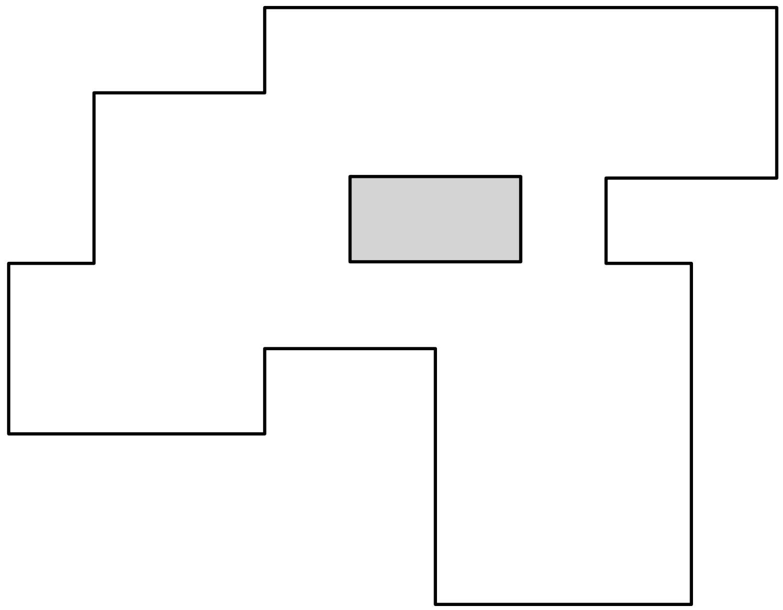


Tiling: Can a given large polyomino P be tiled with copies of a given small polyomino Q?

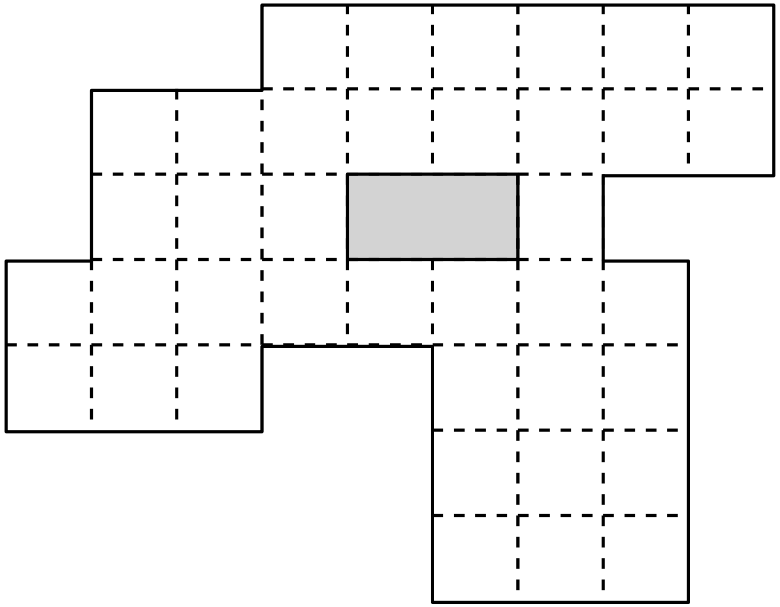
Packing: How many non-overlapping copies of Q can be fit inside P?

Our paper: $Q \in \{$  ,  $\}$

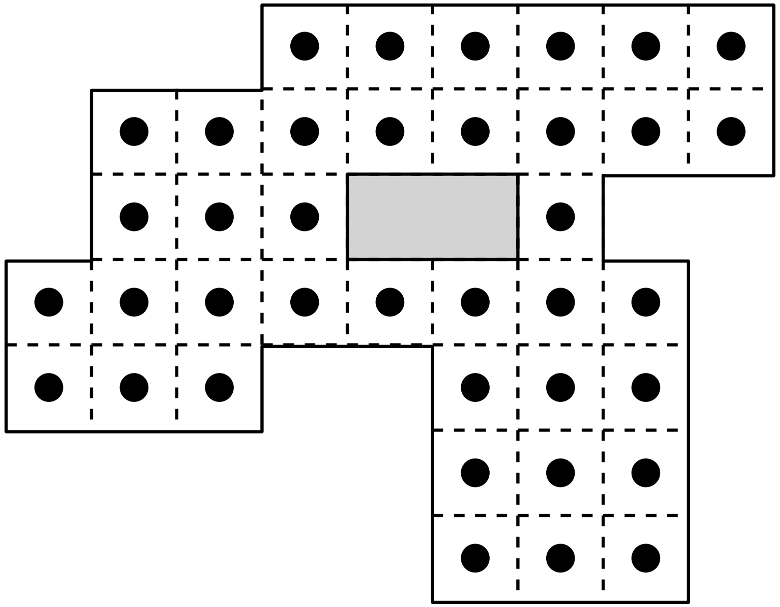
Representing a polyomino



Representing a polyomino



Representing a polyomino



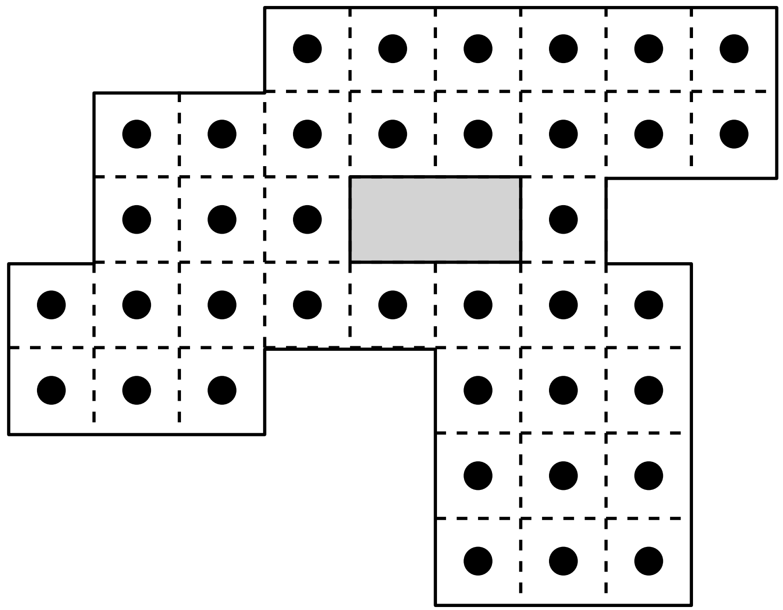
Usual way:

Store coordinates of each cell:

[• , • , • , • , • , • , . . .]

Area representation

Representing a polyomino

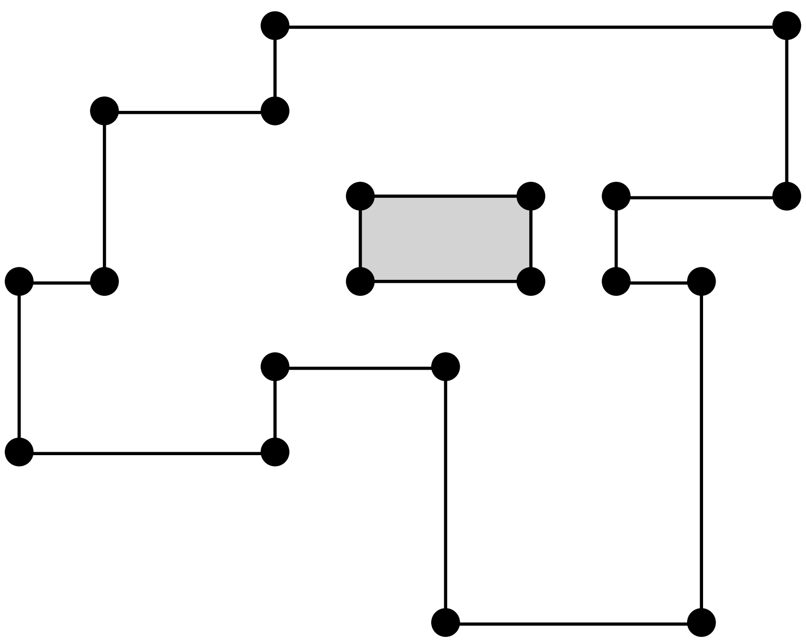


Usual way:

Store coordinates of each cell:

[• , • , • , • , • , • , . . .]

Area representation

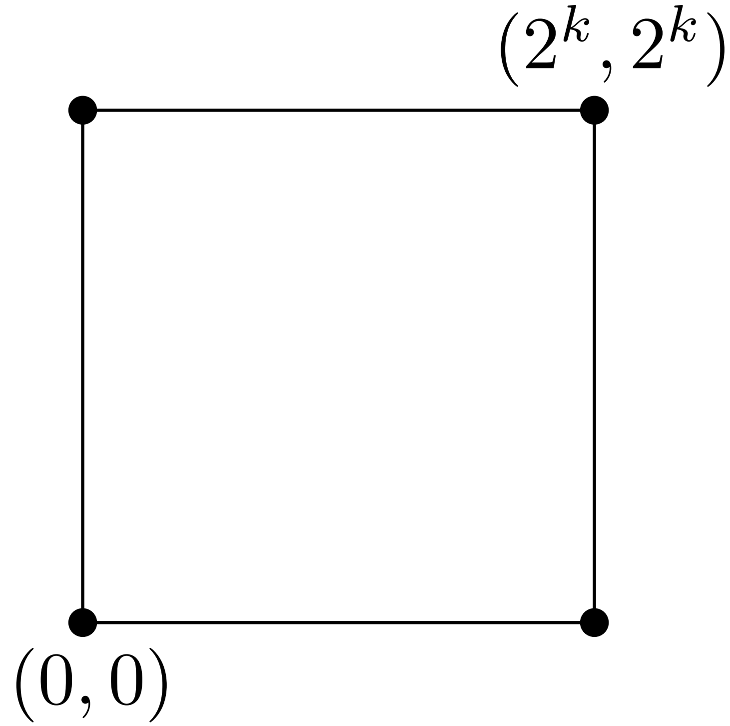


Compact way:

Store coordinates of corners.

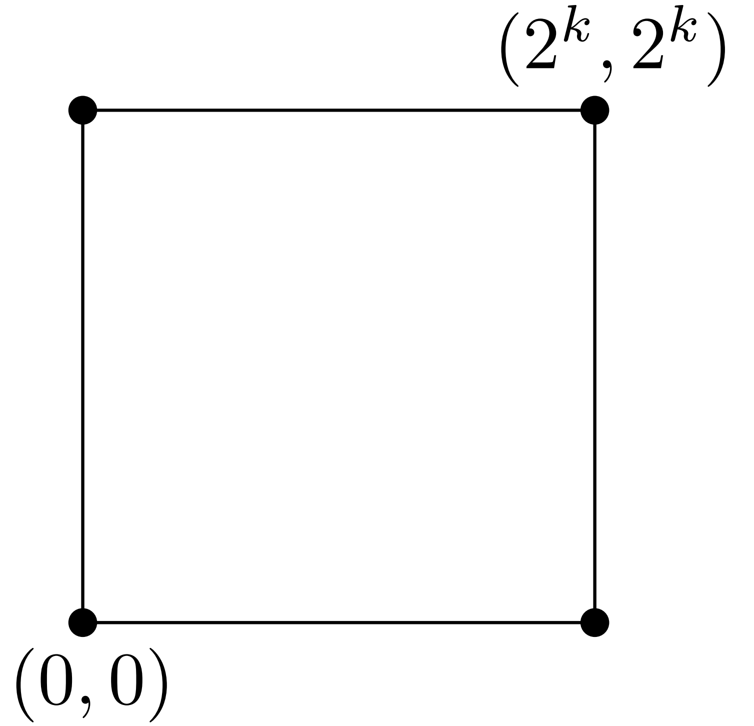
Corner representation

Example

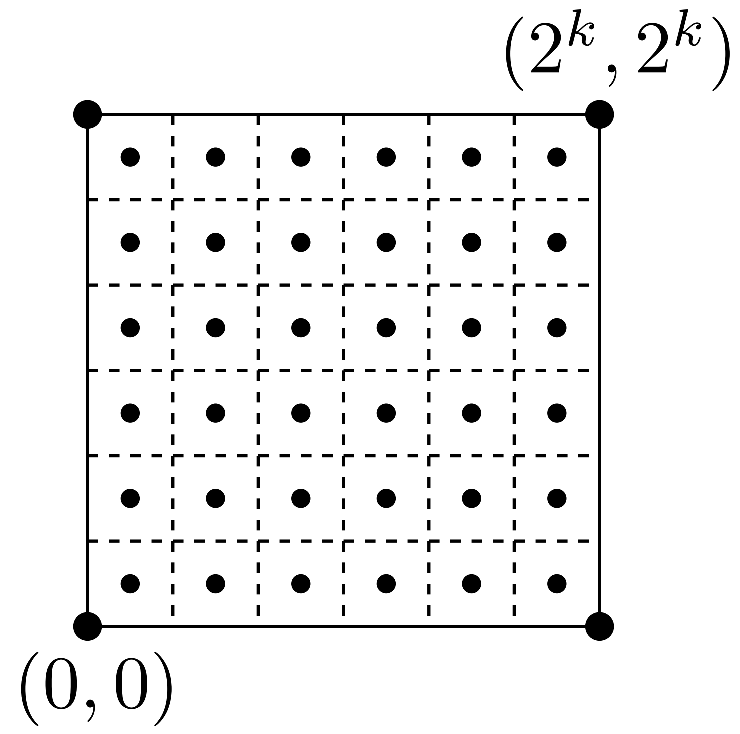


Example

Corner representation:
$[(0,0),(2^k,0),(2^k,2^k),(0,2^k)]$



Example



Corner representation:
$[(0,0),(2^k,0),(2^k,2^k),(0,2^k)]$

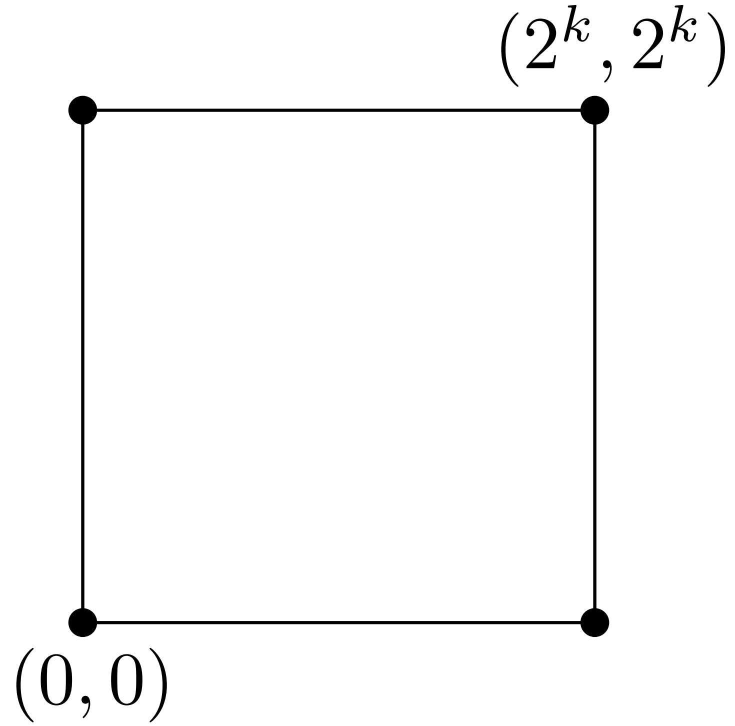
Area representation:
$[(0,0),(1,0),(2,0),\ldots,(2^k,0),$
$(0,1),(1,1),(2,1),\ldots,(2^k,1),$
$\vdots$
$(0,2^k),(1,2^k),(2,2^k),\ldots,(2^k,2^k)]$

Goal

Known algorithms:

Assume area representation $\Rightarrow$

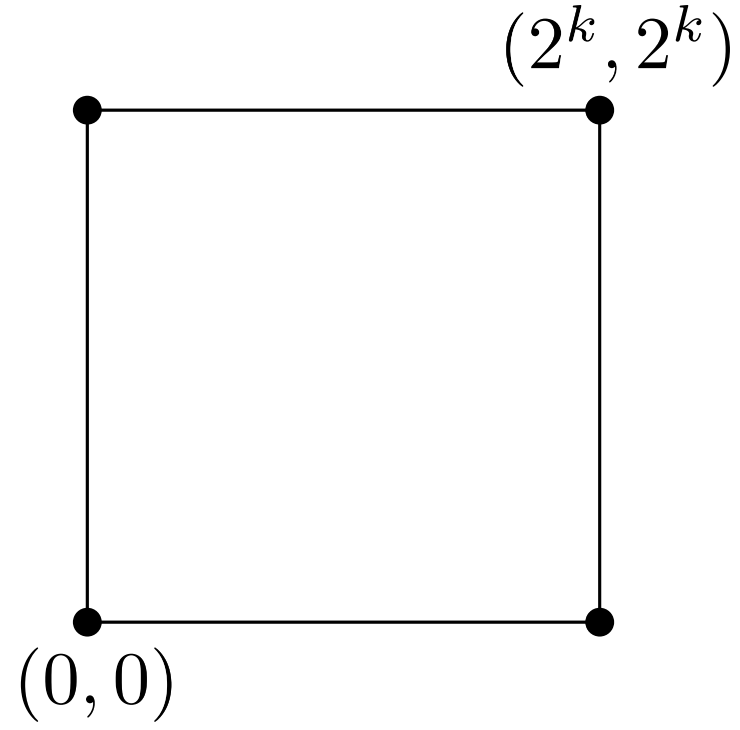
Time polynomial in the area.



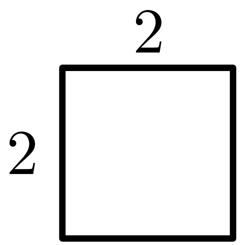
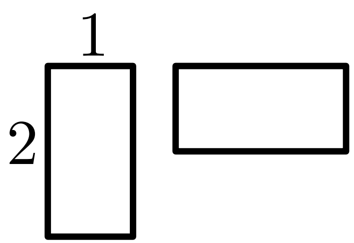
Goal

Known algorithms:
Assume area representation $\Rightarrow$
Time polynomial in the area.

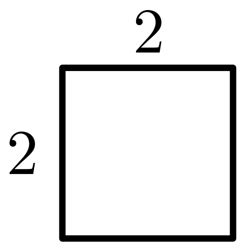
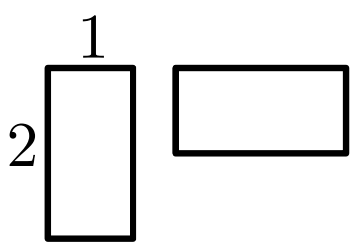
Goal:
Assume corner representation.
Find algorithms with running time $O(\text{poly}(n))$.
n: the number of corners.



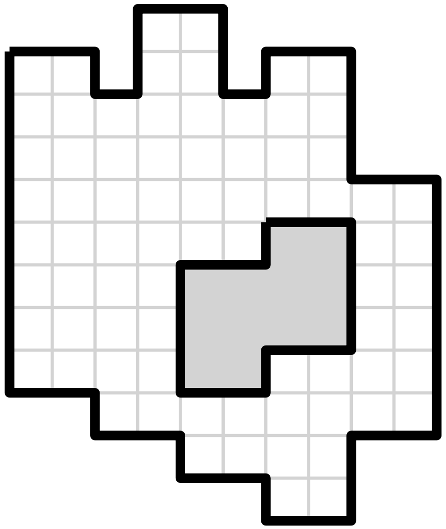
Results

Shapes	Tiling	Packing
	?	NP-complete
	?	?

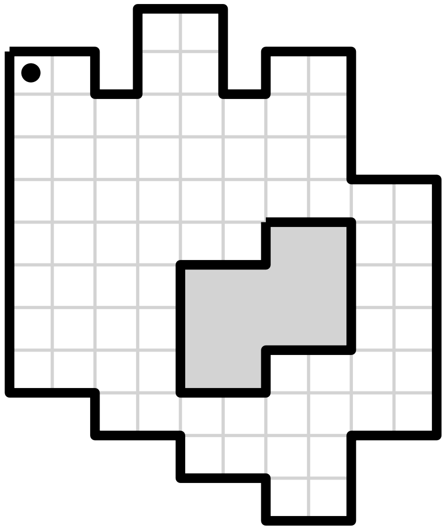
Results

Shapes	Tiling	Packing
	No holes: $O(n)$ Holes: $O(n \log n)$	NP-complete
	$\widetilde{O}(n^3)$	$\widetilde{O}(n^3)$

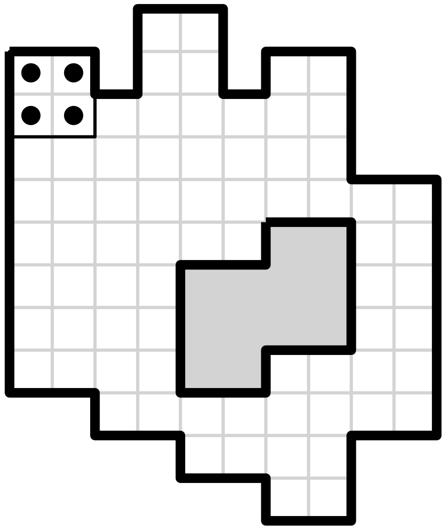
Tiling with 2×2 squares



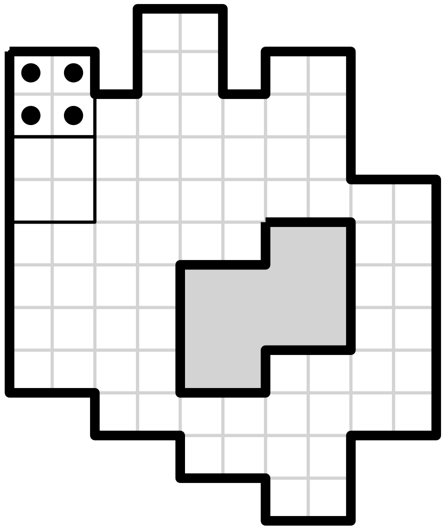
Tiling with 2×2 squares



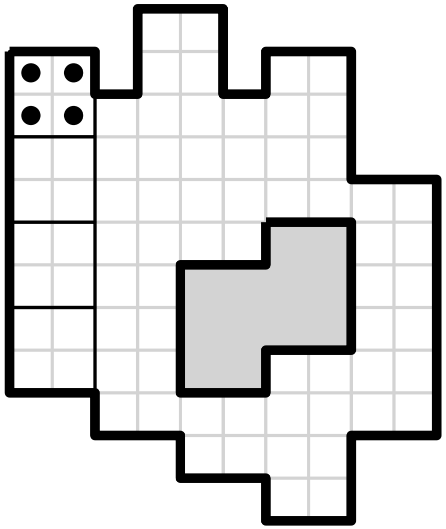
Tiling with 2×2 squares



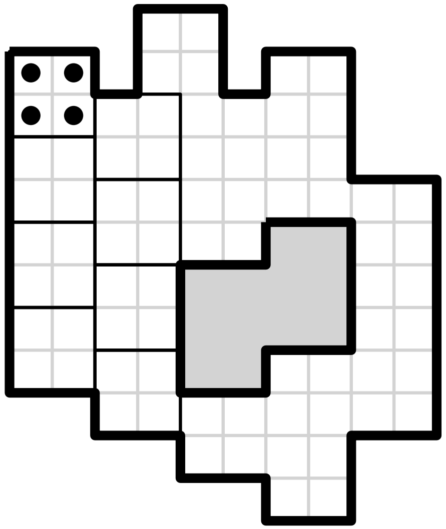
Tiling with 2×2 squares



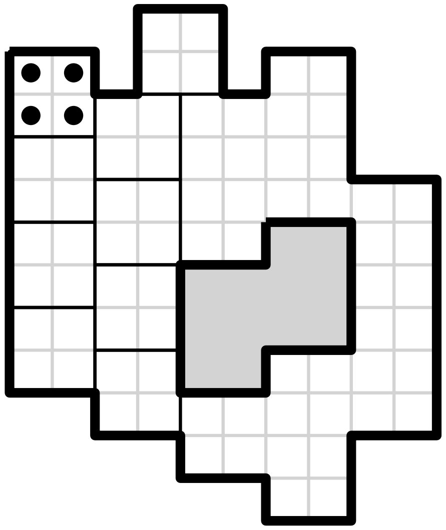
Tiling with 2×2 squares



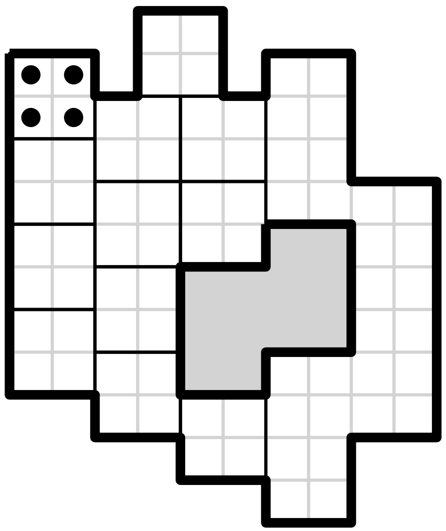
Tiling with 2×2 squares



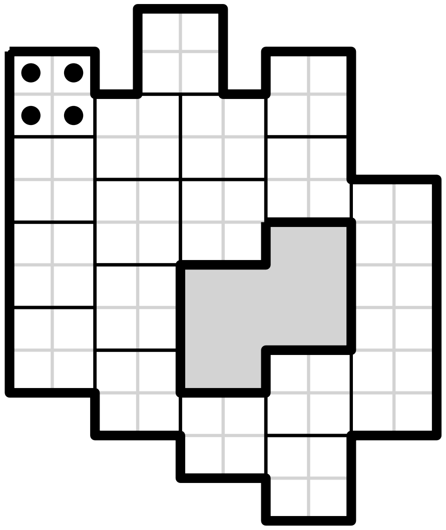
Tiling with 2×2 squares



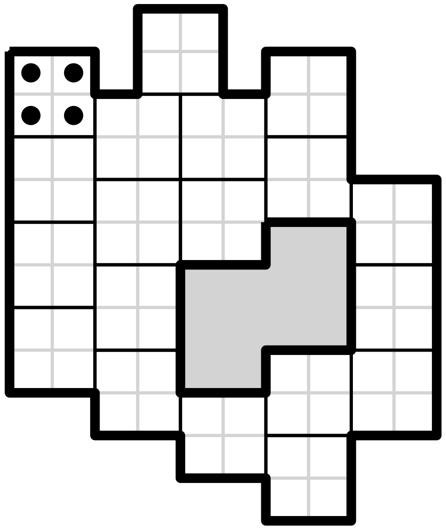
Tiling with 2×2 squares



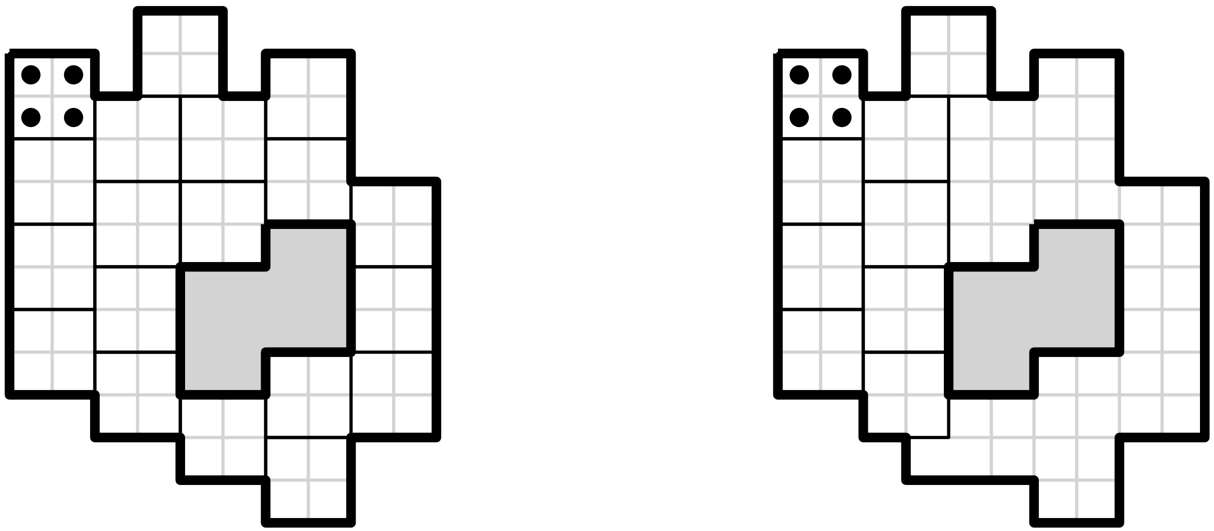
Tiling with 2×2 squares



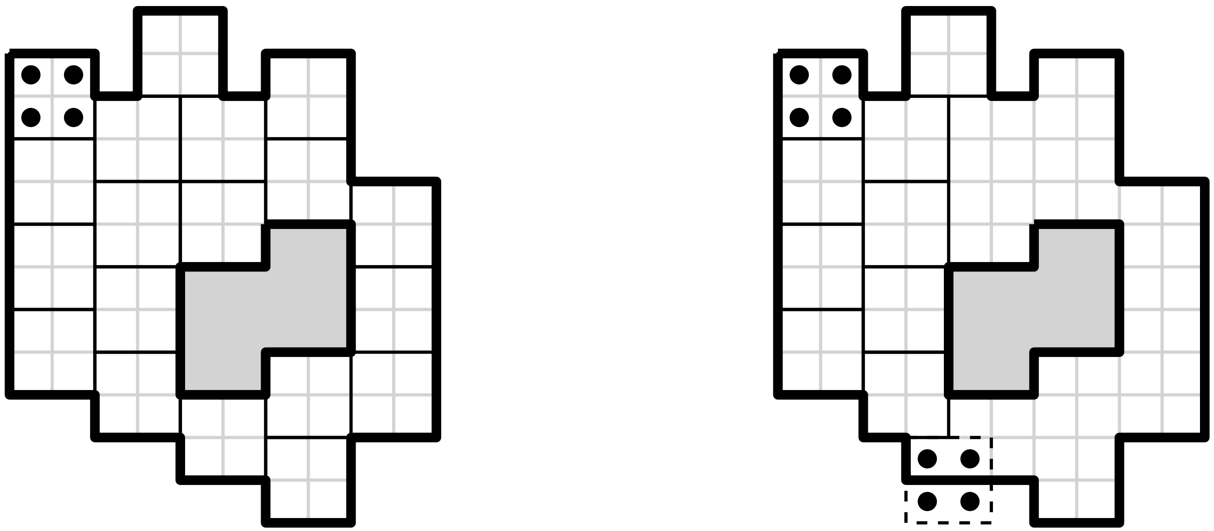
Tiling with 2×2 squares



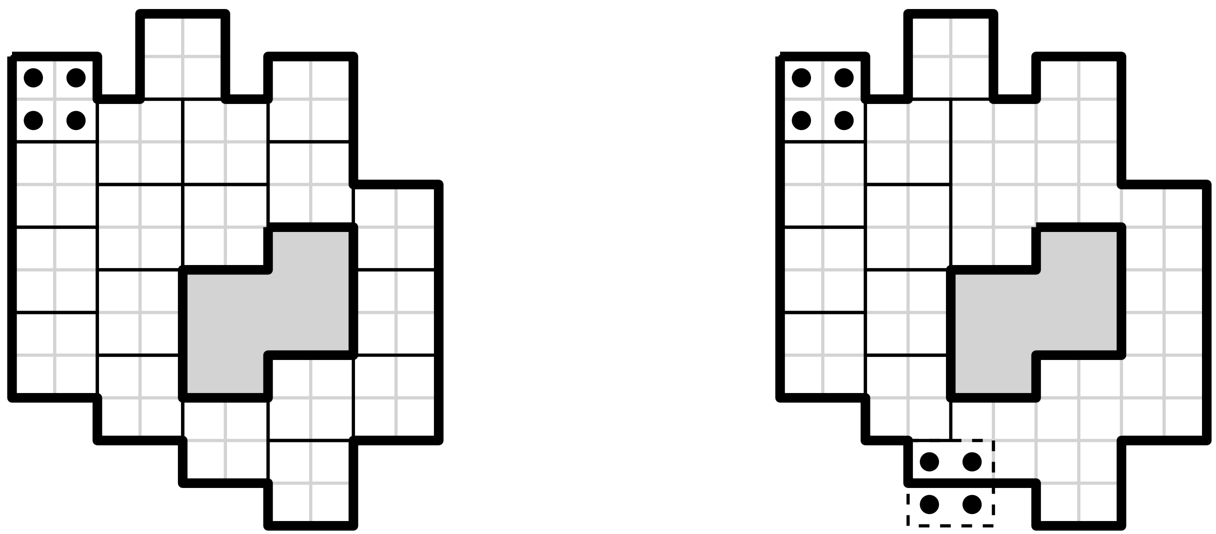
Tiling with 2×2 squares



Tiling with 2×2 squares

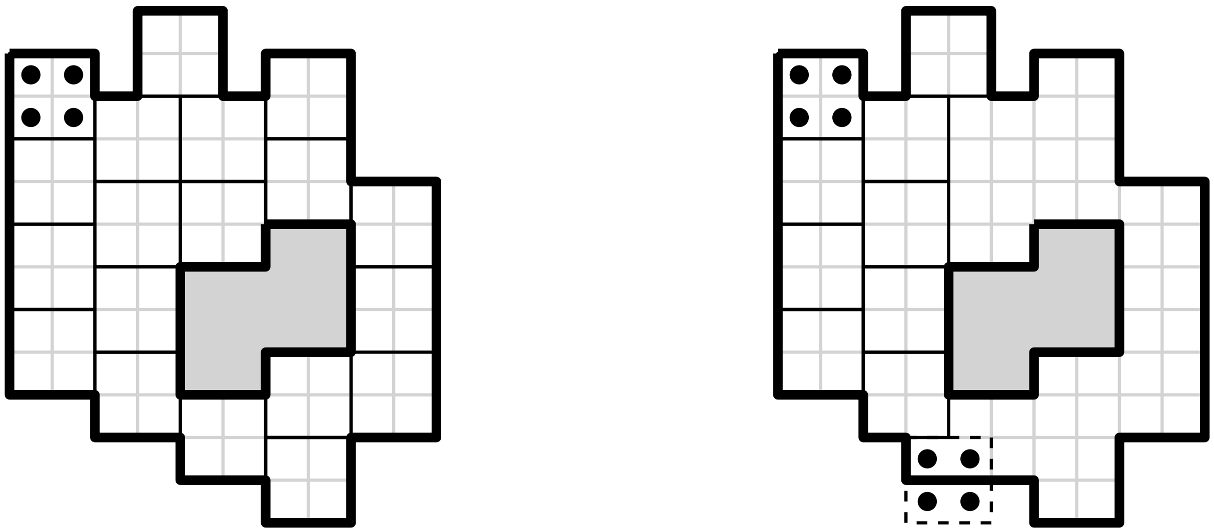


Tiling with 2×2 squares



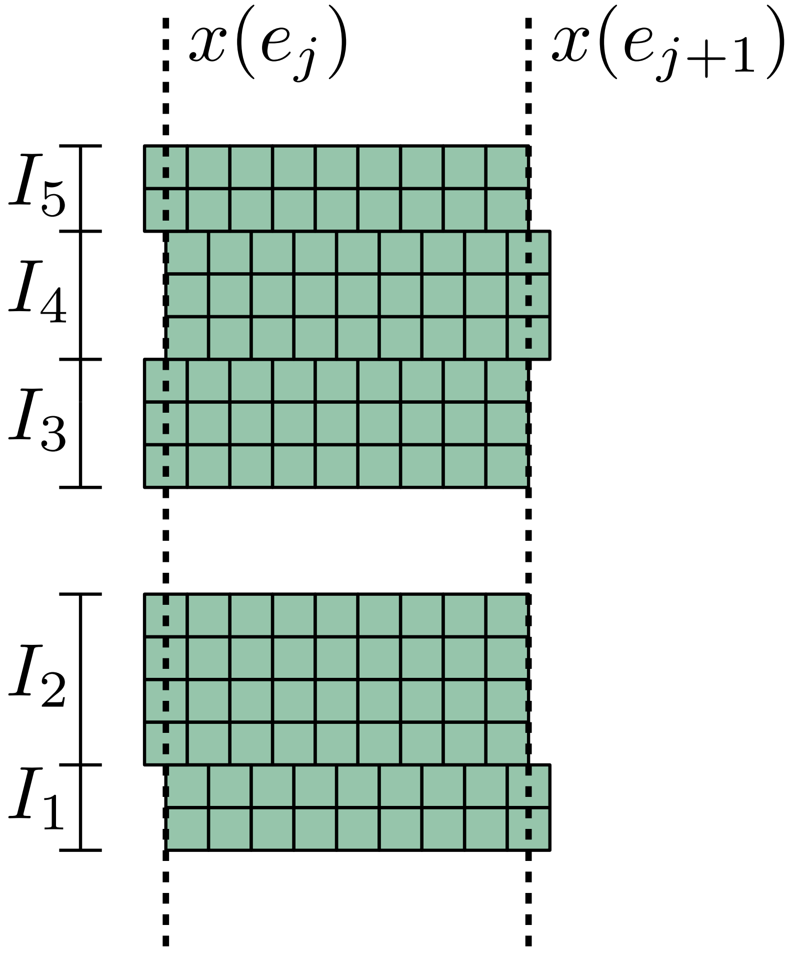
Can be done in $O(A)$ time.

Tiling with 2×2 squares

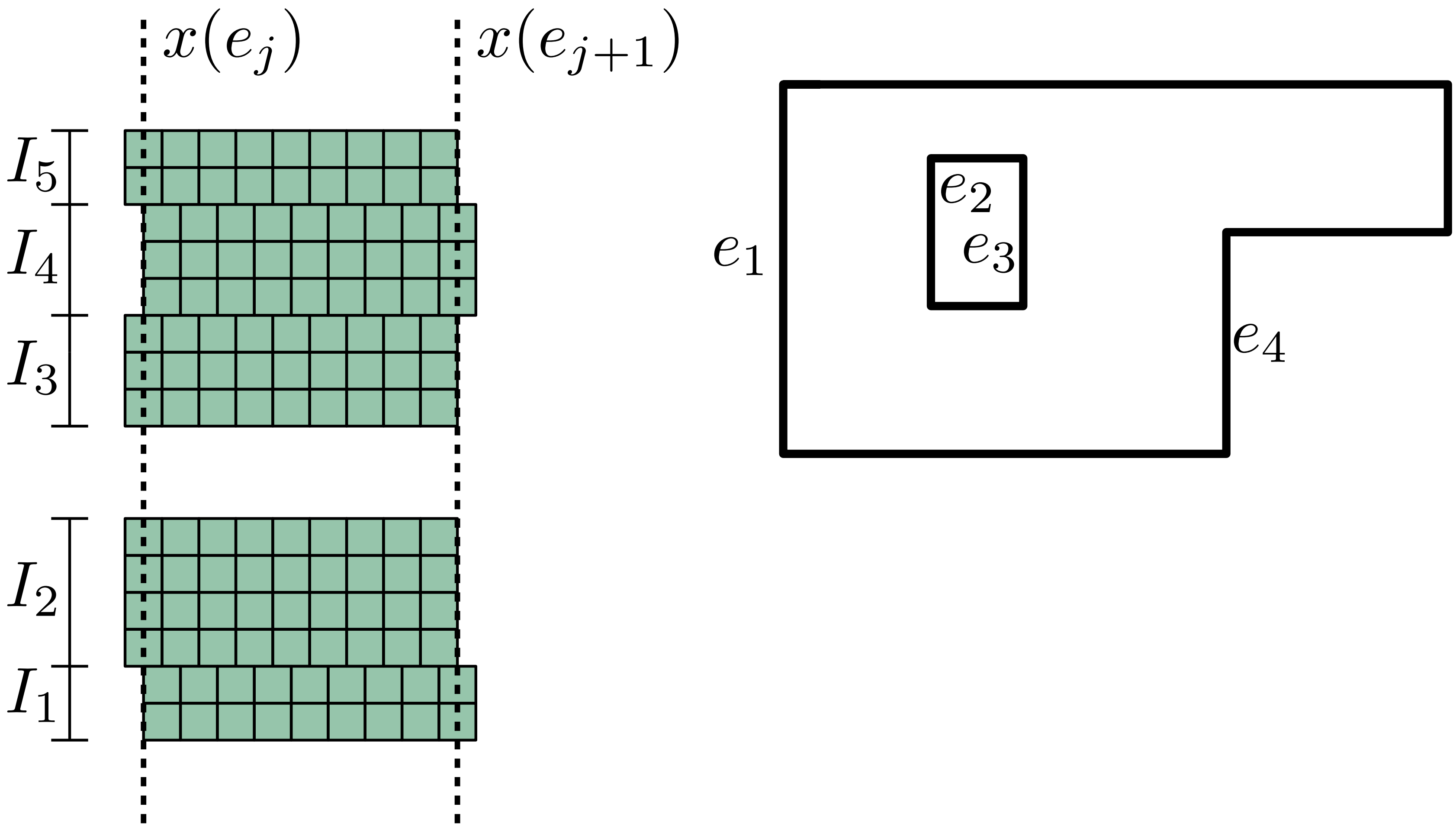


Can be done in $O(A)$ time.

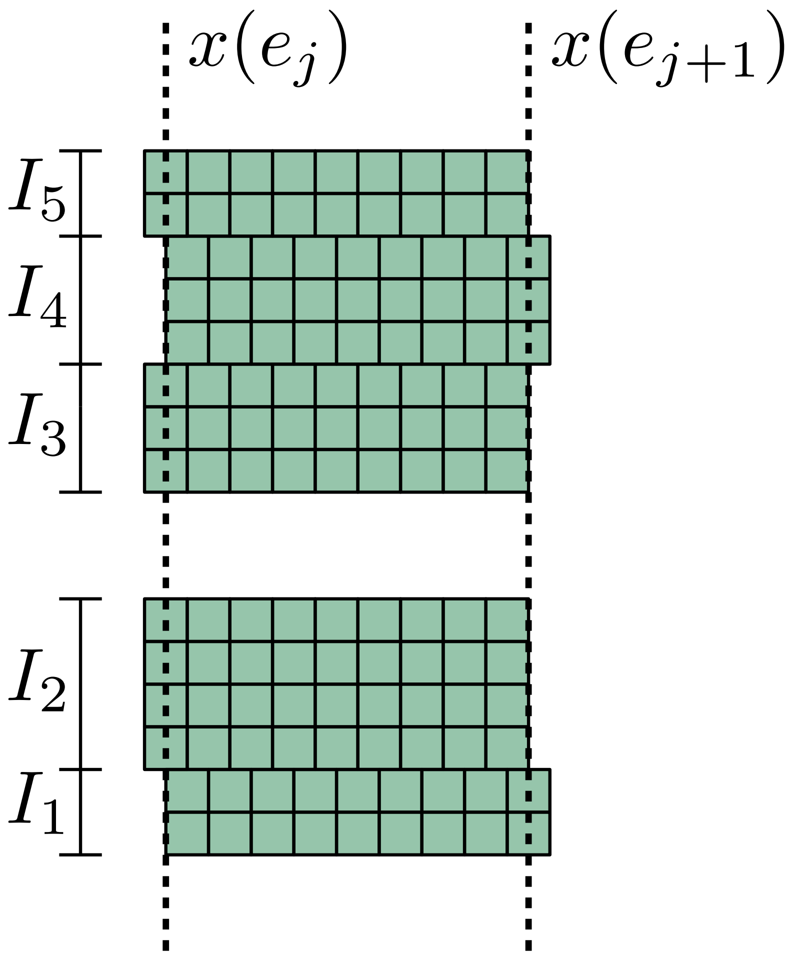
Polynomial-time algorithm but in the area of P!

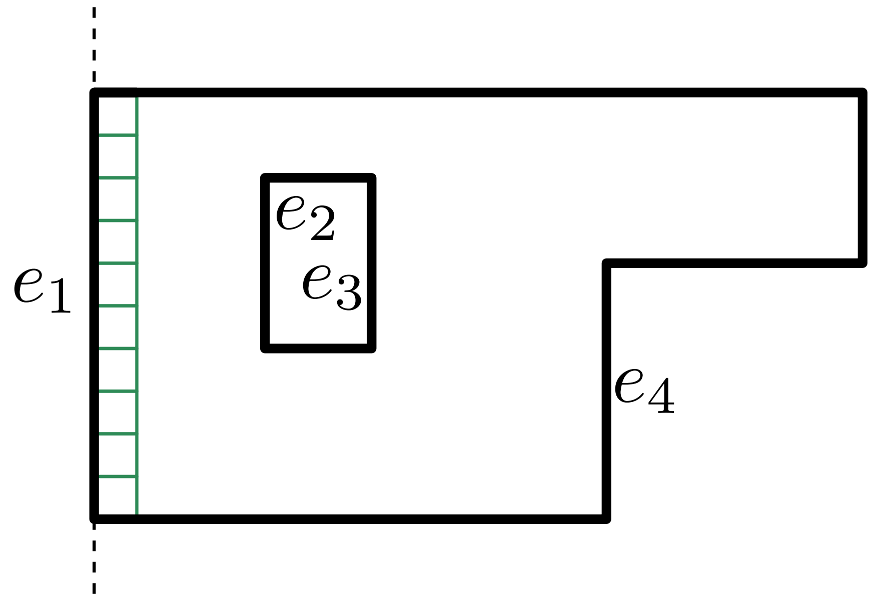


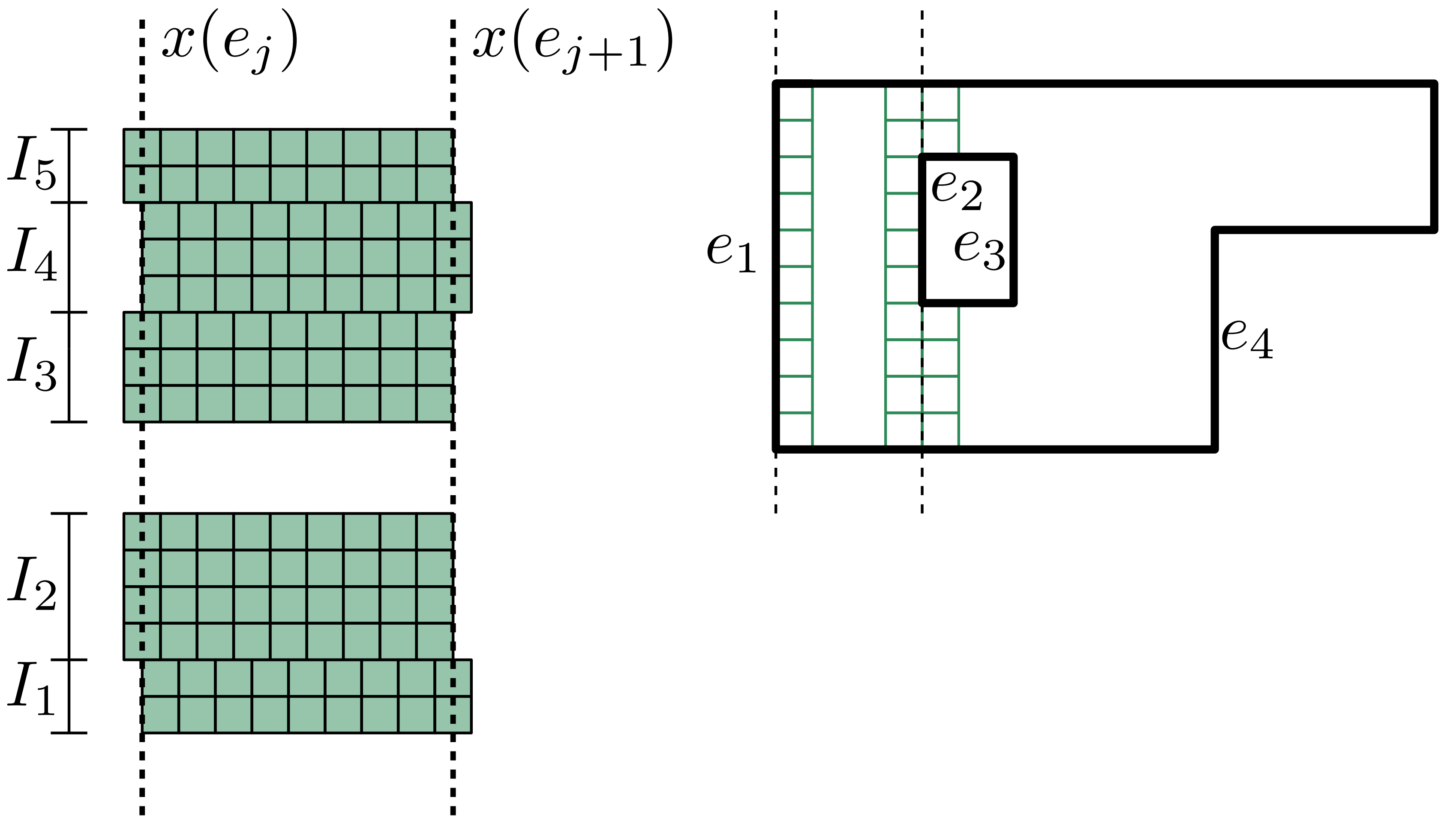
$x(e_j)$
$x(e_{j+1})$
I_5
I_4
I_3
I_2
I_1

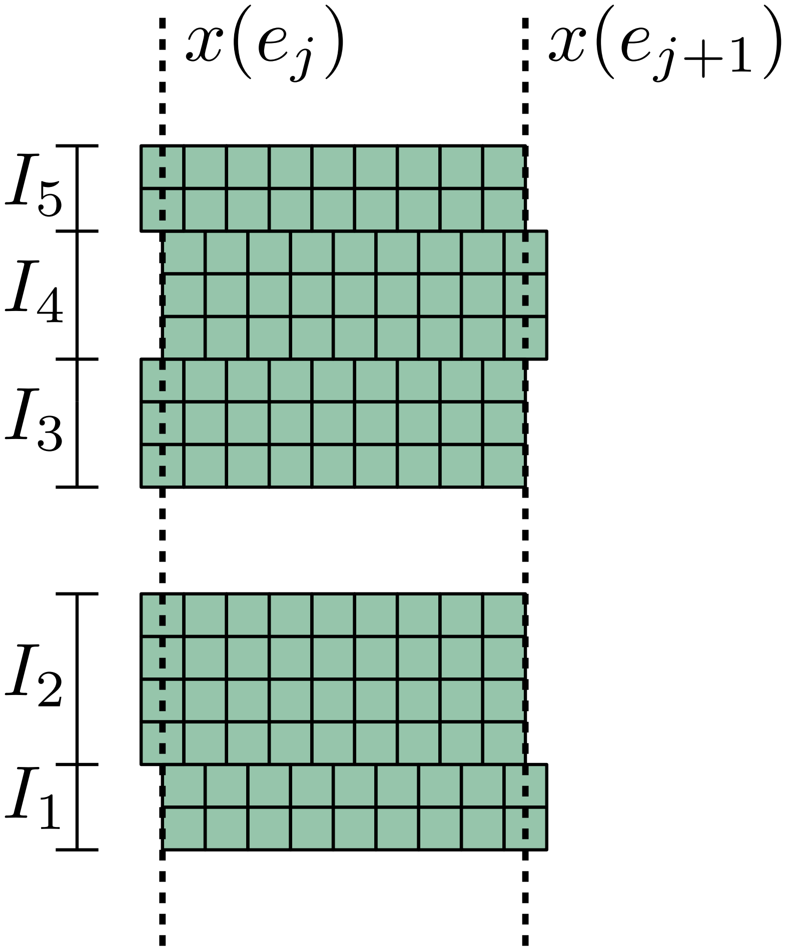


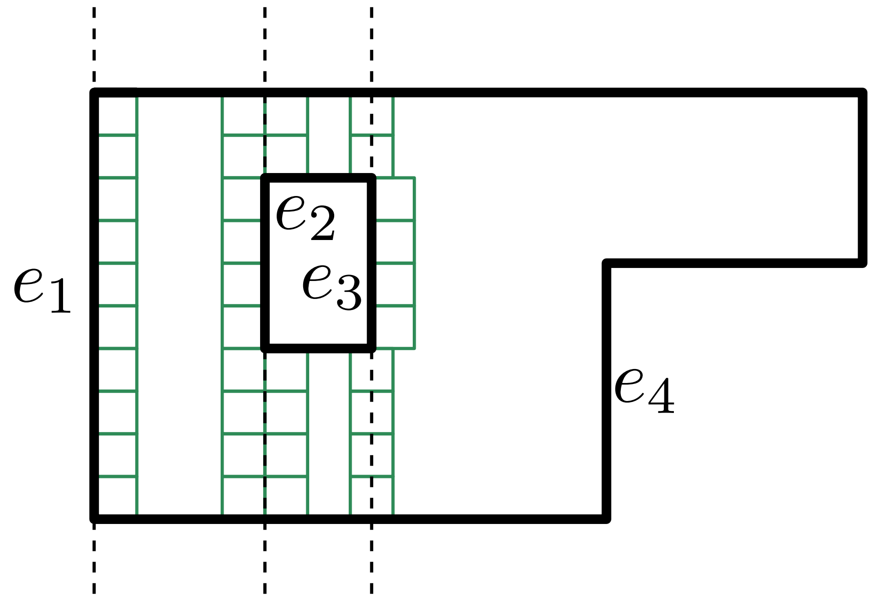
$x(e_j)$
$x(e_{j+1})$
I_5
I_4
I_3
I_2
I_1
e_1
e_2
e_3
e_4

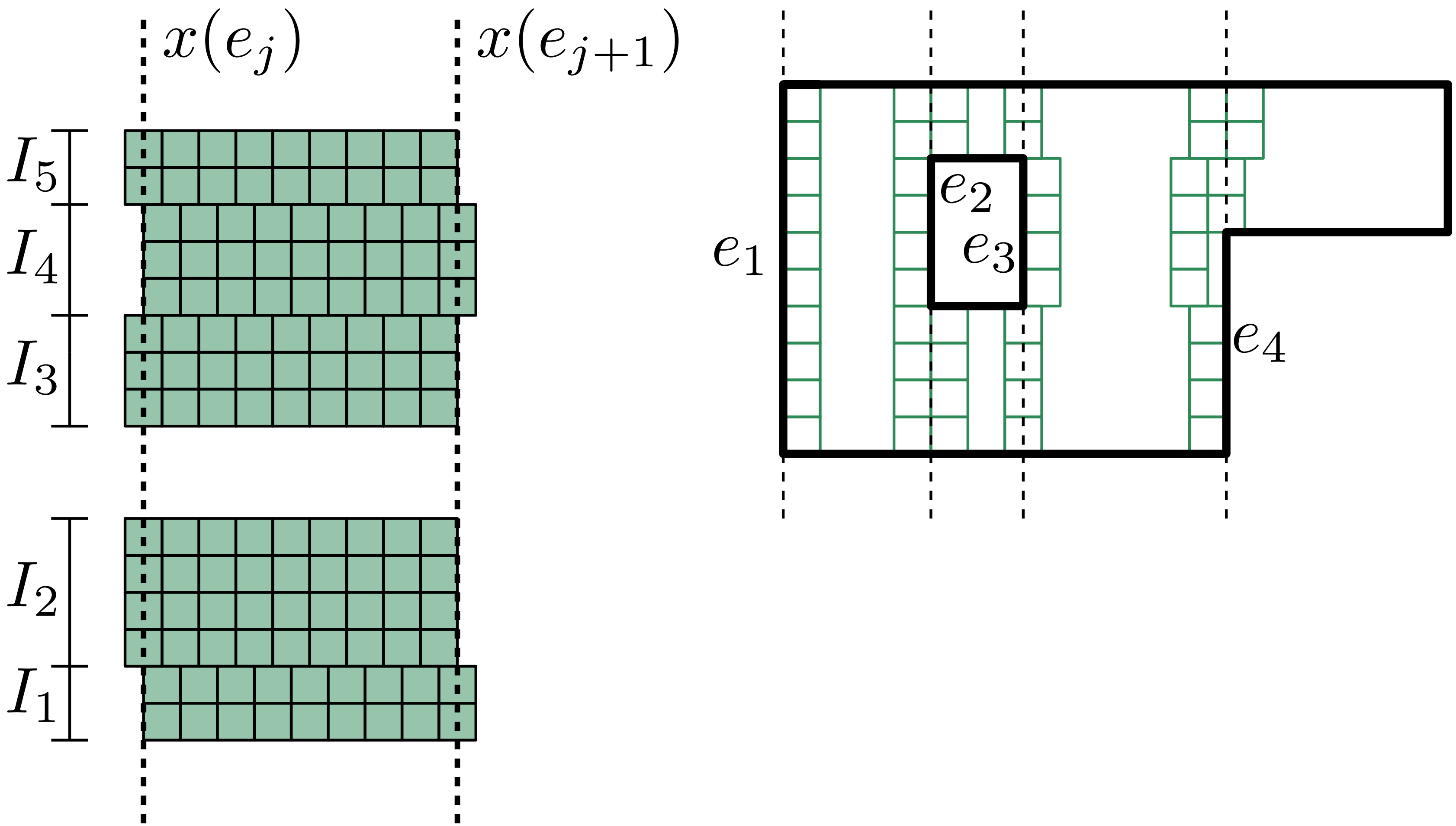

$x(e_j)$
$x(e_{j+1})$
I_5
I_4
I_3
I_2
I_1


e_1
e_2
e_3
e_4

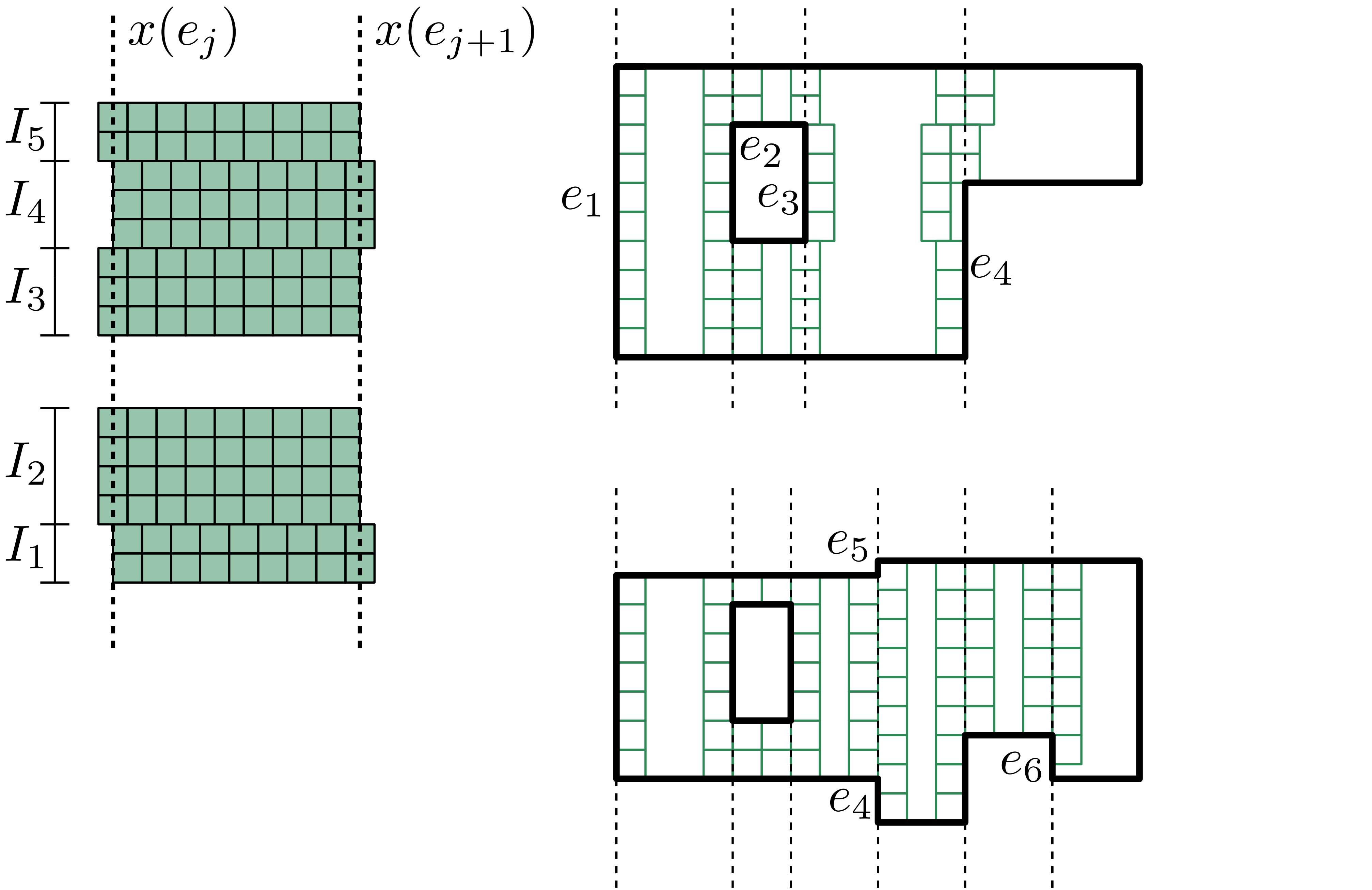

$x(e_j)$
$x(e_{j+1})$
I_5
I_4
I_3
I_2
I_1
e_1
e_2
e_3
e_4


$x(e_j)$
$x(e_{j+1})$
I_5
I_4
I_3
I_2
I_1


e_1
e_2
e_3
e_4

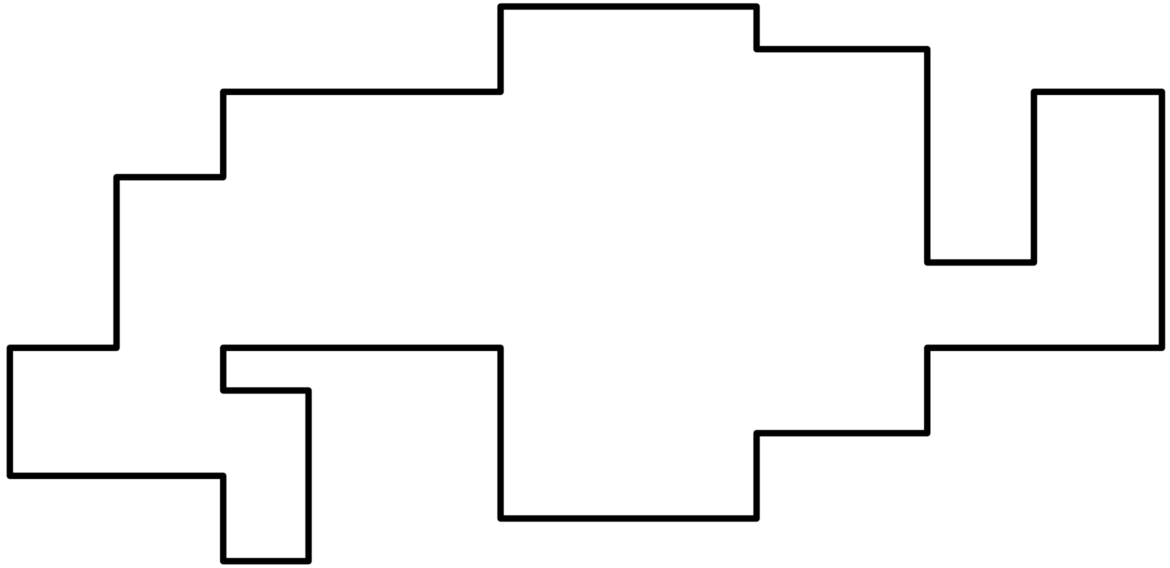


$x(e_j)$
$x(e_{j+1})$
I_5
I_4
I_3
I_2
I_1
e_1
e_2
e_3
e_4

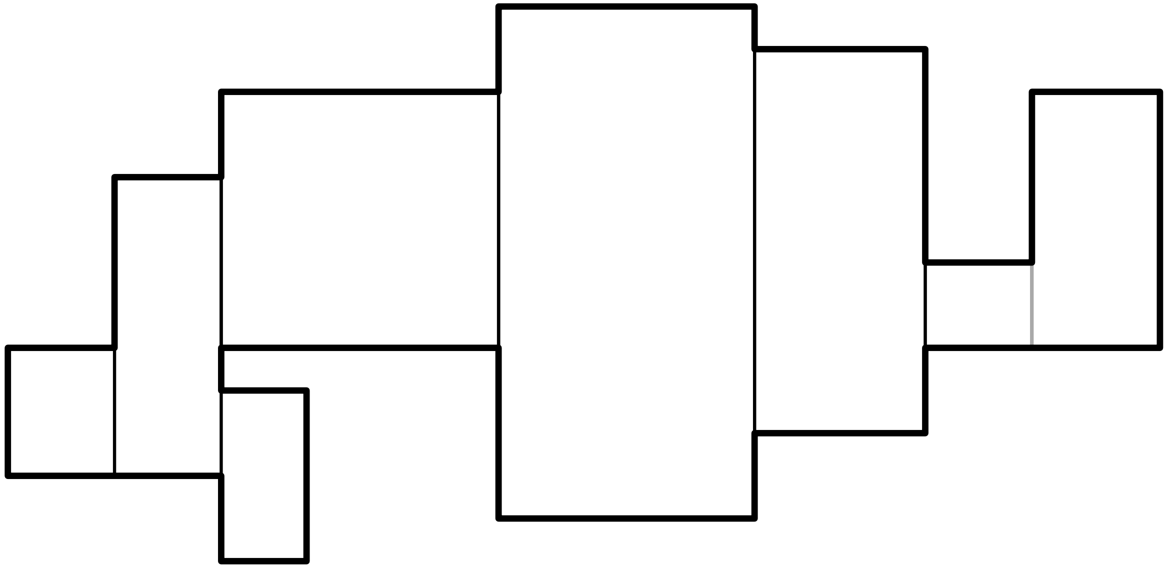


$x(e_j)$
$x(e_{j+1})$
I_5
I_4
I_3
I_2
I_1
e_1
e_2
e_3
e_4
e_5
e_4
e_6

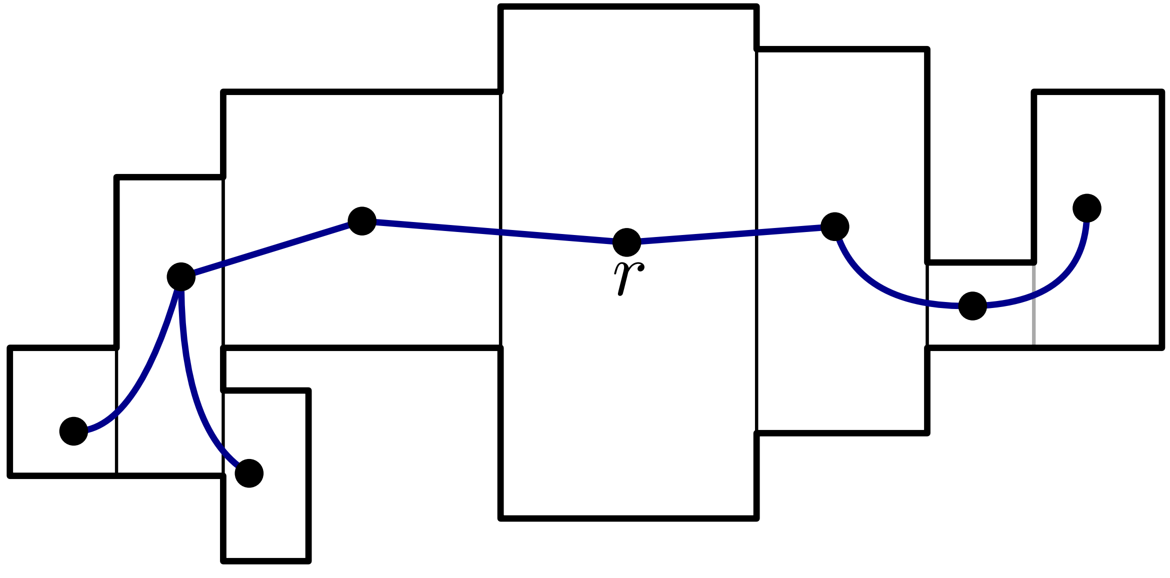
No holes: $O(n)$ time!



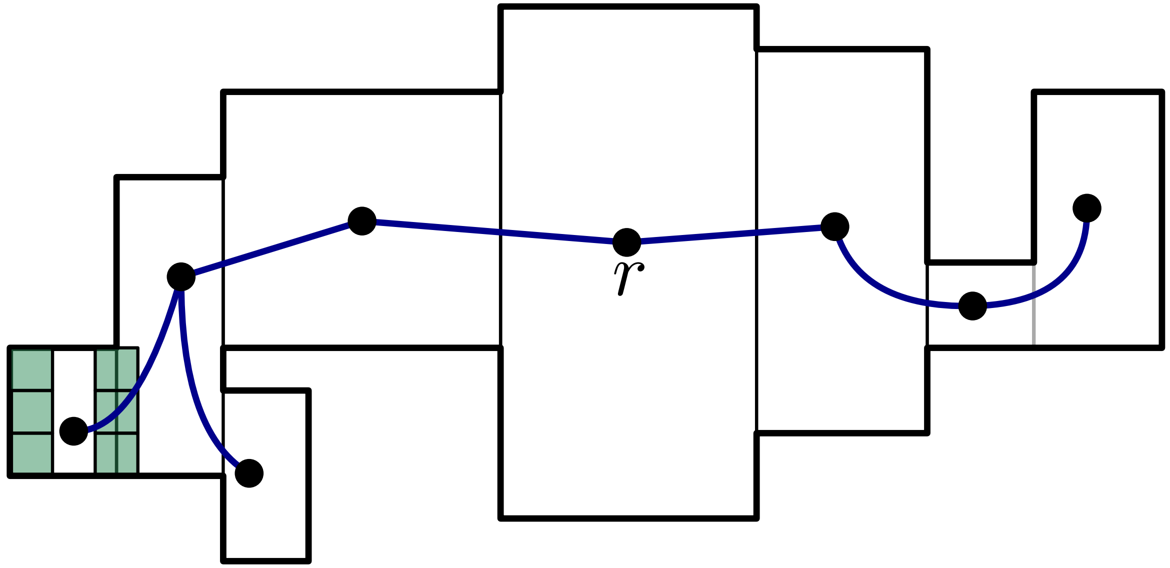
No holes: $O(n)$ time!



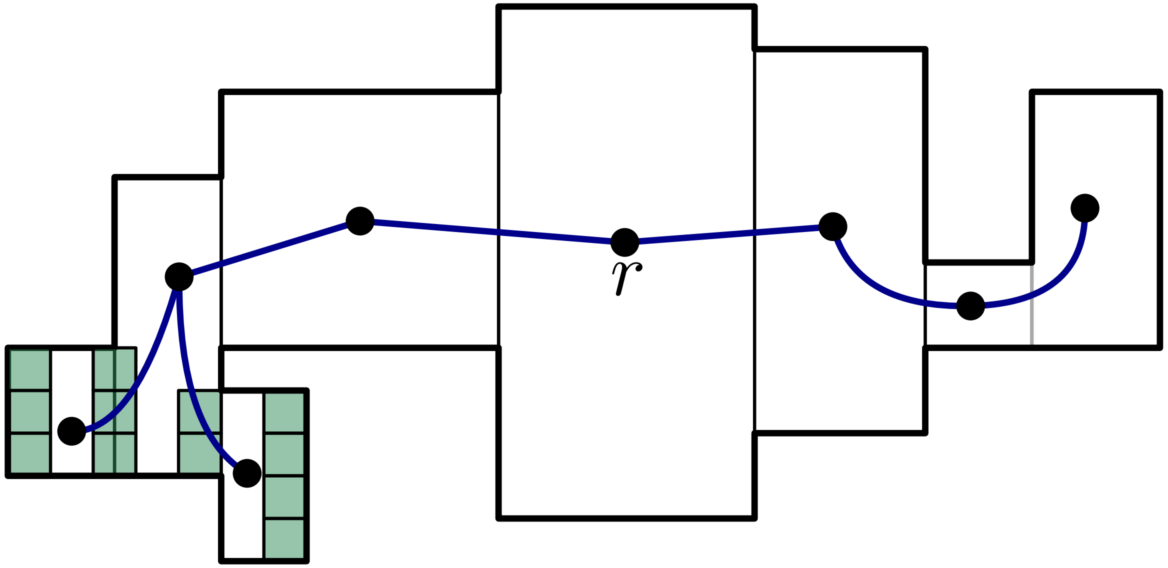
No holes: $O(n)$ time!



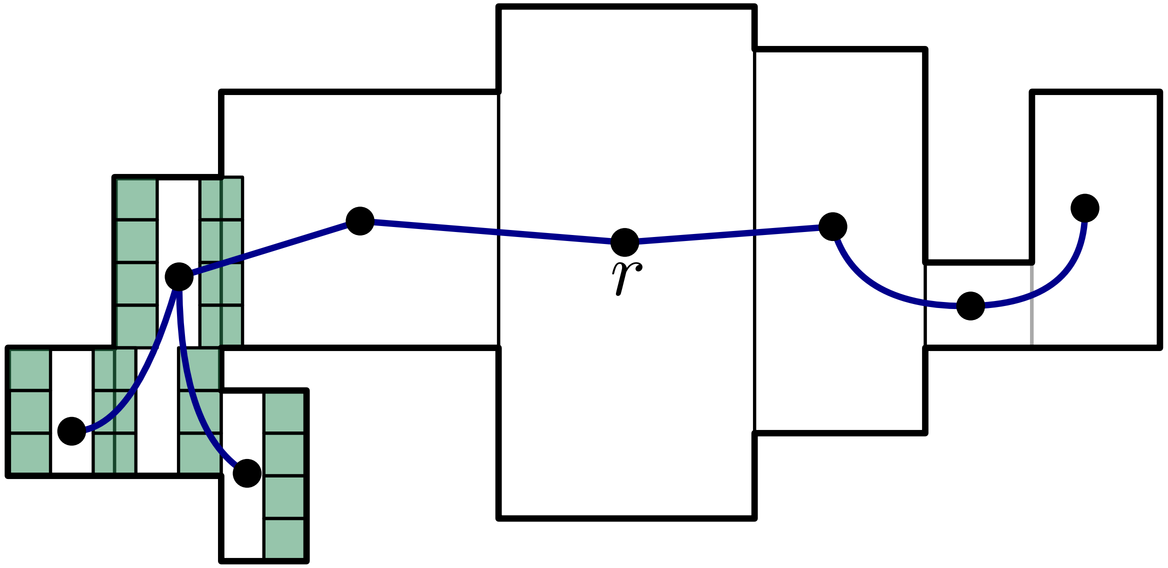
No holes: $O(n)$ time!



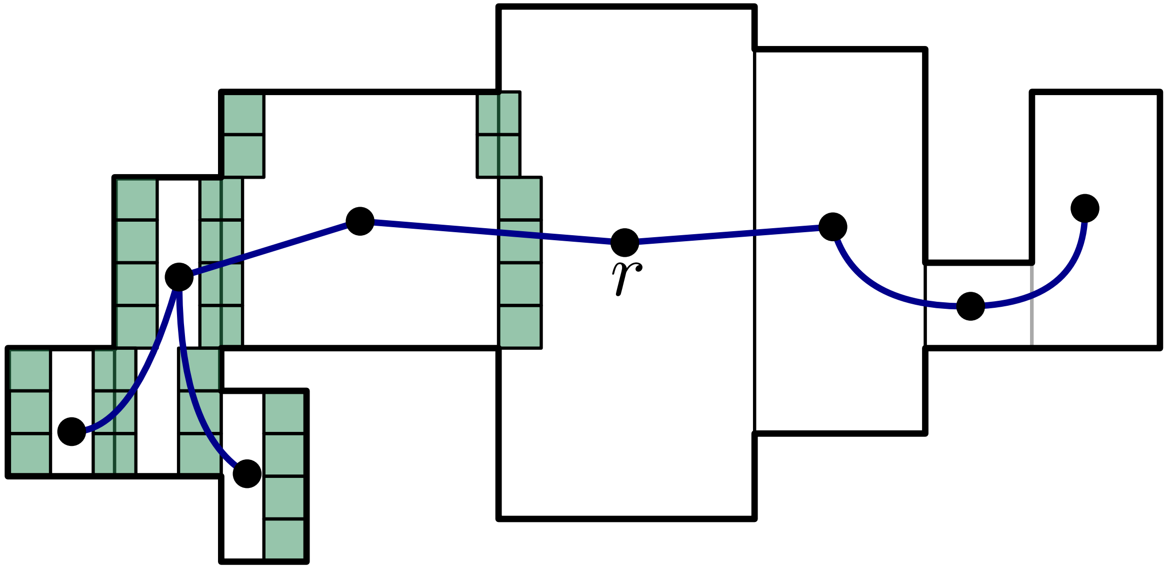
No holes: $O(n)$ time!



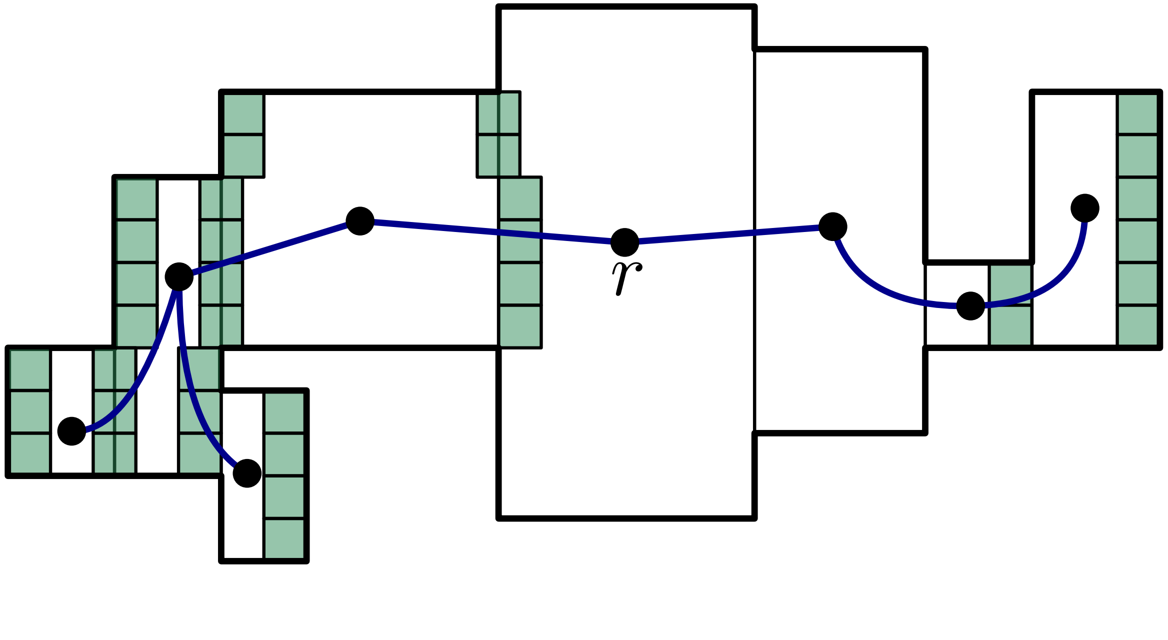
No holes: $O(n)$ time!



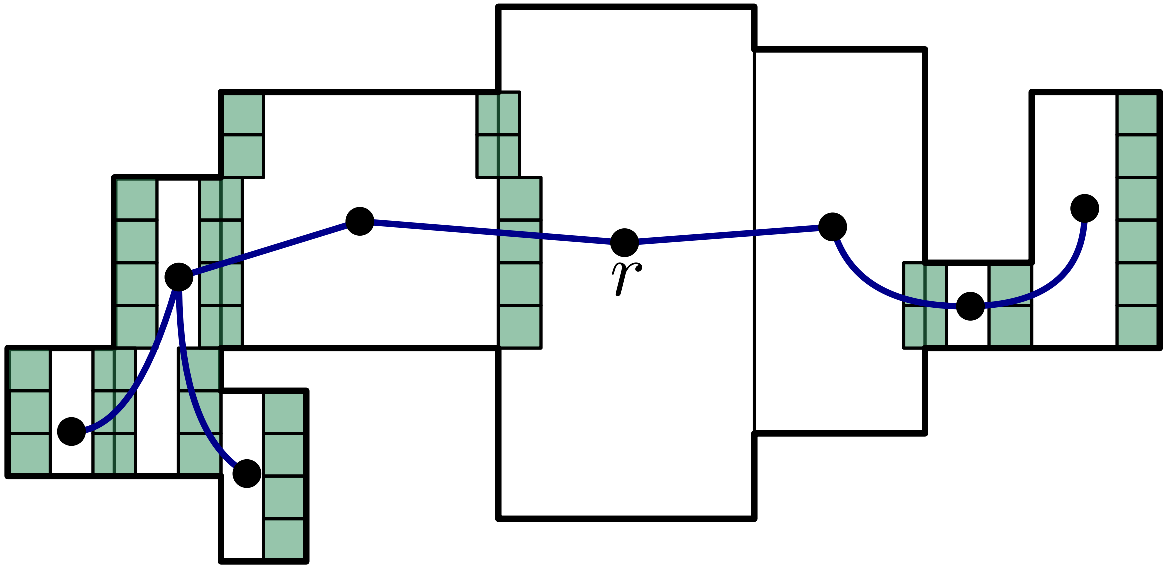
No holes: $O(n)$ time!



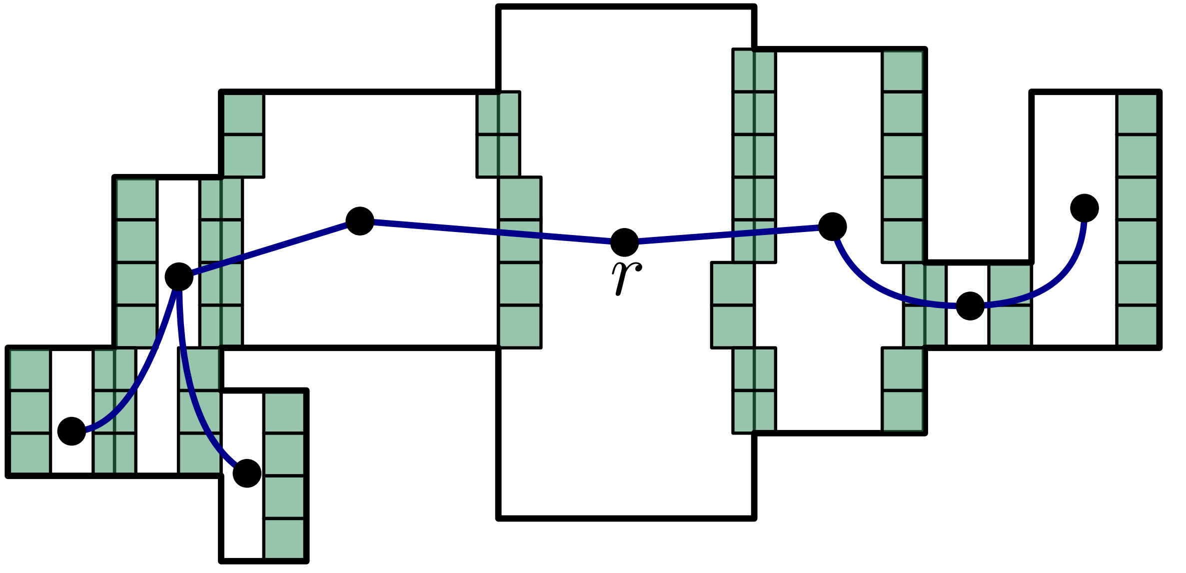
No holes: $O(n)$ time!



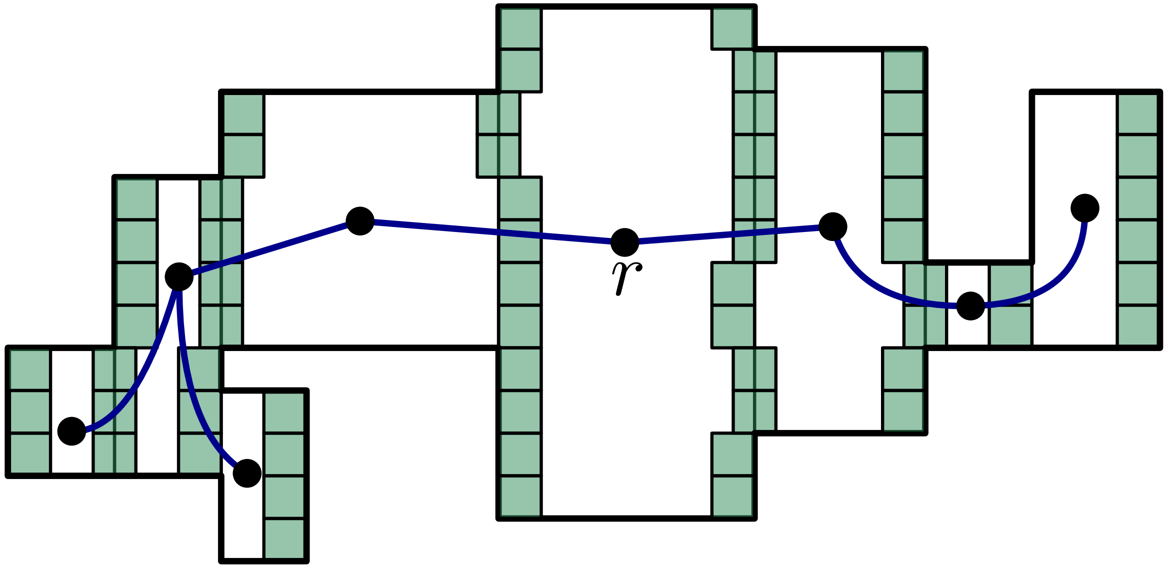
No holes: $O(n)$ time!



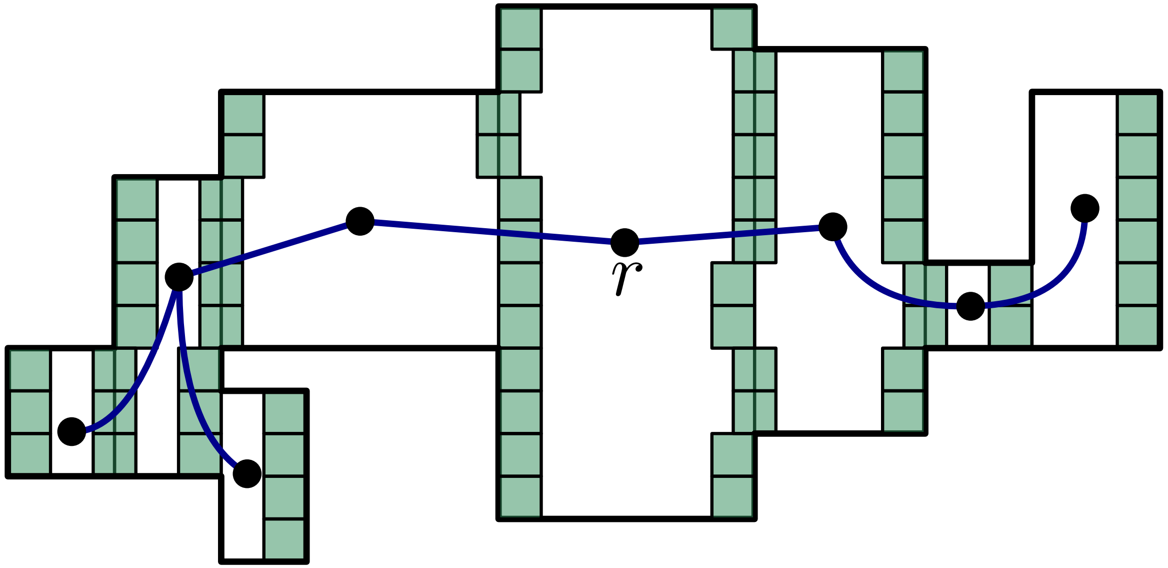
No holes: $O(n)$ time!

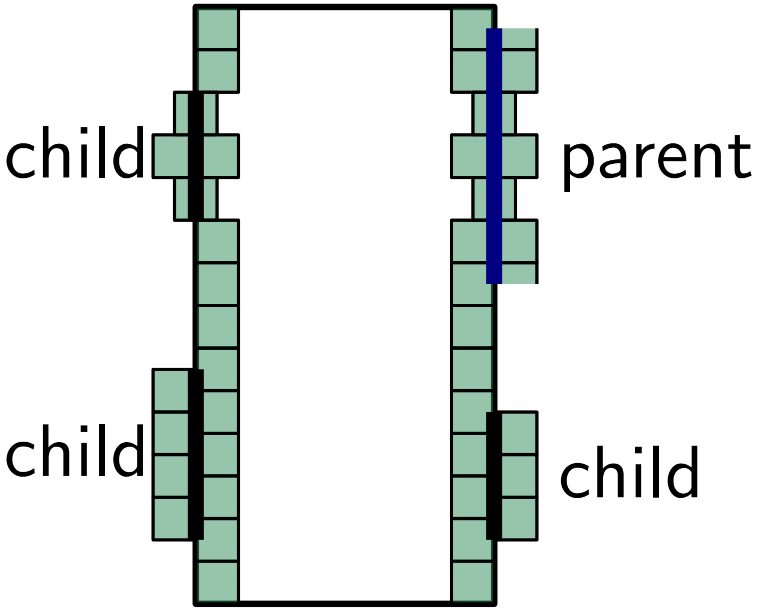


No holes: $O(n)$ time!

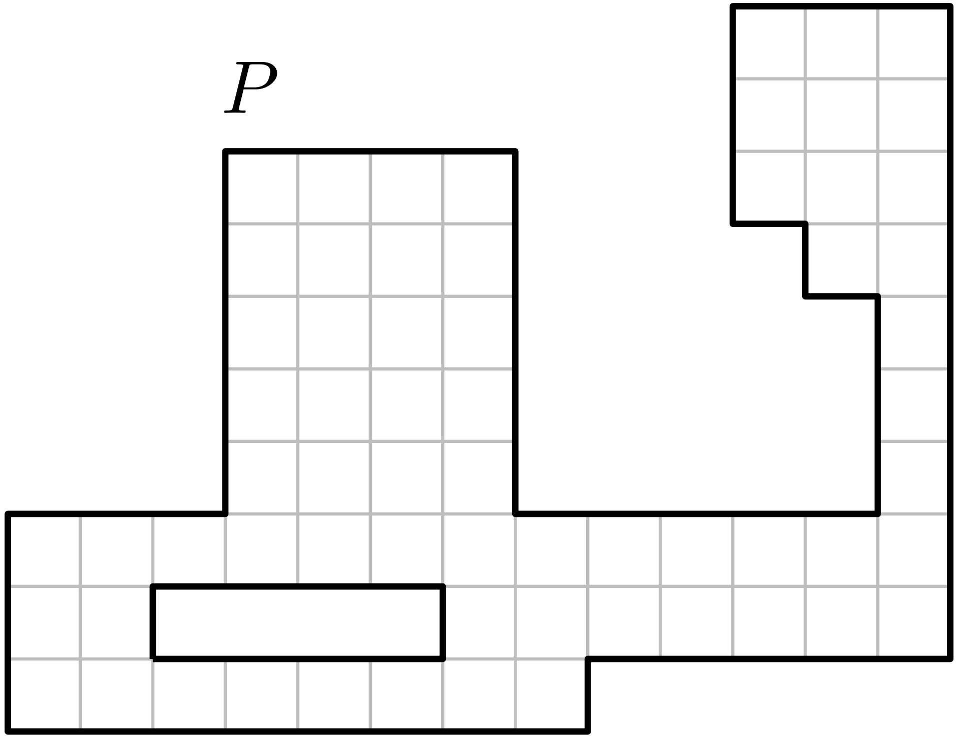


No holes: $O(n)$ time!

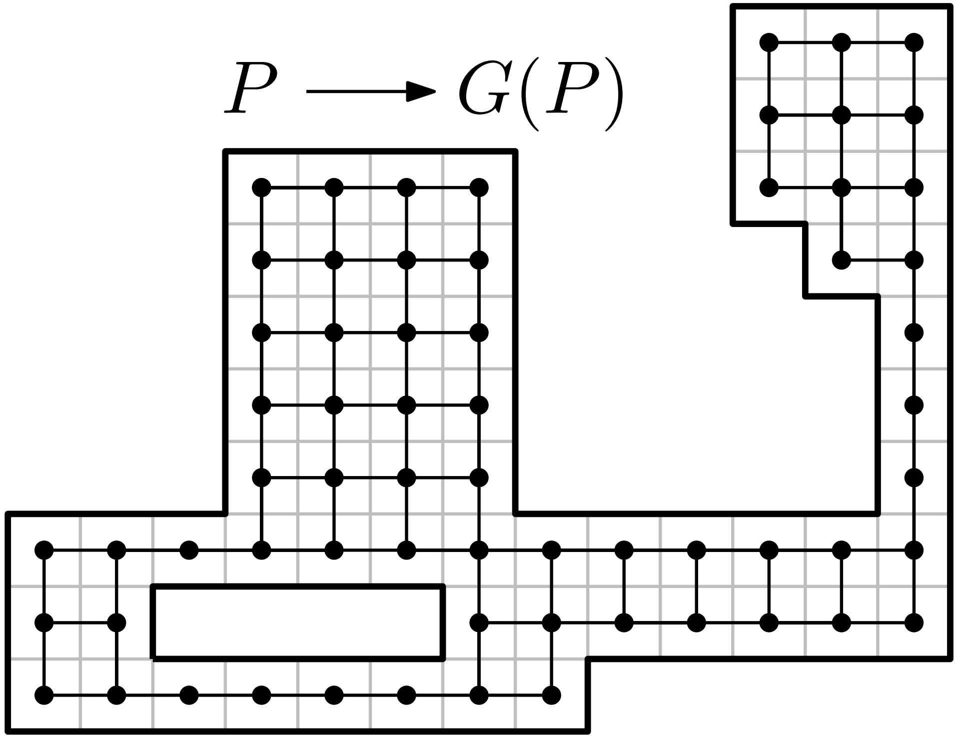




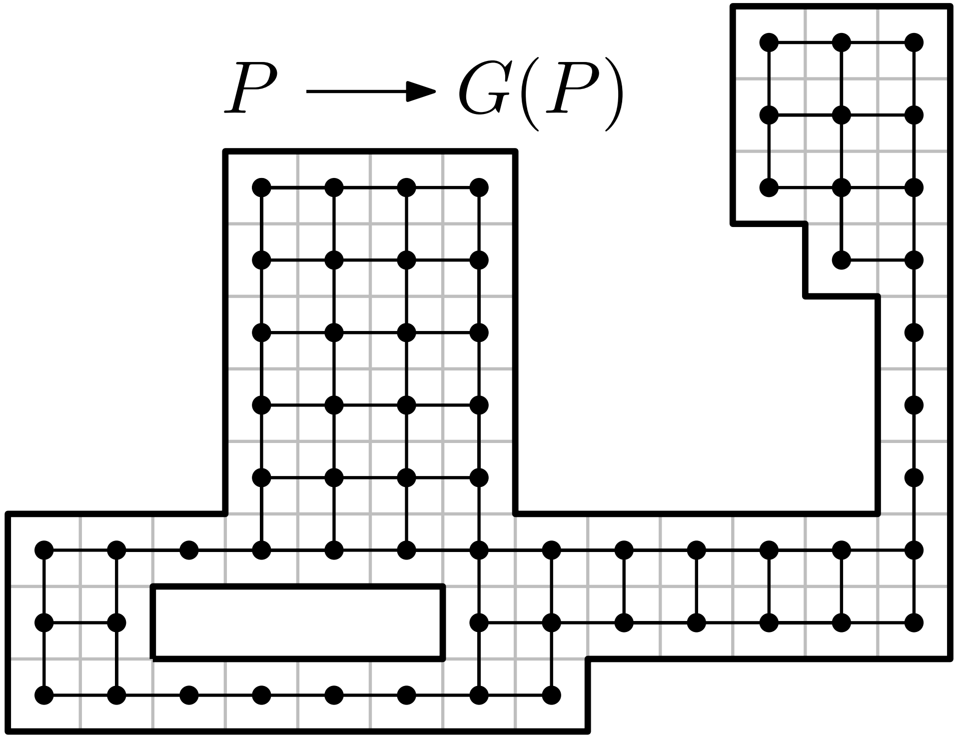
Packing dominos



Packing dominos

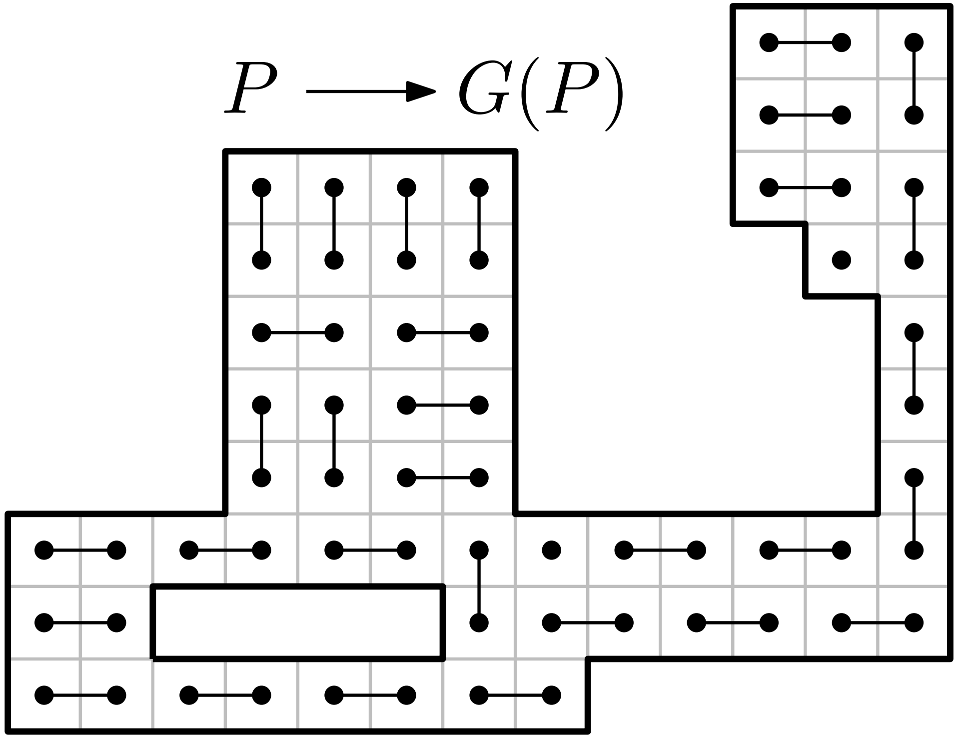


Packing dominos



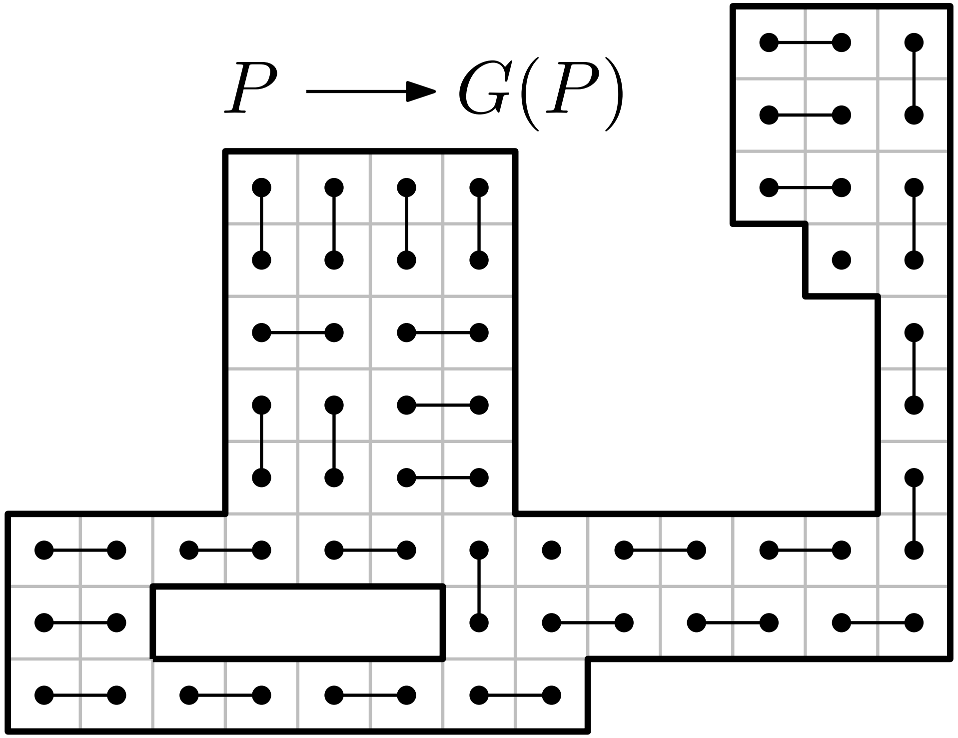
Maximum domino packing of $P \leftrightarrow$ Maximum matching of $G(P)$

Packing dominos



Maximum domino packing of $P \leftrightarrow$ Maximum matching of $G(P)$

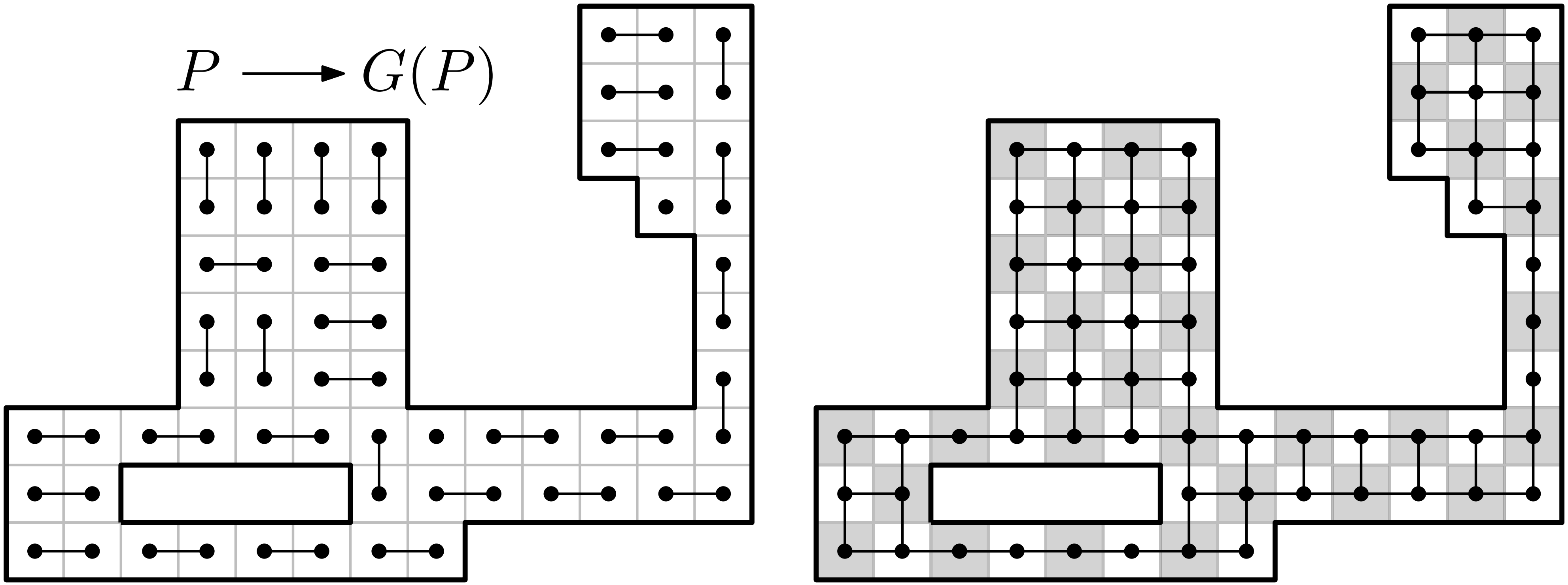
Packing dominos



Maximum domino packing of $P \leftrightarrow$ Maximum matching of $G(P)$

Time $O(A^{3/2})$ for maximum domino packing using Hopcroft-Karp, where A is the *area* of P (Berman et al. '82)

Packing dominos



Maximum domino packing of $P \leftrightarrow$ Maximum matching of $G(P)$

Time $O(A^{3/2})$ for maximum domino packing using Hopcroft-Karp, where A is the *area* of P (Berman et al. '82)

Multiple source multiple sink maximum flow: $\widetilde{O}(A)$ [Borradaile et al., SICOMP 2017].

Related work – Algorithmic

Conway & Lagarias '90 and Thurston '90:
Combinatorial Group Theory approach for deciding tileability.

Related work – Algorithmic

Conway & Lagarias '90 and Thurston '90:
Combinatorial Group Theory approach for deciding tileability.

$\Longrightarrow O(A \log A)$ alg. for tiling with dominos.

Related work – Algorithmic

Conway & Lagarias '90 and Thurston '90:
Combinatorial Group Theory approach for deciding tileability.
$\Longrightarrow O(A \log A)$ alg. for tiling with dominos.

Kenyon & Kenyon '92:
Tiling a hole-free polyomino with $1 \times m$ and $k \times 1$ rectangles in time $O(A)$.
Tiling a hole-free polyomino with $k \times m$ and $m \times k$ rectangles in time $O(A^2)$.

Related work – Algorithmic

Conway & Lagarias '90 and Thurston '90:
Combinatorial Group Theory approach for deciding tileability.
$\Longrightarrow O(A \log A)$ alg. for tiling with dominos.

Kenyon & Kenyon '92:
Tiling a hole-free polyomino with $1 \times m$ and $k \times 1$ rectangles in time $O(A)$.
Tiling a hole-free polyomino with $k \times m$ and $m \times k$ rectangles in time $O(A^2)$.

Remila '05:
Tiling a hole-free polyomino with $k \times m$ and $k' \times m'$ rectangles in time $O(A^2)$

Related work – Algorithmic

All polynomial in the area!

Conway & Lagarias '90 and Thurston '90:
Combinatorial Group Theory approach for deciding tileability.
$\Longrightarrow O(A \log A)$ alg. for tiling with dominos.

Kenyon & Kenyon '92:
Tiling a hole-free polyomino with $1 \times m$ and $k \times 1$ rectangles in time $O(A)$.
Tiling a hole-free polyomino with $k \times m$ and $m \times k$ rectangles in time $O(A^2)$.

Remila '05:
Tiling a hole-free polyomino with $k \times m$ and $k' \times m'$ rectangles in time $O(A^2)$

Related work – Algorithmic

All polynomial
in the area!

Conway & Lagarias '90 and Thurston '90:
Combinatorial Group Theory approach for deciding tileability.
$\Longrightarrow O(A \log A)$ alg. for tiling with dominos.

Kenyon & Kenyon '92:
Tiling a hole-free polyomino with $1 \times m$ and $k \times 1$ rectangles in time $O(A)$.
Tiling a hole-free polyomino with $k \times m$ and $m \times k$ rectangles in time $O(A^2)$.

Remila '05:
Tiling a hole-free polyomino with $k \times m$ and $k' \times m'$ rectangles in time $O(A^2)$

Related work – Hardness

Beauquier et al. '95, Pak & Yang '13, Fowler et al. '81, **Berman et al '90**:
Various hardness results for tiling and packing.

Related work – Algorithmic

All polynomial
in the area!

Conway & Lagarias '90 and Thurston '90:
Combinatorial Group Theory approach for deciding tileability.
$\Longrightarrow O(A \log A)$ alg. for tiling with dominos.

Kenyon & Kenyon '92:
Tiling a hole-free polyomino with $1 \times m$ and $k \times 1$ rectangles in time $O(A)$.
Tiling a hole-free polyomino with $k \times m$ and $m \times k$ rectangles in time $O(A^2)$.

Remila '05:
Tiling a hole-free polyomino with $k \times m$ and $k' \times m'$ rectangles in time $O(A^2)$

Related work – Hardness

Beauquier et al. '95, Pak & Yang '13, Fowler et al. '81, **Berman et al '90**:
Various hardness results for tiling and packing.

Berman et al '90:
Deciding if k 2×2 squares can be packed in a polyomino (with holes) is NP-complete.

Related work – Algorithmic

All polynomial
in the area!

Conway & Lagarias '90 and Thurston '90:
Combinatorial Group Theory approach for deciding tileability.
$\Longrightarrow O(A \log A)$ alg. for tiling with dominos.

Kenyon & Kenyon '92:
Tiling a hole-free polyomino with $1 \times m$ and $k \times 1$ rectangles in time $O(A)$.
Tiling a hole-free polyomino with $k \times m$ and $m \times k$ rectangles in time $O(A^2)$.

Remila '05:
Tiling a hole-free polyomino with $k \times m$ and $k' \times m'$ rectangles in time $O(A^2)$

Related work – Hardness

Beauquier et al. '95, Pak & Yang '13, Fowler et al. '81, **Berman et al '90**:
Various hardness results for tiling and packing.

Berman et al '90:
Deciding if k 2×2 squares can be packed in a polyomino (with holes) is NP-complete.

Berger '66:
Deciding if a finite set of polyominos can tile the plane is Turing-complete

This Talk

Packing Dominos in $\widetilde{O}(n^3)$ time

This Talk

Packing Dominos in $\widetilde{O}(n^3)$ time

Assume no holes

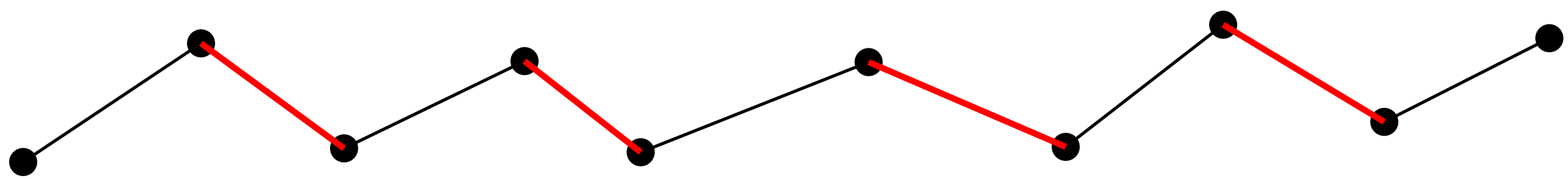
Augmenting paths

Let G be a graph, M a matching of G.

Augmenting paths

Let G be a graph, M a matching of G.

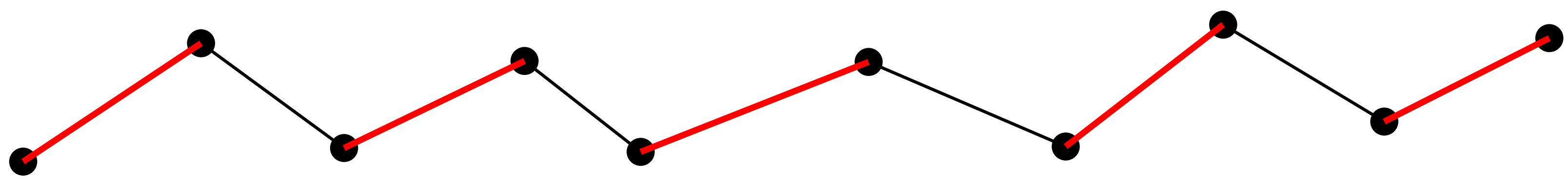
A path $P = v_1, v_2, \ldots, v_{2k}$ of G is **augmenting** if v_1 and v_{2k} are unmatched and $(v_{2i}, v_{2i+1}) \in M$, $i = 1, \ldots, k-1$



Augmenting paths

Let G be a graph, M a matching of G.

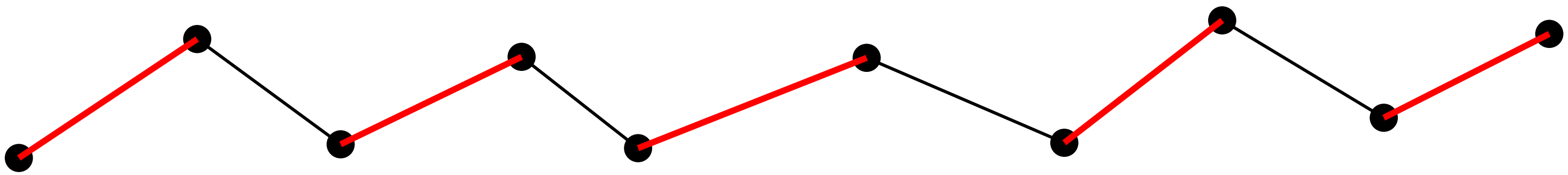
A path $P = v_1, v_2, \ldots, v_{2k}$ of G is **augmenting** if v_1 and v_{2k} are unmatched and $(v_{2i}, v_{2i+1}) \in M$, $i = 1, \ldots, k-1$



Augmenting paths

Let G be a graph, M a matching of G.

A path $P = v_1, v_2, \ldots, v_{2k}$ of G is **augmenting** if v_1 and v_{2k} are unmatched and $(v_{2i}, v_{2i+1}) \in M$, $i = 1, \ldots, k-1$

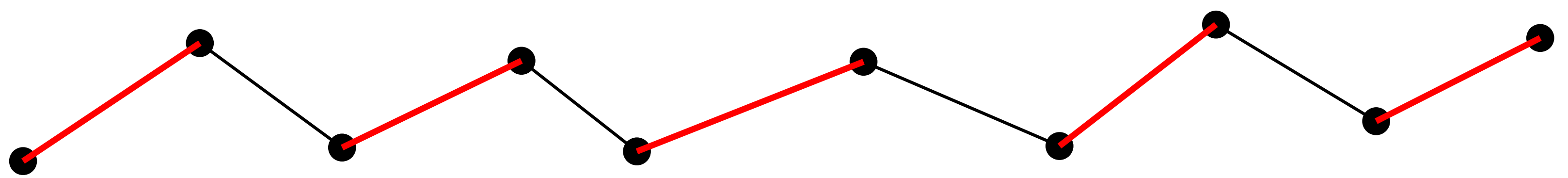


Lemma (Berge). *Let G be a graph and M a matching of G which is not maximum. Then there exists an augmenting path between two unmatched vertices of G.*

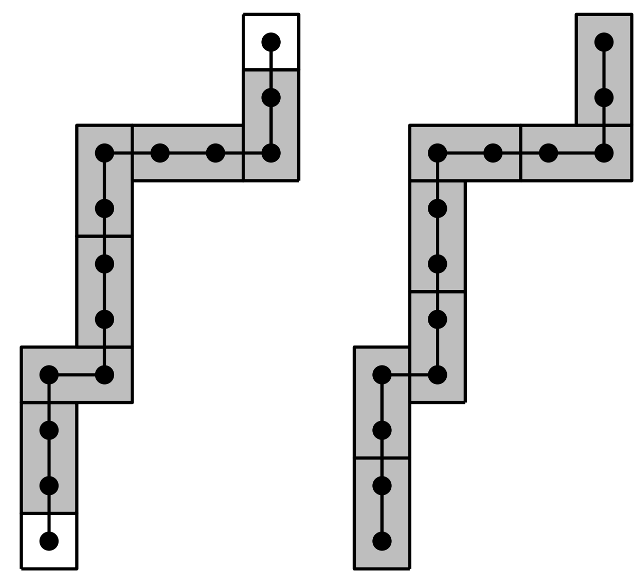
Augmenting paths

Let G be a graph, M a matching of G.

A path $P = v_1, v_2, \ldots, v_{2k}$ of G is **augmenting** if v_1 and v_{2k} are unmatched and $(v_{2i}, v_{2i+1}) \in M$, $i = 1, \ldots, k-1$

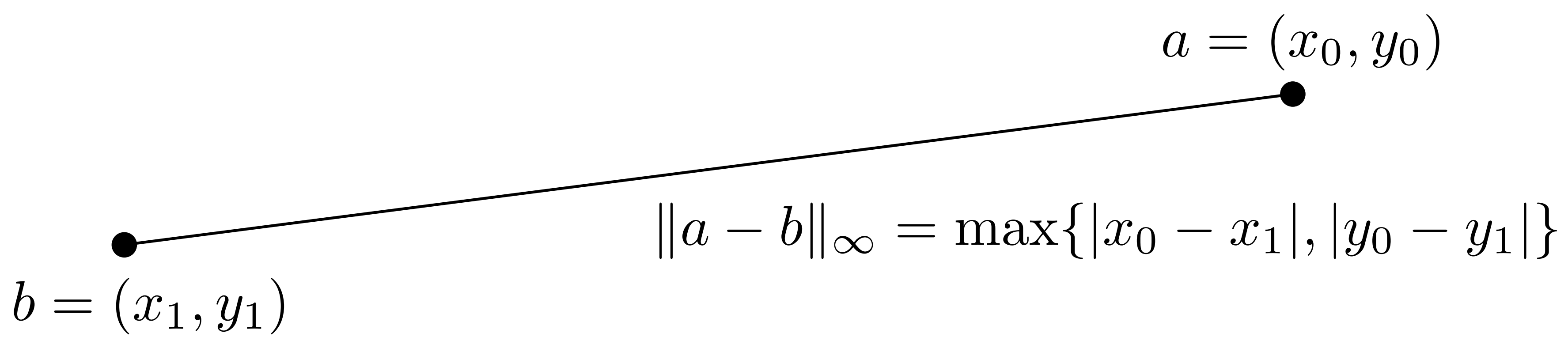


Lemma (Berge). *Let G be a graph and M a matching of G which is not maximum. Then there exists an augmenting path between two unmatched vertices of G.*



Other Preliminaries

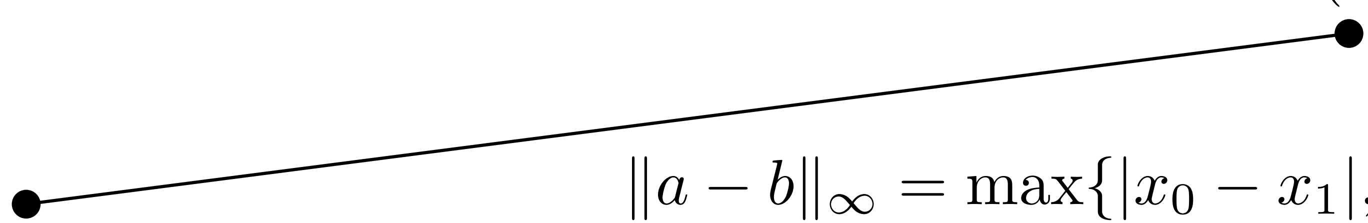
For $A, B \subset \mathbf{R}^2$, we define $d(A,B) = \inf_{(a,b)\in A\times B} \|a-b\|_\infty$



Other Preliminaries

For $A, B \subset \mathbf{R}^2$, we define $d(A,B) = \inf_{(a,b)\in A\times B} \|a-b\|_\infty$

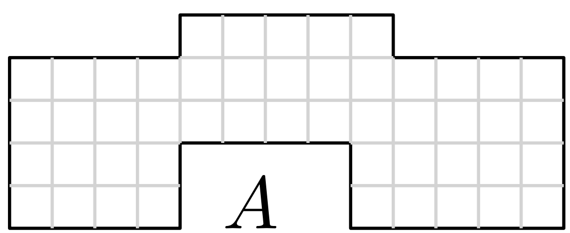
For $A \subset \mathbf{R}^2$, $r \geq 0$ we define the offsets
$B(A,r) = \{x \in \mathbf{R}^2 \mid d(A,x) \leq r\}$ and $B(A,-r) = B(A^c,r)^c$

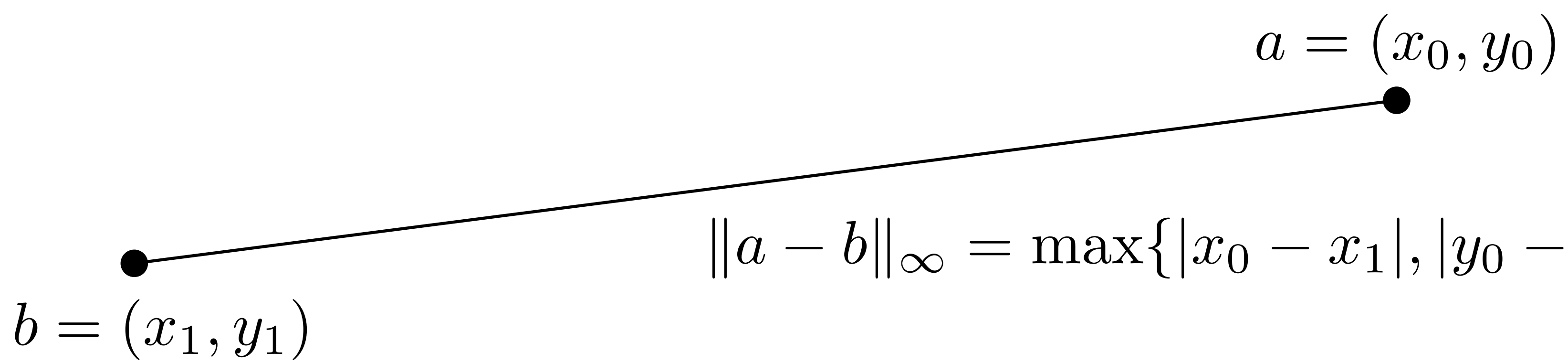


Other Preliminaries

For $A, B \subset \mathbf{R}^2$, we define $d(A, B) = \inf_{(a,b) \in A \times B} \|a - b\|_\infty$

For $A \subset \mathbf{R}^2$, $r \geq 0$ we define the offsets
$B(A, r) = \{x \in \mathbf{R}^2 \mid d(A, x) \leq r\}$ and $B(A, -r) = B(A^c, r)^c$



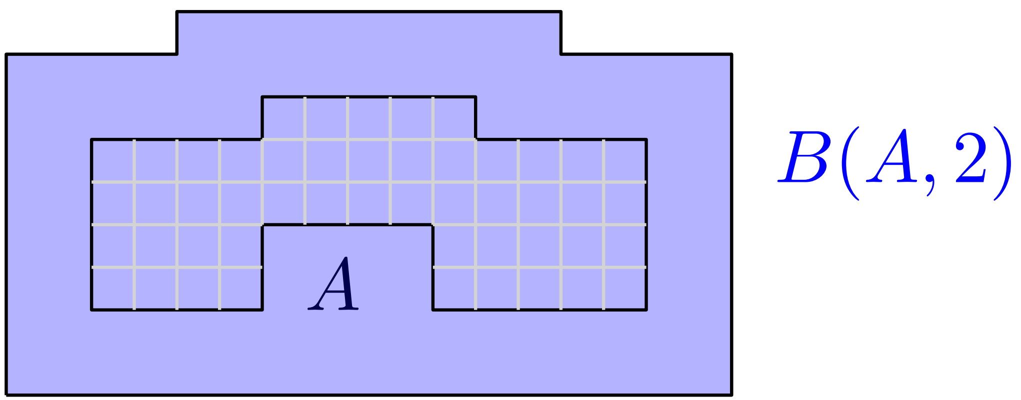


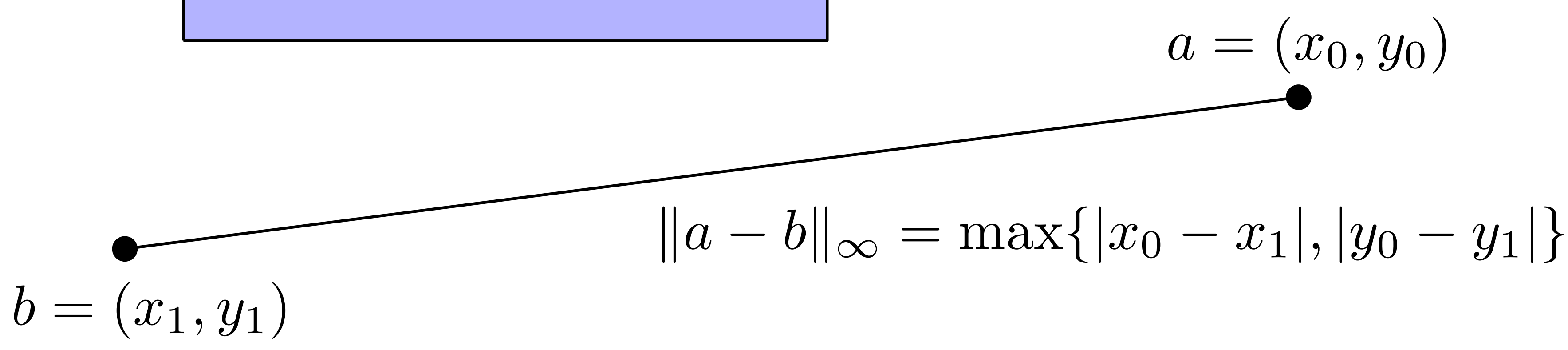
$$\|a - b\|_\infty = \max\{|x_0 - x_1|, |y_0 - y_1|\}$$

Other Preliminaries

For $A, B \subset \mathbf{R}^2$, we define $d(A,B) = \inf_{(a,b)\in A\times B} \|a-b\|_\infty$

For $A \subset \mathbf{R}^2$, $r \geq 0$ we define the offsets
$B(A,r) = \{x \in \mathbf{R}^2 \mid d(A,x) \leq r\}$ and $B(A,-r) = B(A^c,r)^c$

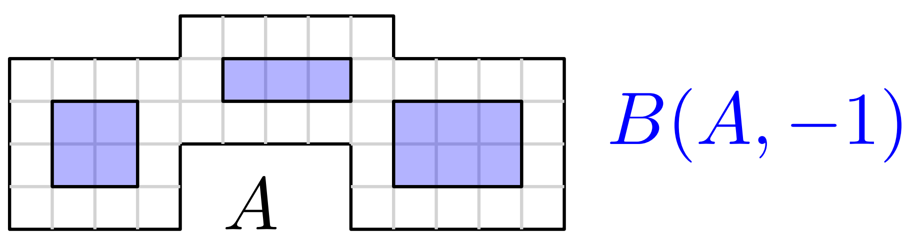


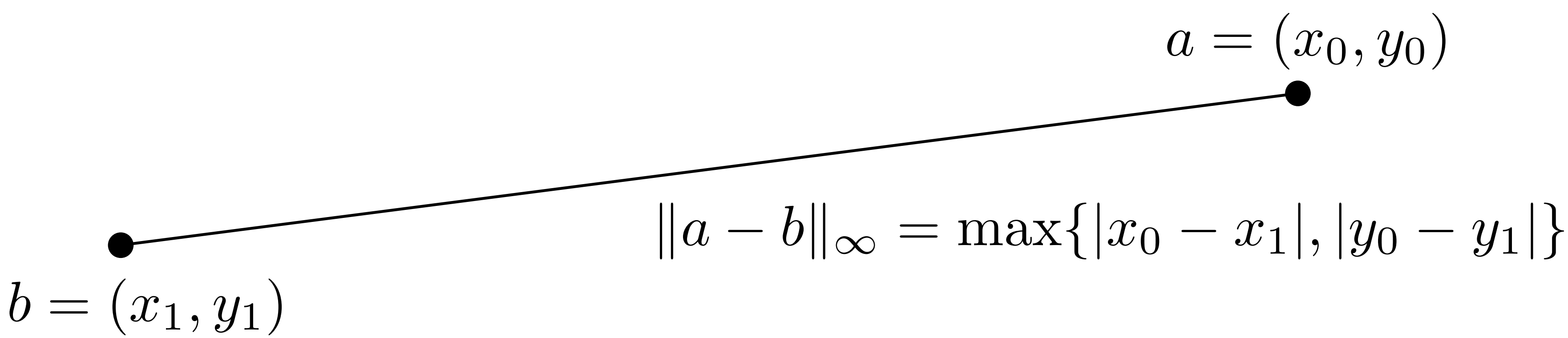


Other Preliminaries

For $A, B \subset \mathbf{R}^2$, we define $d(A,B) = \inf_{(a,b)\in A\times B} \|a-b\|_\infty$

For $A \subset \mathbf{R}^2$, $r \geq 0$ we define the offsets
$B(A,r) = \{x \in \mathbf{R}^2 \mid d(A,x) \leq r\}$ and $B(A,-r) = B(A^c,r)^c$



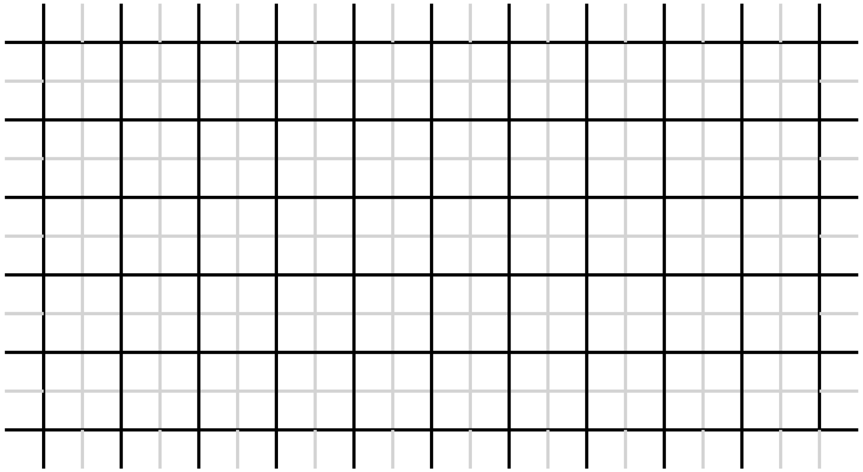


Other Preliminaries

A polyomino $P \subset \mathbf{R}^2$ has **consistent parity** if all first coordinates of corners of P have the same parity and vice versa for the second coordinates

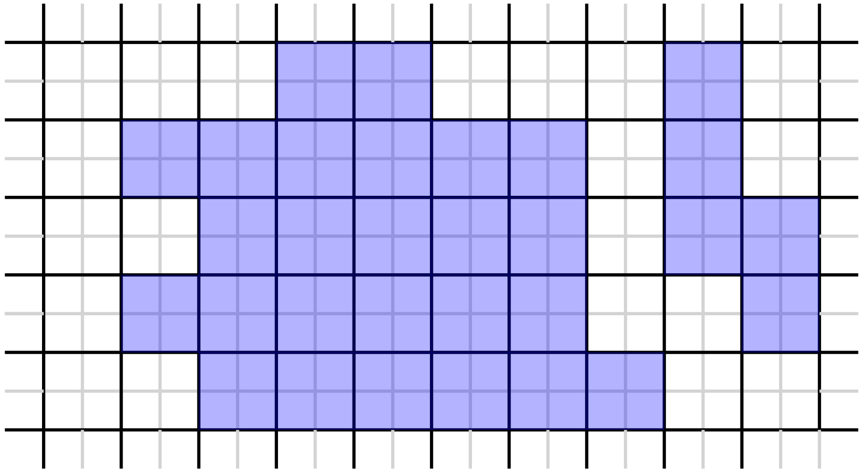
Other Preliminaries

A polyomino $P \subset \mathbf{R}^2$ has **consistent parity** if all first coordinates of corners of P have the same parity and vice versa for the second coordinates



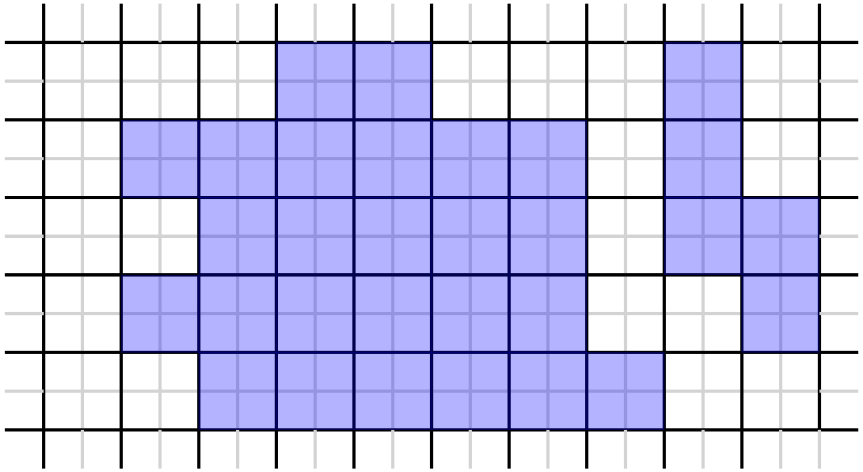
Other Preliminaries

A polyomino $P \subset \mathbf{R}^2$ has **consistent parity** if all first coordinates of corners of P have the same parity and vice versa for the second coordinates



Other Preliminaries

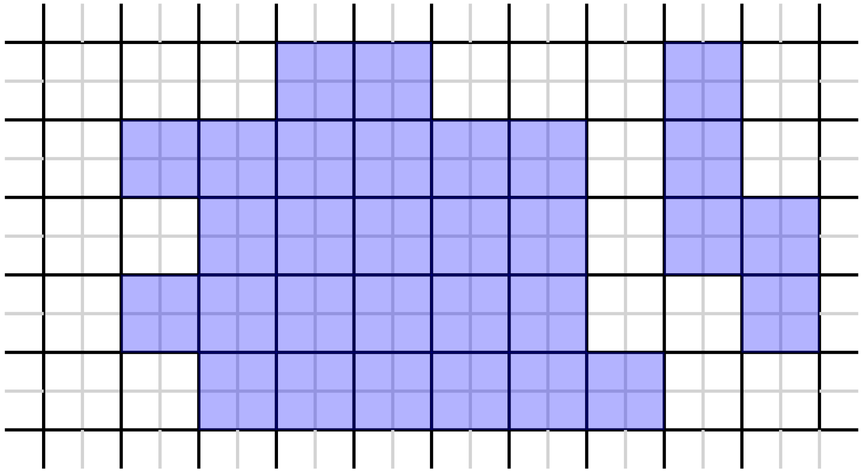
A polyomino $P \subset \mathbf{R}^2$ has **consistent parity** if all first coordinates of corners of P have the same parity and vice versa for the second coordinates



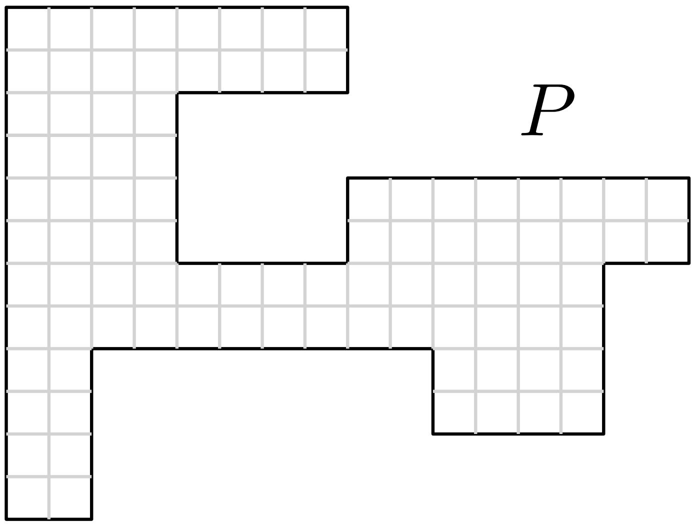
Observation: *If P has no holes and consistent parity then each component of $P \setminus B(P, -1)$ is Hamiltonian.*

Other Preliminaries

A polyomino $P \subset \mathbf{R}^2$ has **consistent parity** if all first coordinates of corners of P have the same parity and vice versa for the second coordinates □

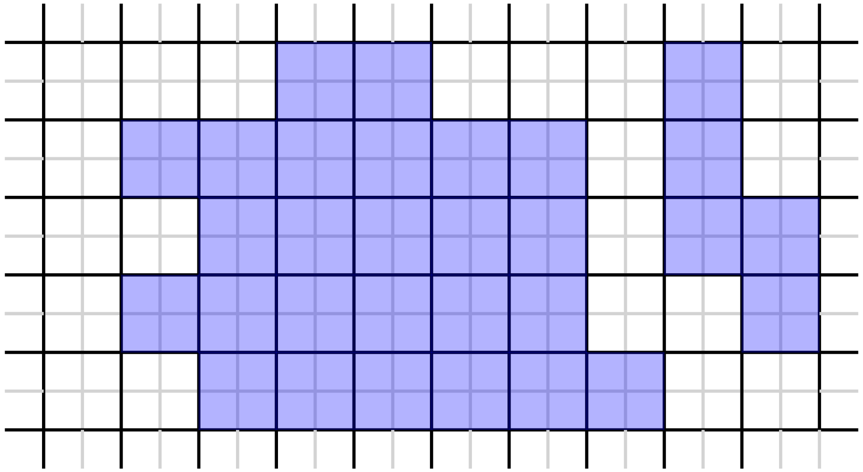


Observation: *If P has no holes and consistent parity then each component of $P \setminus B(P, -1)$ is Hamiltonian.*

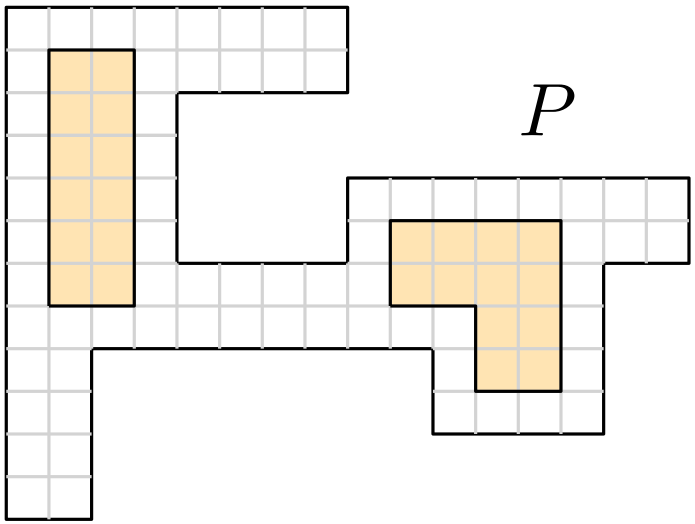


Other Preliminaries

A polyomino $P \subset \mathbf{R}^2$ has **consistent parity** if all first coordinates of corners of P have the same parity and vice versa for the second coordinates

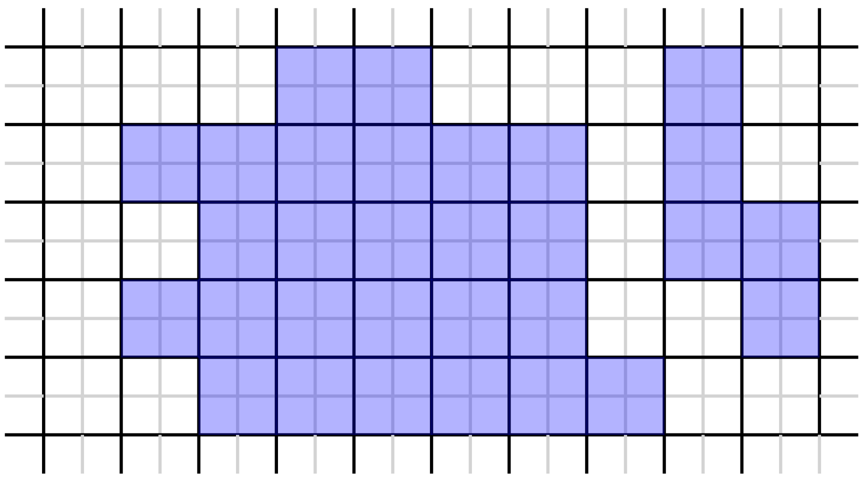


Observation: *If P has no holes and consistent parity then each component of $P \setminus B(P, -1)$ is Hamiltonian.*

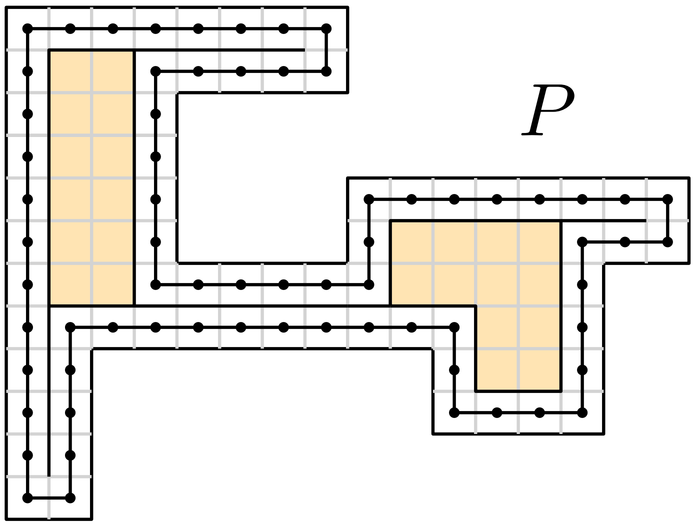


Other Preliminaries

A polyomino $P \subset \mathbf{R}^2$ has **consistent parity** if all first coordinates of corners of P have the same parity and vice versa for the second coordinates

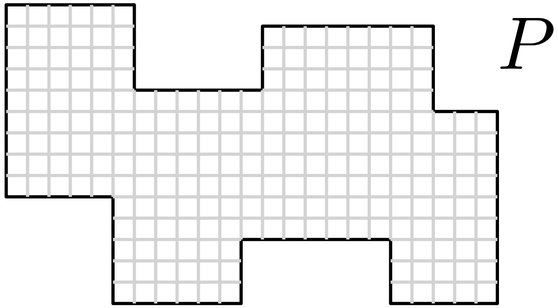


Observation: *If P has no holes and consistent parity then each component of $P \setminus B(P, -1)$ is Hamiltonian.*



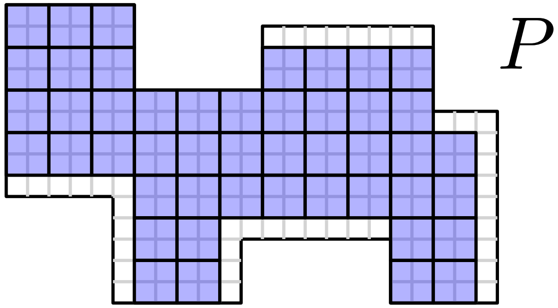
Main Structural Result 1

Let subpolyomino $P_0 \subseteq P$ be maximal with consistent parity.



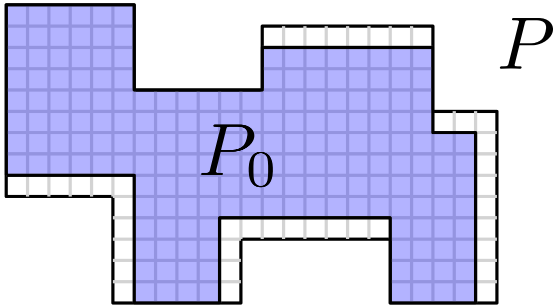
Main Structural Result 1

Let subpolyomino $P_0 \subseteq P$ be maximal with consistent parity.



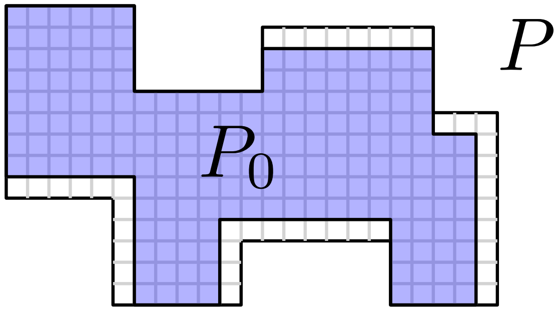
Main Structural Result 1

Let subpolyomino $P_0 \subseteq P$ be maximal with consistent parity.



Main Structural Result 1

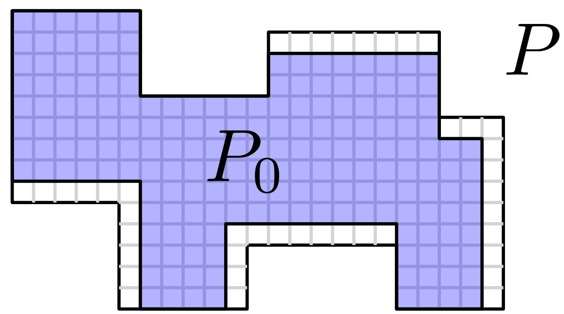
Let subpolyomino $P_0 \subseteq P$ be maximal with consistent parity.



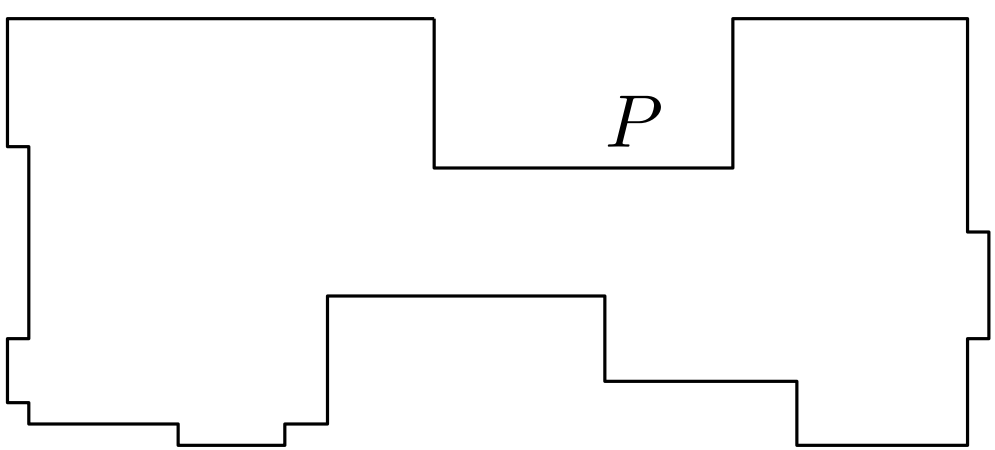
Let $r = \lfloor n/2 \rfloor$ and $Q = B(P_0, -r)$. Note that Q has consistent parity.

Main Structural Result 1

Let subpolyomino $P_0 \subseteq P$ be maximal with consistent parity.

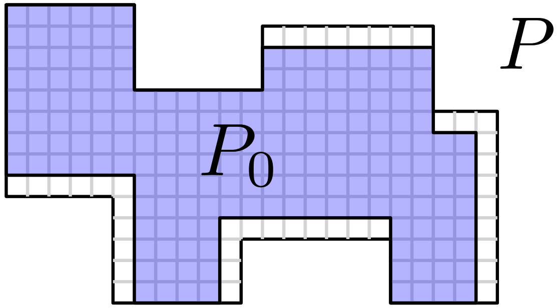


Let $r = \lfloor n/2 \rfloor$ and $Q = B(P_0, -r)$. Note that Q has consistent parity.

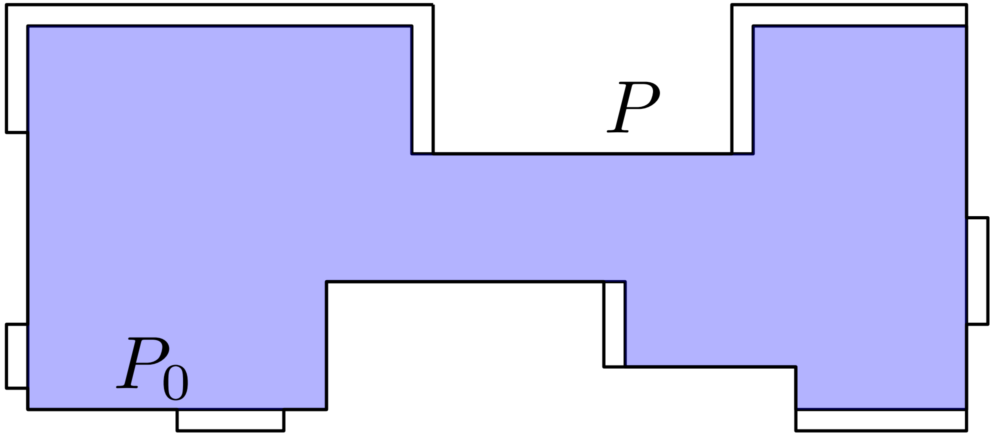


Main Structural Result 1

Let subpolyomino $P_0 \subseteq P$ be maximal with consistent parity.

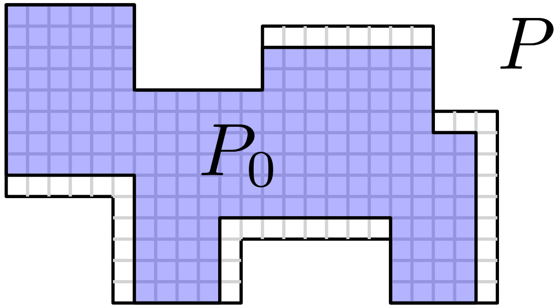


Let $r = \lfloor n/2 \rfloor$ and $Q = B(P_0, -r)$. Note that Q has consistent parity.

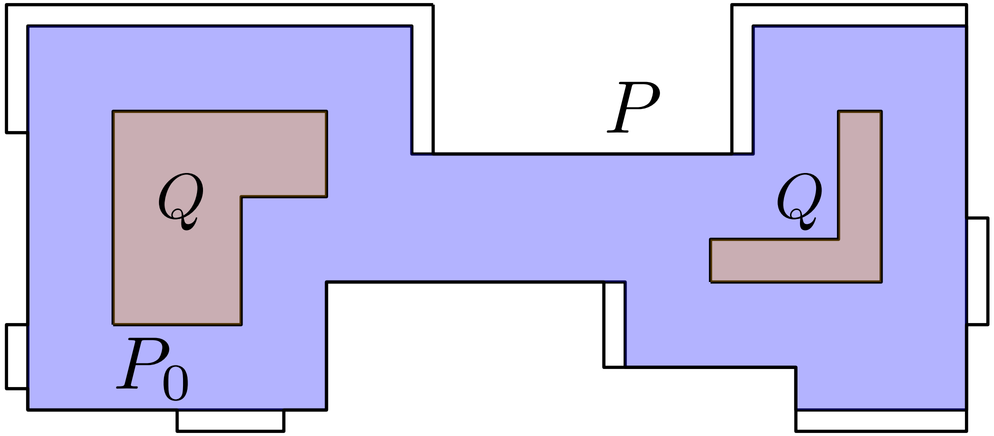


Main Structural Result 1

Let subpolyomino $P_0 \subseteq P$ be maximal with consistent parity.

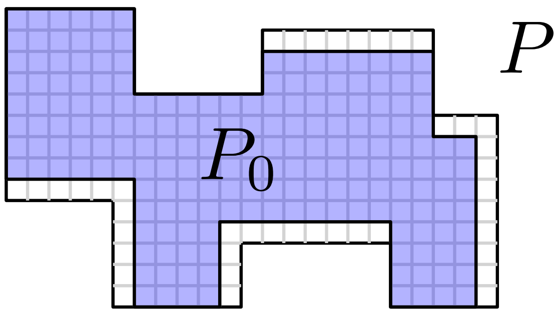


Let $r = \lfloor n/2 \rfloor$ and $Q = B(P_0, -r)$. Note that Q has consistent parity.

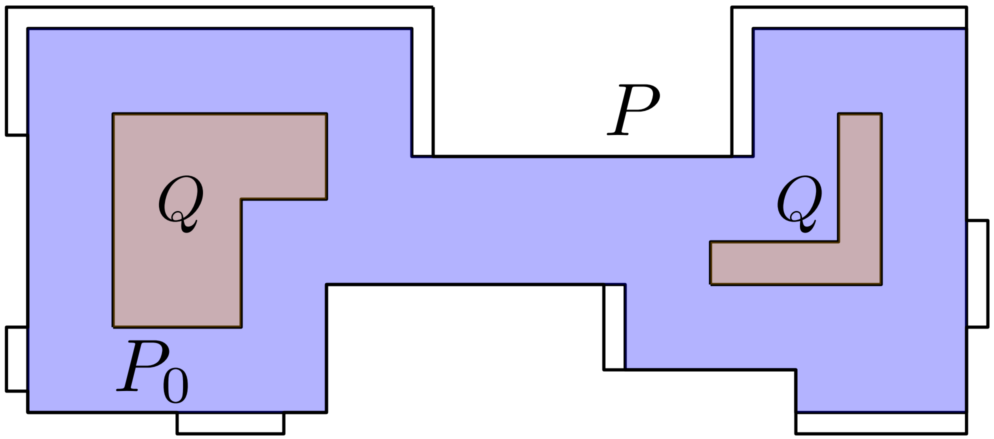


Main Structural Result 1

Let subpolyomino $P_0 \subseteq P$ be maximal with consistent parity.



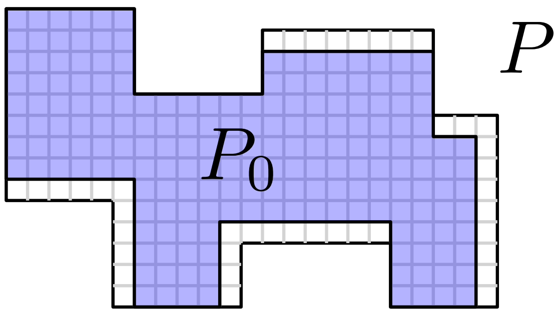
Let $r = \lfloor n/2 \rfloor$ and $Q = B(P_0, -r)$. Note that Q has consistent parity.



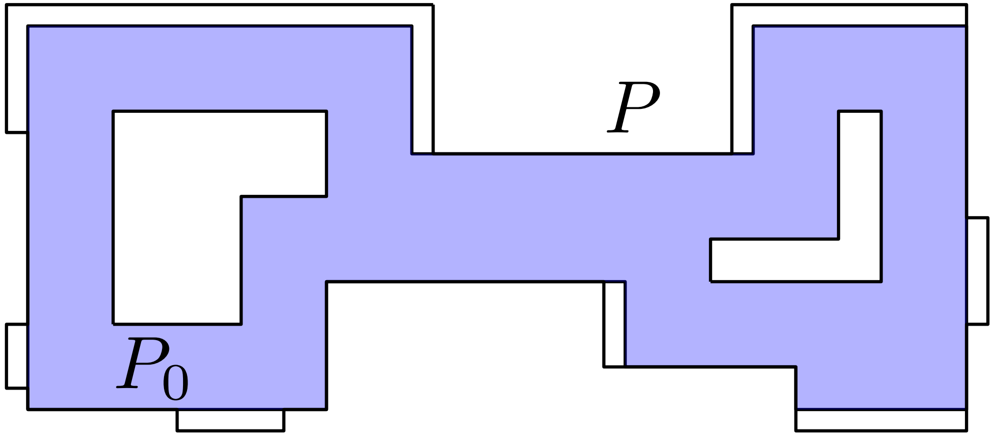
Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Main Structural Result 1

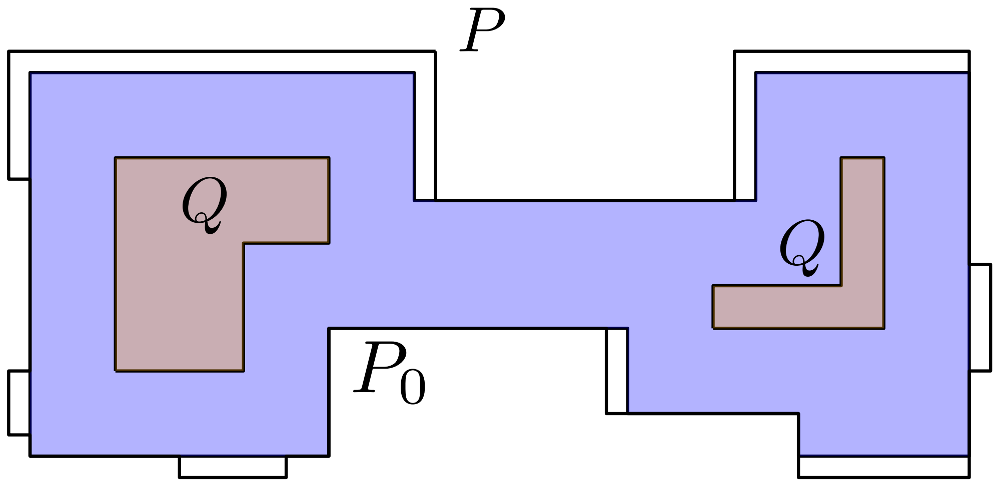
Let subpolyomino $P_0 \subseteq P$ be maximal with consistent parity.



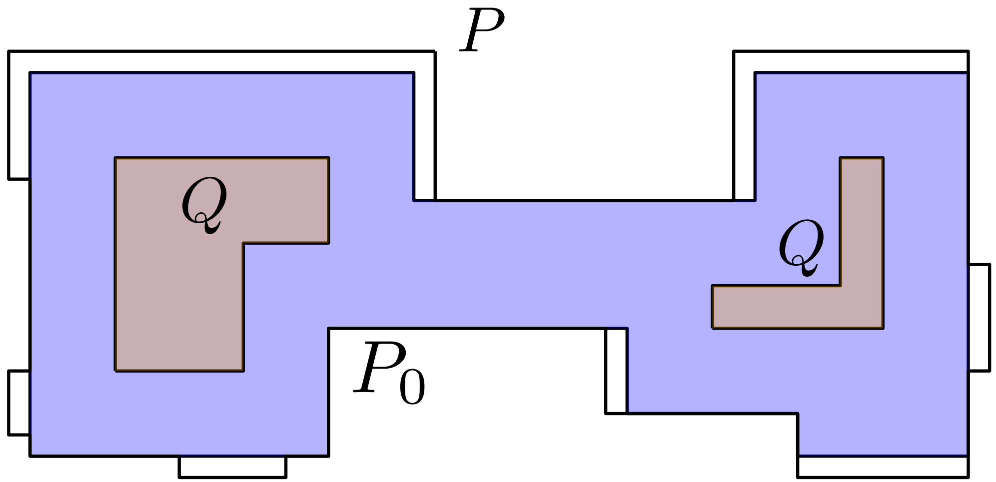
Let $r = \lfloor n/2 \rfloor$ and $Q = B(P_0, -r)$. Note that Q has consistent parity.



Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

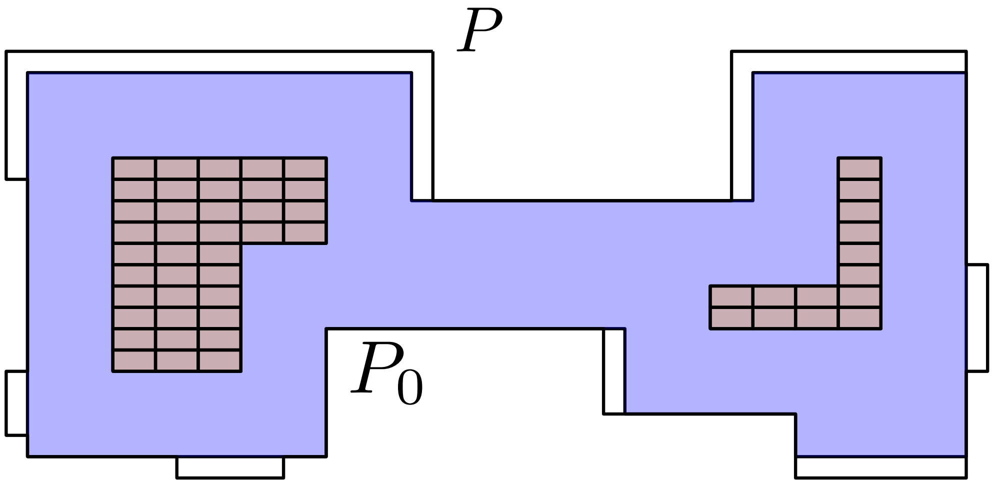


Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*



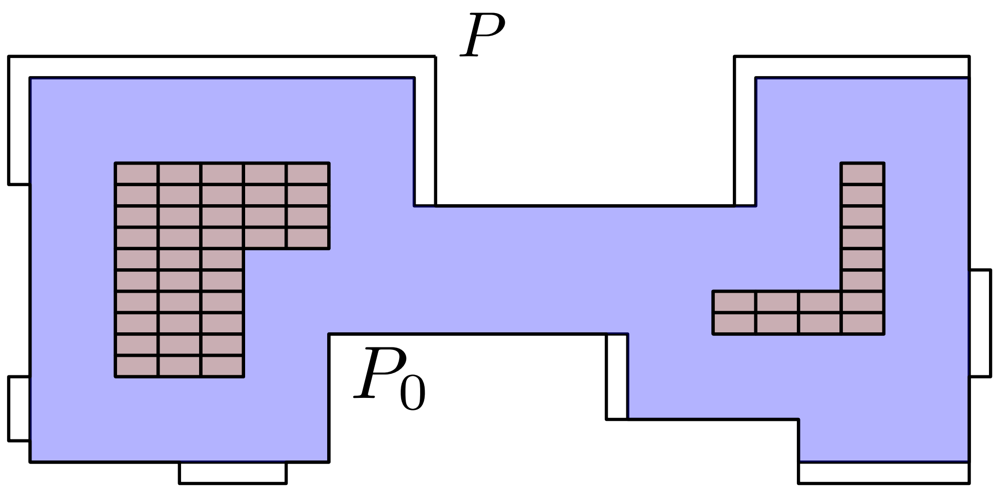
Let $\mathcal{Q}$ be any tiling of Q

Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*



Let $\mathcal{Q}$ be any tiling of Q

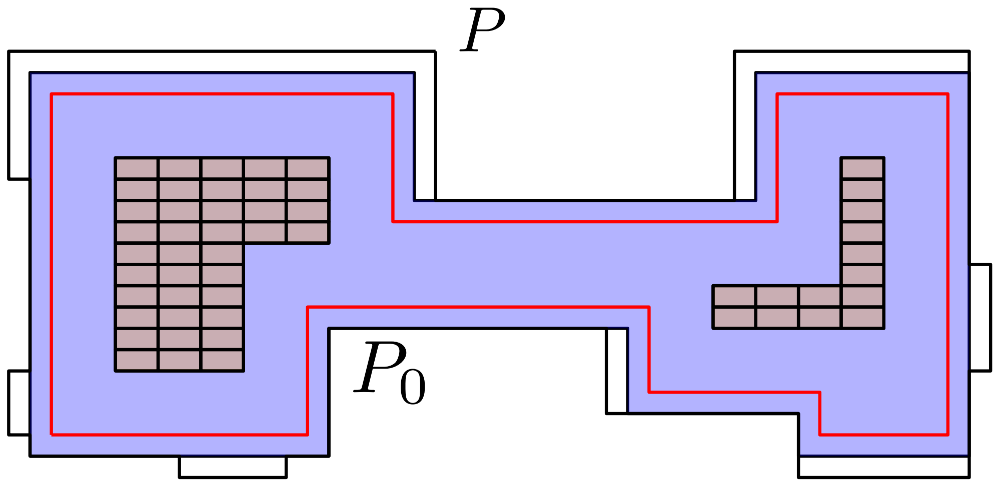
Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*



Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

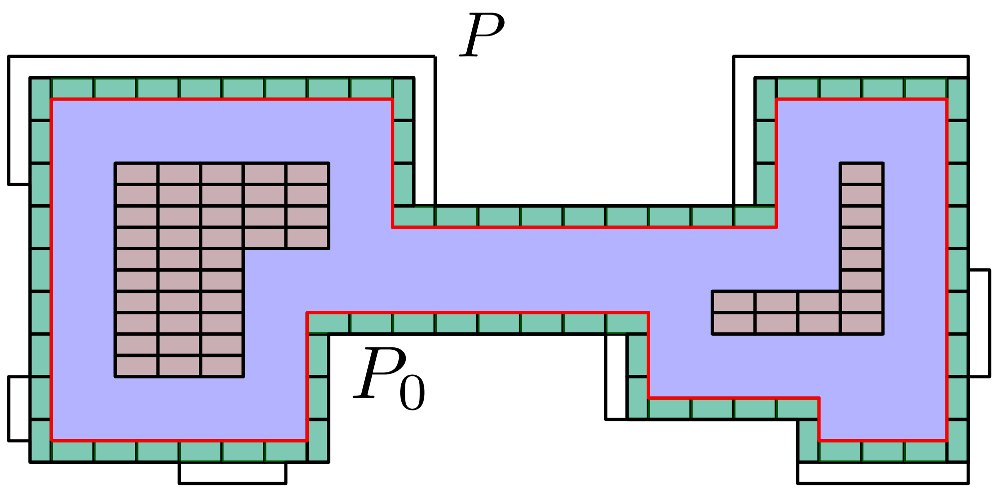
Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.



Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

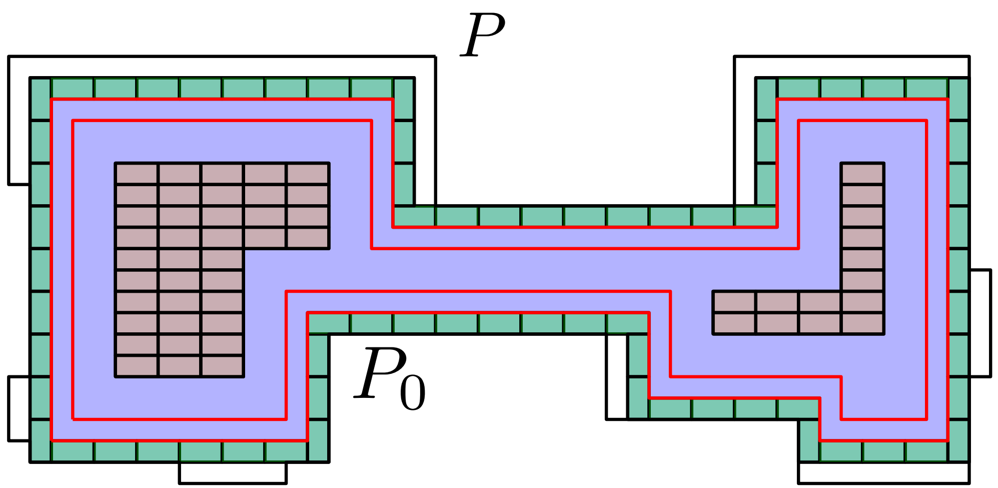
Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.



Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

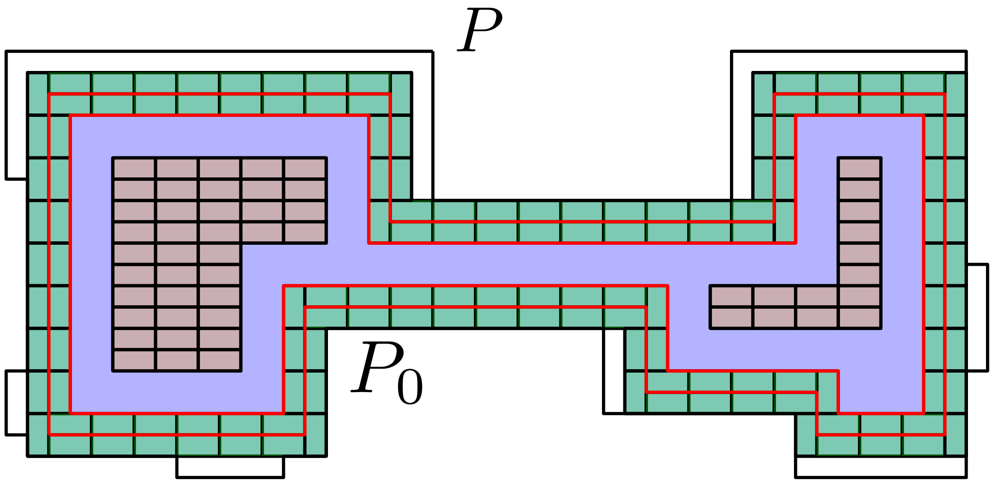
Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.



Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

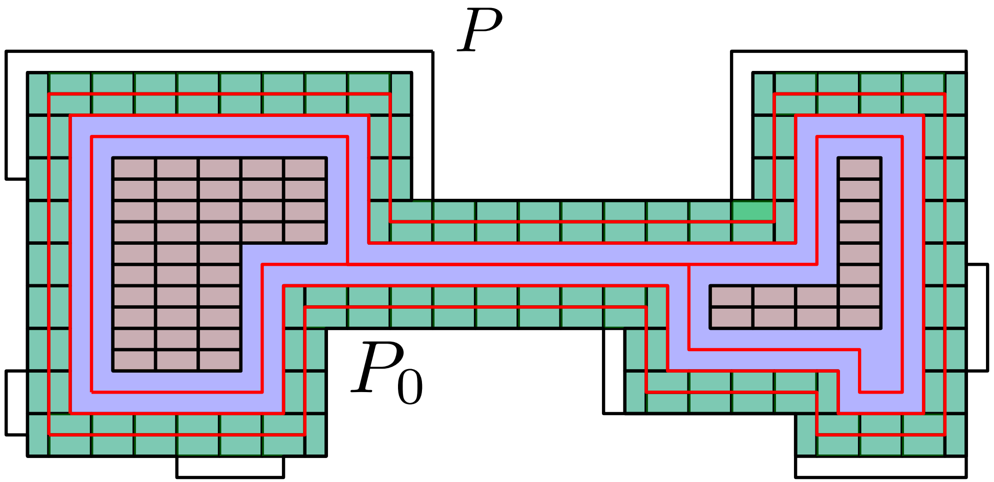
Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.



Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

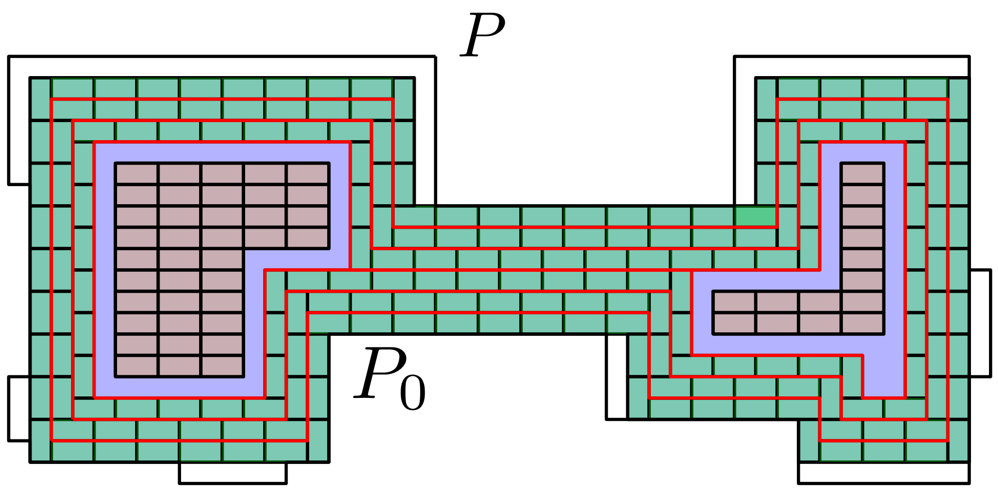
Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.



Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

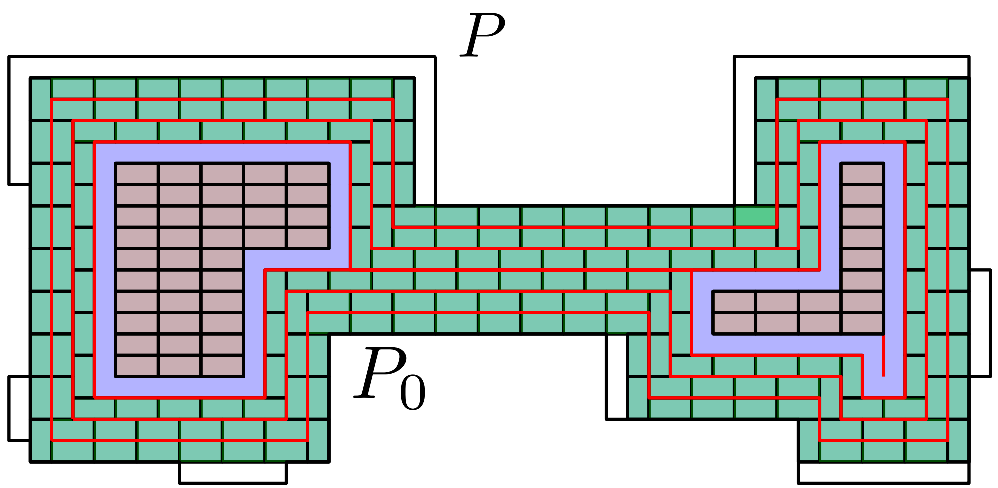
Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.



Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

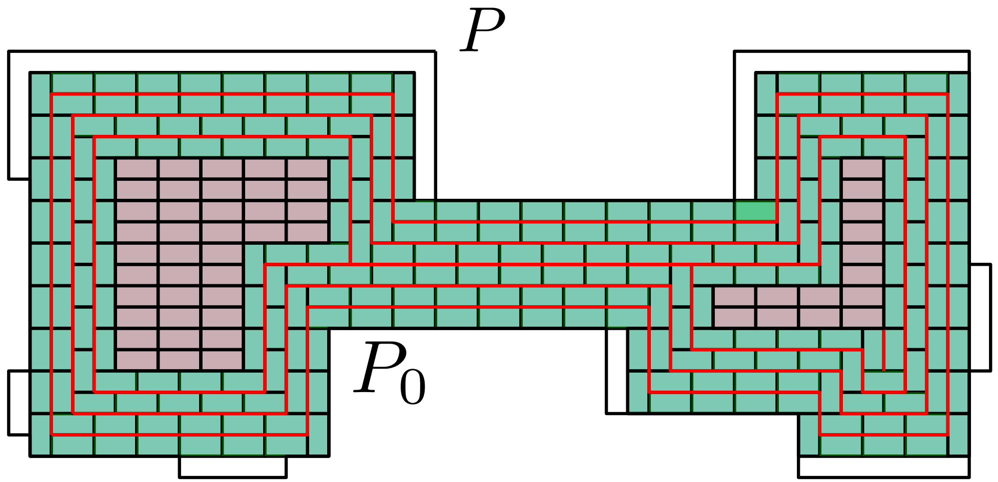
Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.



Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

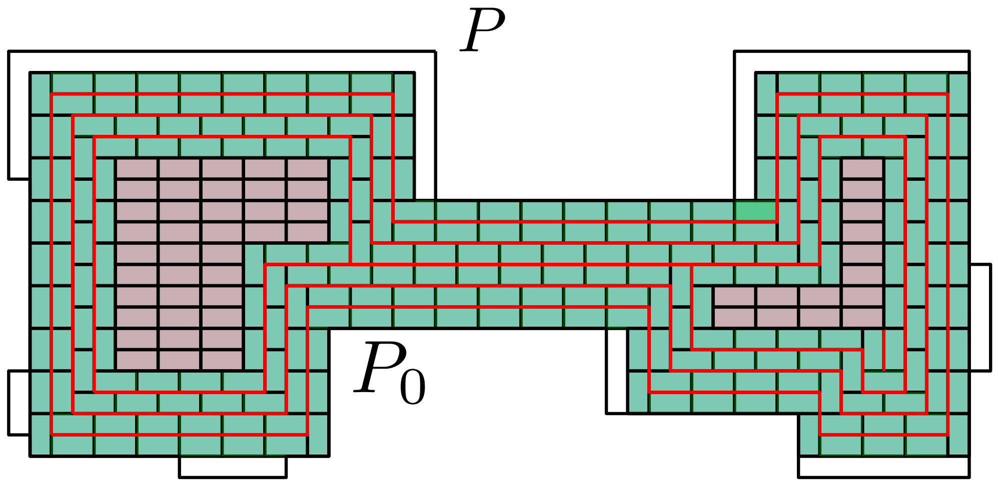
Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.



Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.

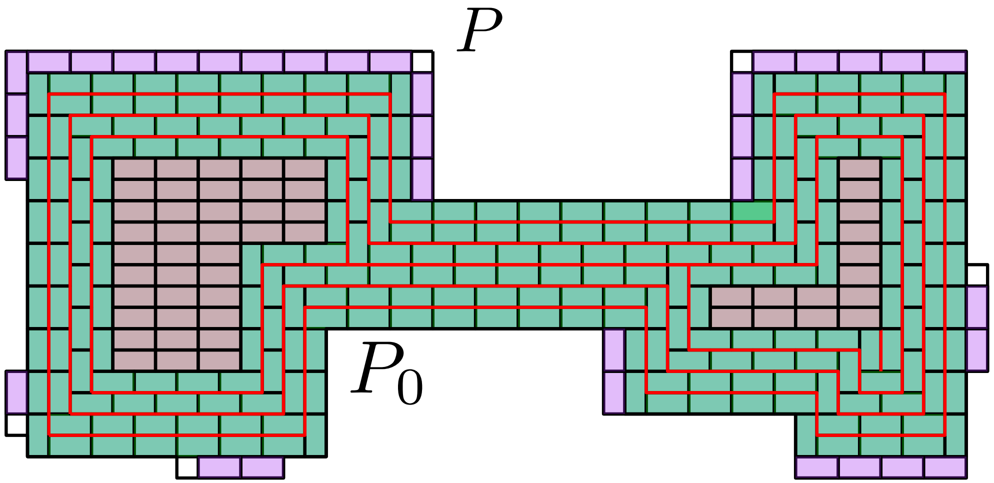


Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.

Finally pack dominos into $P \setminus P_0$, leaving at most n uncovered cells.

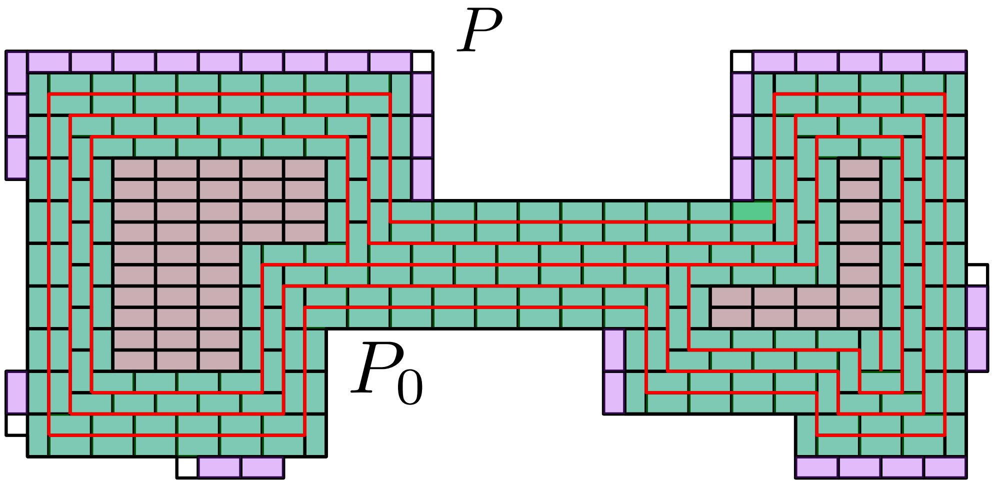


Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.

Finally pack dominos into $P \setminus P_0$, leaving at most n uncovered cells.



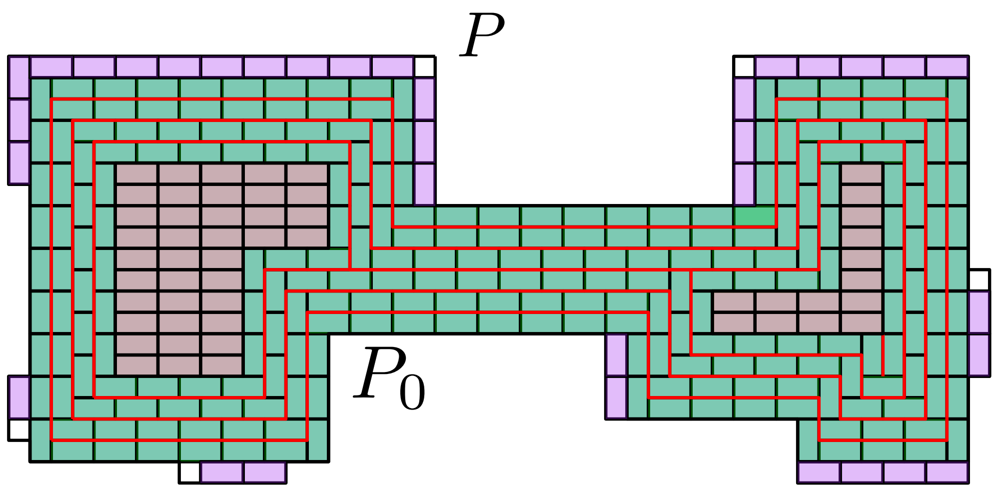
Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Let $\mathcal{Q}$ be any tiling of Q

Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.

Finally pack dominos into $P \setminus P_0$, leaving at most n uncovered cells.

If covering is not maximum, use Berge's Lemma.



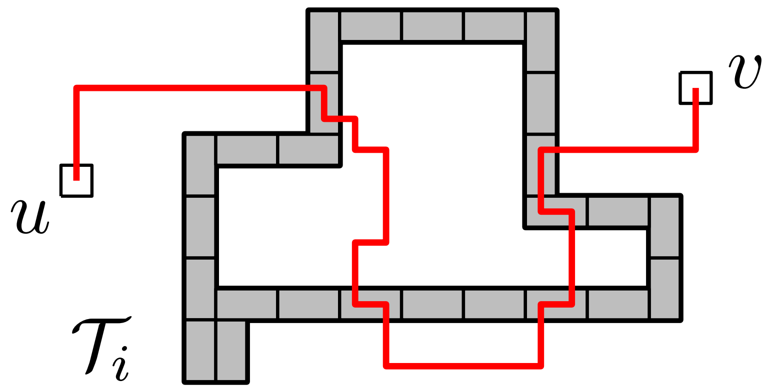
Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

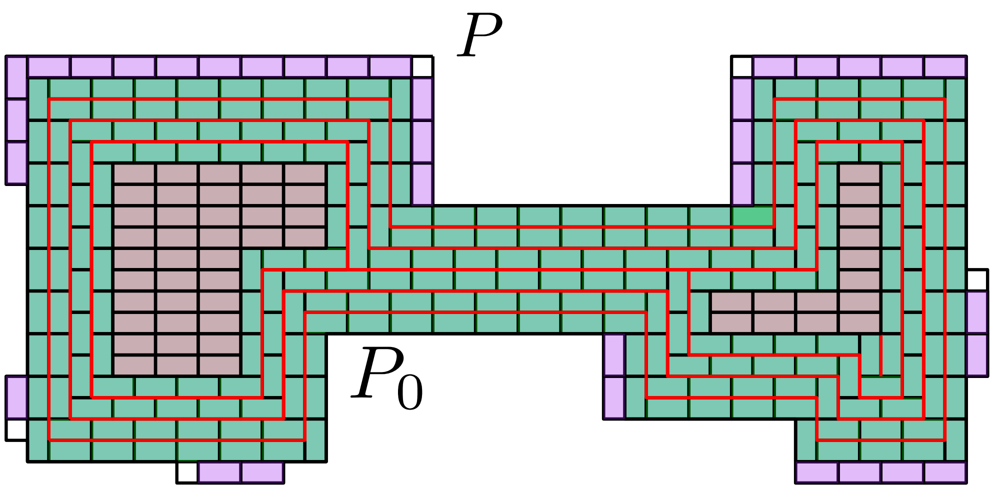
Let $\mathcal{Q}$ be any tiling of Q

Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.

Finally pack dominos into $P \setminus P_0$, leaving at most n uncovered cells.

If covering is not maximum, use Berge's Lemma.





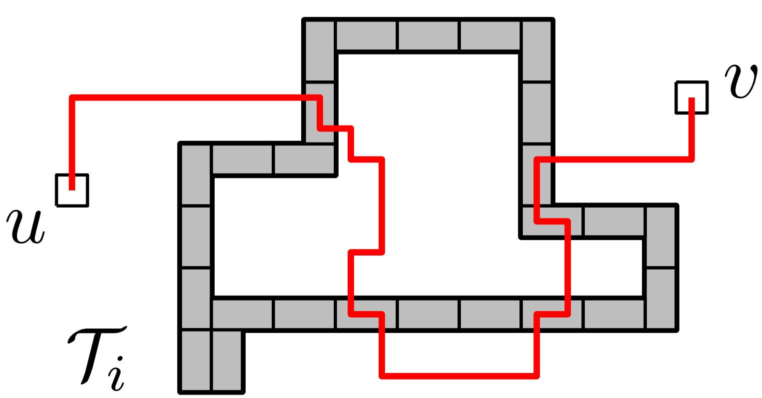
Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

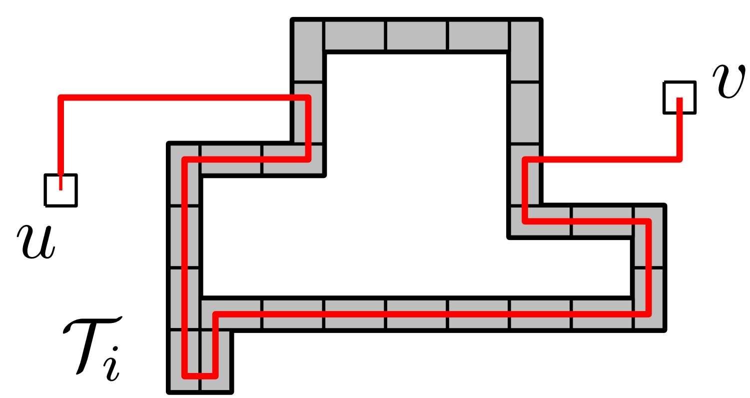
Let $\mathcal{Q}$ be any tiling of Q

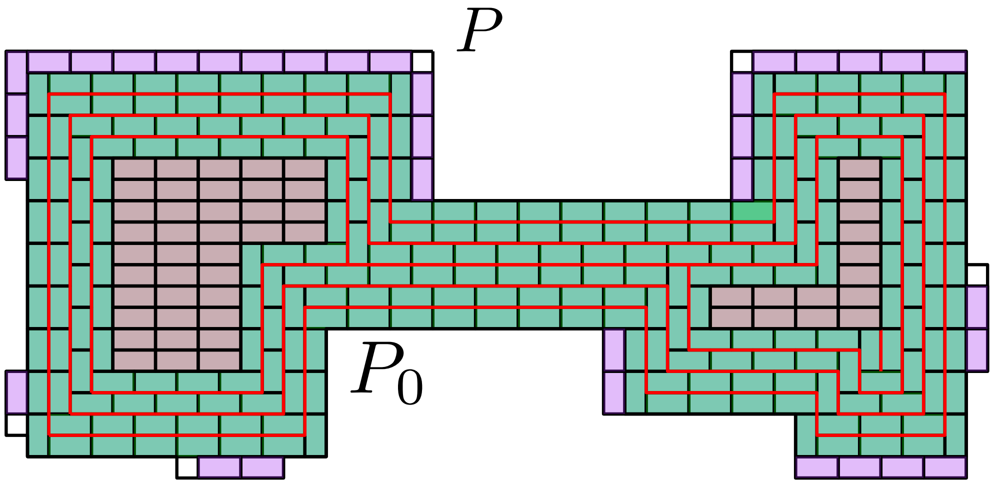
Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.

Finally pack dominos into $P \setminus P_0$, leaving at most n uncovered cells.

If covering is not maximum, use Berge's Lemma.







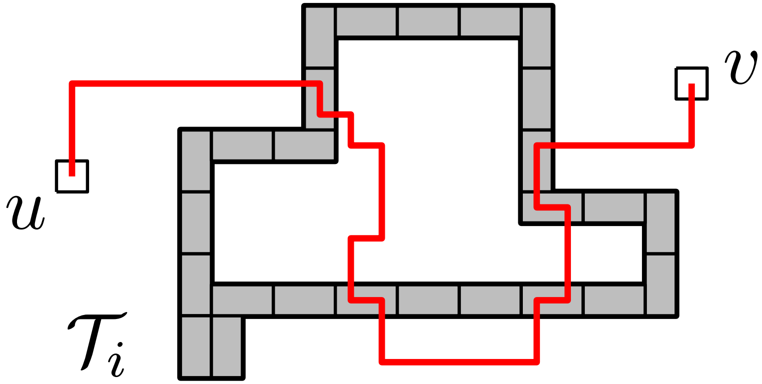
Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

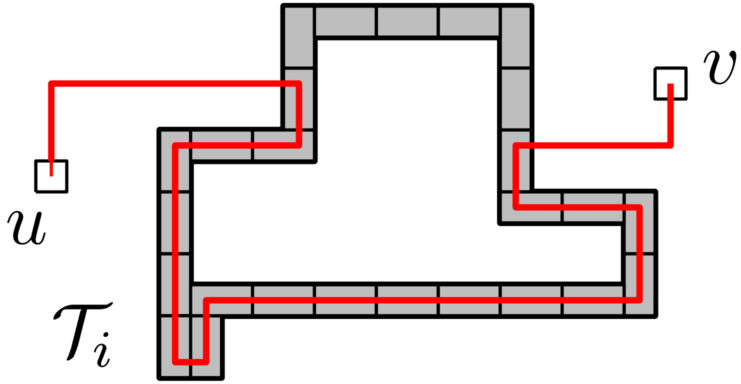
Let $\mathcal{Q}$ be any tiling of Q

Tile $P_0 \setminus Q$ layer by layer. Tilings $\mathcal{T}_1, \ldots, \mathcal{T}_r$.

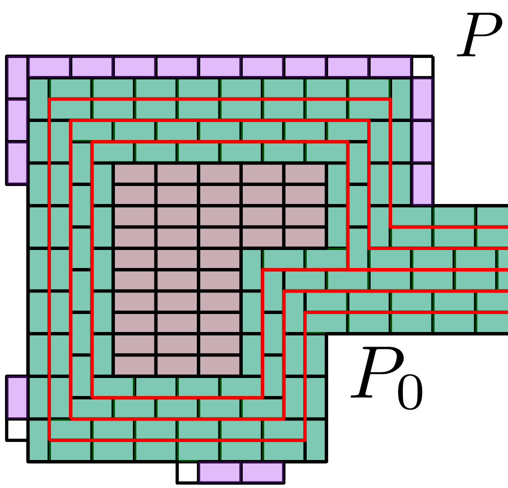
Finally pack dominos into $P \setminus P_0$, leaving at most n uncovered cells.

If covering is not maximum, use Berge's Lemma.



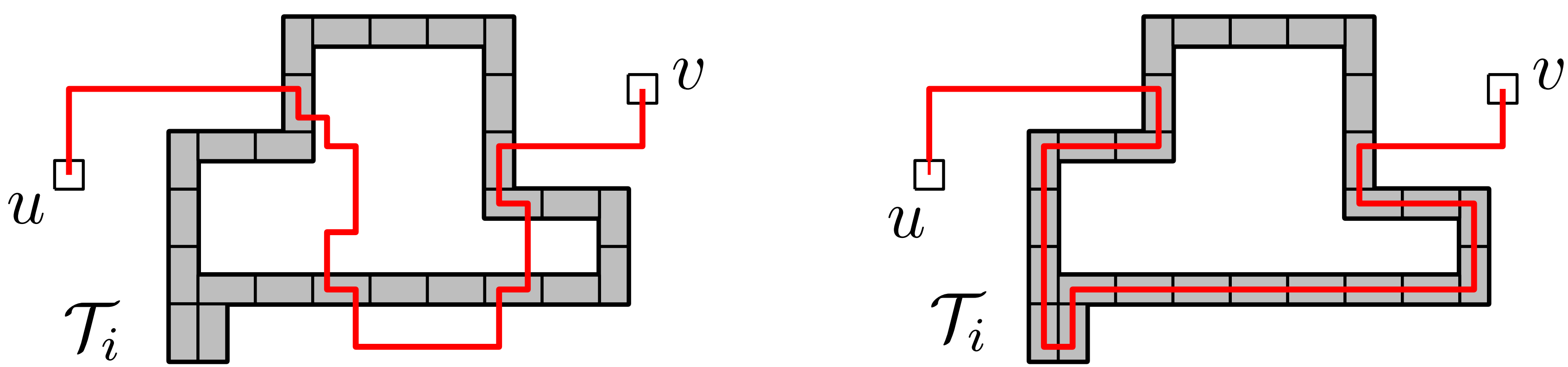


Has to repeat at most $r = \lfloor n/2 \rfloor$ times



Main Lemma. *There exists a maximum domino packing of P restricting to a tiling of Q.*

Importantly:
P has no holes $\Rightarrow$
u and v are both 'outside' each of the Hamiltonian cycles.

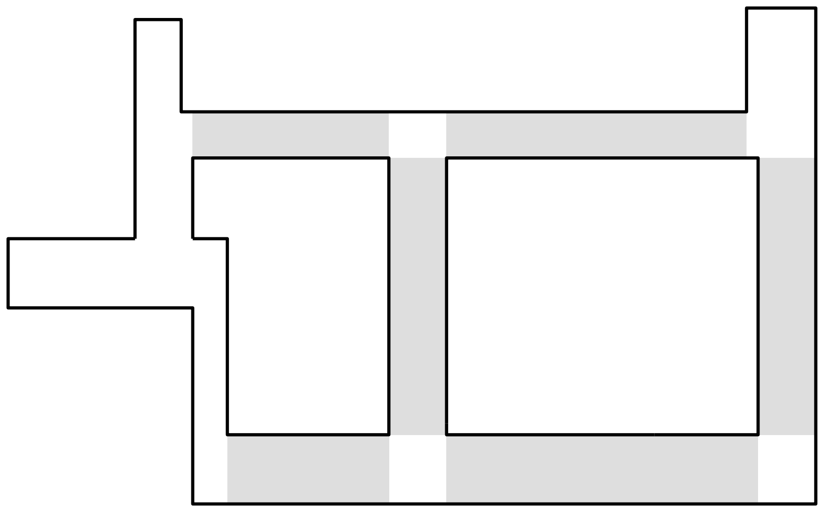


Has to repeat at most $r = \lfloor n/2 \rfloor$ times

The reduced instance

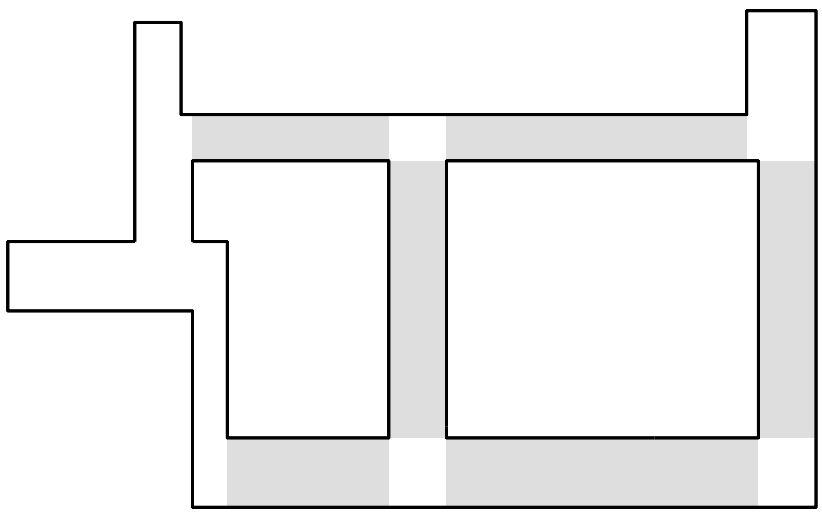
The reduced instance

Issue: There can be exponentially long and narrow 'pipes' $\Rightarrow$ he size can be exponential.



The reduced instance

Issue: There can be exponentially long and narrow 'pipes' $\Rightarrow$
he size can be exponential.

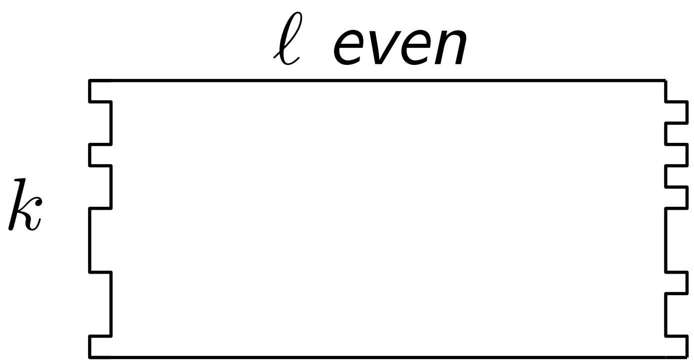


However, any point of $P' = P \setminus Q$ is of distance $O(n)$ to $\partial P'$

Structural Result 2: Shortening Pipes

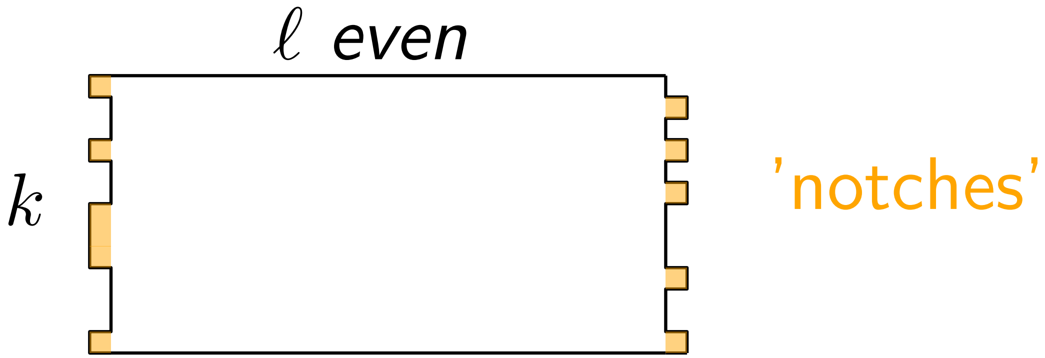
Structural Result 2: Shortening Pipes

Consider 'pipes' of the form:



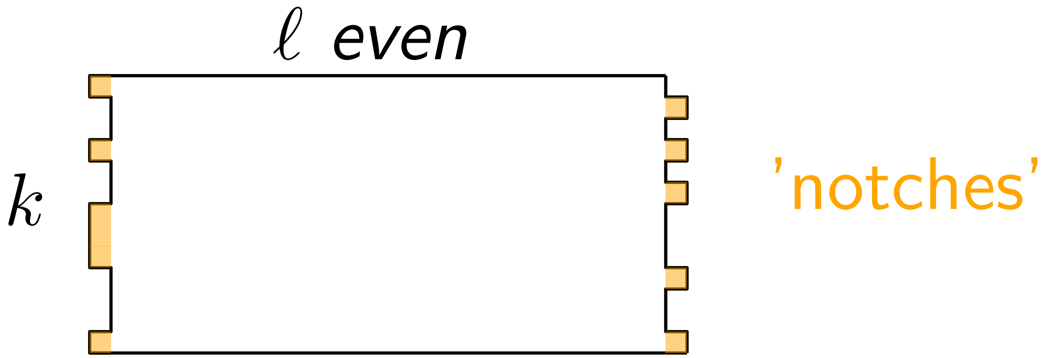
Structural Result 2: Shortening Pipes

Consider 'pipes' of the form:



Structural Result 2: Shortening Pipes

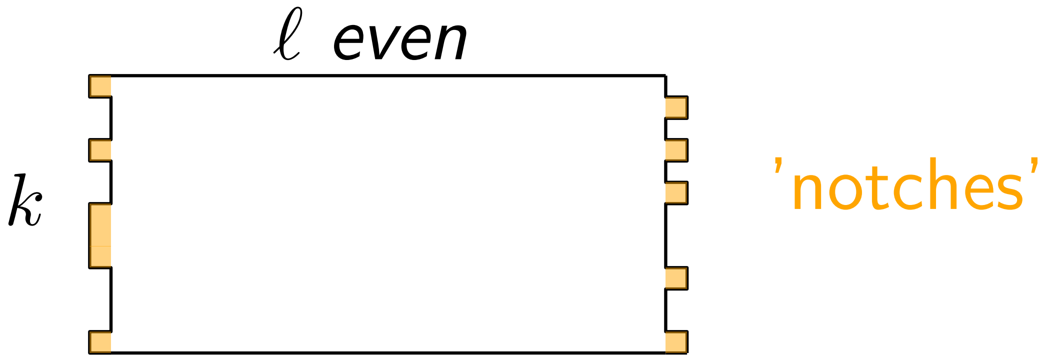
Consider 'pipes' of the form:



Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

Structural Result 2: Shortening Pipes

Consider 'pipes' of the form:

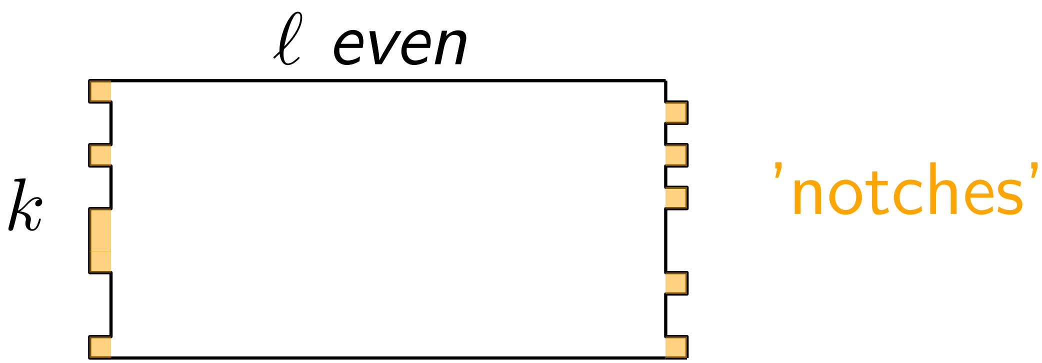


Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

Lemma. *If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing is $b - w$*

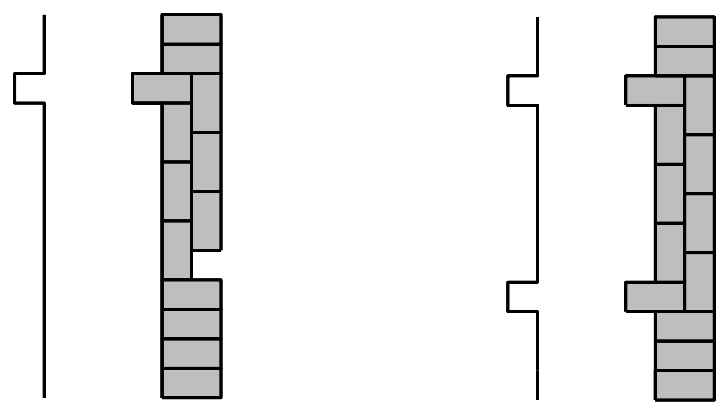
Structural Result 2: Shortening Pipes

Consider 'pipes' of the form:



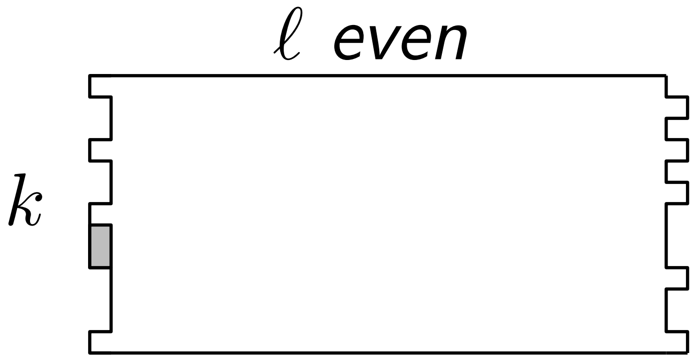
Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

Lemma. *If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing is $b - w$*



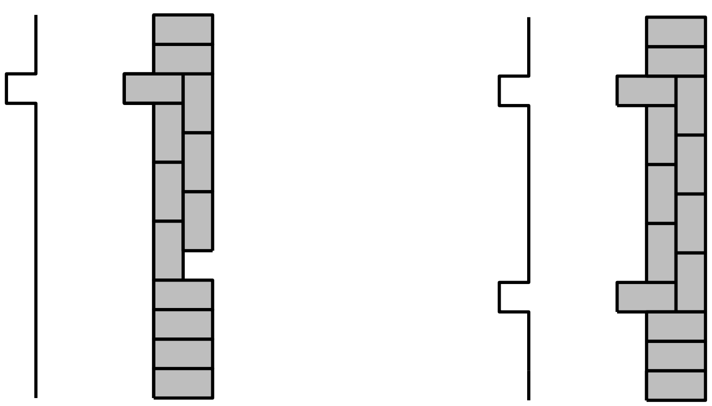
Structural Result 2: Shortening Pipes

Consider 'pipes' of the form:



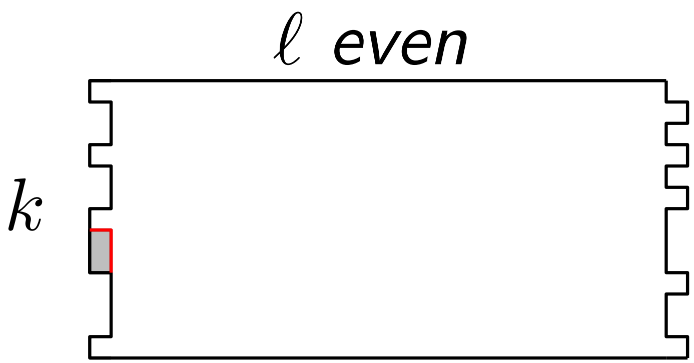
Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

Lemma. *If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing is $b - w$*



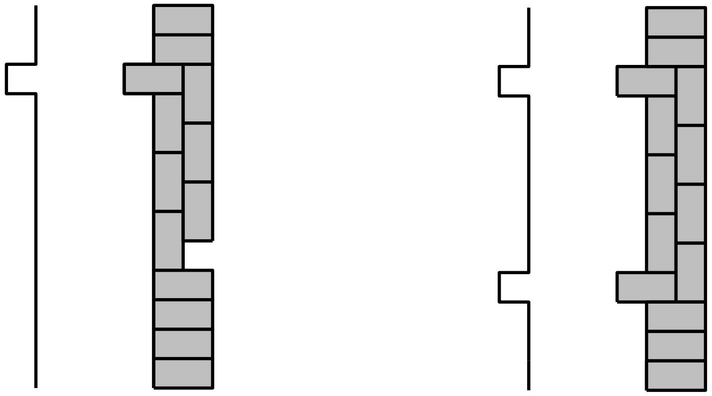
Structural Result 2: Shortening Pipes

Consider 'pipes' of the form:



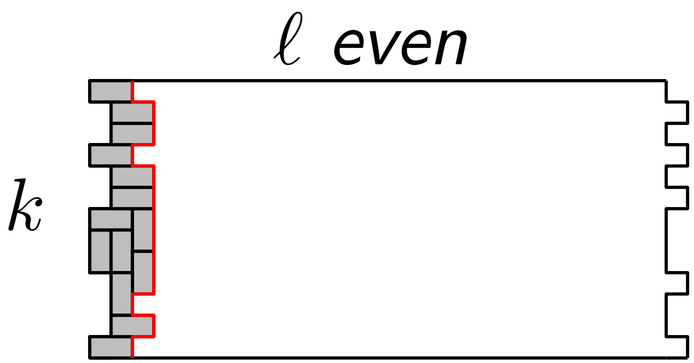
Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

Lemma. *If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing is $b - w$*



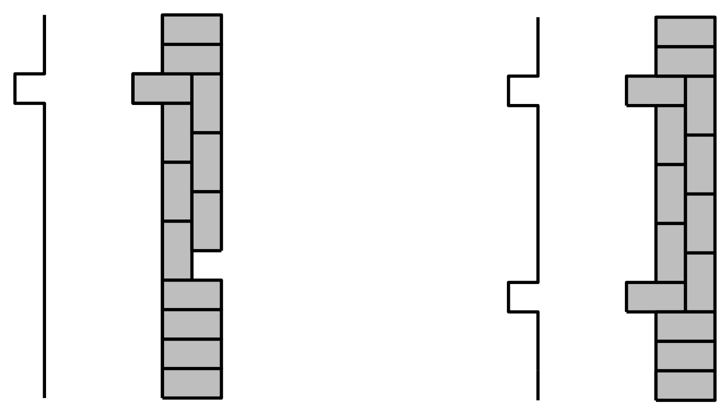
Structural Result 2: Shortening Pipes

Consider 'pipes' of the form:



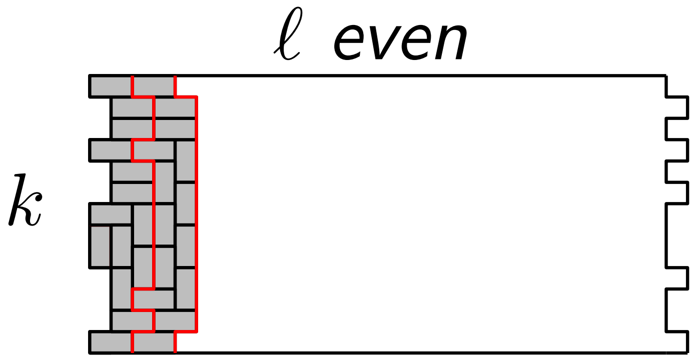
Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

Lemma. *If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing is $b - w$*



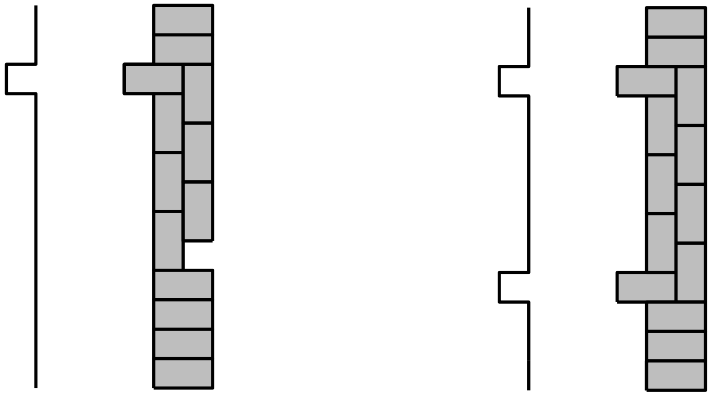
Structural Result 2: Shortening Pipes

Consider 'pipes' of the form:



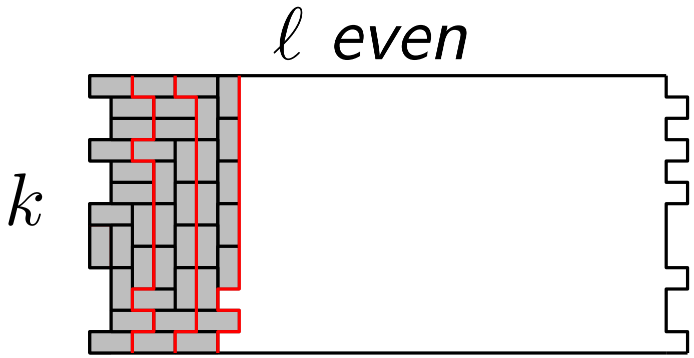
Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

Lemma. *If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing is $b - w$*



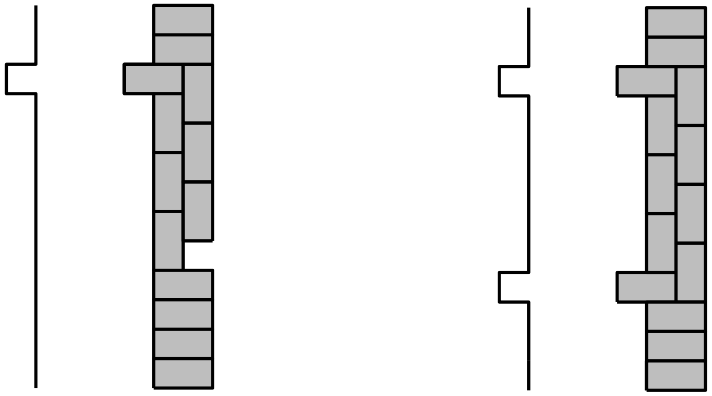
Structural Result 2: Shortening Pipes

Consider 'pipes' of the form:



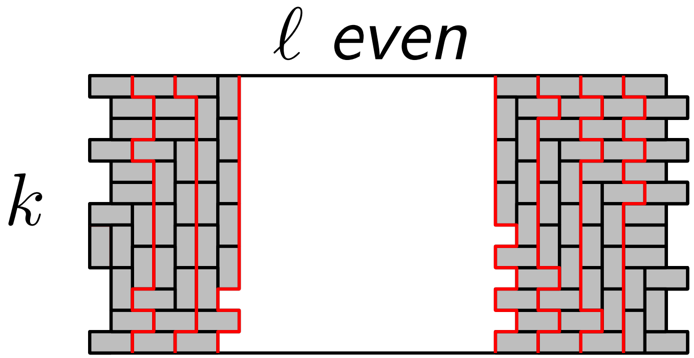
Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

Lemma. *If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing is $b - w$*



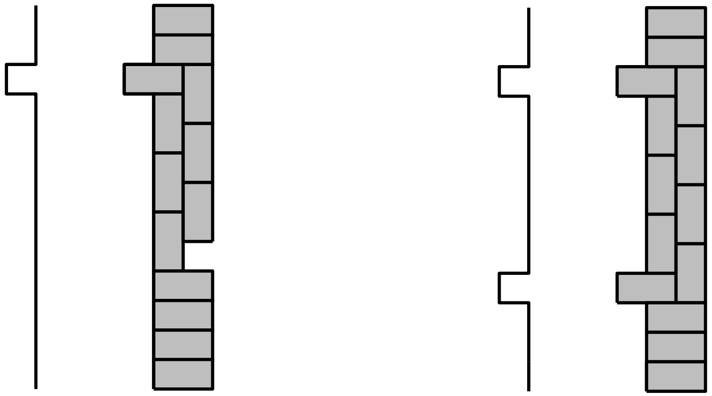
Structural Result 2: Shortening Pipes

Consider 'pipes' of the form:



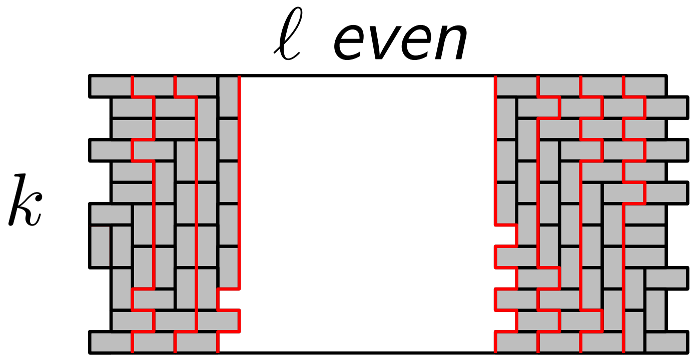
Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

Lemma. *If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing is $b - w$*



Structural Result 2: Shortening Pipes

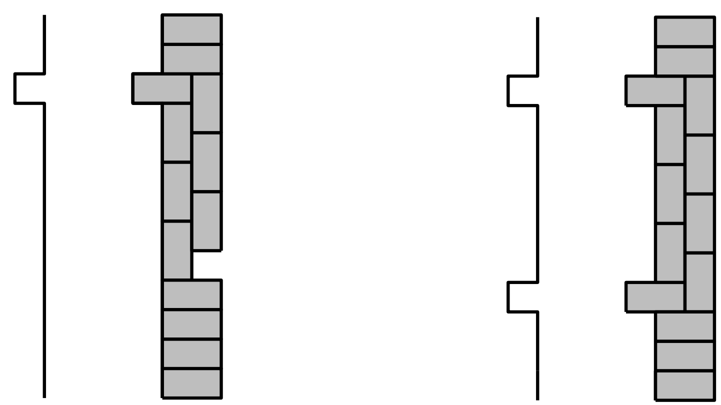
Consider 'pipes' of the form:



Now fill in horizontal dominos

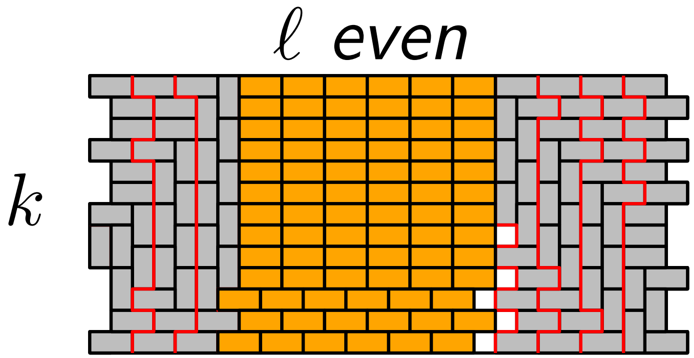
Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

Lemma. *If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing is $b - w$*



Structural Result 2: Shortening Pipes

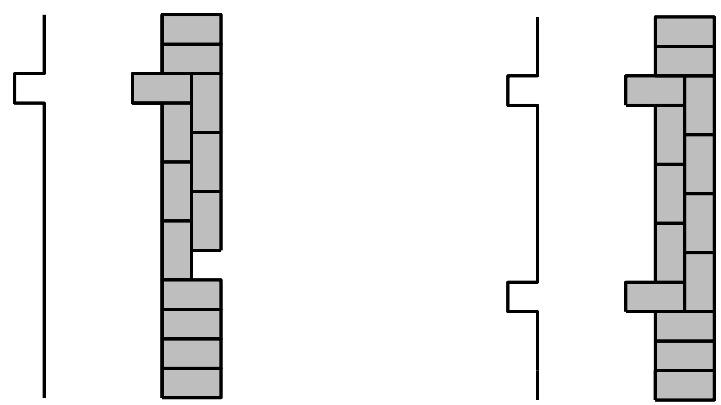
Consider 'pipes' of the form:



Now fill in horizontal dominos

Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

Lemma. *If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing is $b - w$*

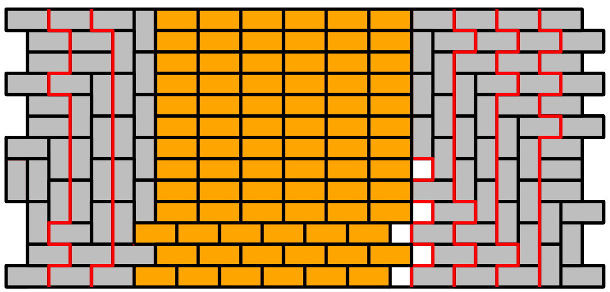


Structural Result 2: Shortening Pipes

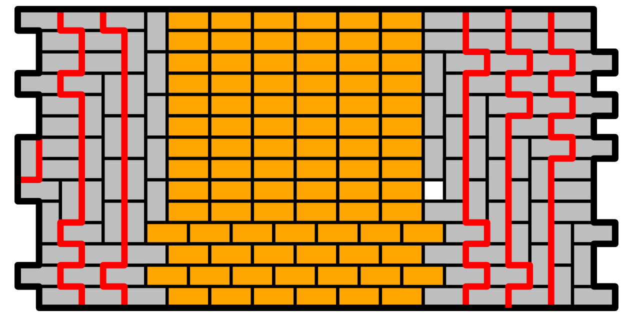
Consider 'pipes' of the form:

ℓ *even*

k

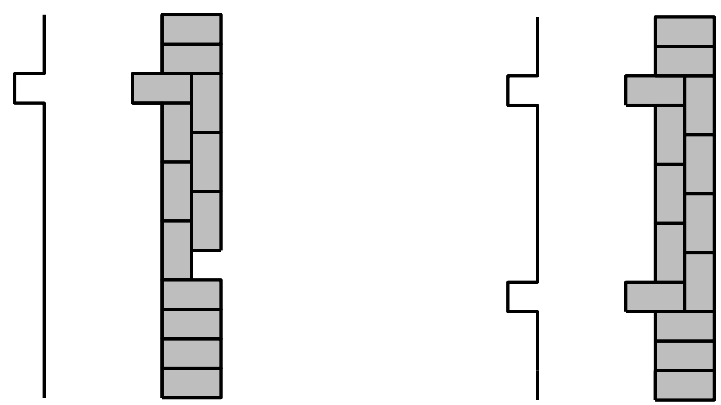


Now fill in horizontal dominos

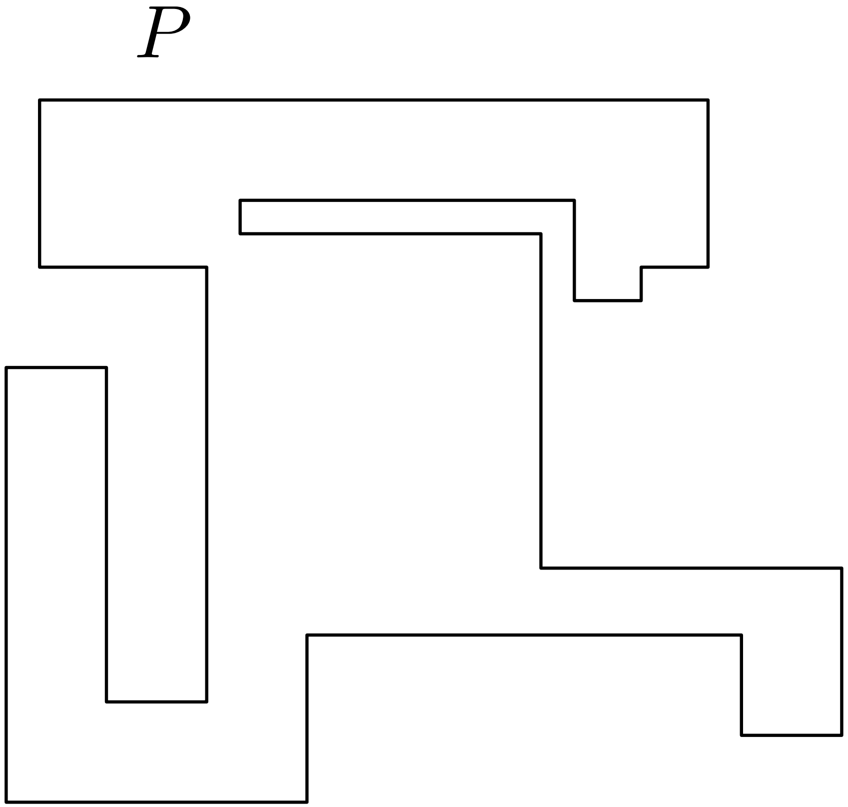


Color black and white in chessboard fashion with b black cells and w white cells. Assume $b \geq w$.

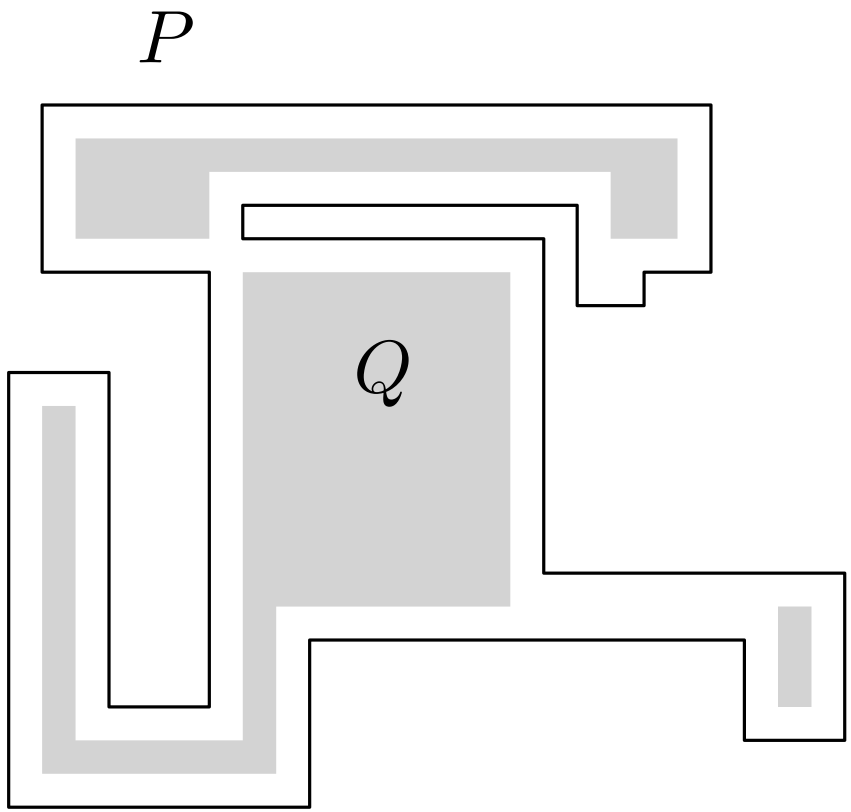
Lemma. *If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing is $b - w$*



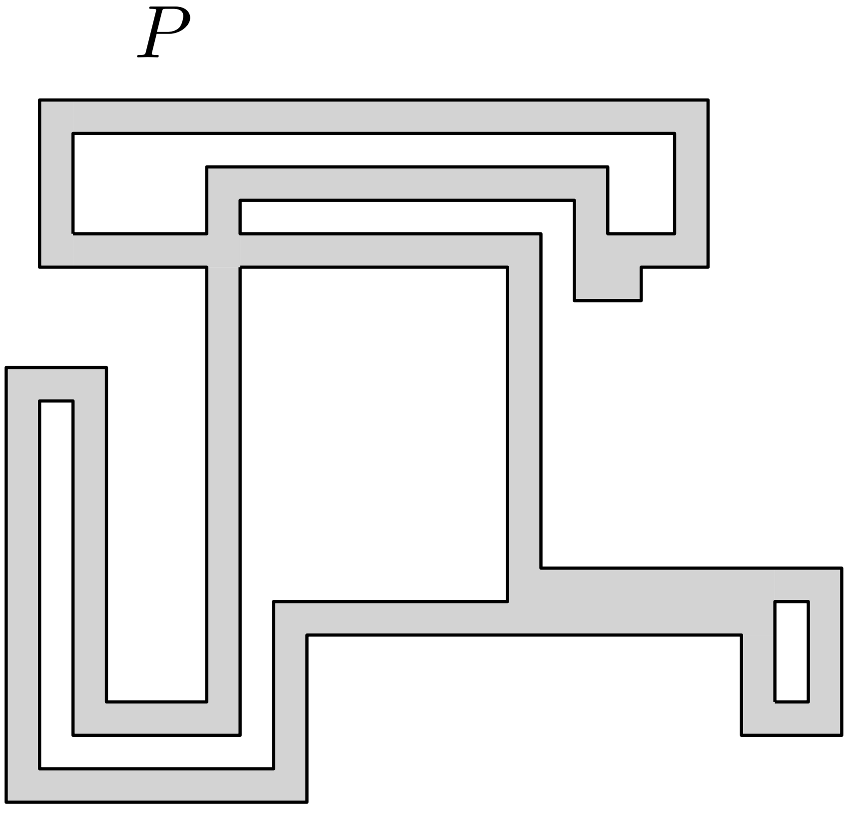
The final reduction



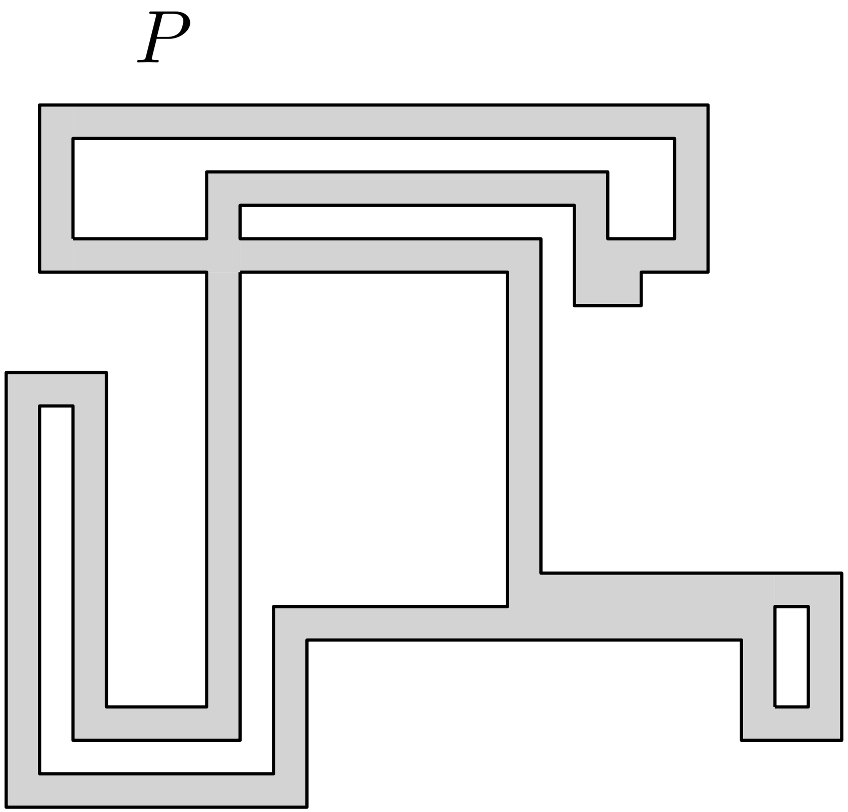
The final reduction



The final reduction

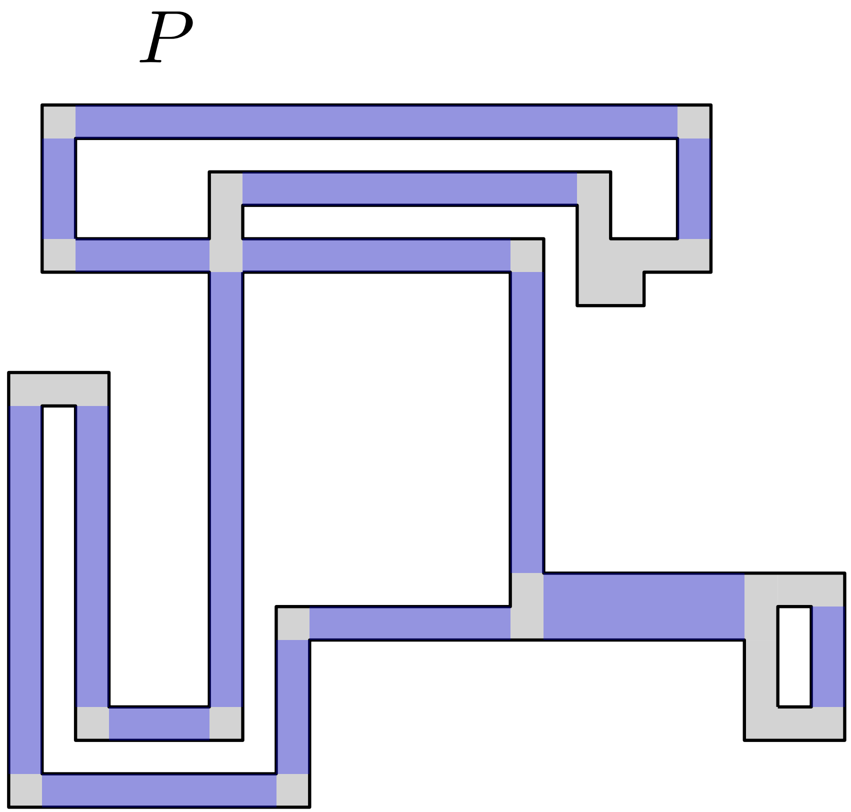


The final reduction



Find all pipes of length at least twice their width.

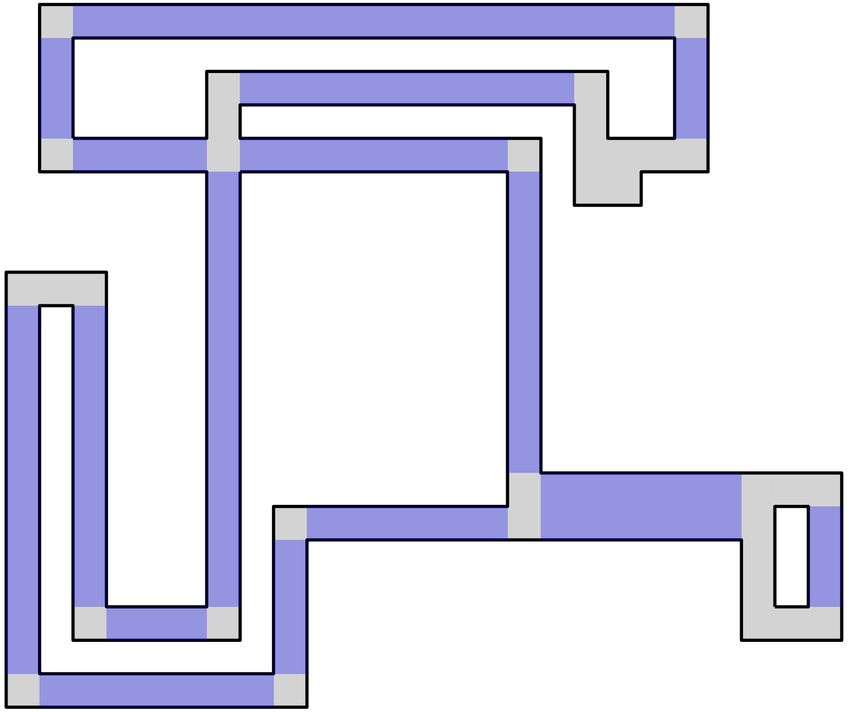
The final reduction



Find all pipes of length at least twice their width.

The final reduction

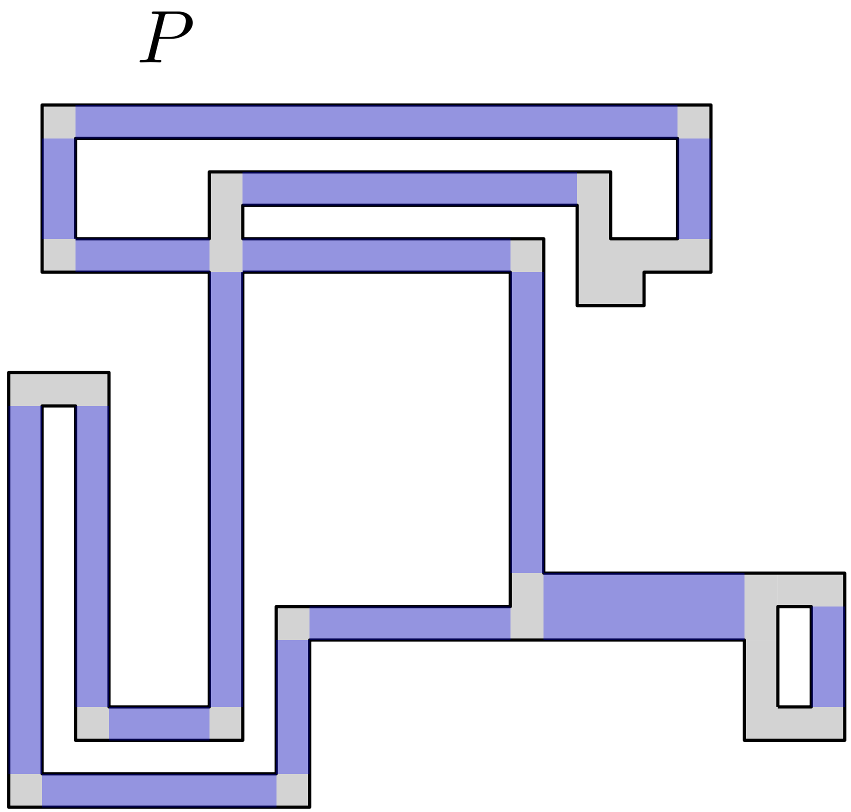
P



Find all pipes of length at least twice their width.

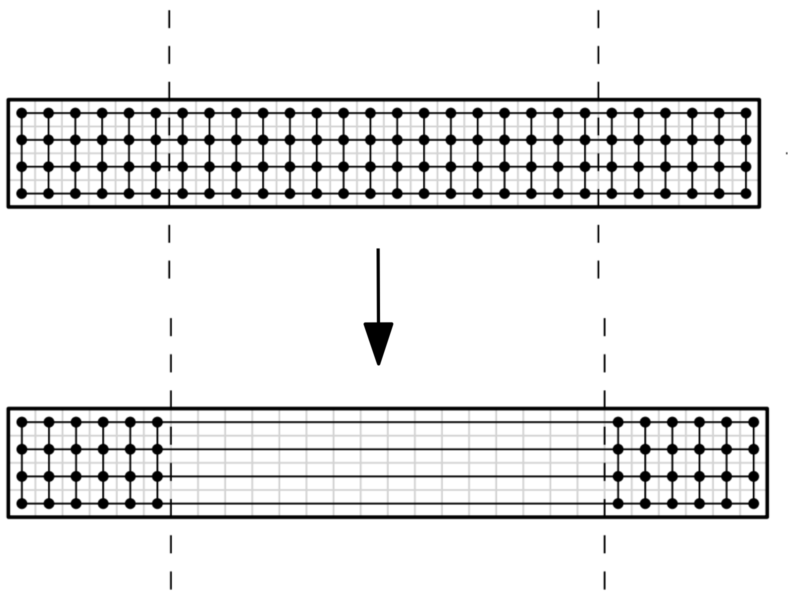
Perform the following operation on each pipe of $G(P')$

The final reduction

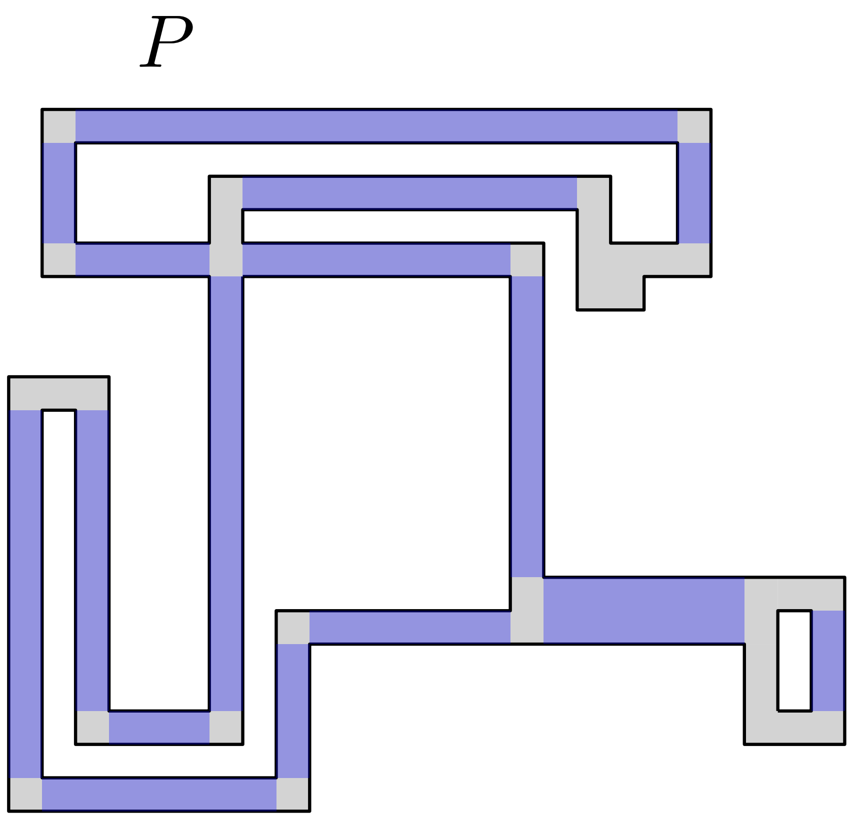


Find all pipes of length at least twice their width.

Perform the following operation on each pipe of $G(P')$

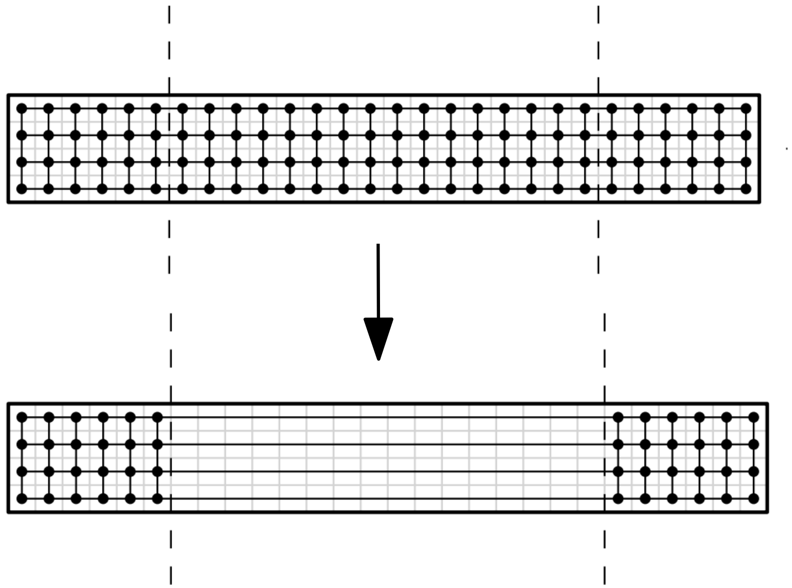


The final reduction



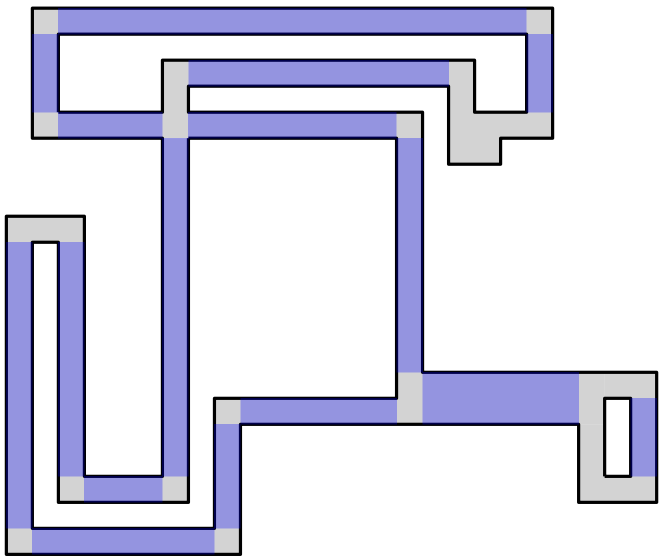
Find all pipes of length at least twice their width.

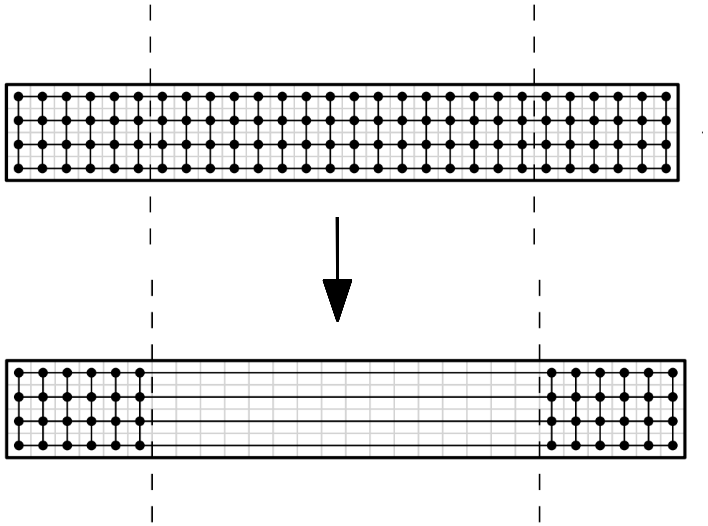
Perform the following operation on each pipe of $G(P')$



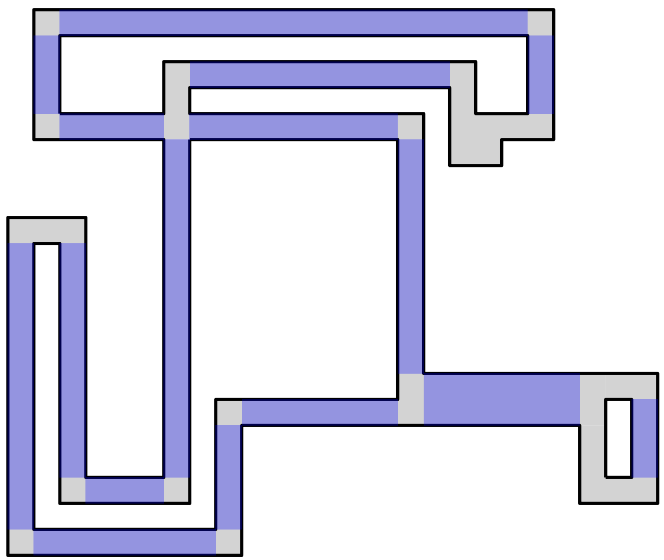
Lemma ensures that no. of unmatched vertices in a maximum matching remains the same

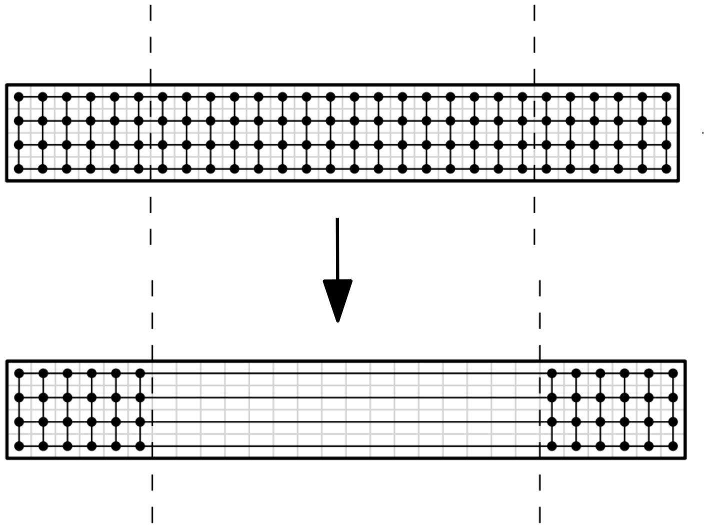
Summing up the reduction





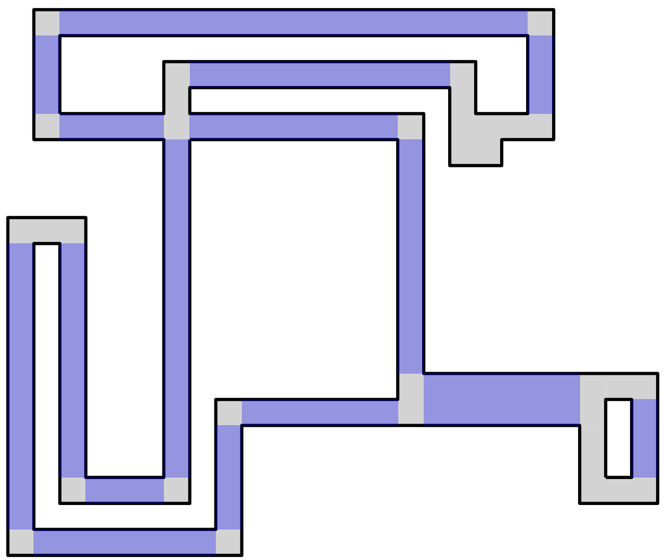
Summing up the reduction

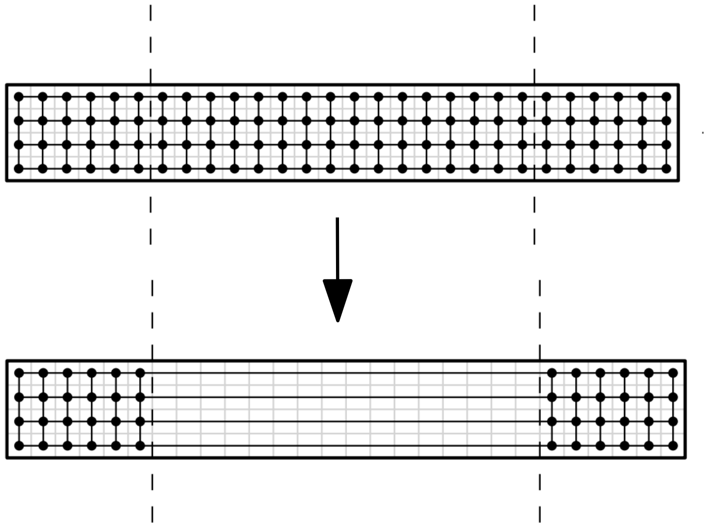




In reduced instance G^*, each vertex is of distance $O(n)$ to a corner.

Summing up the reduction

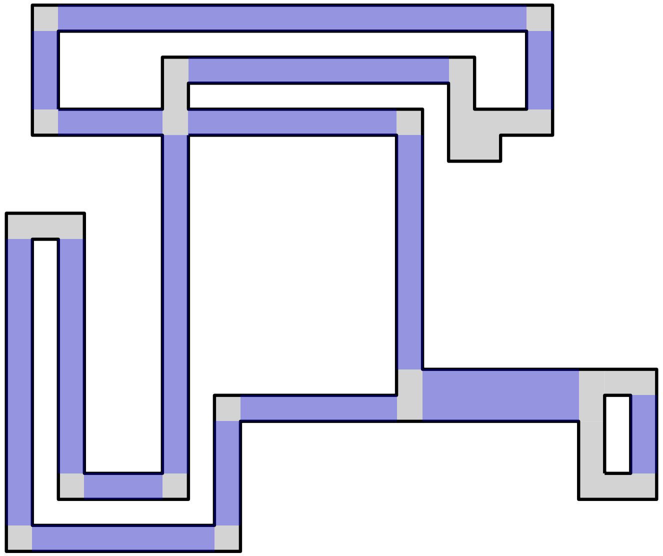


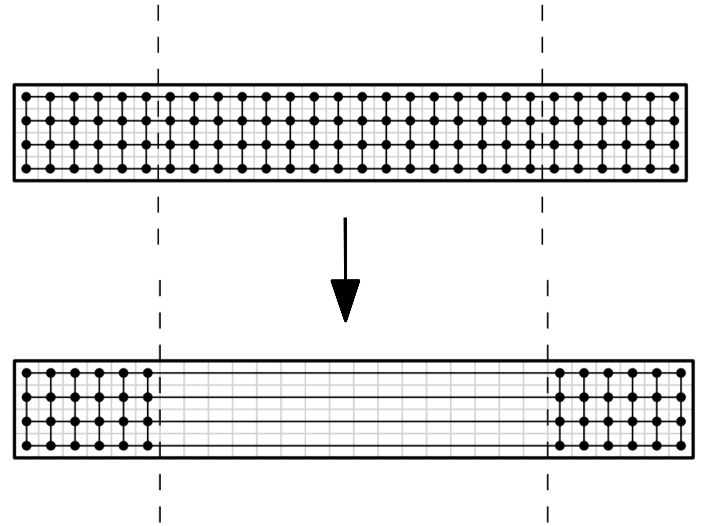


In reduced instance G^*, each vertex is of distance $O(n)$ to a corner.

Thus, G^* has order $O(n^3)$

Summing up the reduction



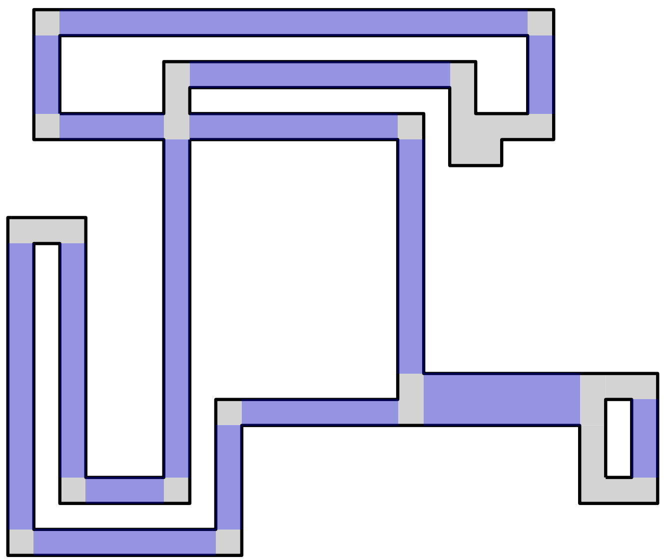


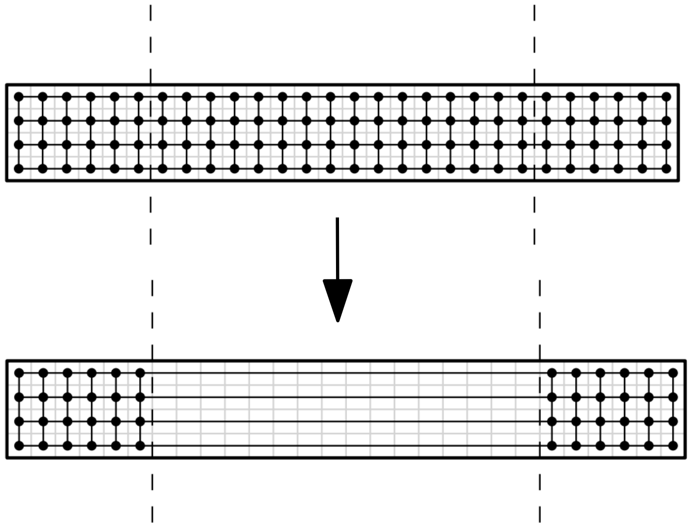
In reduced instance G^*, each vertex is of distance $O(n)$ to a corner.

Thus, G^* has order $O(n^3)$

G^* is planar and bipartite.

Summing up the reduction





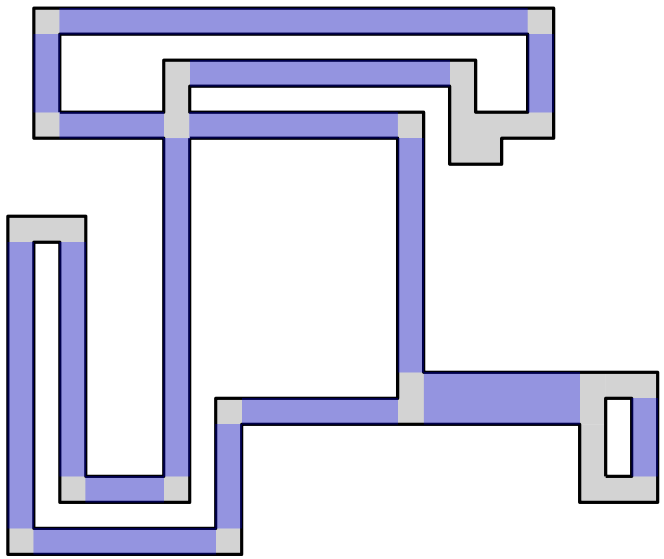
In reduced instance G^*, each vertex is of distance $O(n)$ to a corner.

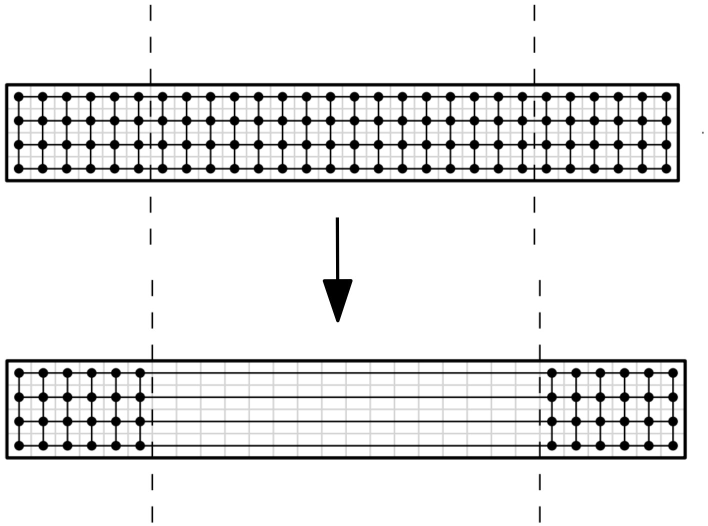
Thus, G^* has order $O(n^3)$

G^* is planar and bipartite.

Find maximum matching M using a multiple-source multiple-sink maximum flow alg., $O(n^3 \log^3 n)$ time.

Summing up the reduction





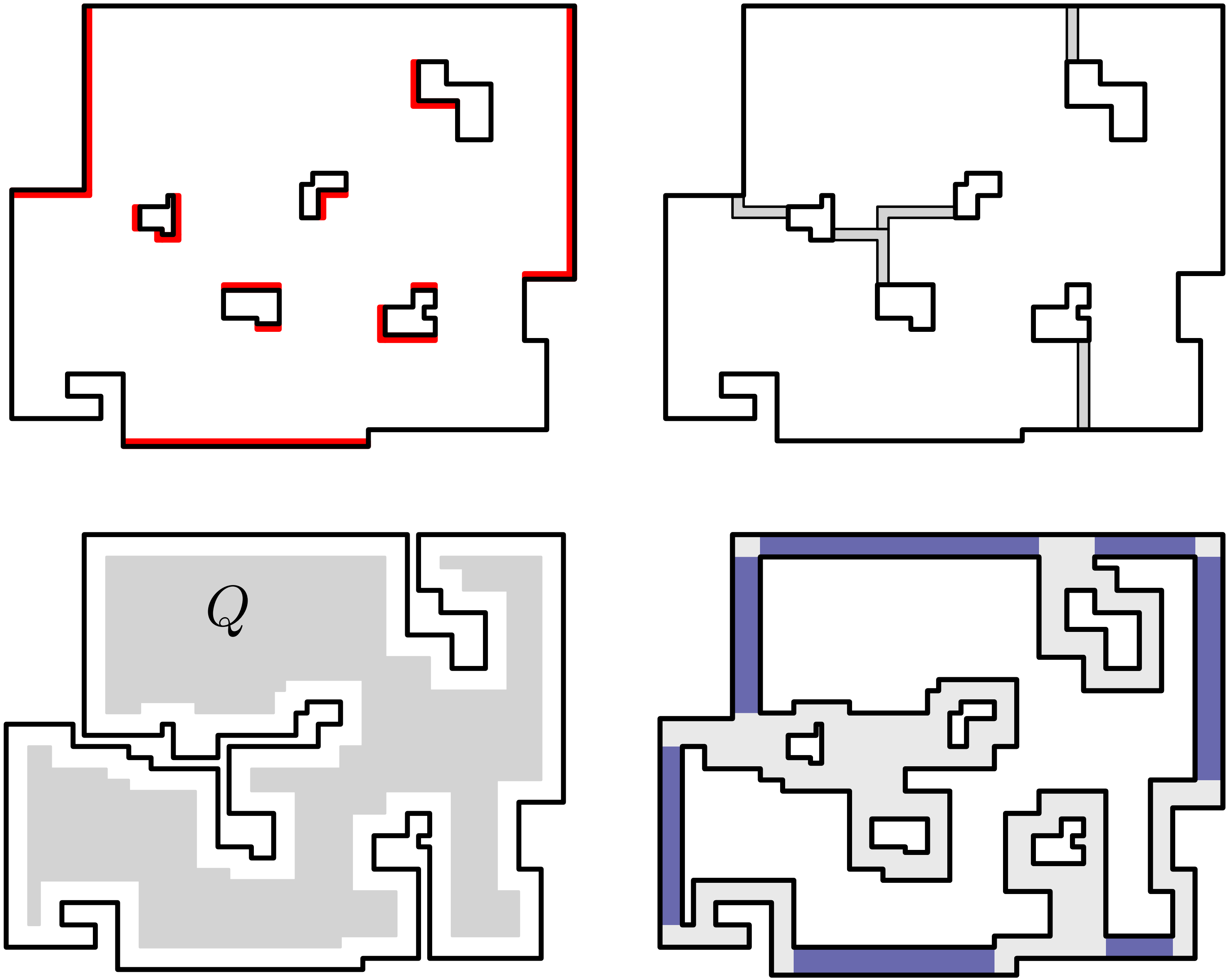
In reduced instance G^*, each vertex is of distance $O(n)$ to a corner.
Thus, G^* has order $O(n^3)$
G^* is planar and bipartite.
Find maximum matching M using a multiple-source multiple-sink maximum flow alg., $O(n^3 \log^3 n)$ time.

Return $|M| + \frac{\text{area}(P) - V(G^*)}{2}$.

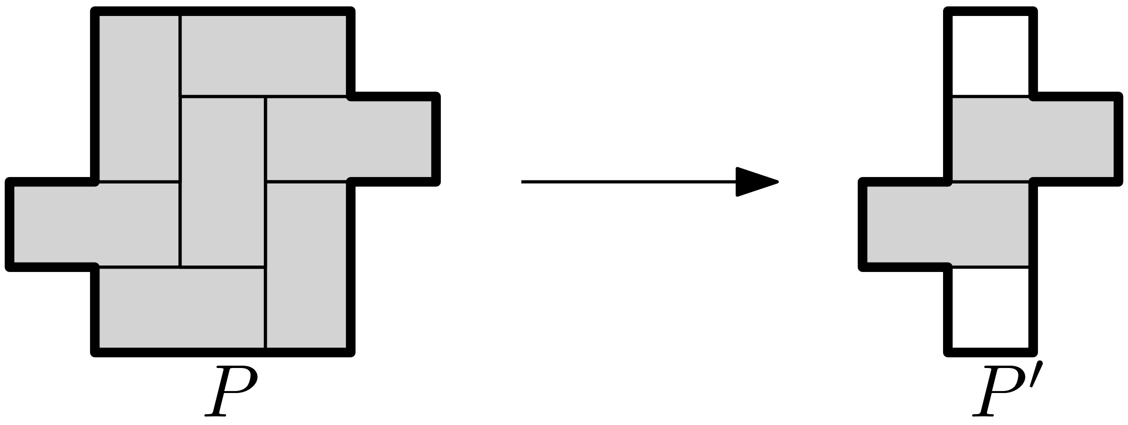
The total running time

	Running time:
Compute P_0	$O(n \log n)$
Compute offset	$O(n \log n)$
Find long pipes	$O(n \log n)$
Find maximum matching	$O(n^3 \log^3 n)$

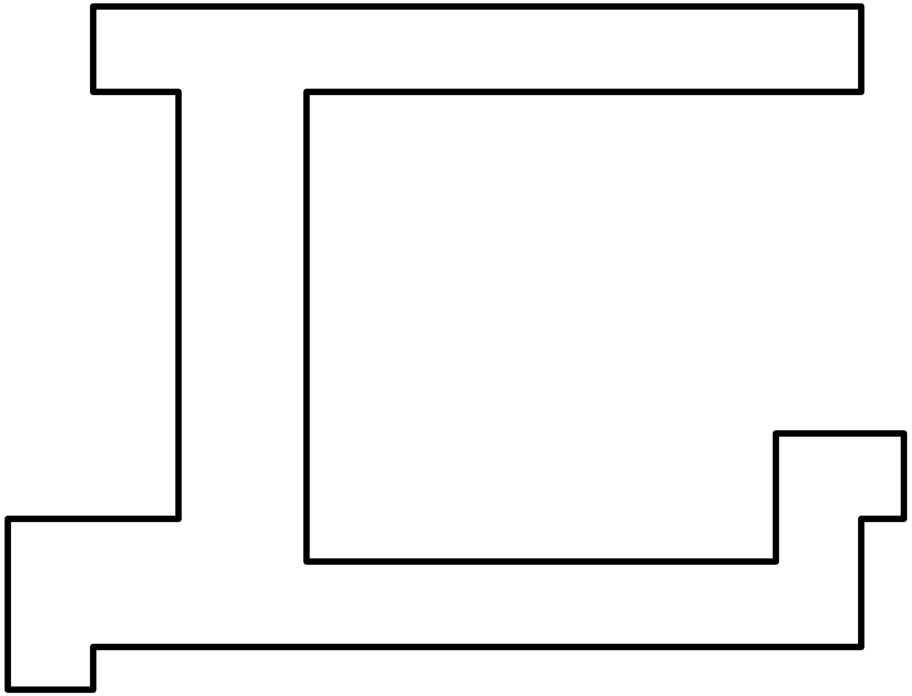
What if there are holes?



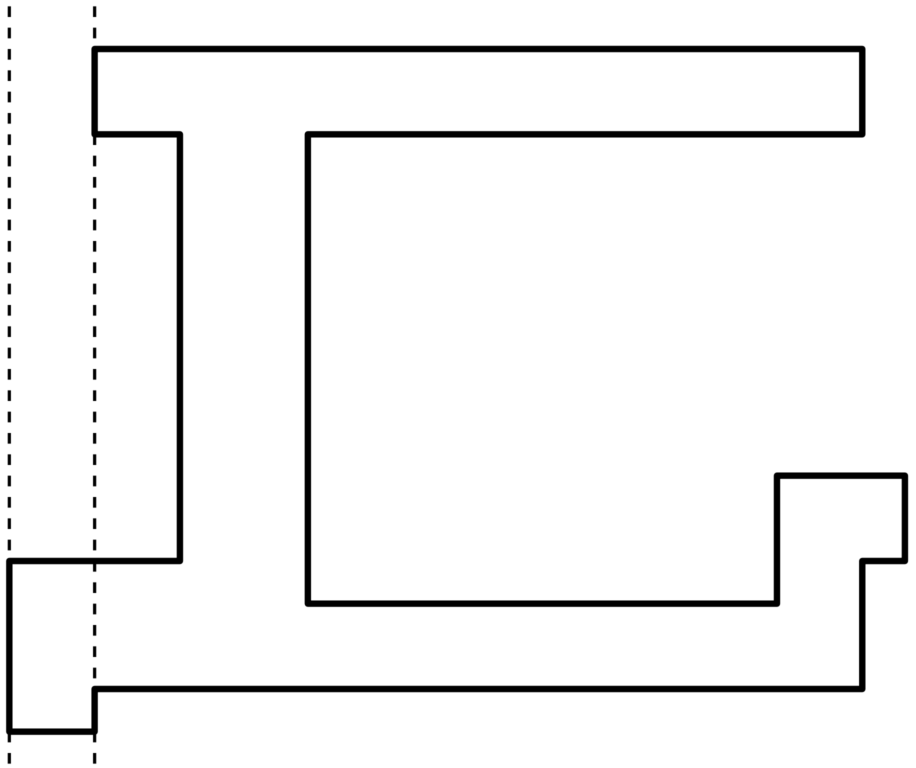
Simple algorithm



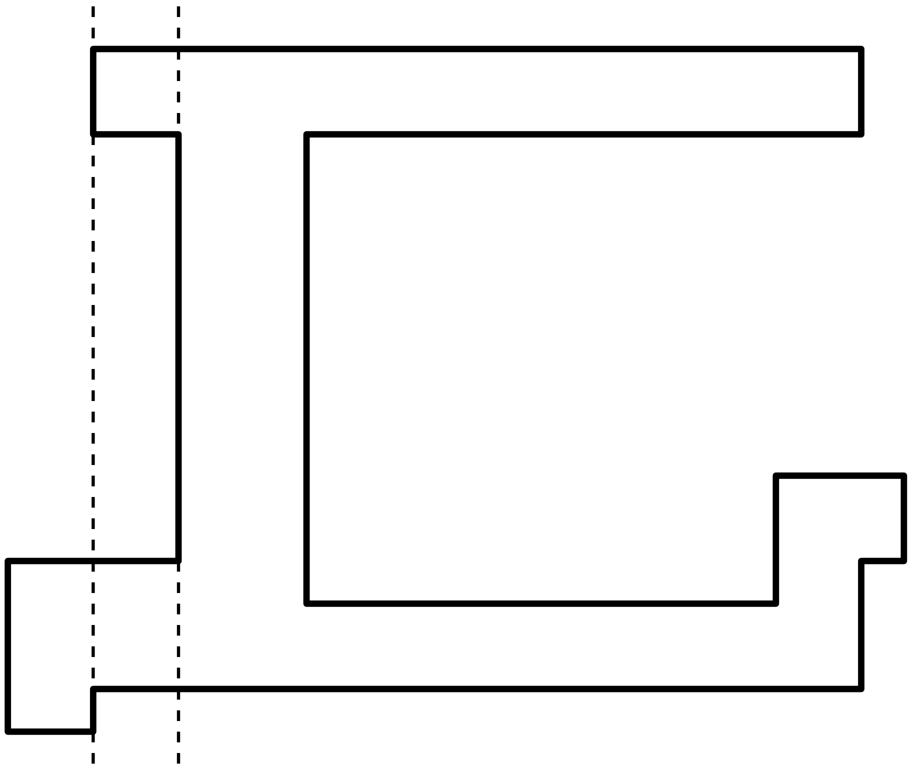
Simple algorithm



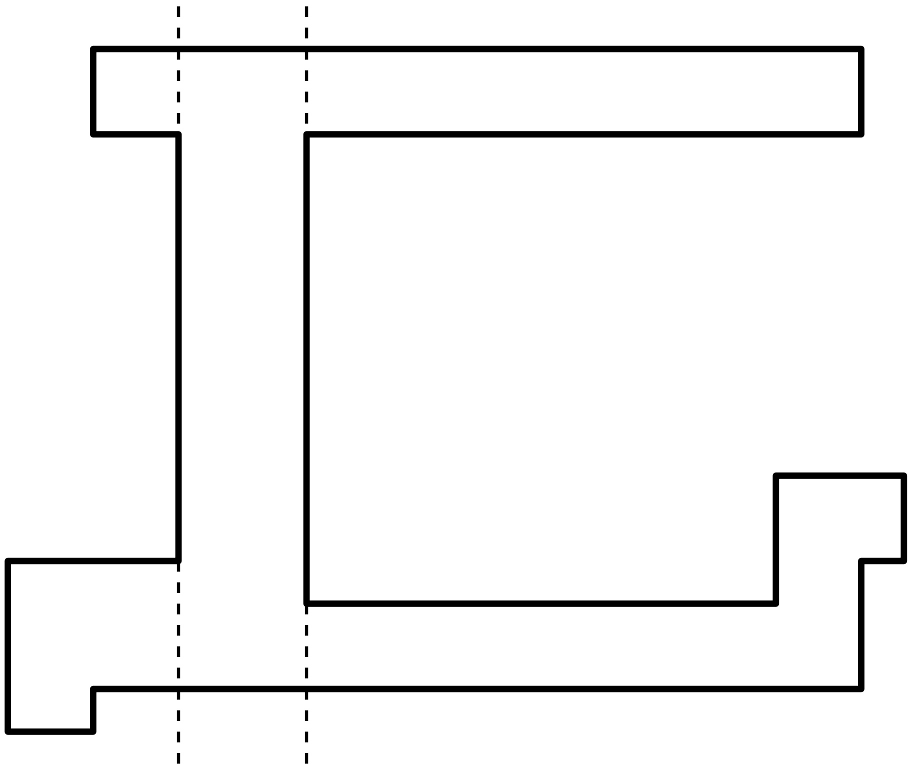
Simple algorithm



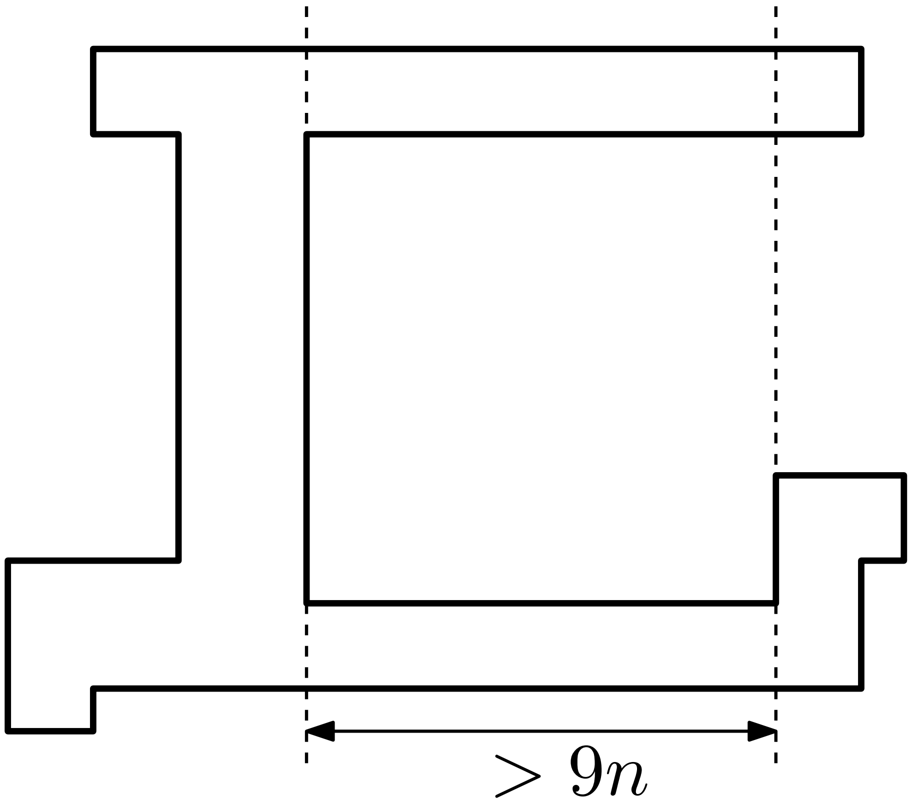
Simple algorithm



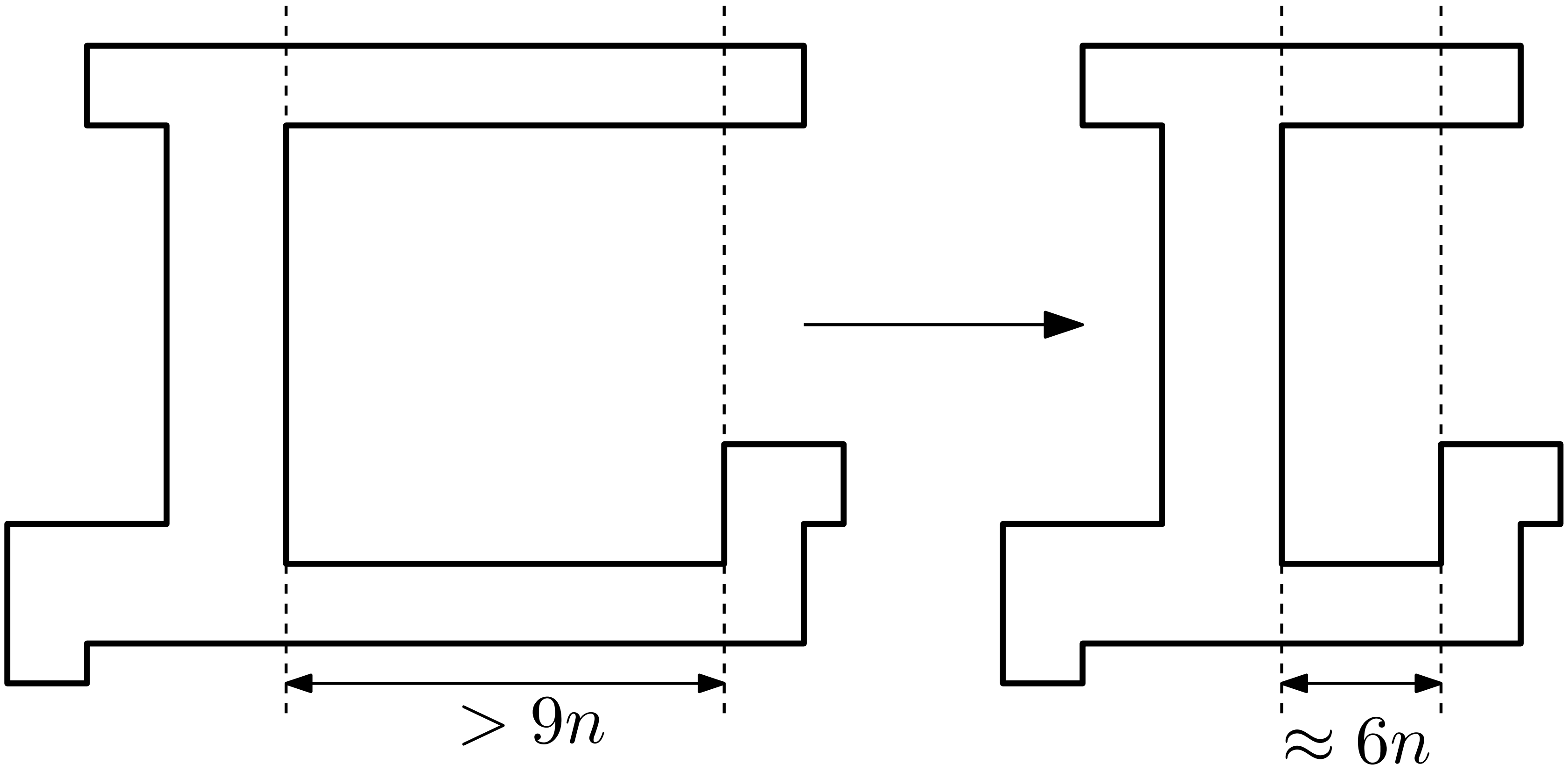
Simple algorithm



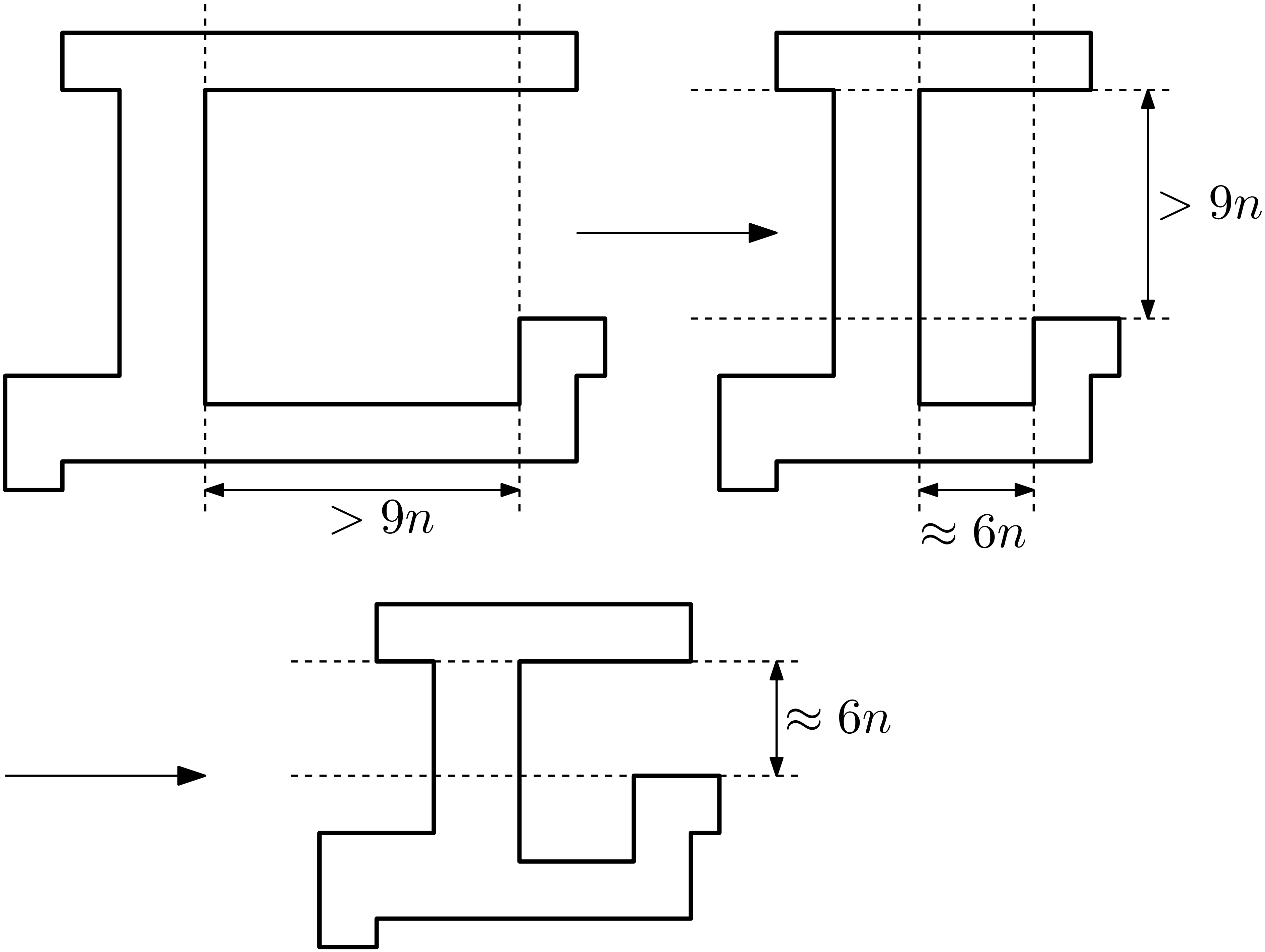
Simple algorithm



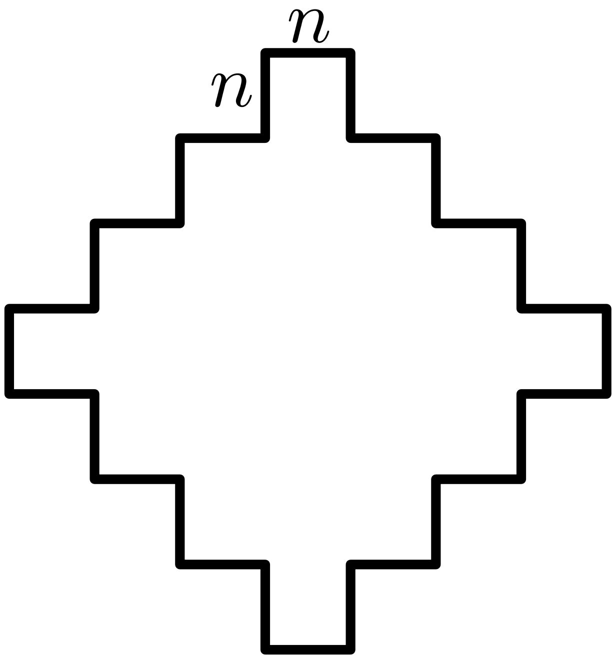
Simple algorithm



Simple algorithm

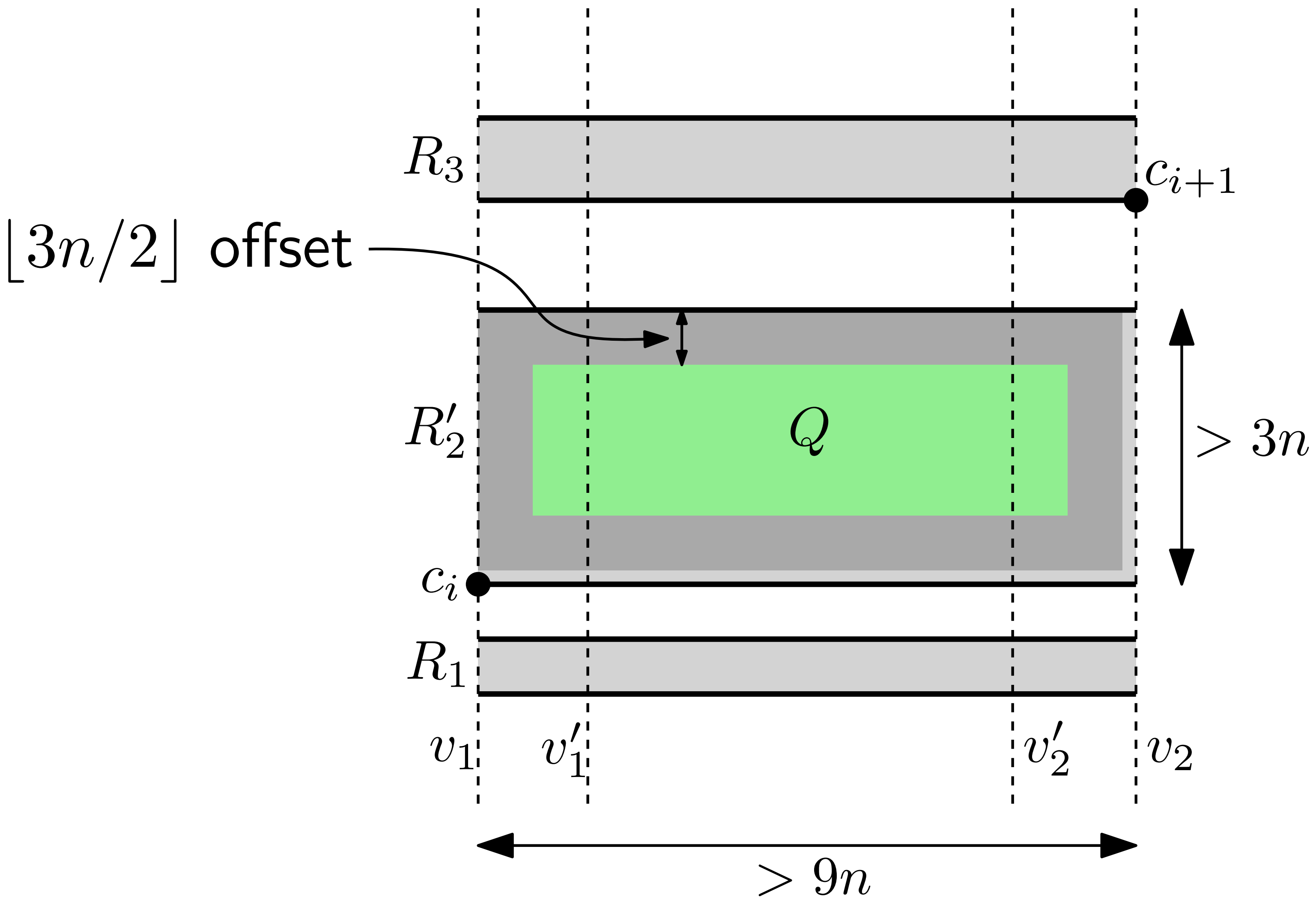


Running time



$$\widetilde{O}(n^4)$$

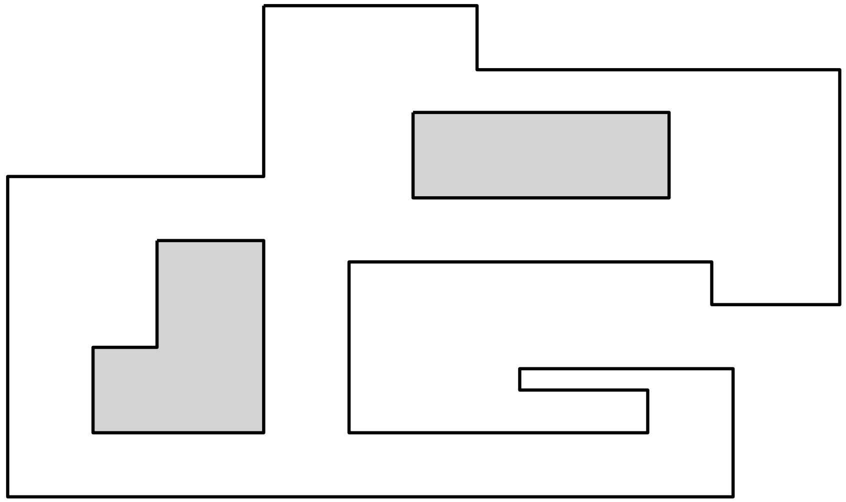
Correctness of simple algorithm



Open Problems

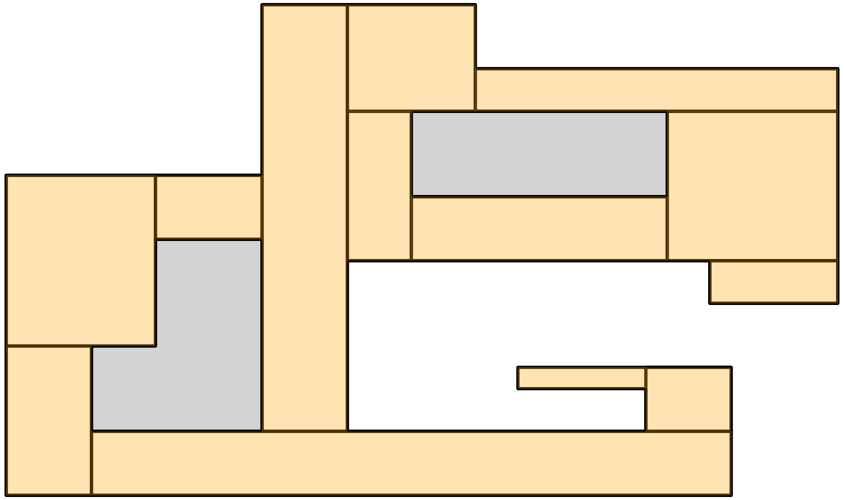
Open Problems

Can domino tiling/packing be solved faster with a reduction to a flow problem?



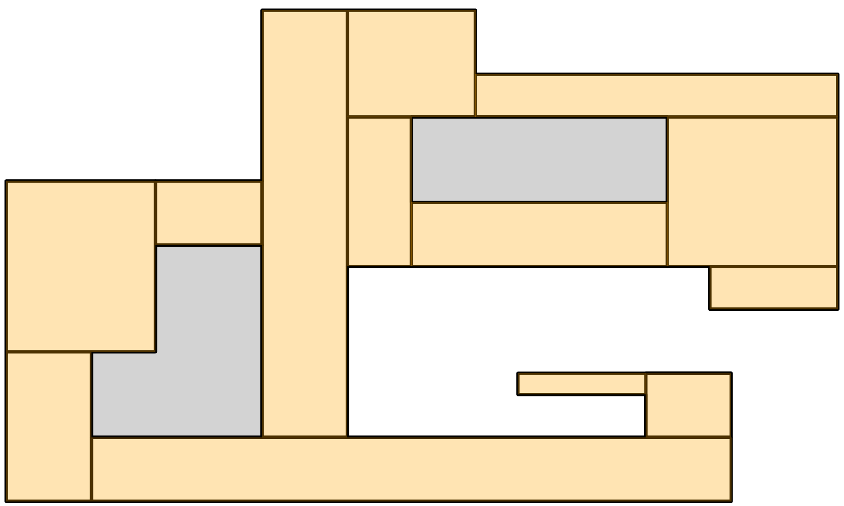
Open Problems

Can domino tiling/packing be solved faster with a reduction to a flow problem?



Open Problems

Can domino tiling/packing be solved faster with a reduction to a flow problem?



Packing 2×2 squares is NP-complete when P has holes. Can it be solved in polynomial time if P is hole-free?