Approximation Algorithms

Lecture 9:
An Approximation Scheme for Euclidean TSP

Part I:
The Traveling Salesman Problem

Alexander Wolff

Winter term 2022/23

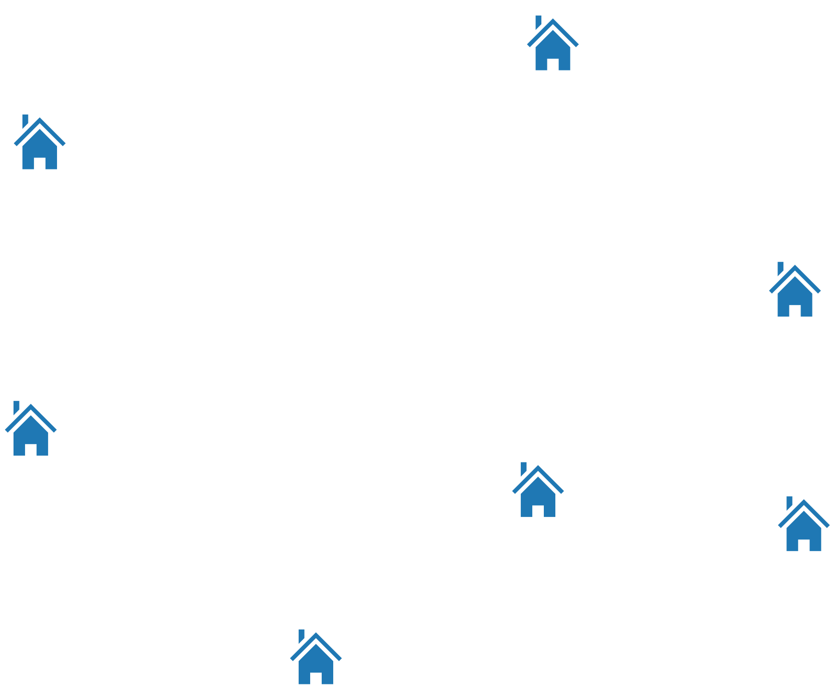
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

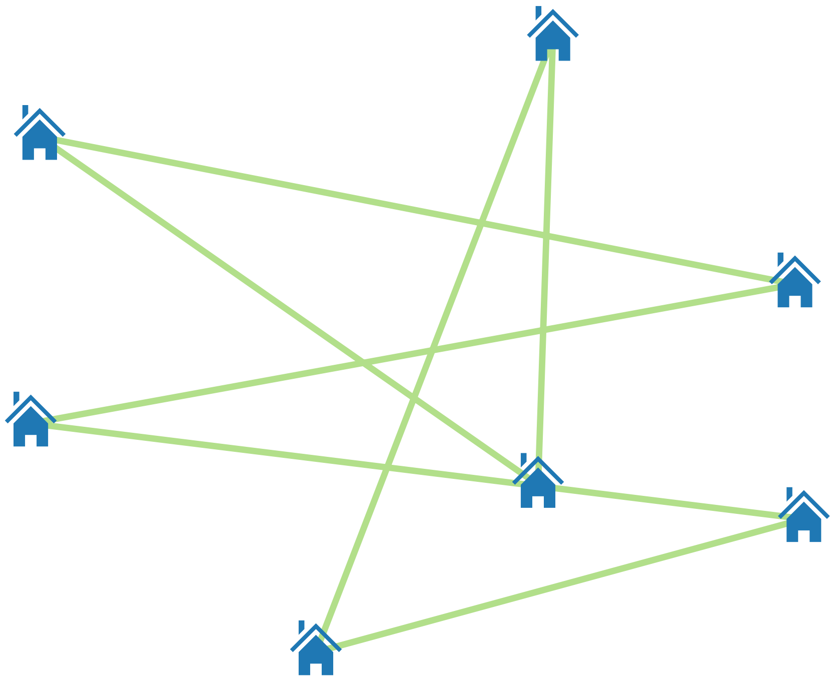


Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle)

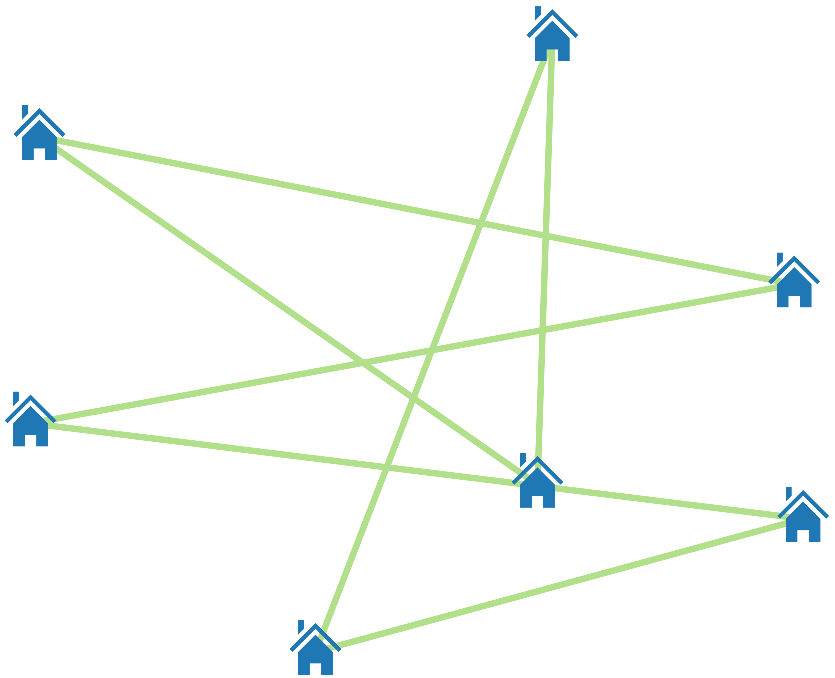


Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle)

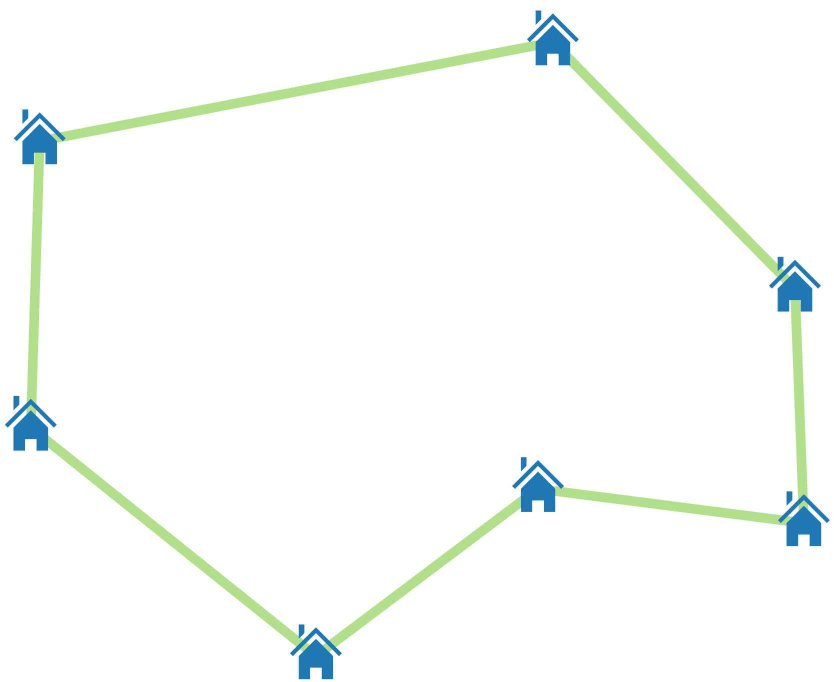


Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

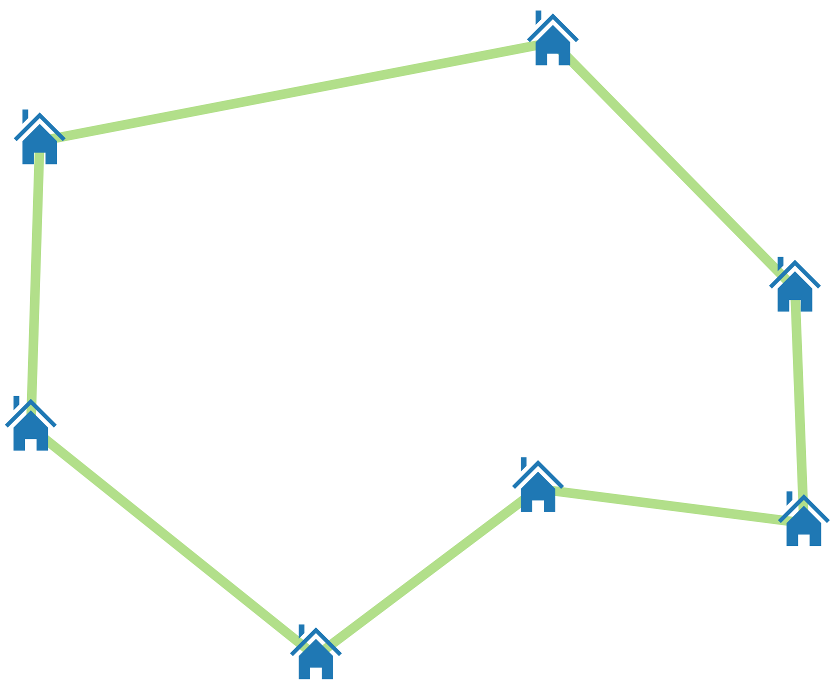


Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.



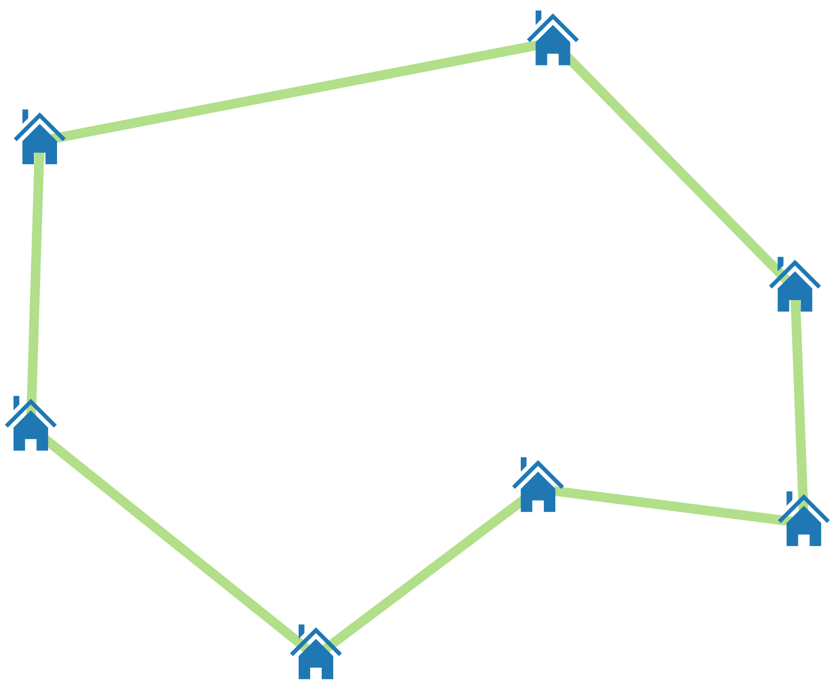
For every polynomial $p(n)$, TSP cannot be approximated within factor (unless $P = NP$).

Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.



For every polynomial $p(n)$, TSP cannot be approximated within factor $2^{p(n)}$ (unless $P = NP$).

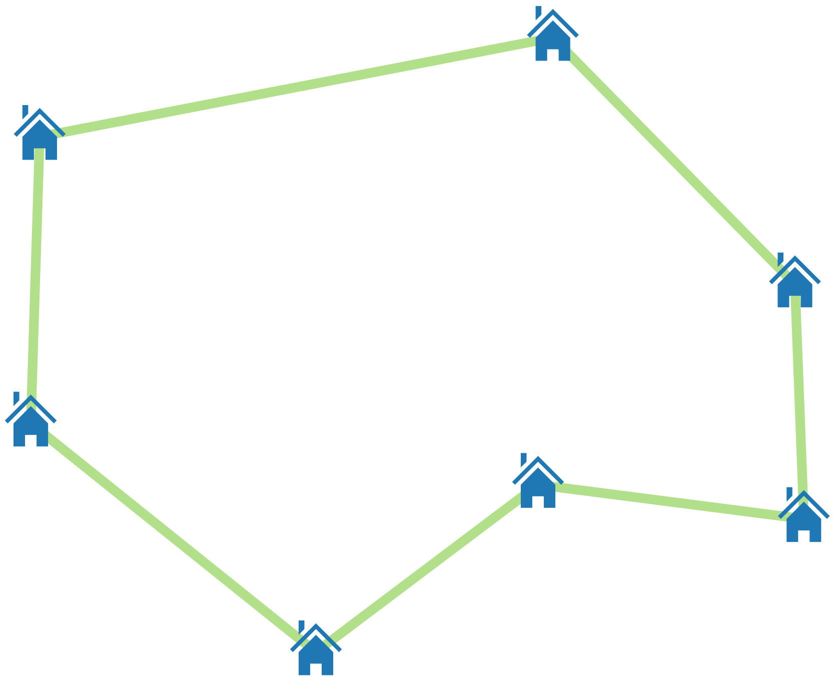
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Distance between two points?



For every polynomial $p(n)$, TSP cannot be approximated within factor $2^{p(n)}$ (unless $P = NP$).

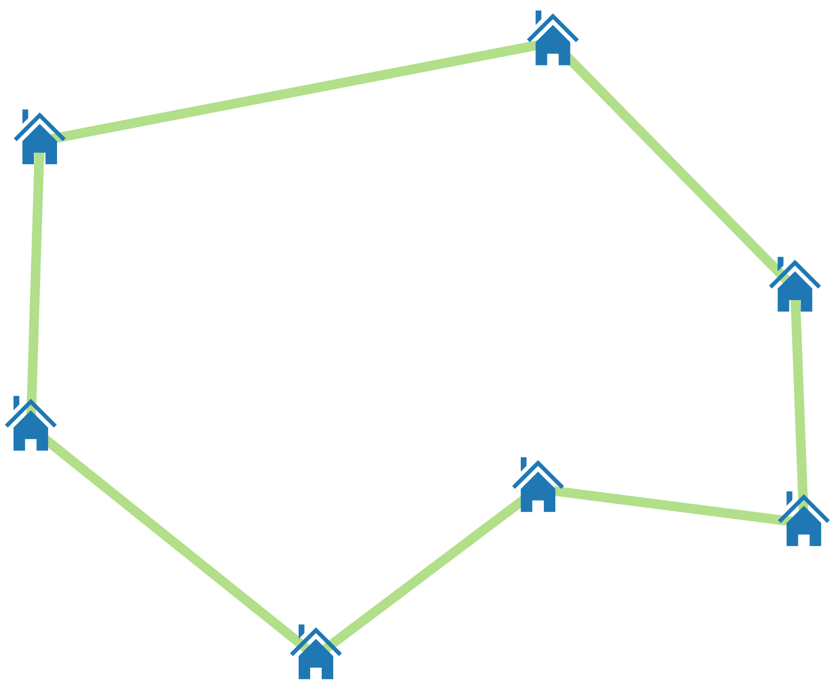
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Distance between two points?



For every polynomial $p(n)$, TSP cannot be approximated within factor $2^{p(n)}$ (unless $P = NP$).

There is a $3/2$-approximation algorithm for Metric TSP.

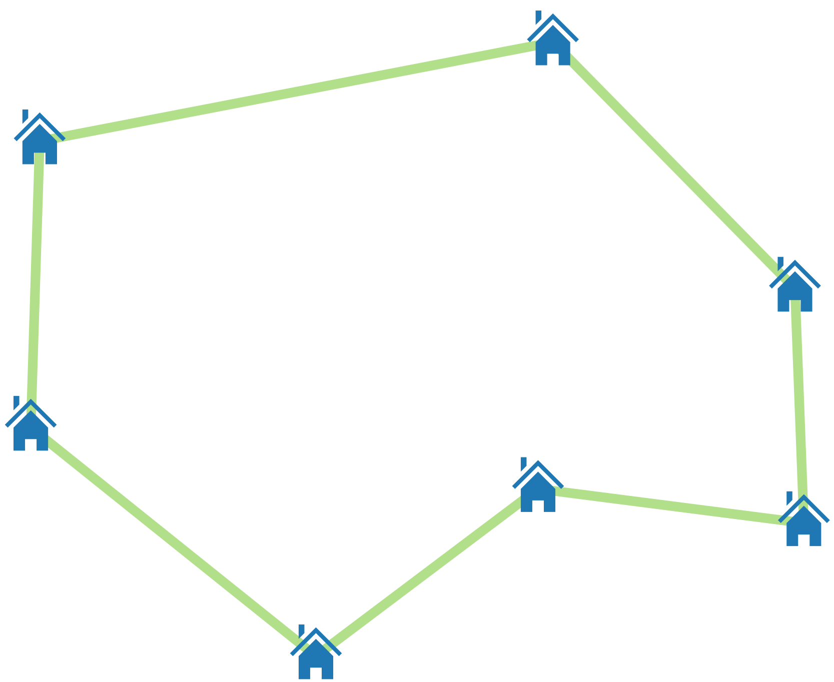
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Distance between two points?



For every polynomial $p(n)$, TSP cannot be approximated within factor $2^{p(n)}$ (unless $P = NP$).

There is a $3/2$-approximation algorithm for Metric TSP.

Metric TSP cannot be approximated within factor $123/122$ (unless $P = NP$).

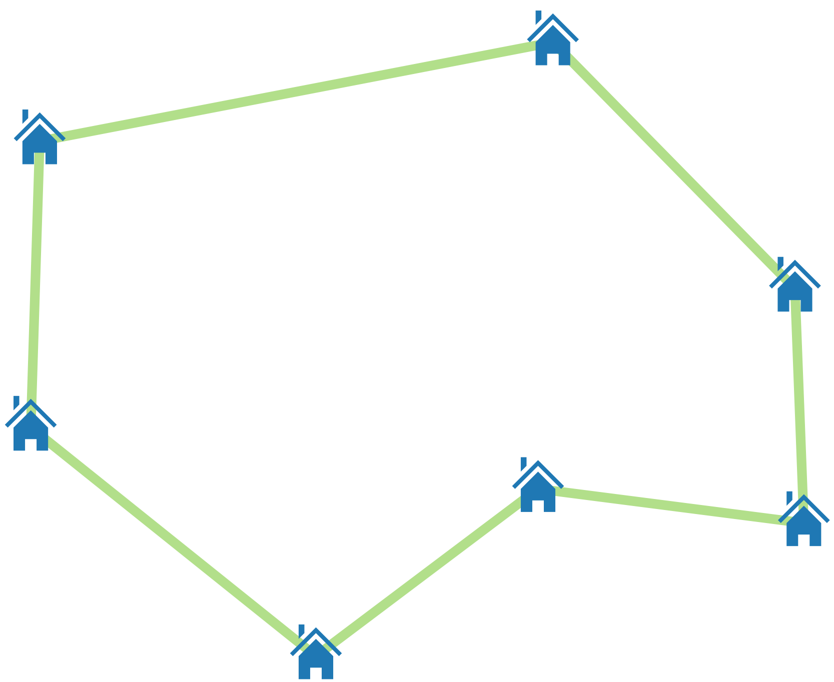
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Let's assume that the salesman flies $\Rightarrow$ Euclidean distances.



For every polynomial $p(n)$, TSP cannot be approximated within factor $2^{p(n)}$ (unless $\mathrm{P} = \mathrm{NP}$).

There is a $3/2$-approximation algorithm for Metric TSP.

Metric TSP cannot be approximated within factor $123/122$ (unless $\mathrm{P} = \mathrm{NP}$).

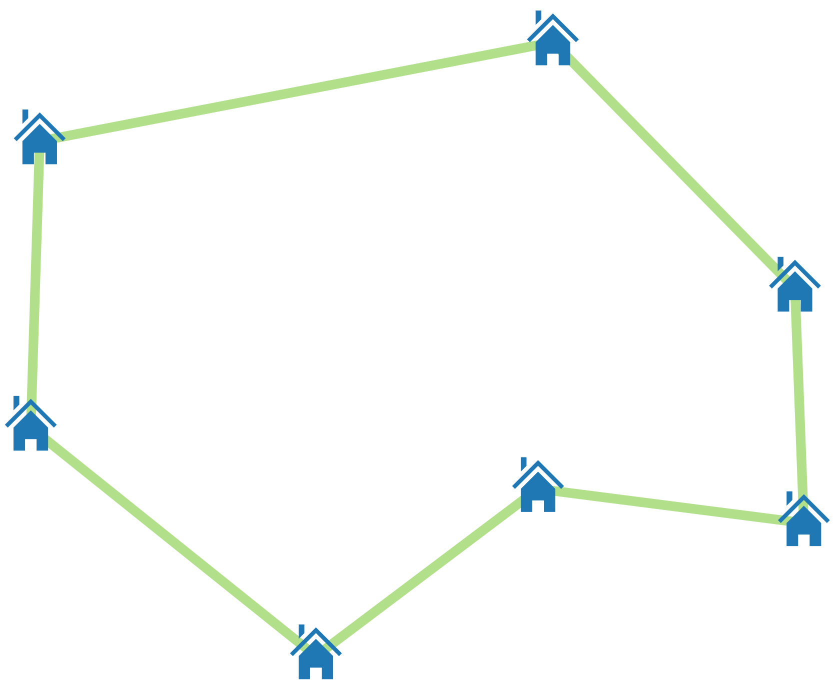
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Let's assume that the salesman flies $\Rightarrow$ Euclidean distances.



Simplifying Assumptions

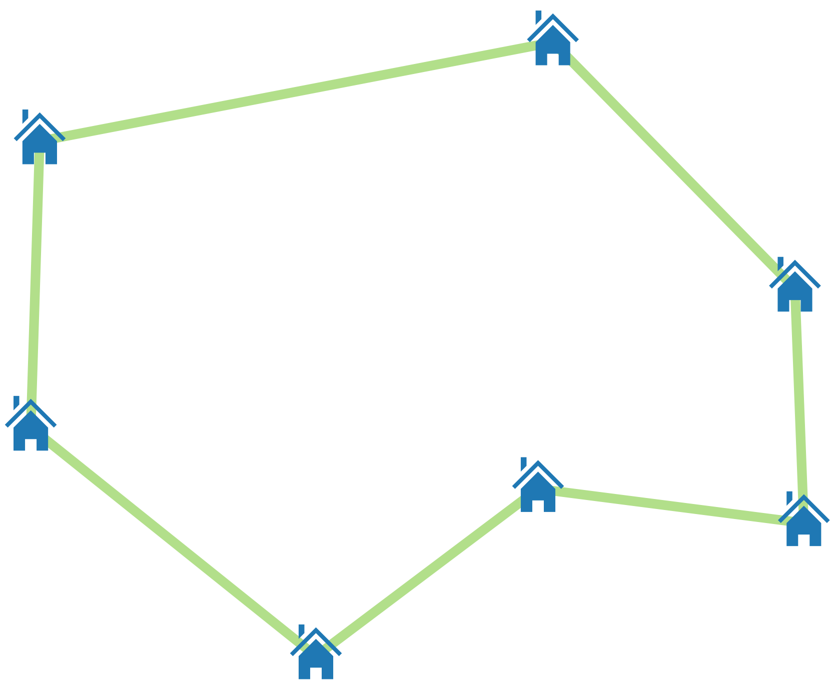
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Let's assume that the salesman flies $\Rightarrow$ Euclidean distances.



Simplifying Assumptions

- Houses inside $(L \times L)$-square

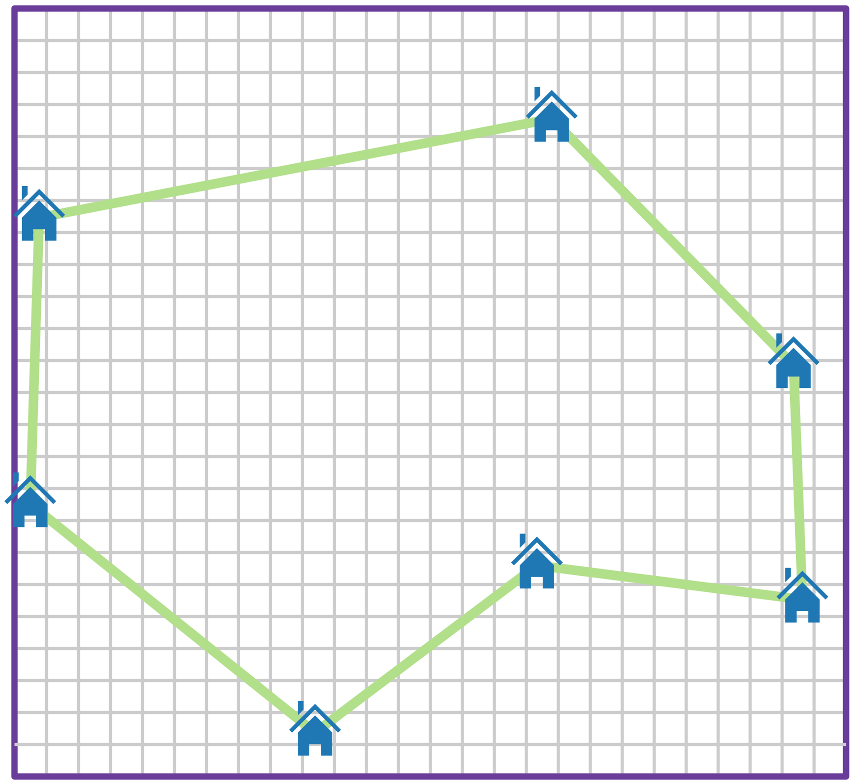
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Let's assume that the salesman flies $\Rightarrow$ Euclidean distances.



Simplifying Assumptions

- Houses inside $(L \times L)$-square

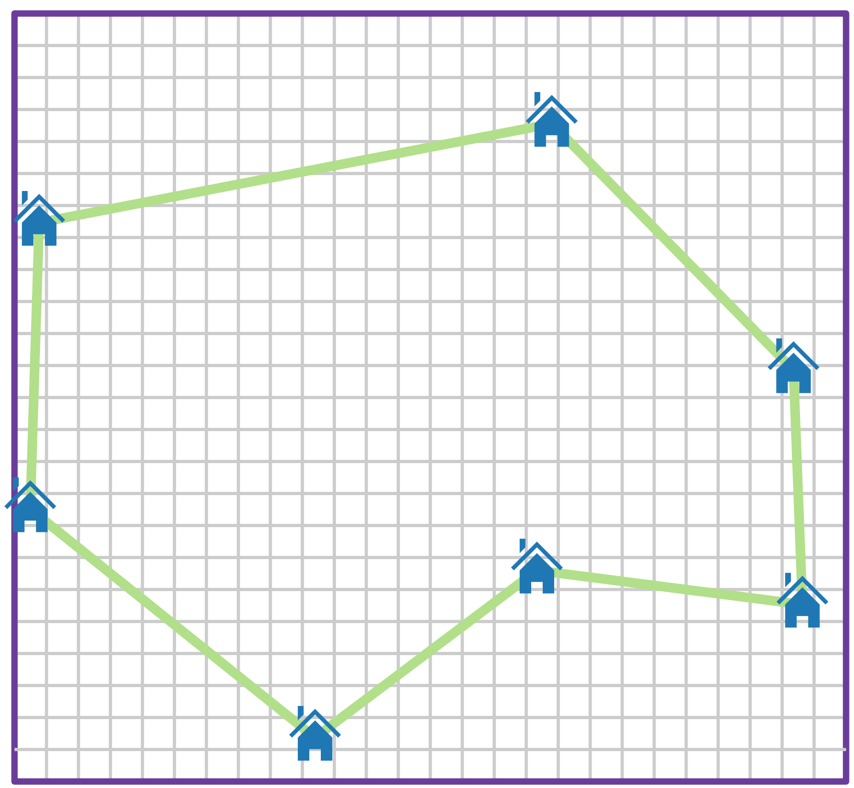
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Let's assume that the salesman flies $\Rightarrow$ Euclidean distances.



Simplifying Assumptions

- Houses inside $(L \times L)$-square
- $L := 4n^2$

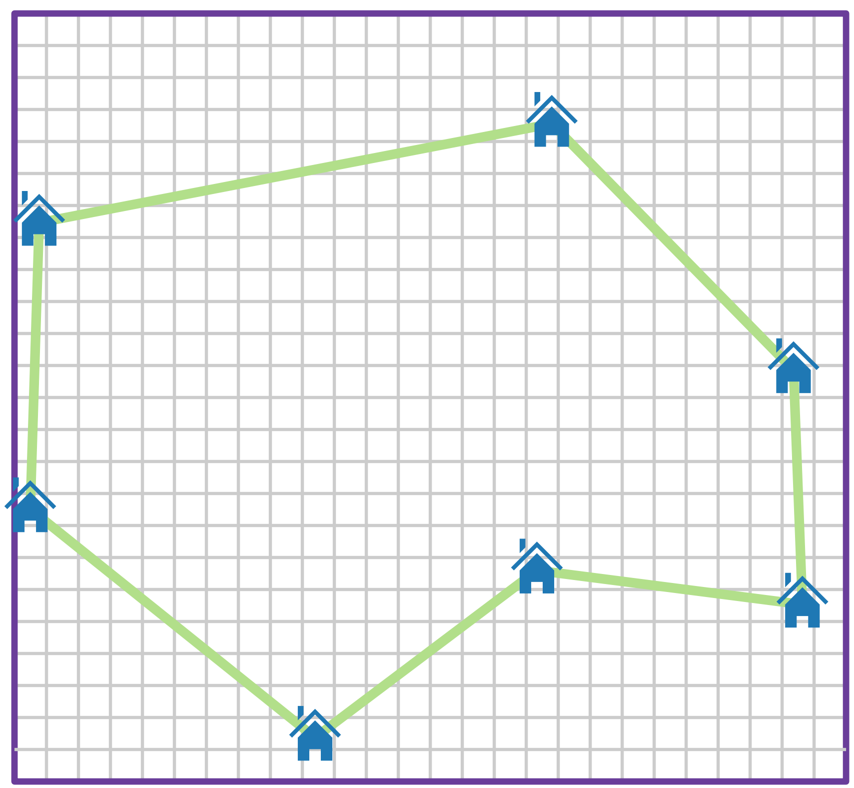
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Let's assume that the salesman flies $\Rightarrow$ Euclidean distances.



Simplifying Assumptions

- Houses inside $(L \times L)$-square
- $L := 4n^2 = 2^k$;

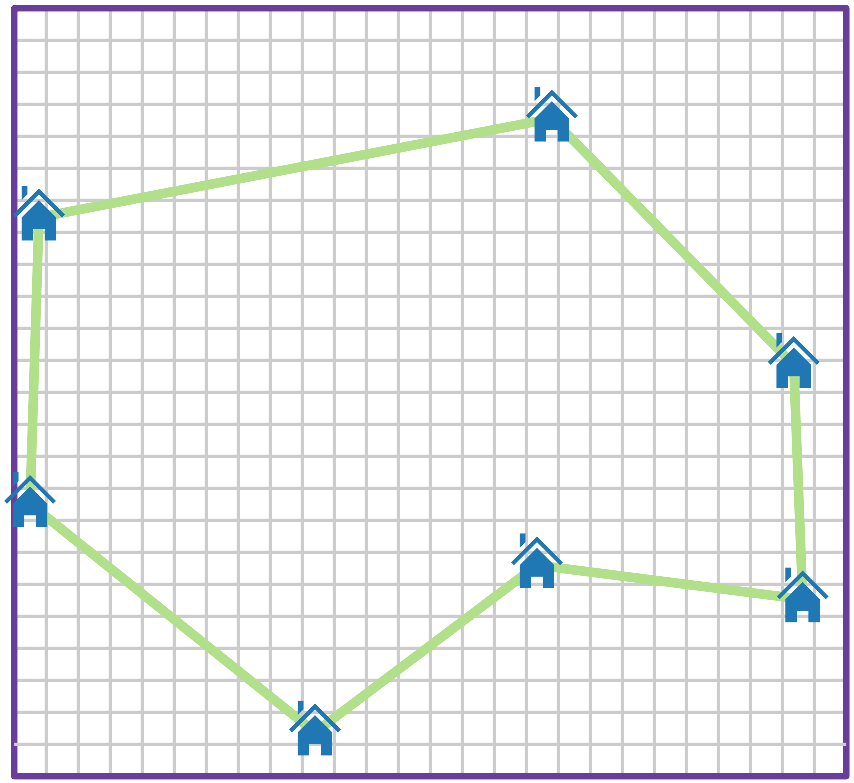
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Let's assume that the salesman flies $\Rightarrow$ Euclidean distances.



Simplifying Assumptions

- Houses inside ($L \times L$)-square
- $L := 4n^2 = 2^k$; $k = 2 + 2\log_2 n$

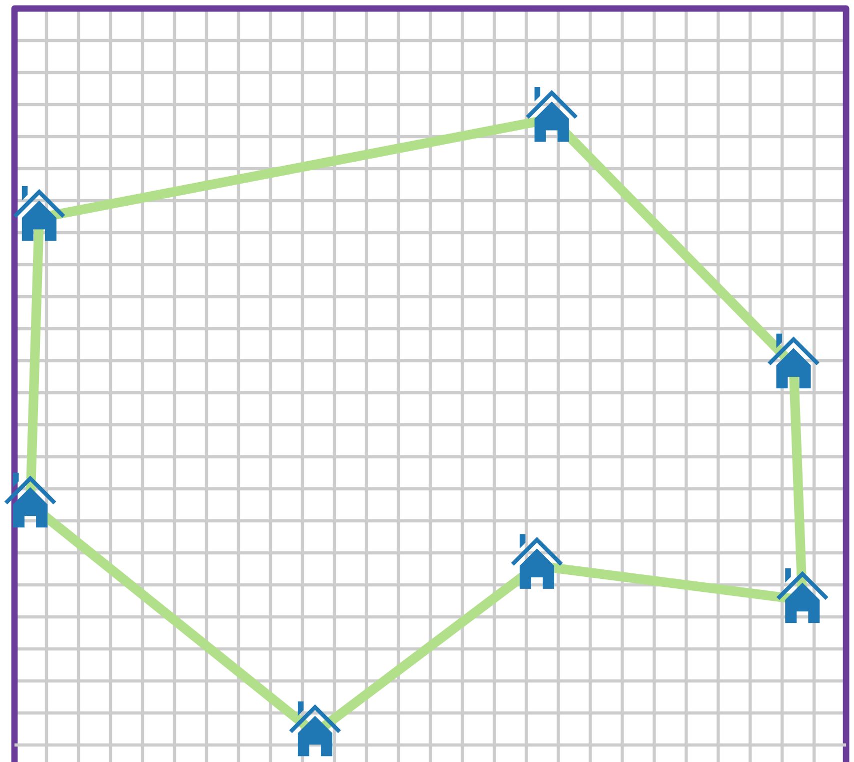
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Let's assume that the salesman flies $\Rightarrow$ Euclidean distances.



Simplifying Assumptions

- Houses inside $(L \times L)$-square
- $L := 4n^2 = 2^k$; $k = 2 + 2\log_2 n$
- integer coordinates

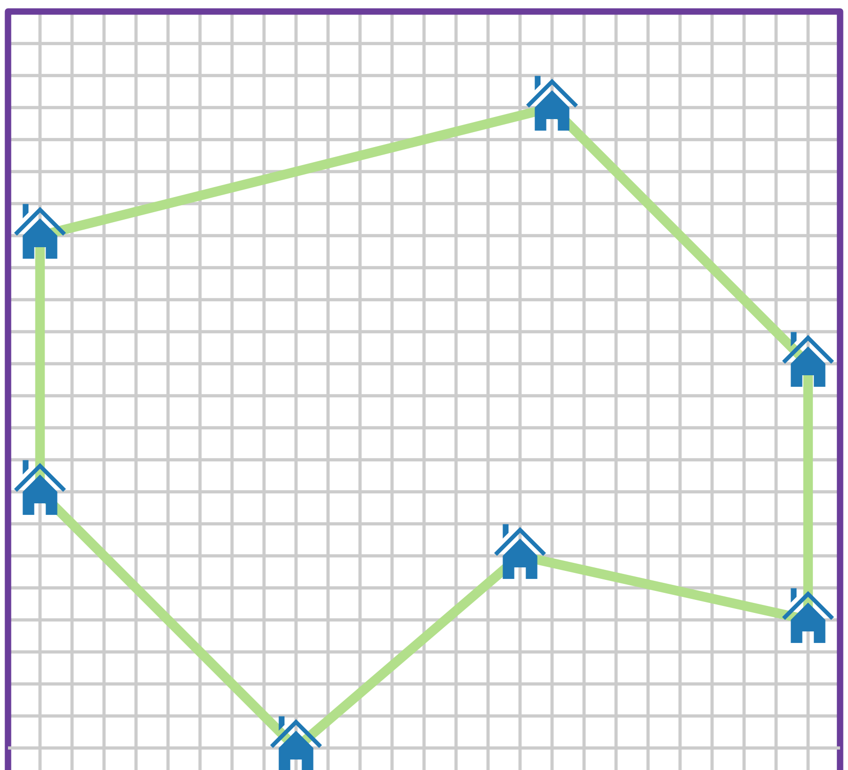
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Let's assume that the salesman flies $\Rightarrow$ Euclidean distances.



Simplifying Assumptions

- Houses inside $(L \times L)$-square
- $L := 4n^2 = 2^k$; $k = 2 + 2\log_2 n$
- integer coordinates

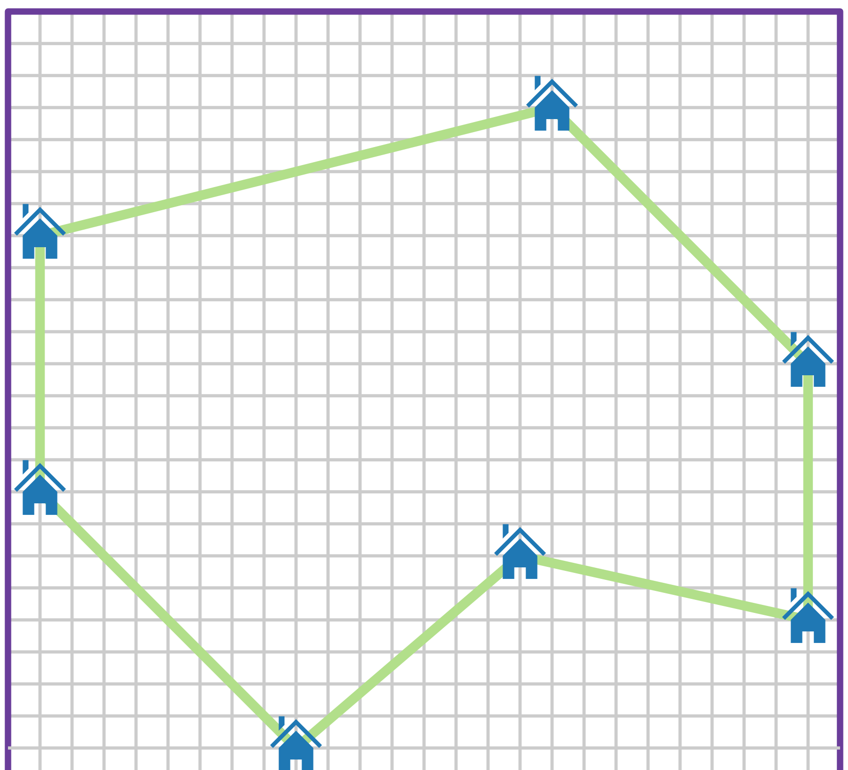
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Let's assume that the salesman flies $\Rightarrow$ Euclidean distances.



Simplifying Assumptions

- Houses inside $(L \times L)$-square
- $L := 4n^2 = 2^k$; $k = 2 + 2\log_2 n$
- integer coordinates

("justification": homework)

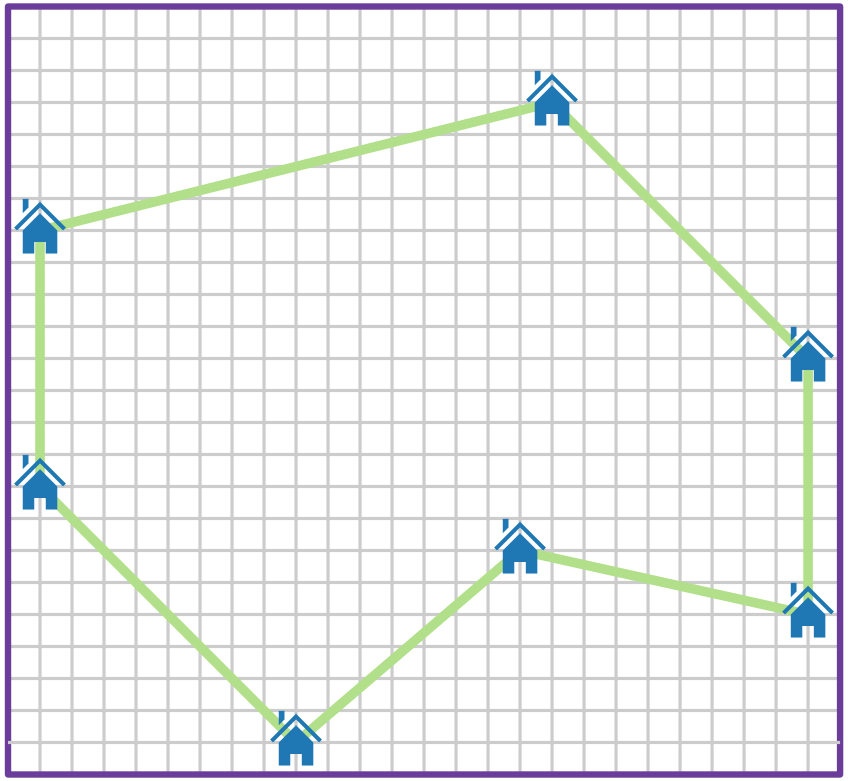
Traveling Salesman Problem (TSP)

Question: What's the fastest way to deliver all parcels to their destination?

Given: A set of n houses (points) in $\mathbb{R}^2$.

Task: Find a tour (Hamiltonian cycle) of min. length.

Let's assume that the salesman flies $\Rightarrow$ Euclidean distances.



Simplifying Assumptions

- Houses inside $(L \times L)$-square
- $L := 4n^2 = 2^k$; $k = 2 + 2\log_2 n$
- integer coordinates

("justification": homework)

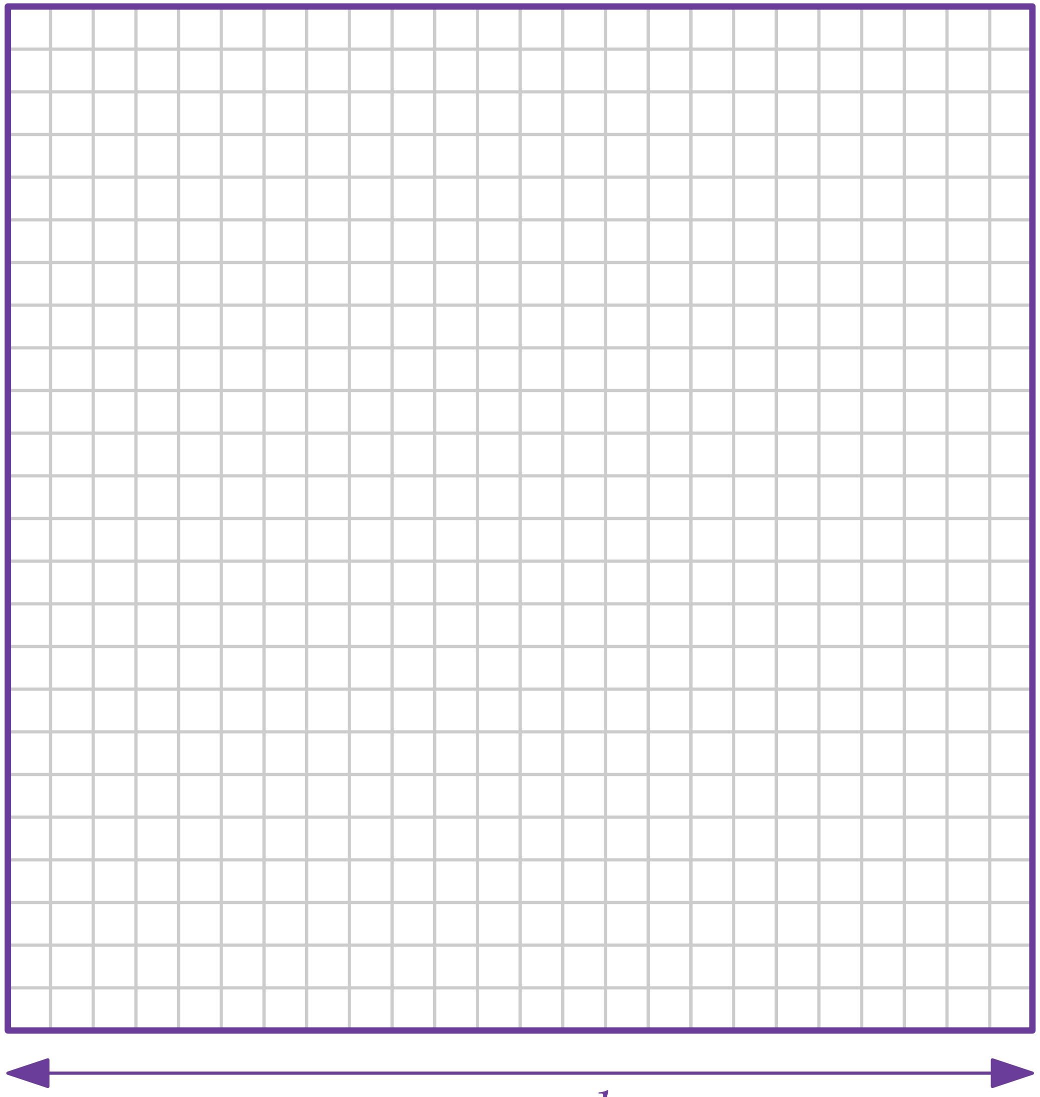
Goal: $(1+\varepsilon)$-approximation!

Approximation Algorithms

Lecture 9:
A PTAS for Euclidean TSP

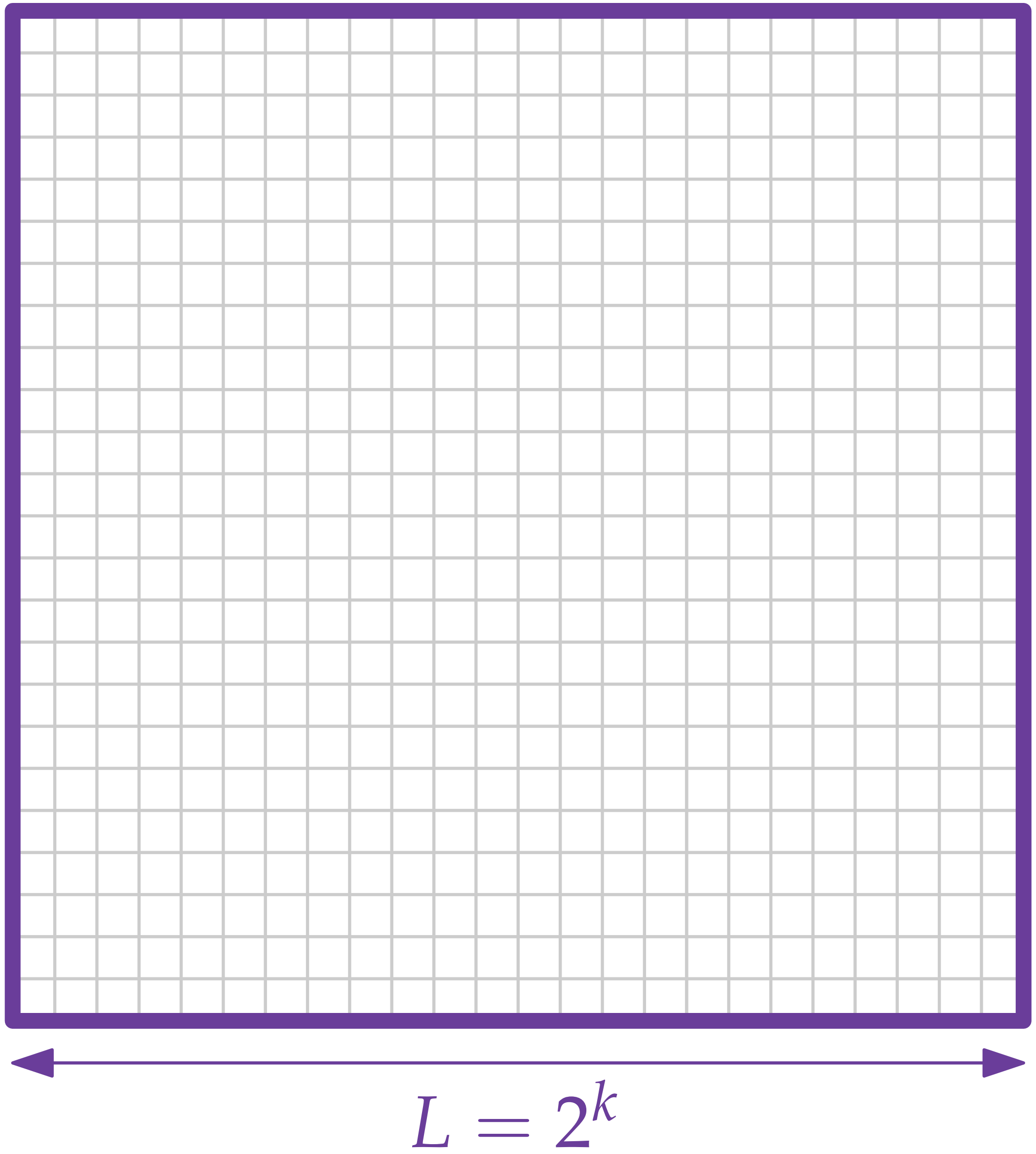
Part II:
Dissection

Basic Dissection



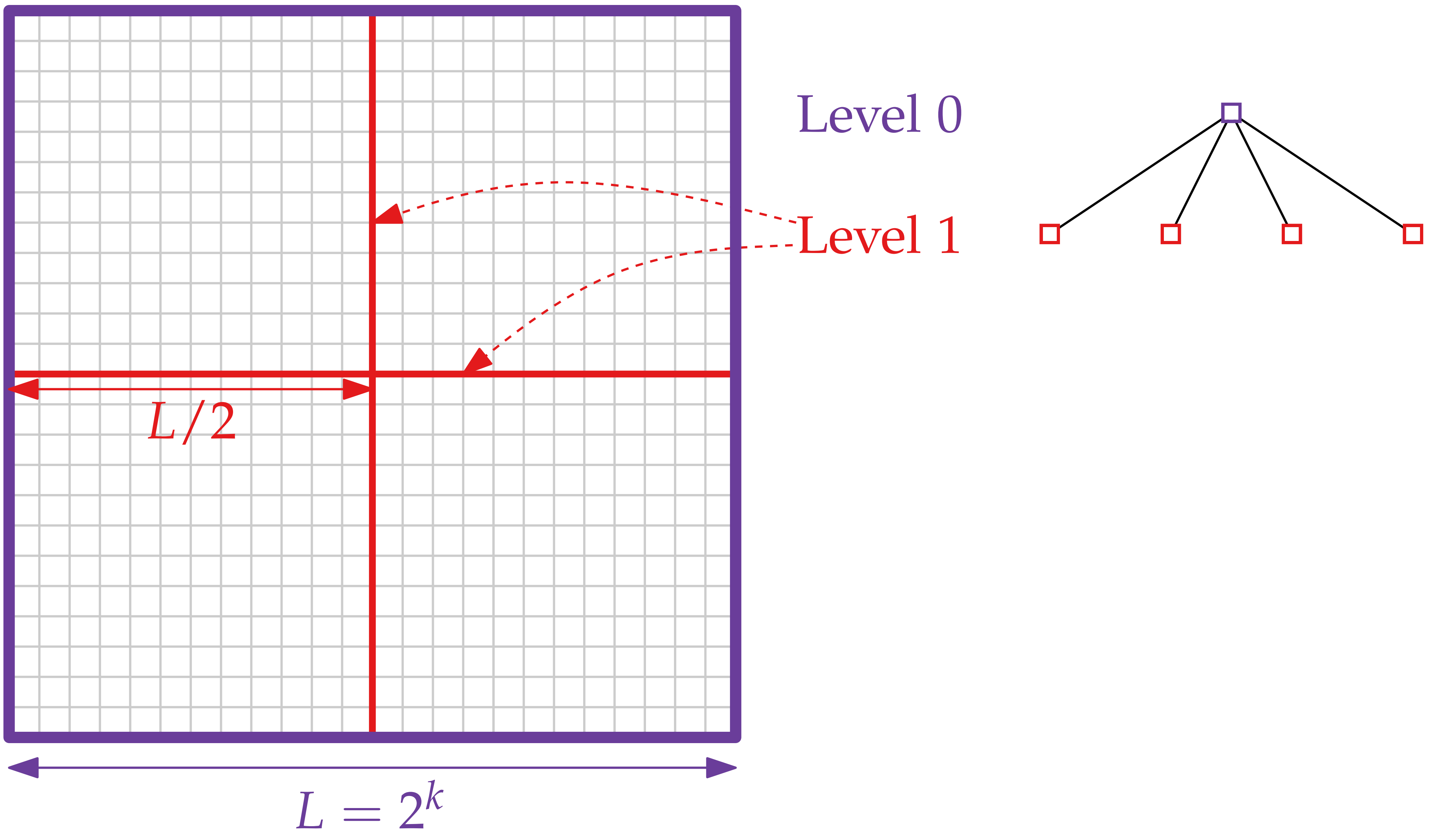
$L = 2^k$

Basic Dissection

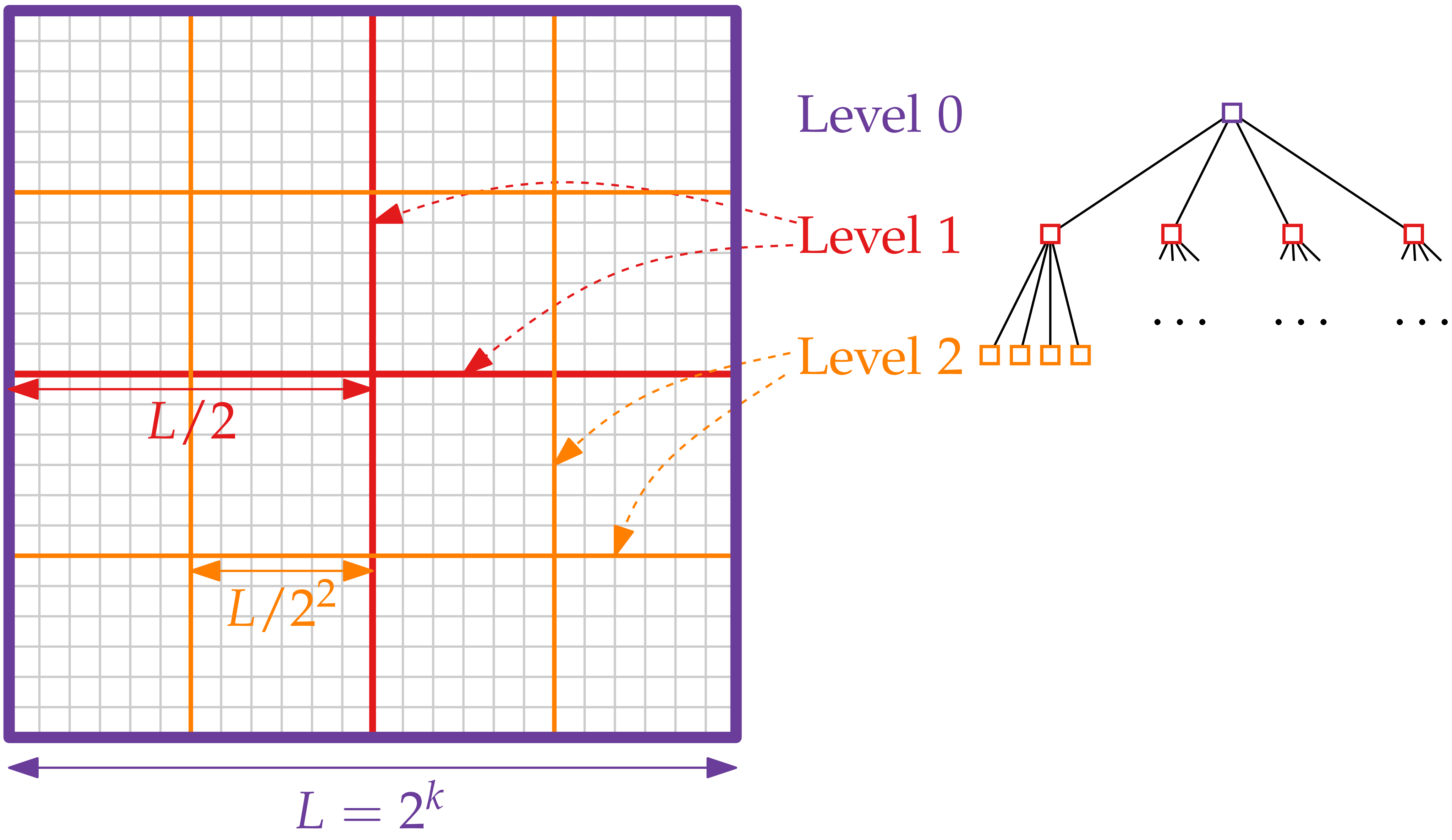


Level 0 ▫

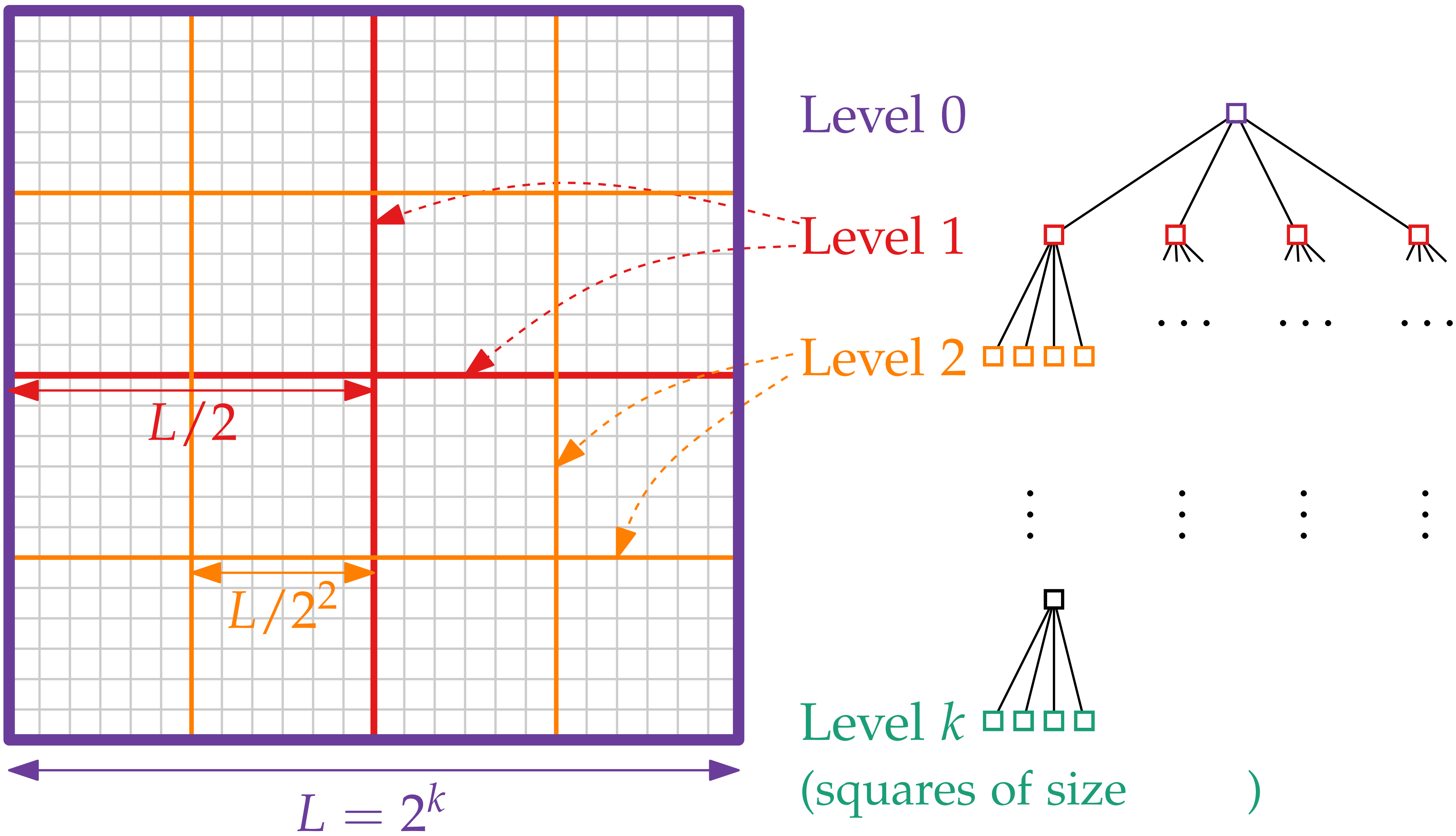
Basic Dissection



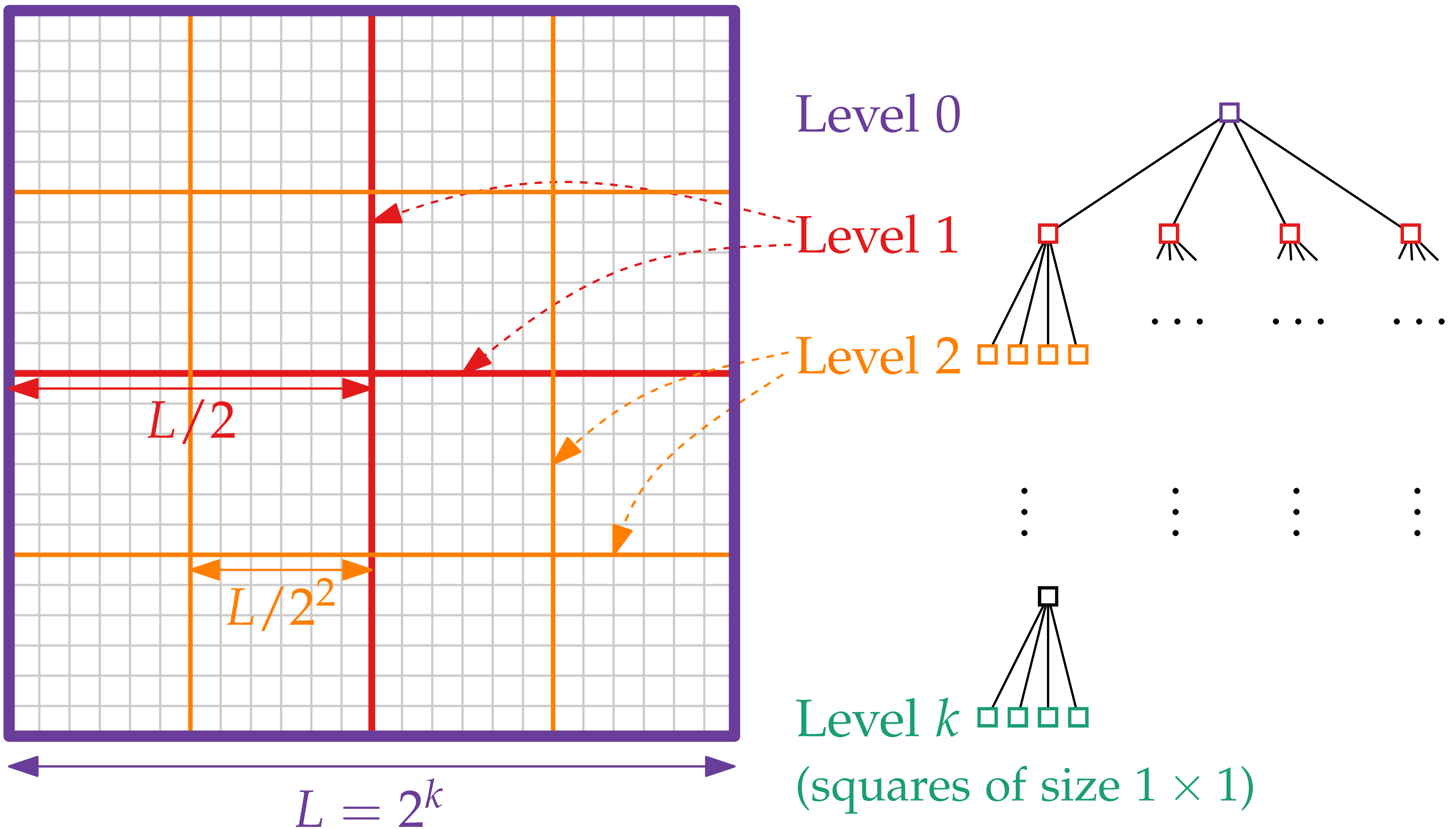
Basic Dissection



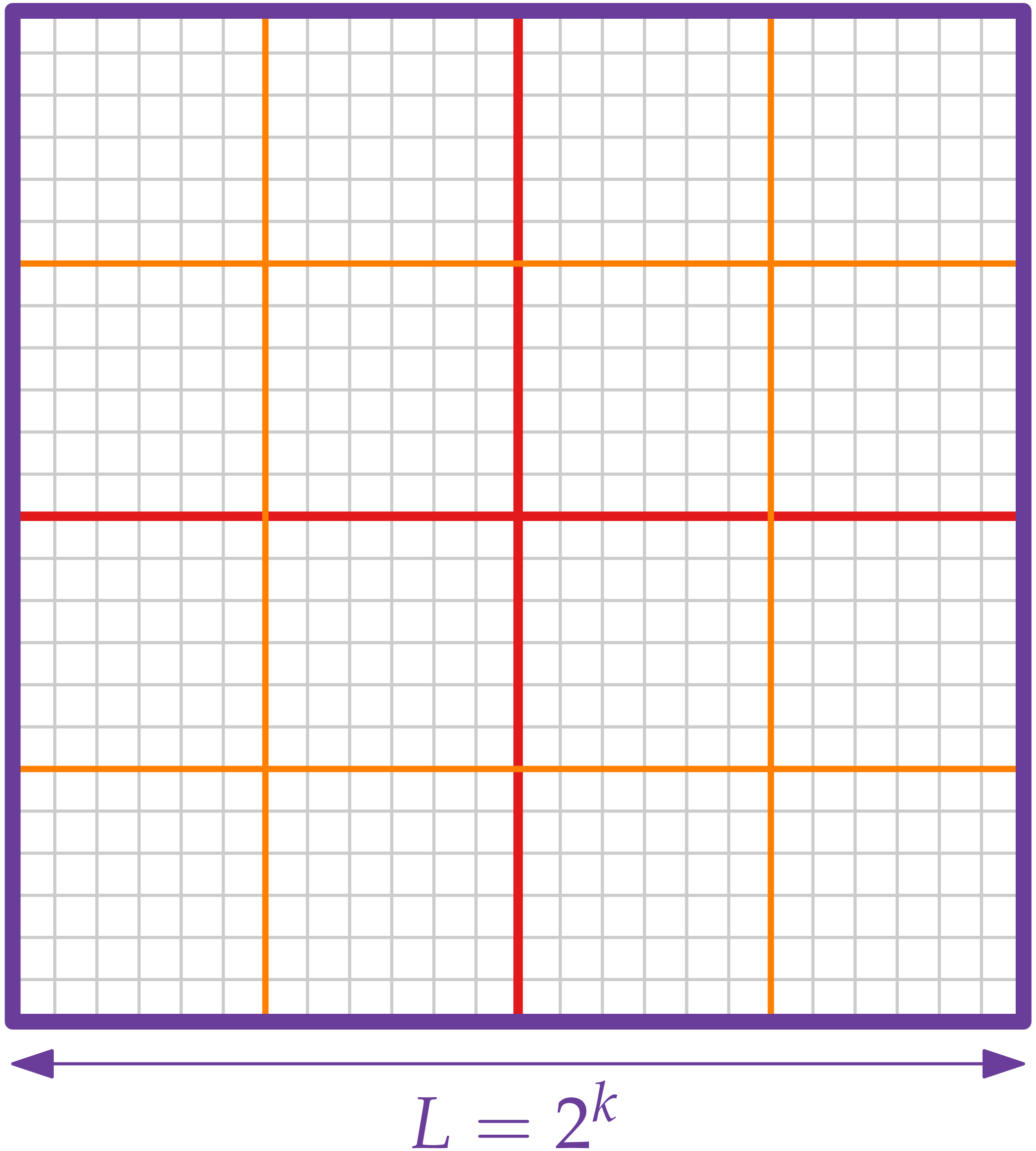
Basic Dissection



Basic Dissection

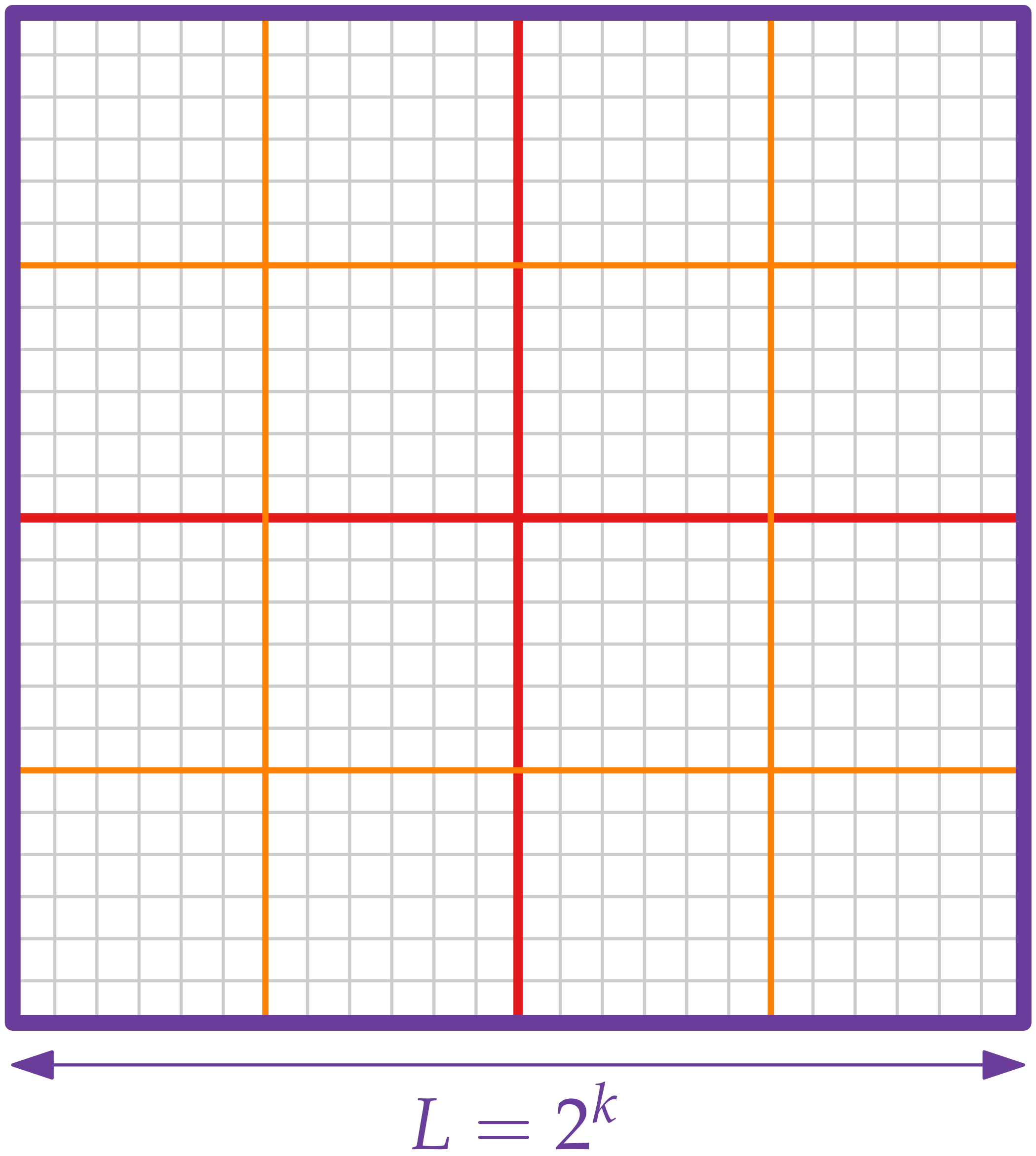


Portals



- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

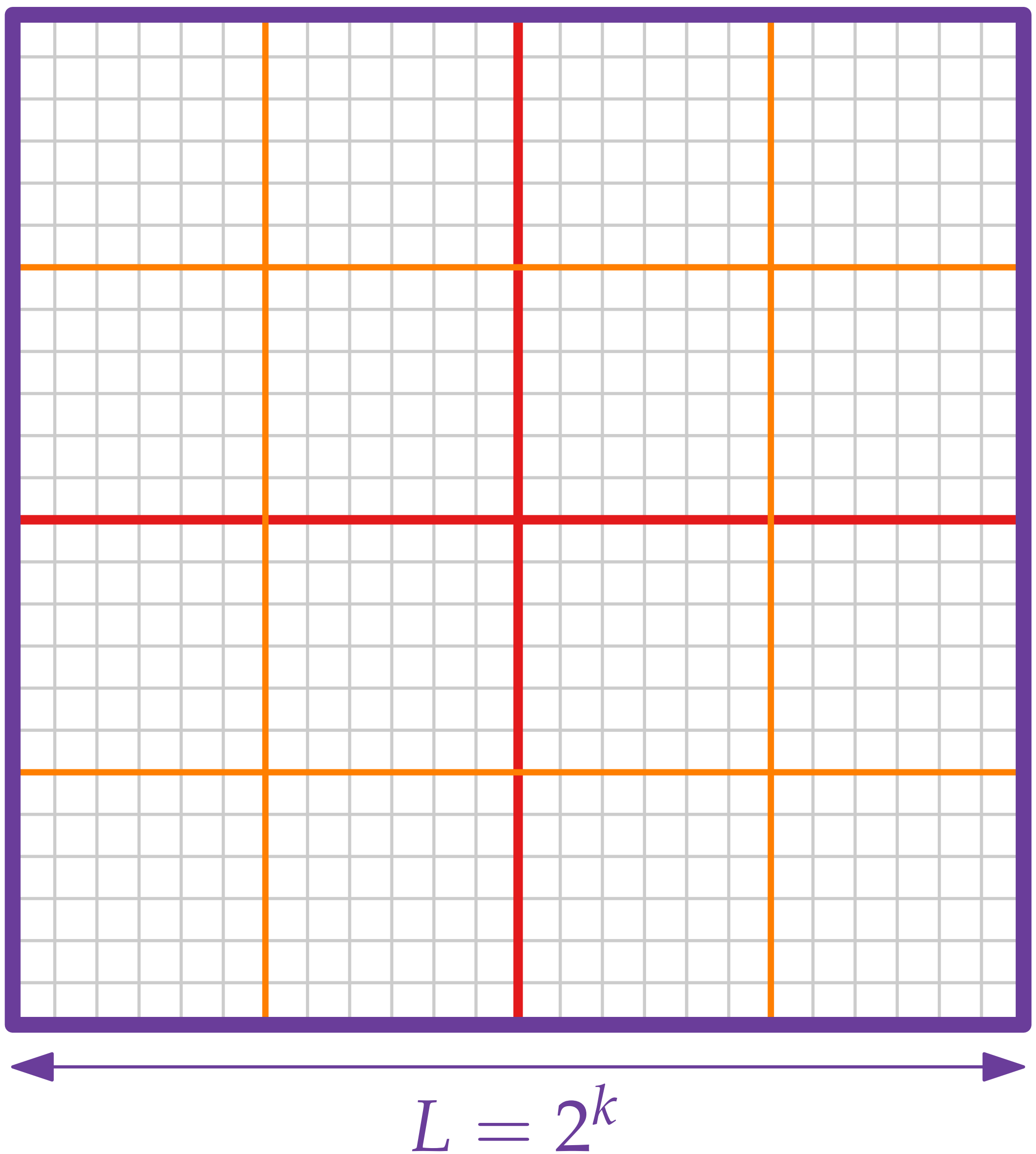
Portals



- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

  Recall that $k = 2 + 2\log_2 n$.

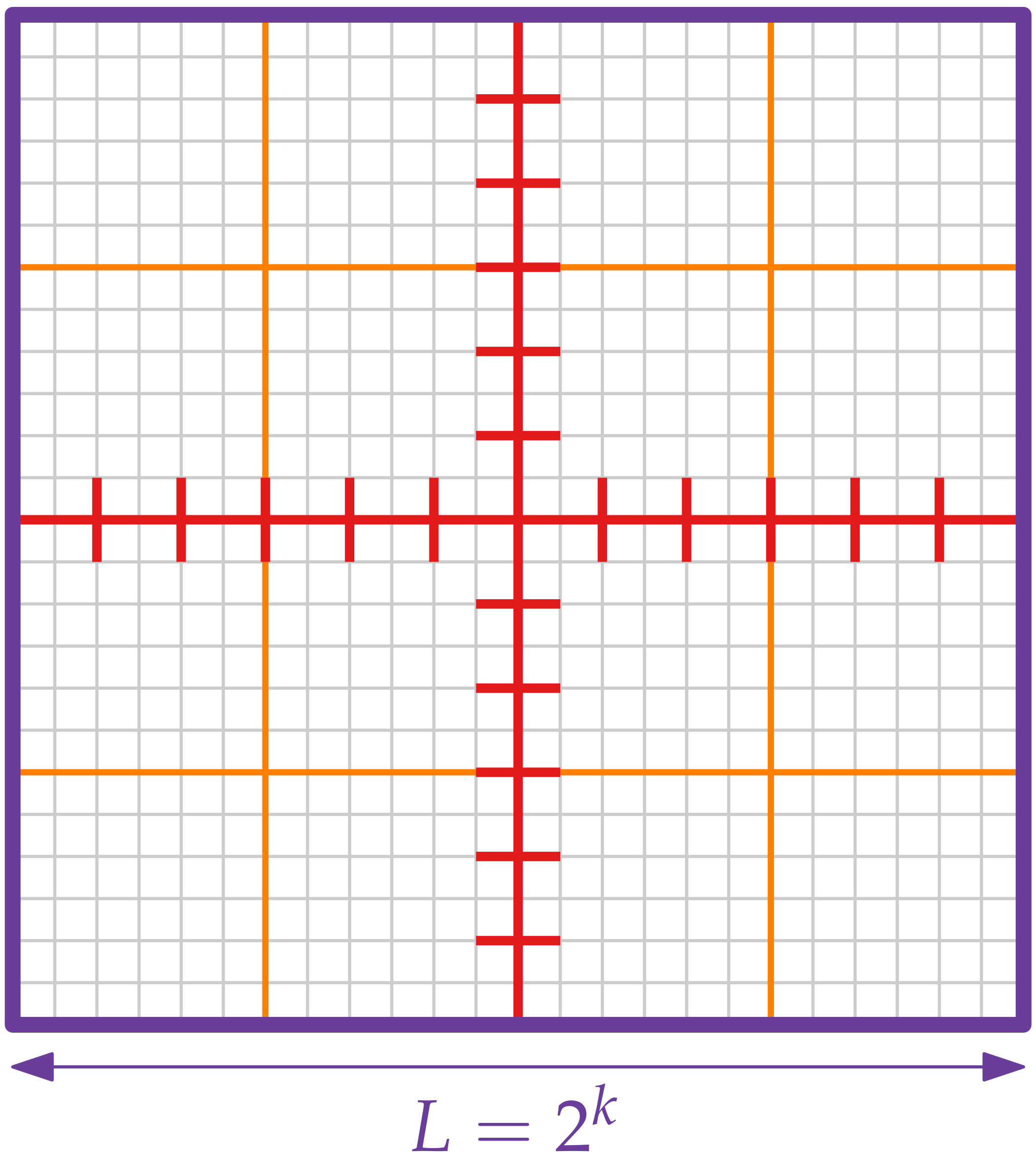
Portals



- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

  Recall that $k = 2 + 2\log_2 n$.
  $\Rightarrow m \in O((\log n)/\varepsilon)$

Portals

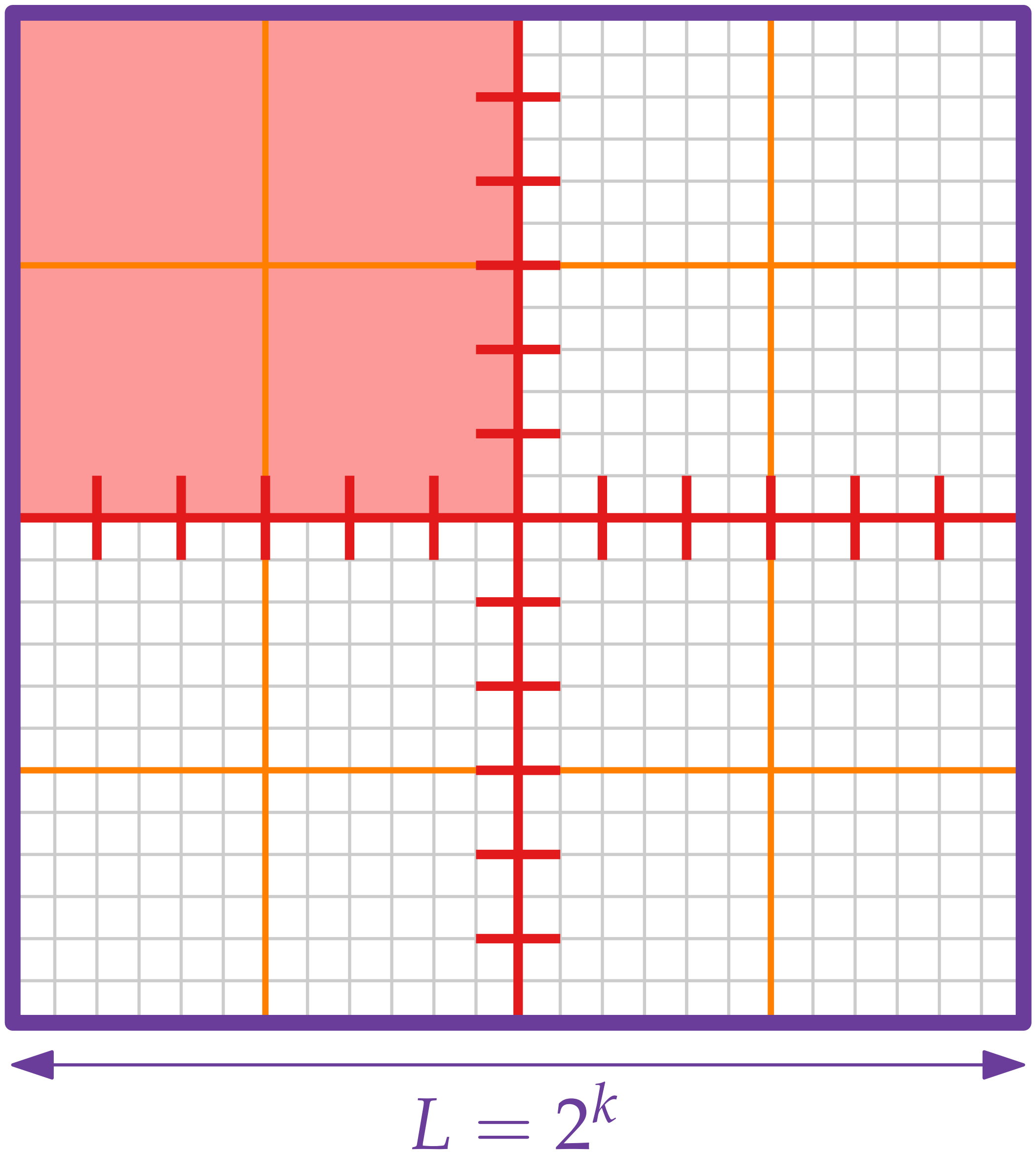


- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

  Recall that $k = 2 + 2\log_2 n$.
  $\Rightarrow m \in O((\log n)/\varepsilon)$

- **Portals** on level-i line are at a distance of $L/(2^i m)$.

Portals



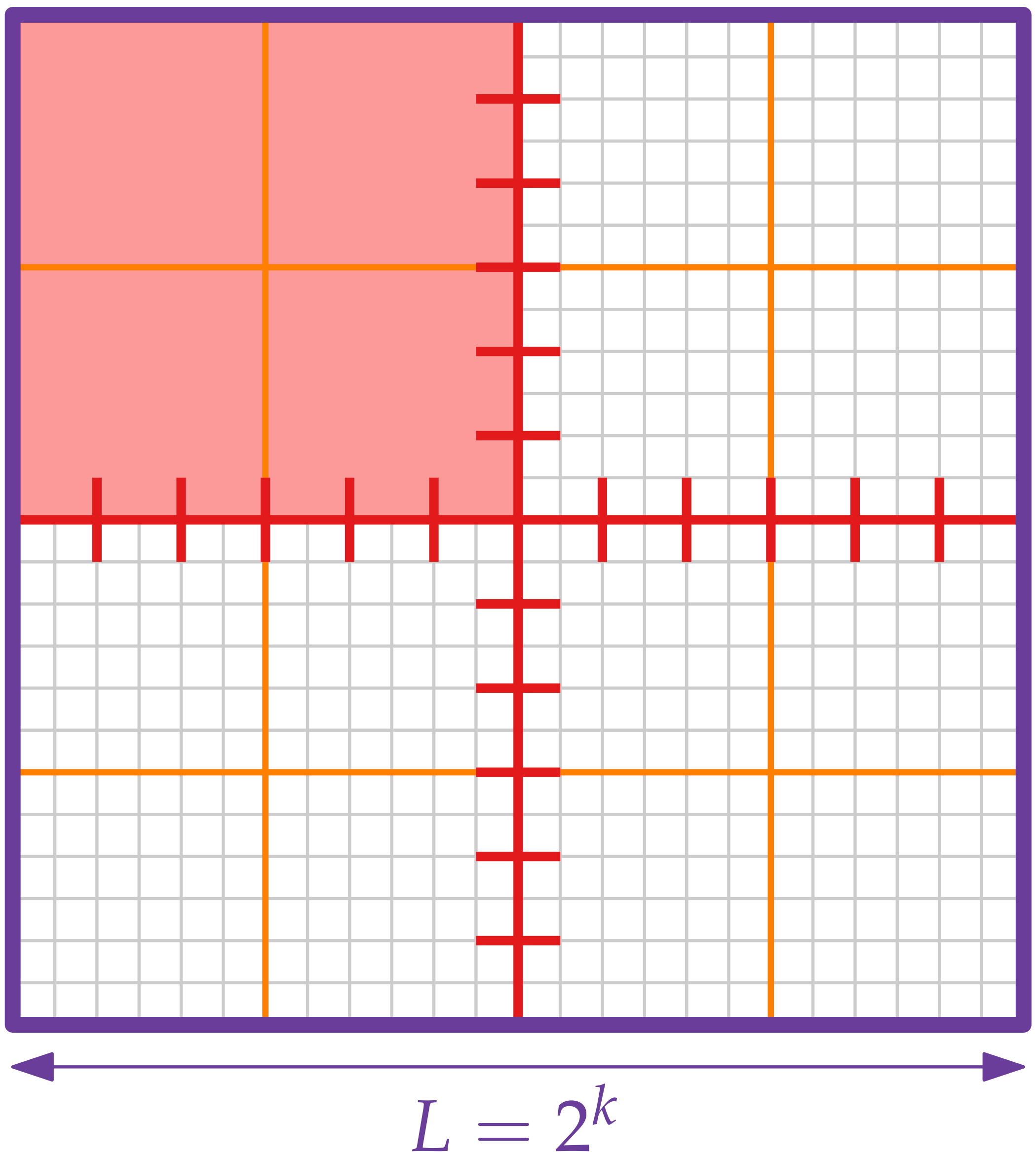
- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

  Recall that $k = 2 + 2\log_2 n$.
  $\Rightarrow m \in O((\log n)/\varepsilon)$

- **Portals** on level-i line are at a distance of $L/(2^i m)$.

- Every level-i square has size

Portals



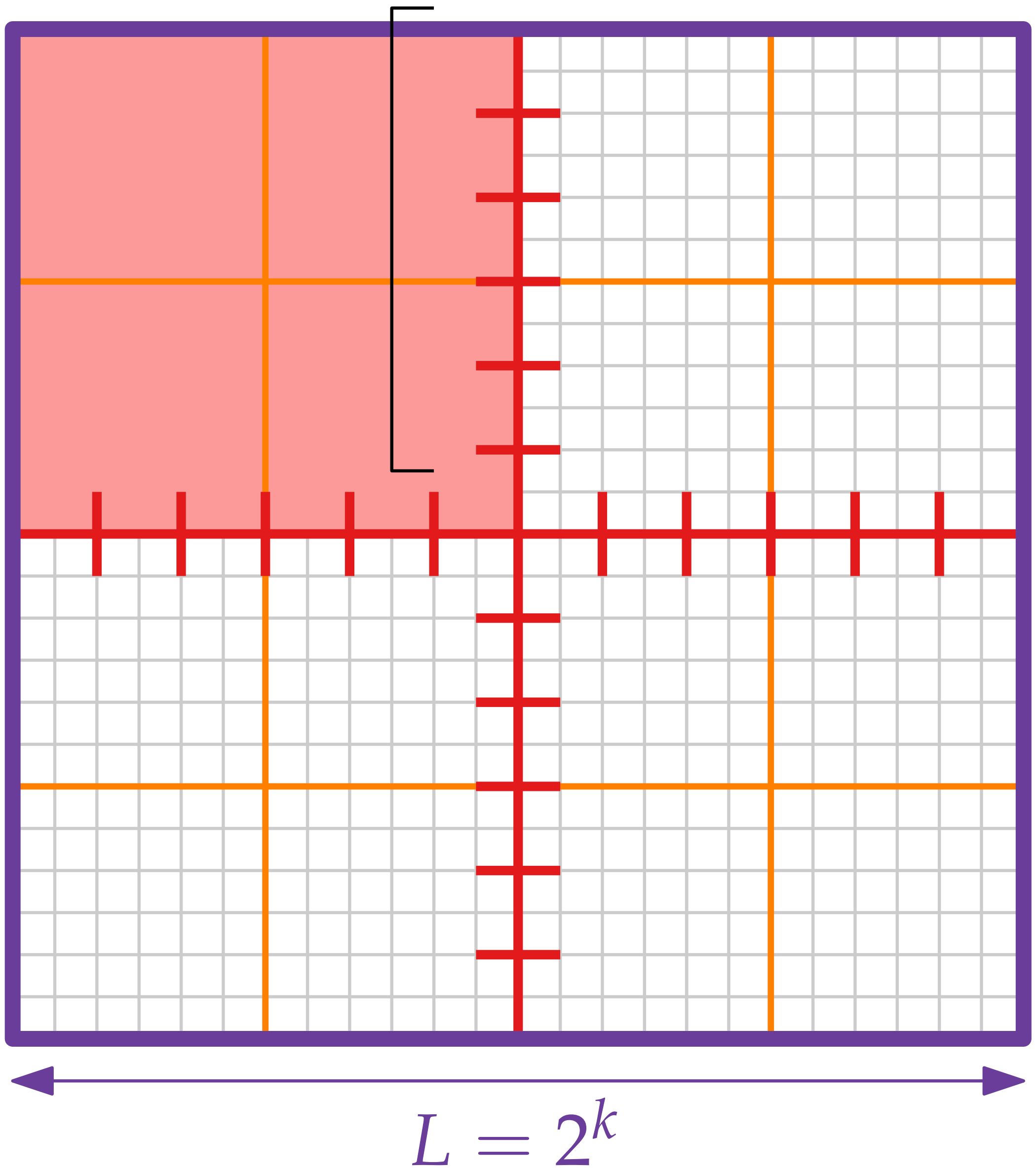
- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

  Recall that $k = 2 + 2\log_2 n$.
  $\Rightarrow m \in O((\log n)/\varepsilon)$

- **Portals** on level-i line are at a distance of $L/(2^i m)$.

- Every level-i square has size $L/2^i \times L/2^i$.

Portals



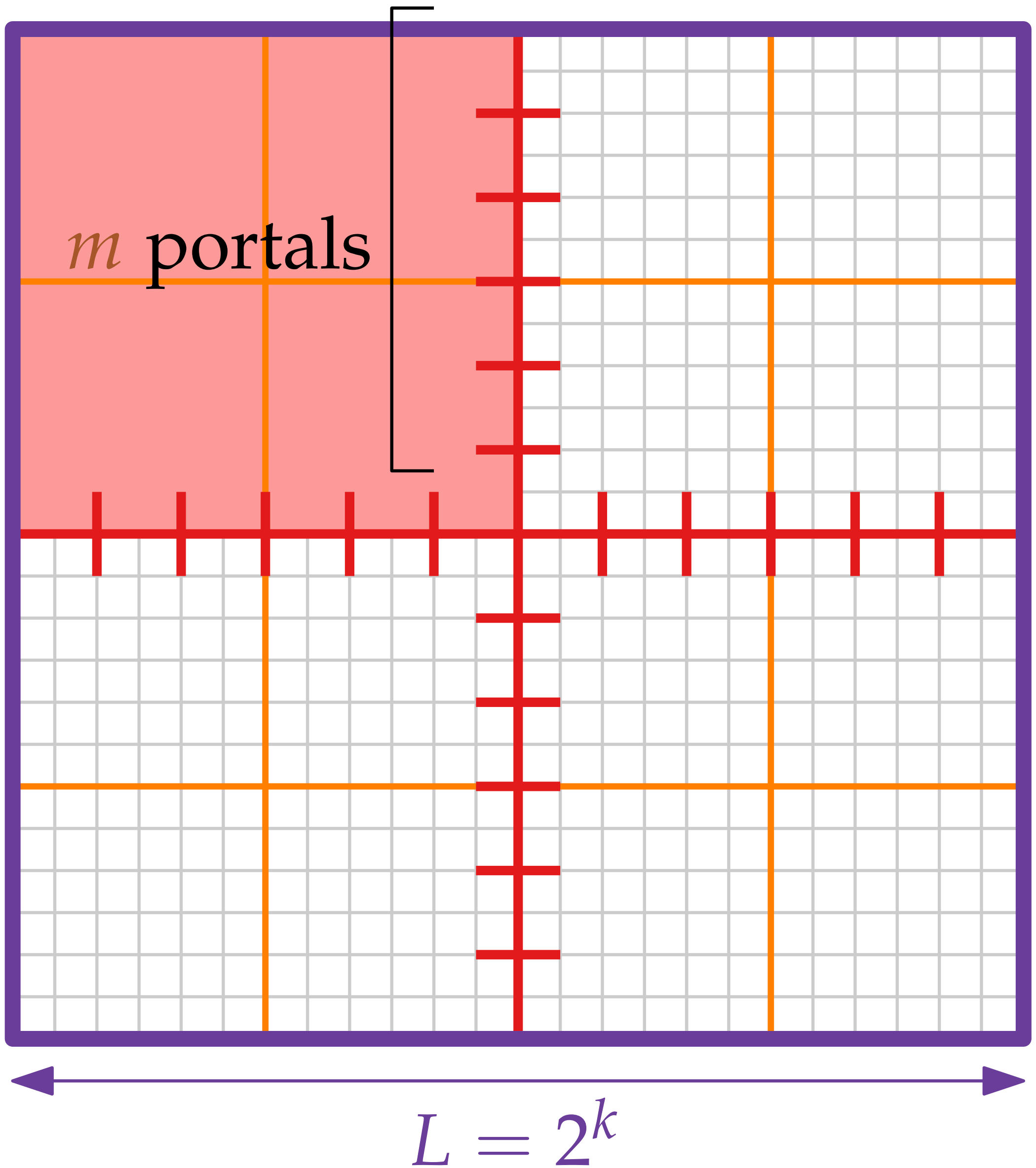
- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

  Recall that $k = 2 + 2\log_2 n$.
  $\Rightarrow m \in O((\log n)/\varepsilon)$

- **Portals** on level-i line are at a distance of $L/(2^i m)$.

- Every level-i square has size $L/2^i \times L/2^i$.

Portals



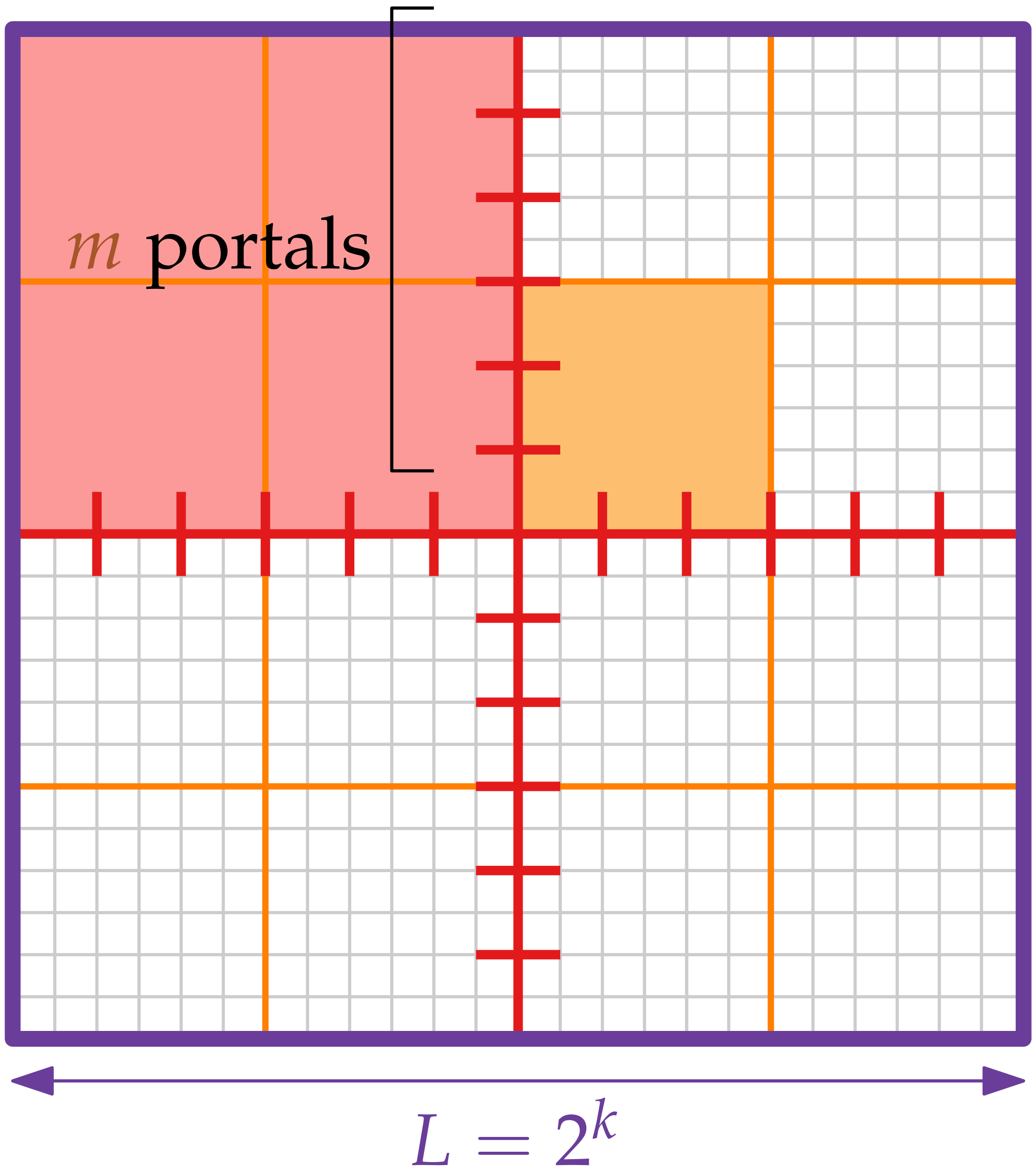
- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

  Recall that $k = 2 + 2\log_2 n$.
  $\Rightarrow m \in O((\log n)/\varepsilon)$

- **Portals** on level-i line are at a distance of $L/(2^i m)$.

- Every level-i square has size $L/2^i \times L/2^i$.

Portals



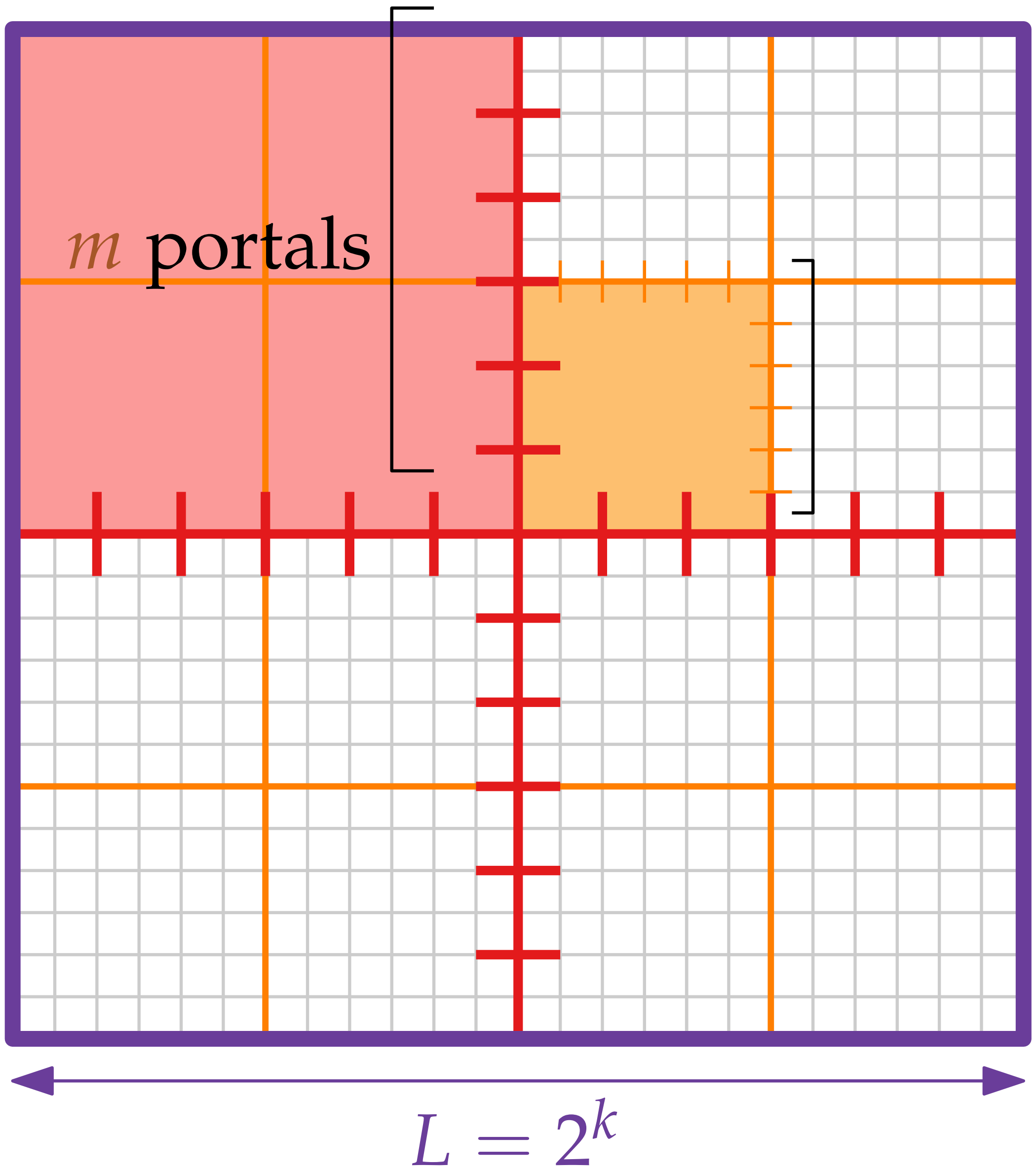
- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

  Recall that $k = 2 + 2\log_2 n$.
  $\Rightarrow m \in O((\log n)/\varepsilon)$

- **Portals** on level-i line are at a distance of $L/(2^i m)$.

- Every level-i square has size $L/2^i \times L/2^i$.

Portals



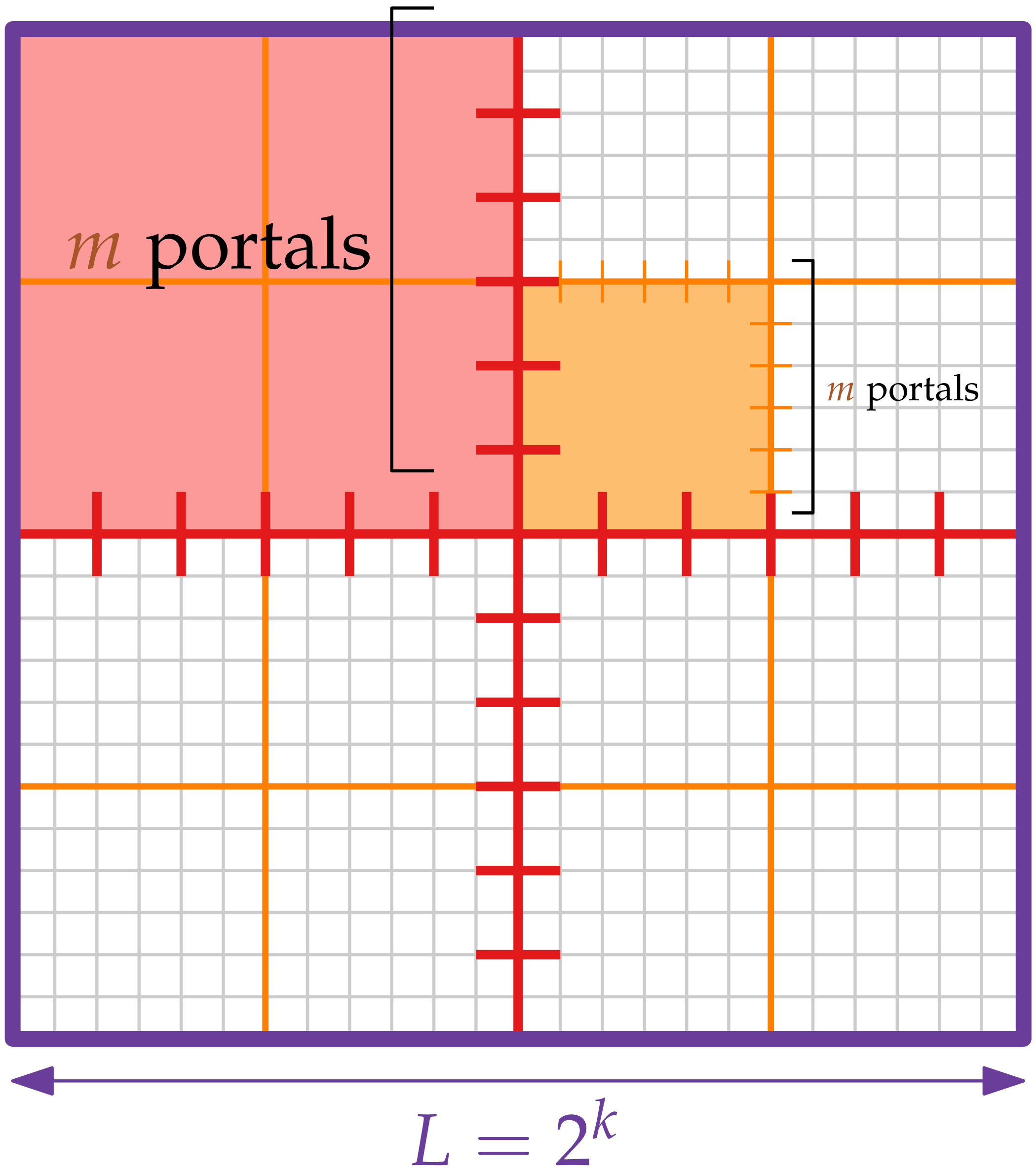
- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

  Recall that $k = 2 + 2\log_2 n$.
  $\Rightarrow m \in O((\log n)/\varepsilon)$

- **Portals** on level-i line are at a distance of $L/(2^i m)$.

- Every level-i square has size $L/2^i \times L/2^i$.

Portals



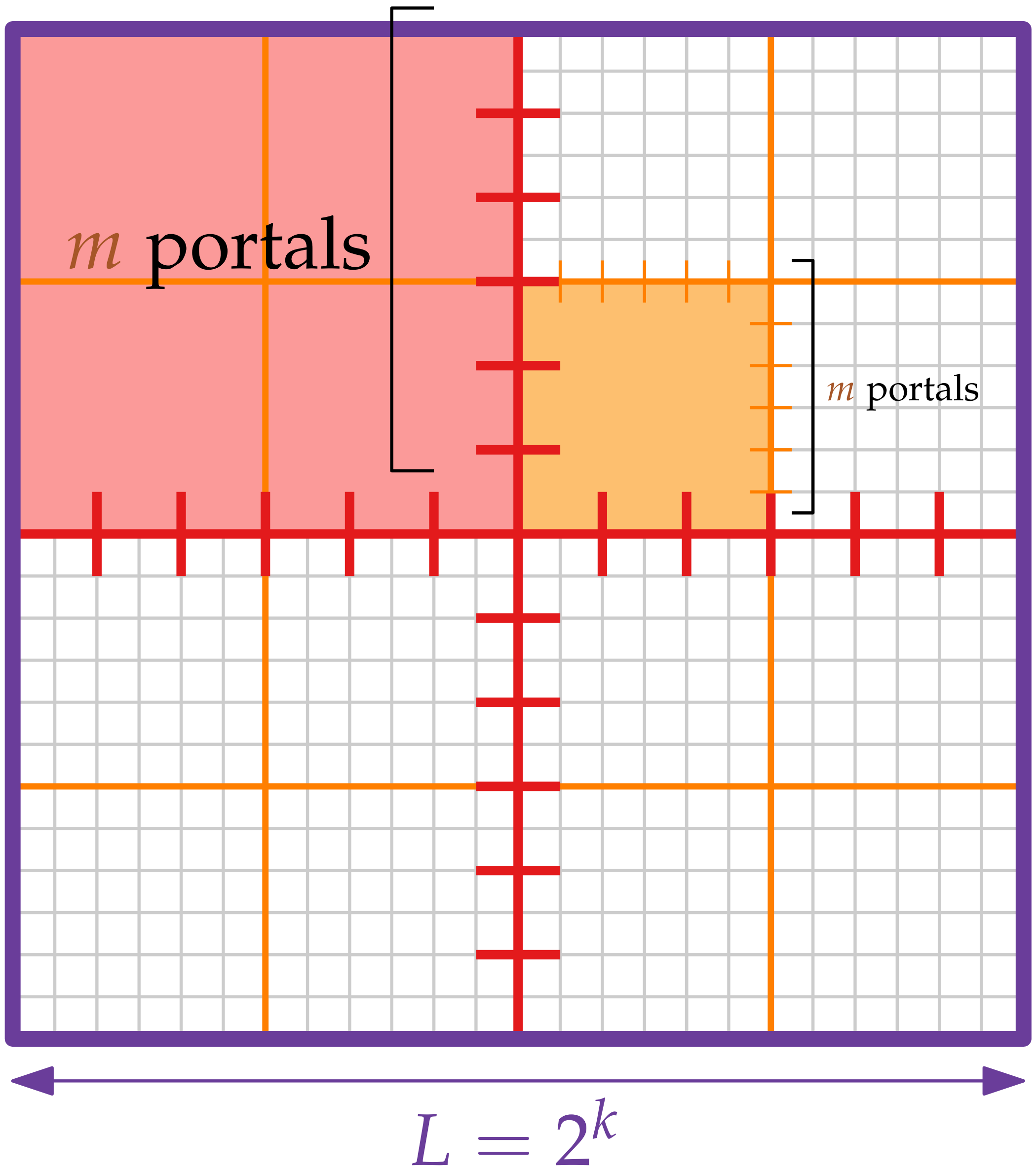
- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

  Recall that $k = 2 + 2\log_2 n$.
  $\Rightarrow m \in O((\log n)/\varepsilon)$

- **Portals** on level-i line are at a distance of $L/(2^i m)$.

- Every level-i square has size $L/2^i \times L/2^i$.

Portals



- Let m be a power of 2 in the interval $[k/\varepsilon, 2k/\varepsilon]$.

  Recall that $k = 2 + 2\log_2 n$.
  $\Rightarrow m \in O((\log n)/\varepsilon)$

- **Portals** on level-i line are at a distance of $L/(2^i m)$.

- Every level-i square has size $L/2^i \times L/2^i$.

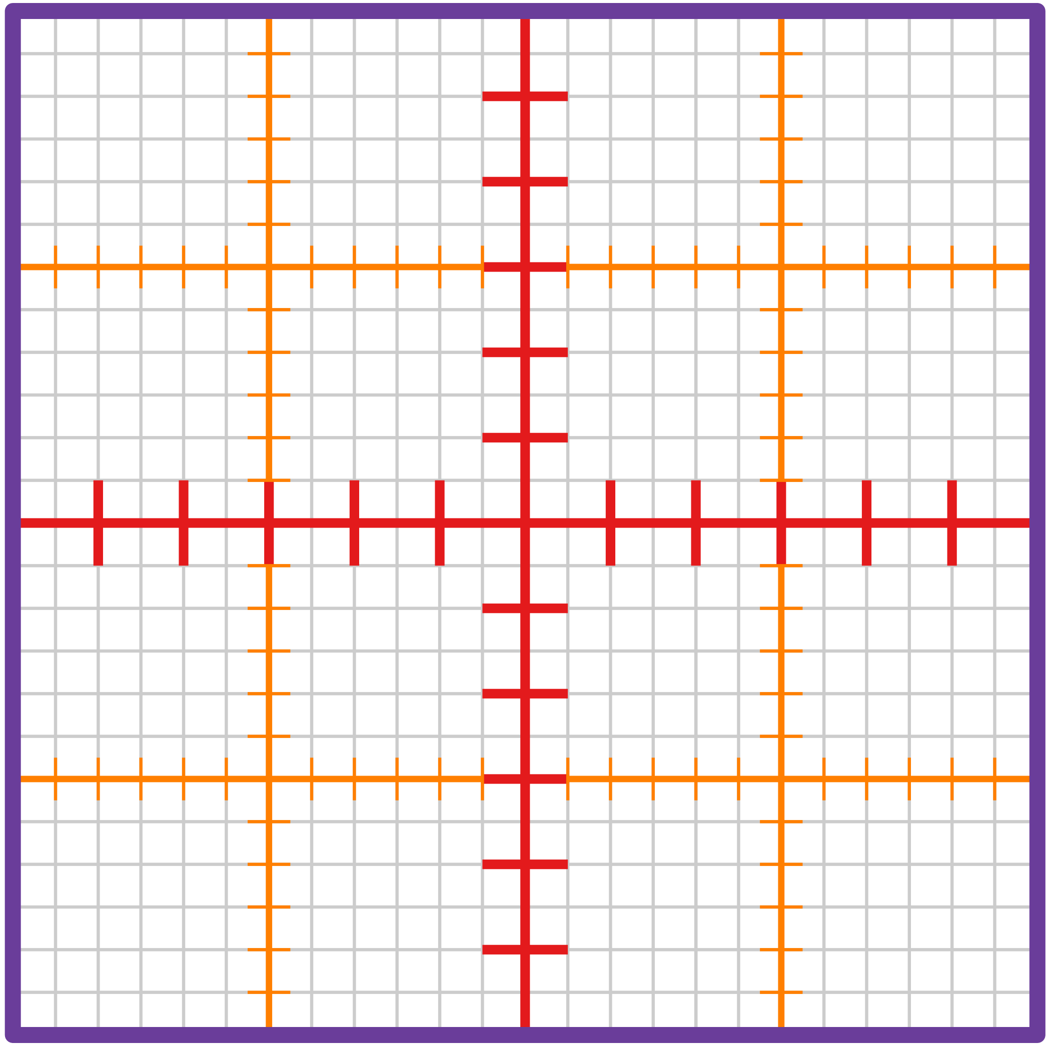
- A level-i square has $\leq 4m$ portals on its boundary.

Approximation Algorithms

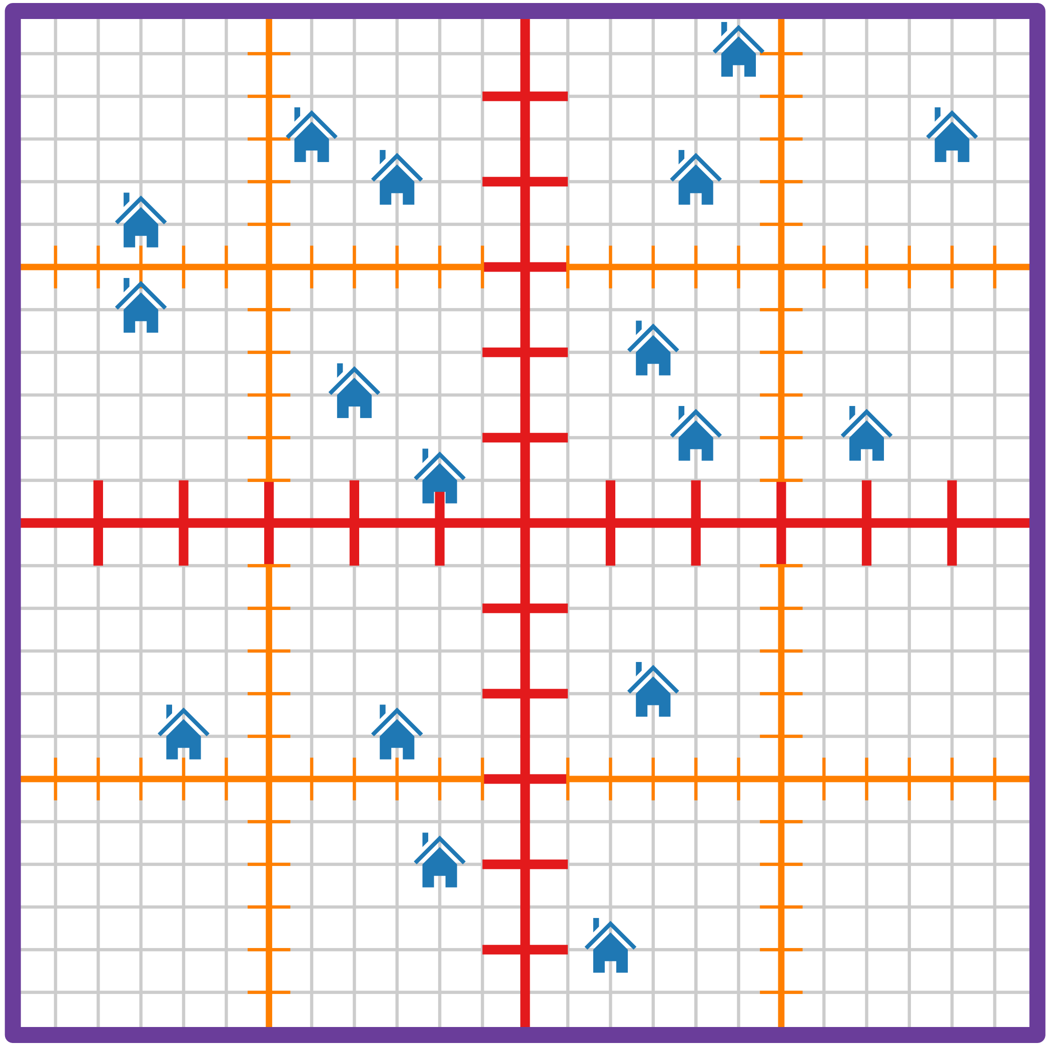
Lecture 9:
A PTAS for Euclidean TSP

Part III:
Well-Behaved Tours

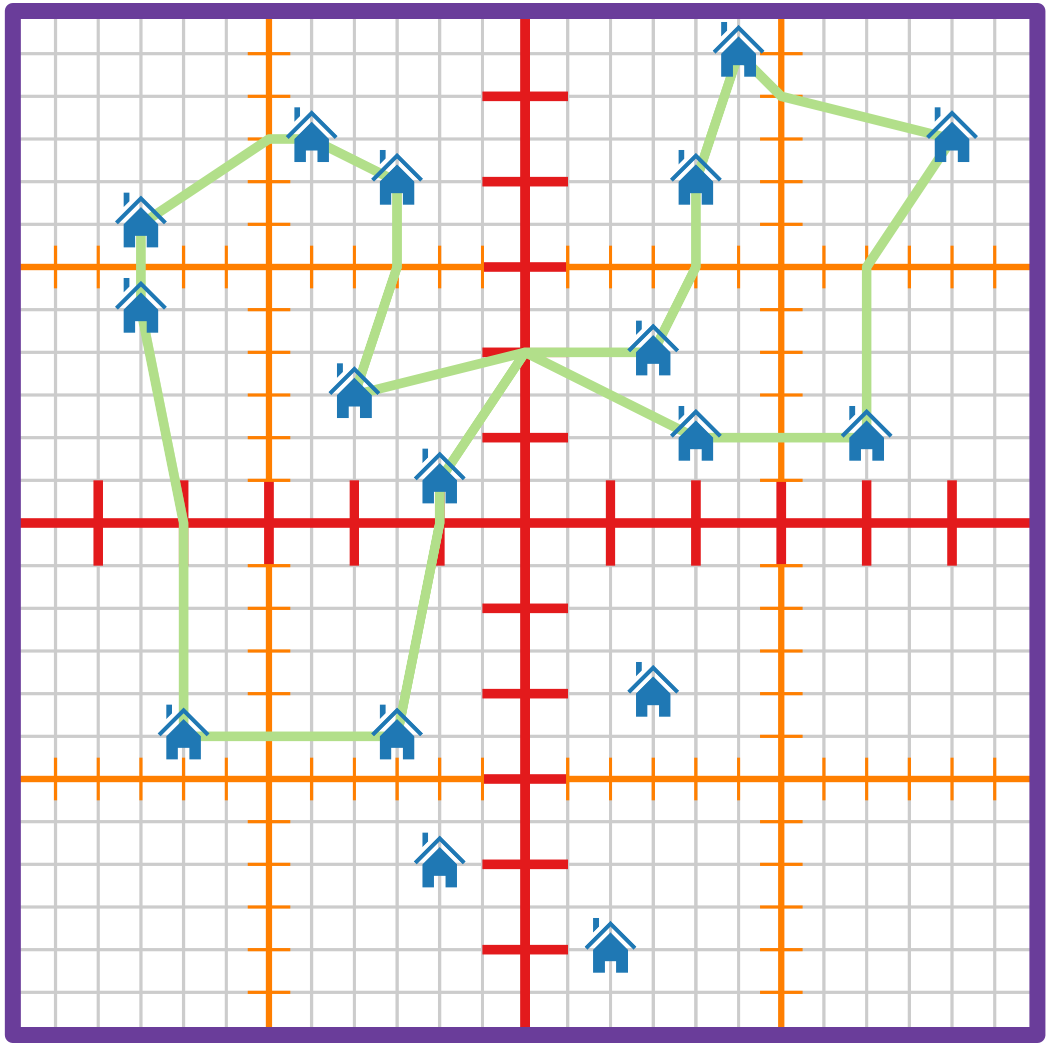
Well-Behaved Tours



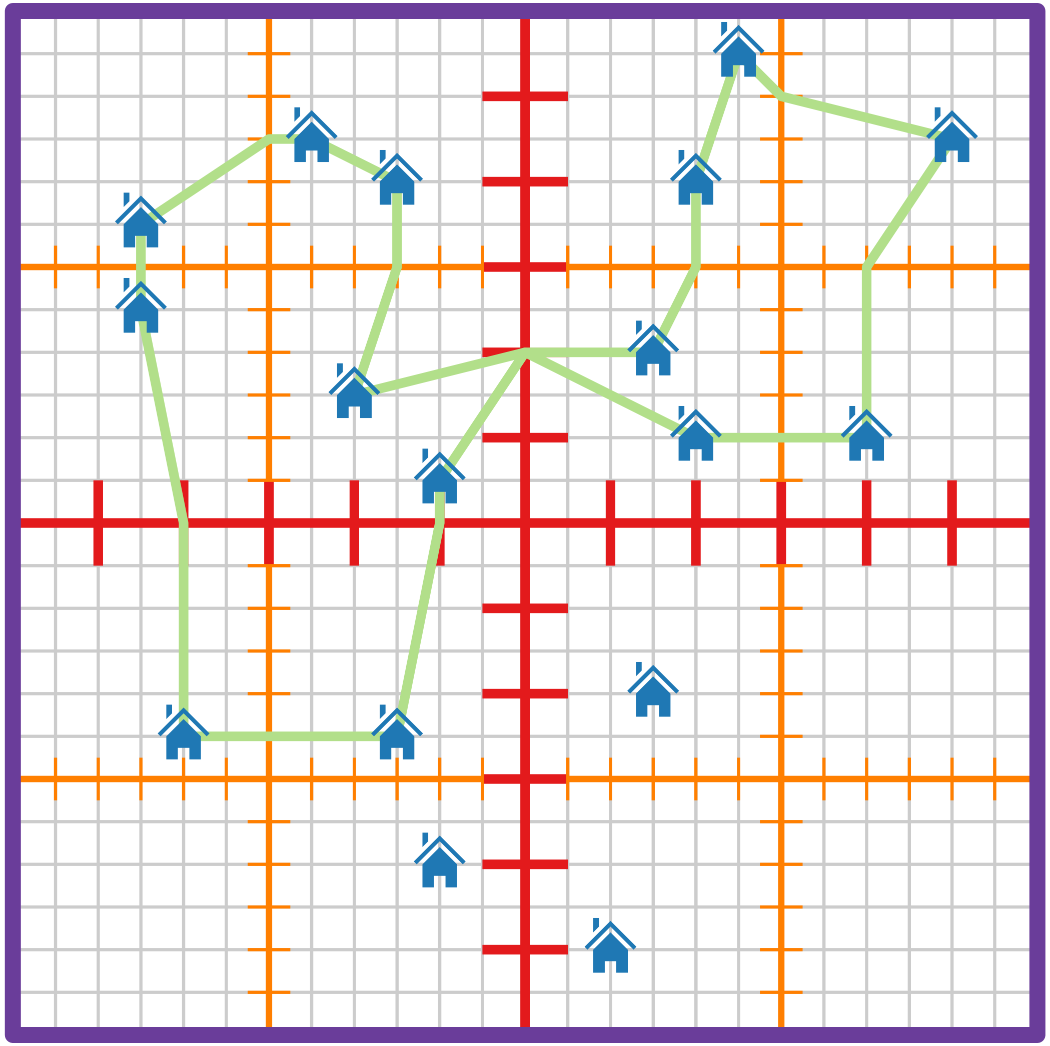
Well-Behaved Tours



Well-Behaved Tours

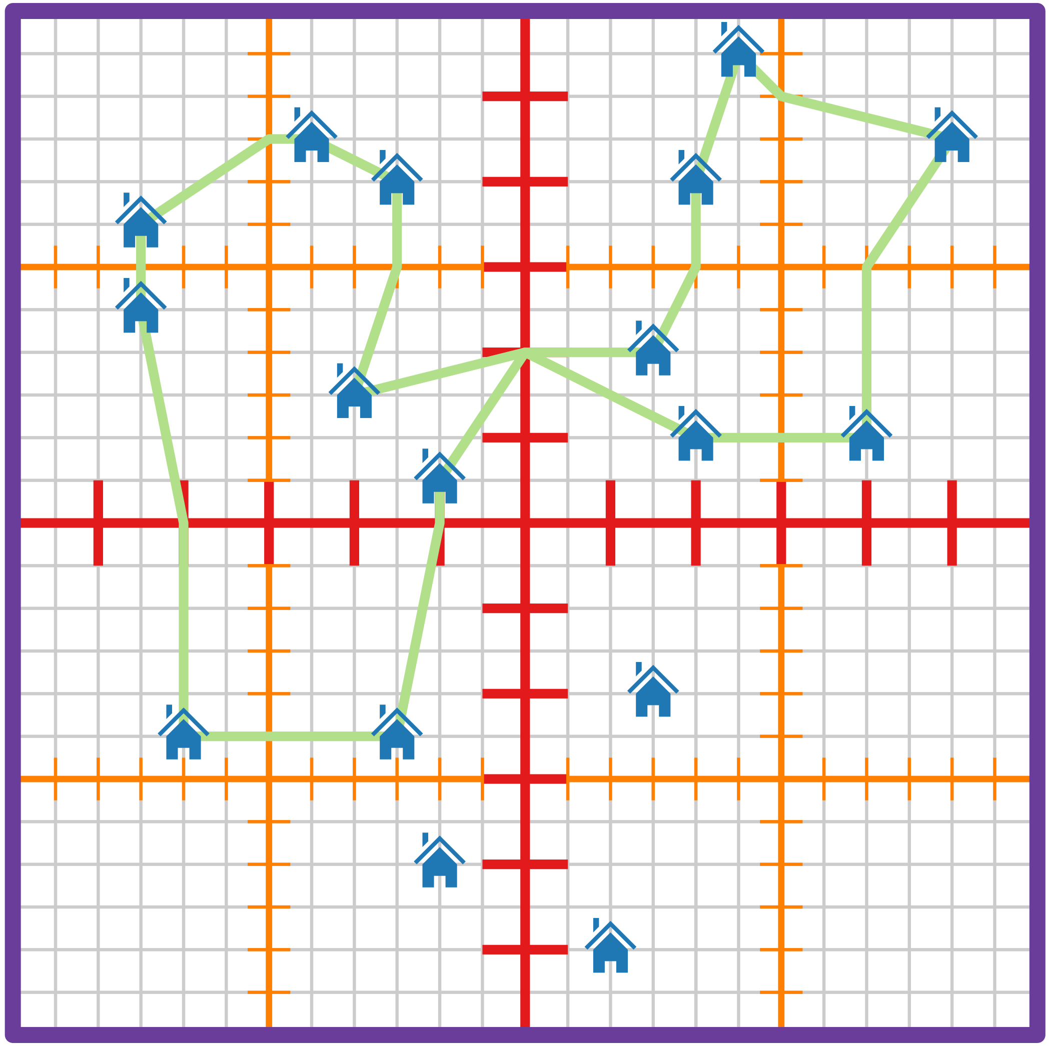


Well-Behaved Tours



A tour is *well-behaved* if

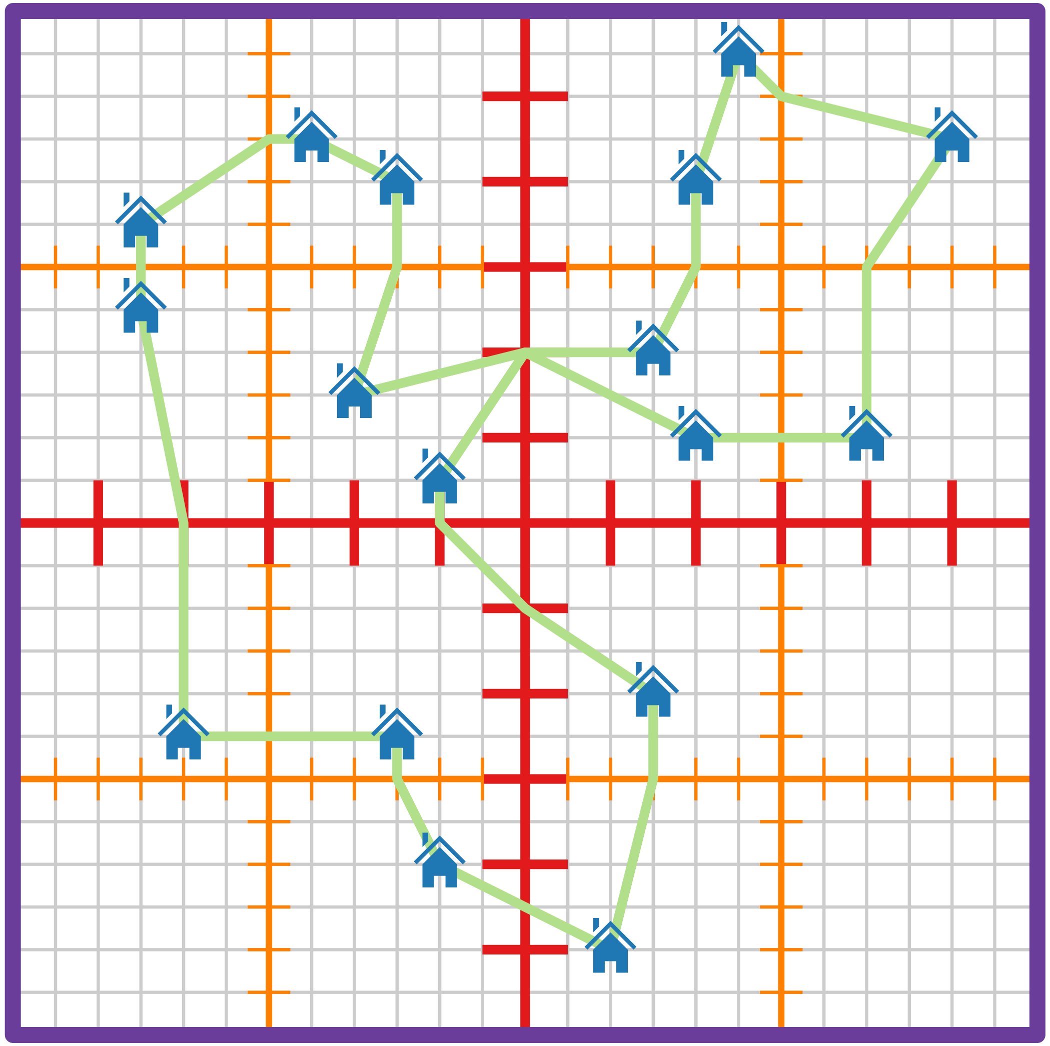
Well-Behaved Tours



A tour is *well-behaved* if

- it involves all houses

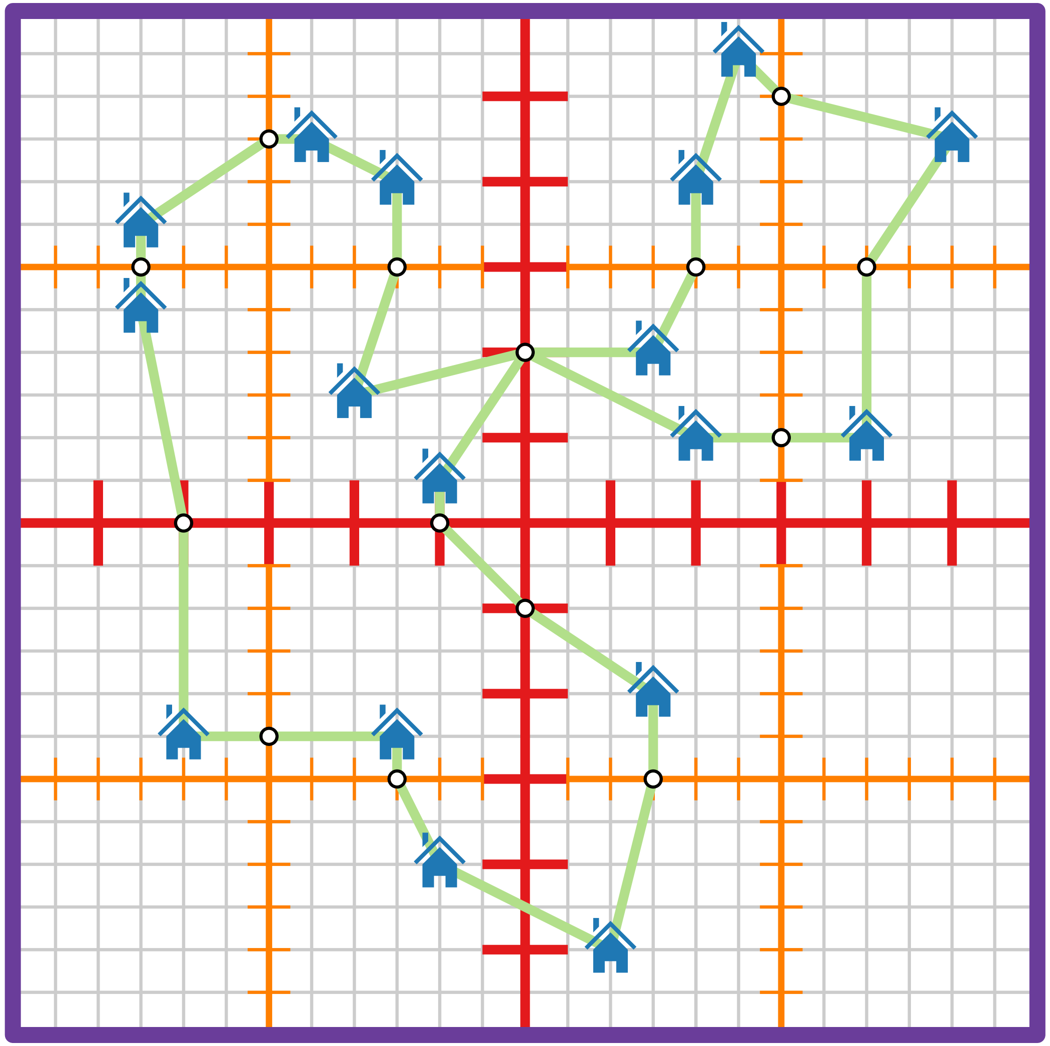
Well-Behaved Tours



A tour is *well-behaved* if

- it involves all houses

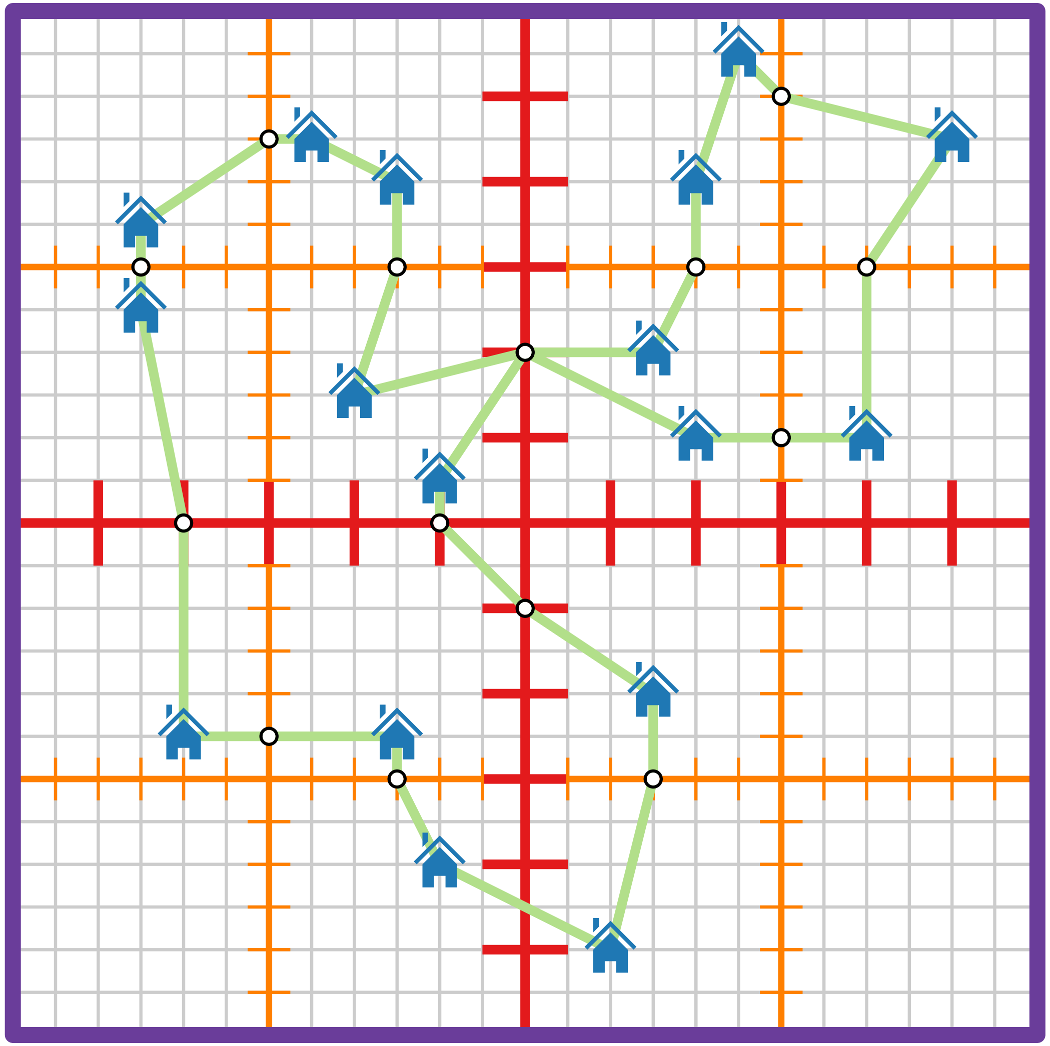
Well-Behaved Tours



A tour is *well-behaved* if

- it involves all houses and a subset of the portals,

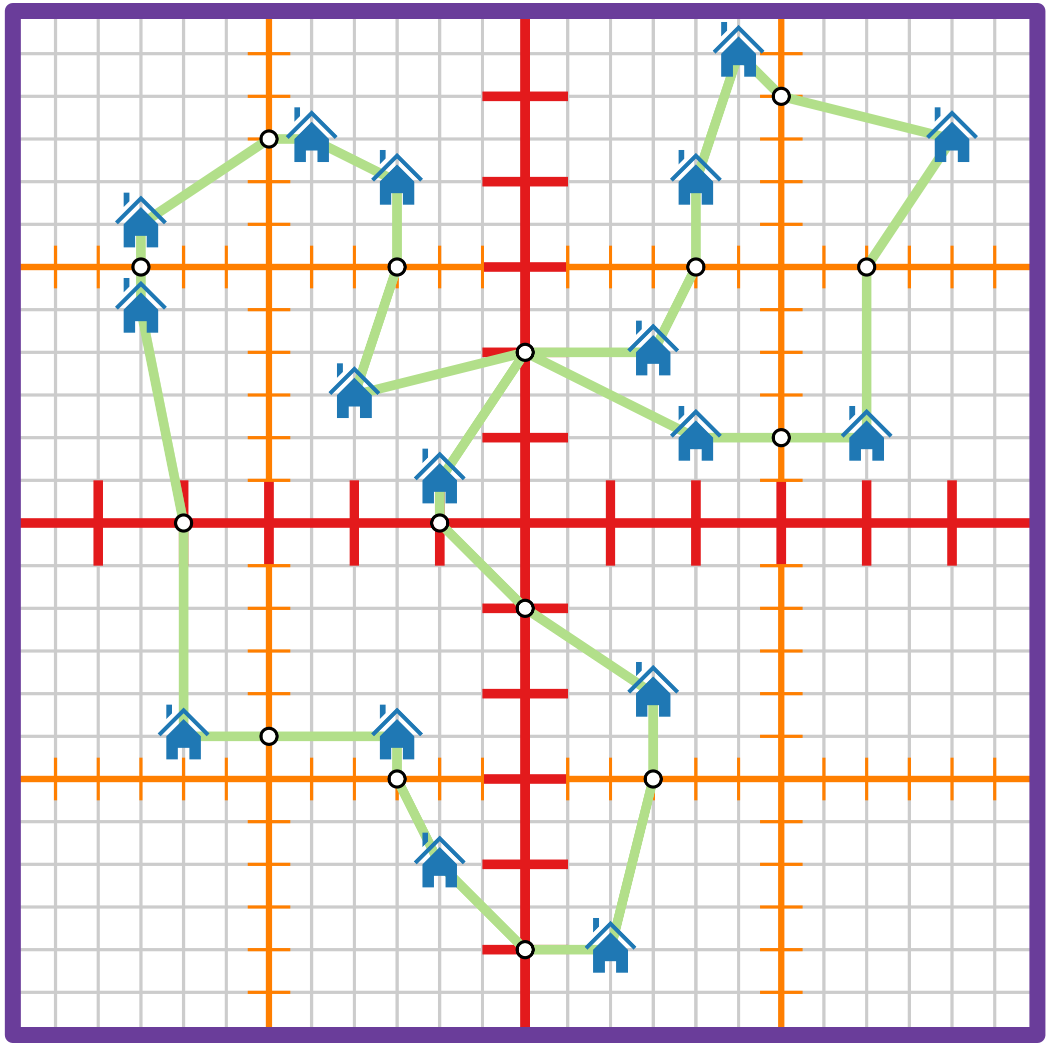
Well-Behaved Tours



A tour is *well-behaved* if

- it involves all houses and a subset of the portals,
- no edge of the tour crosses a line of the basic dissection,

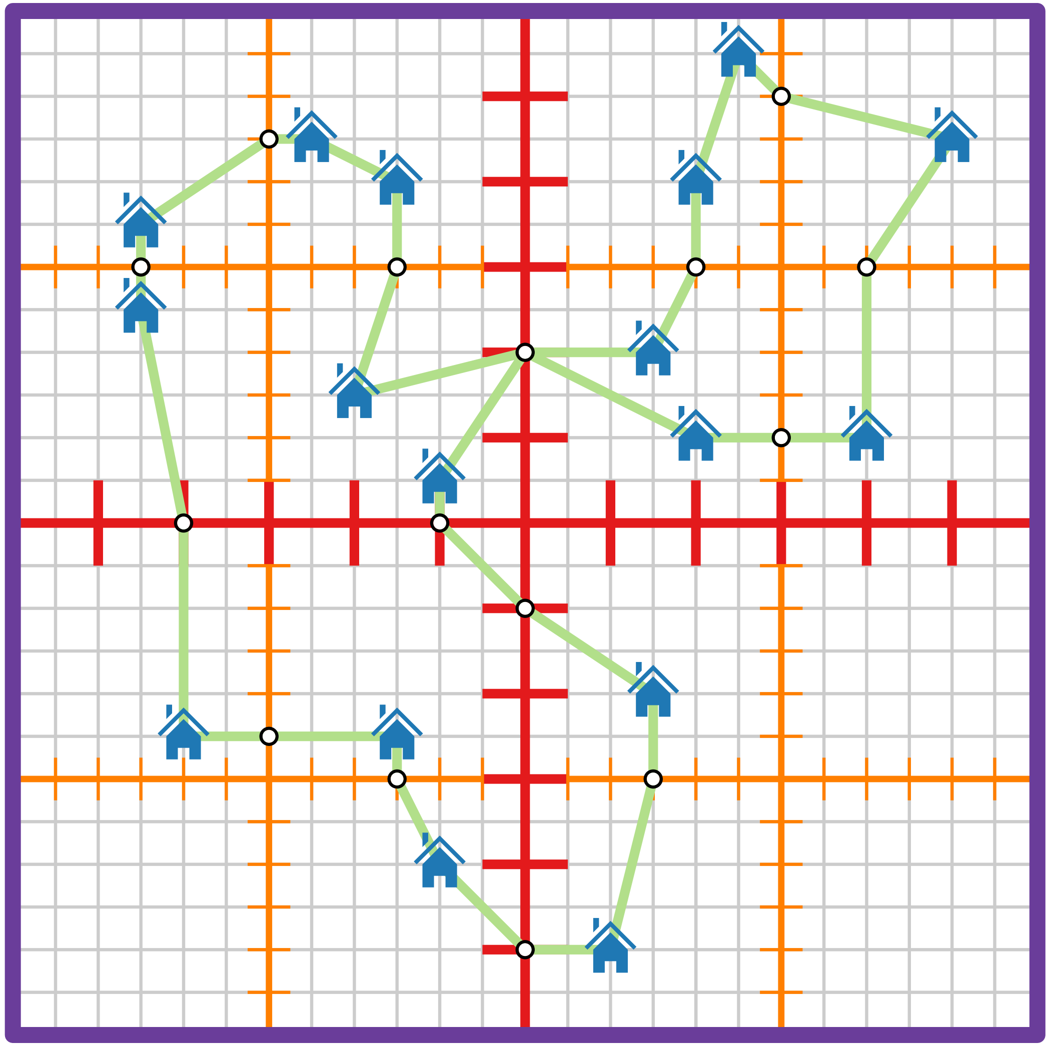
Well-Behaved Tours



A tour is *well-behaved* if

- it involves all houses and a subset of the portals,
- no edge of the tour crosses a line of the basic dissection,

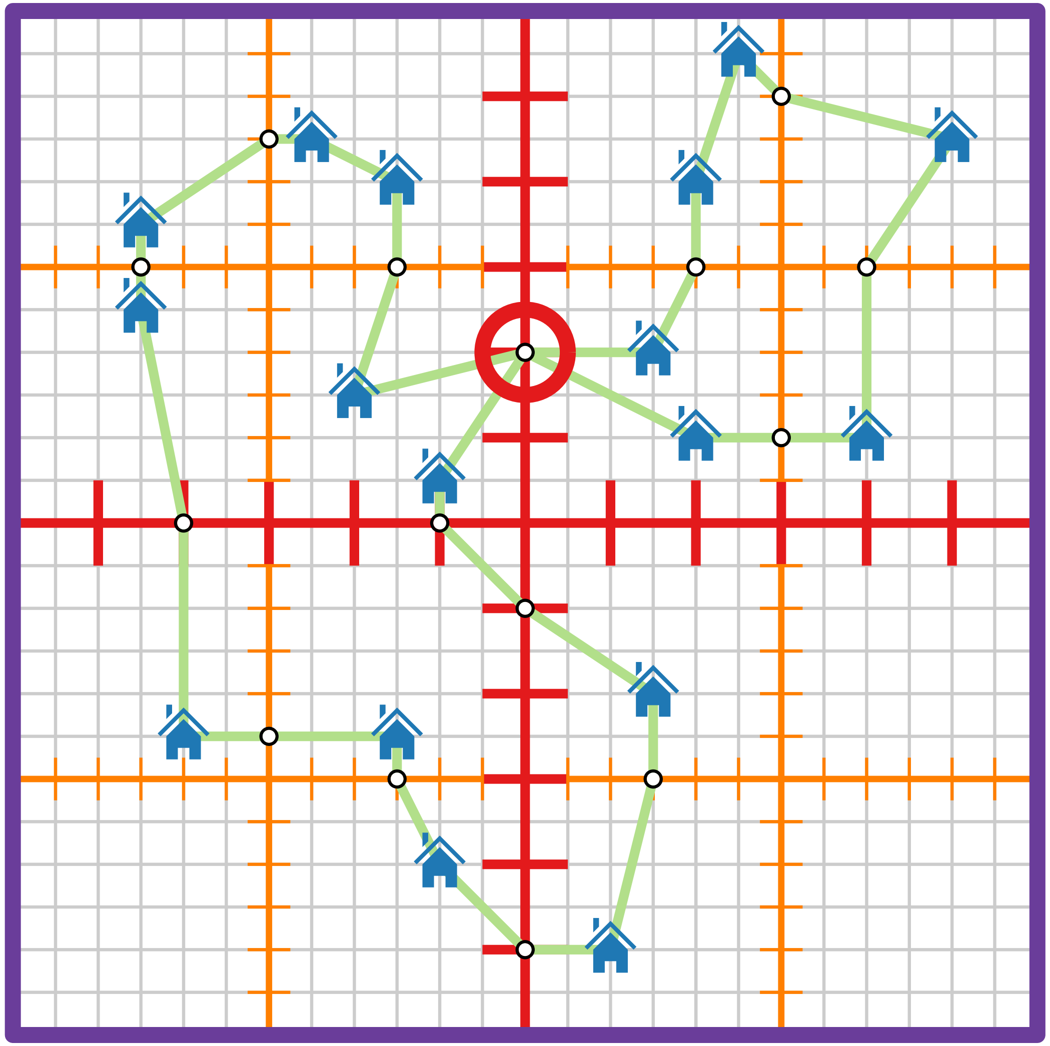
Well-Behaved Tours



A tour is *well-behaved* if

- it involves all houses and a subset of the portals,
- no edge of the tour crosses a line of the basic dissection,
- it is crossing-free.

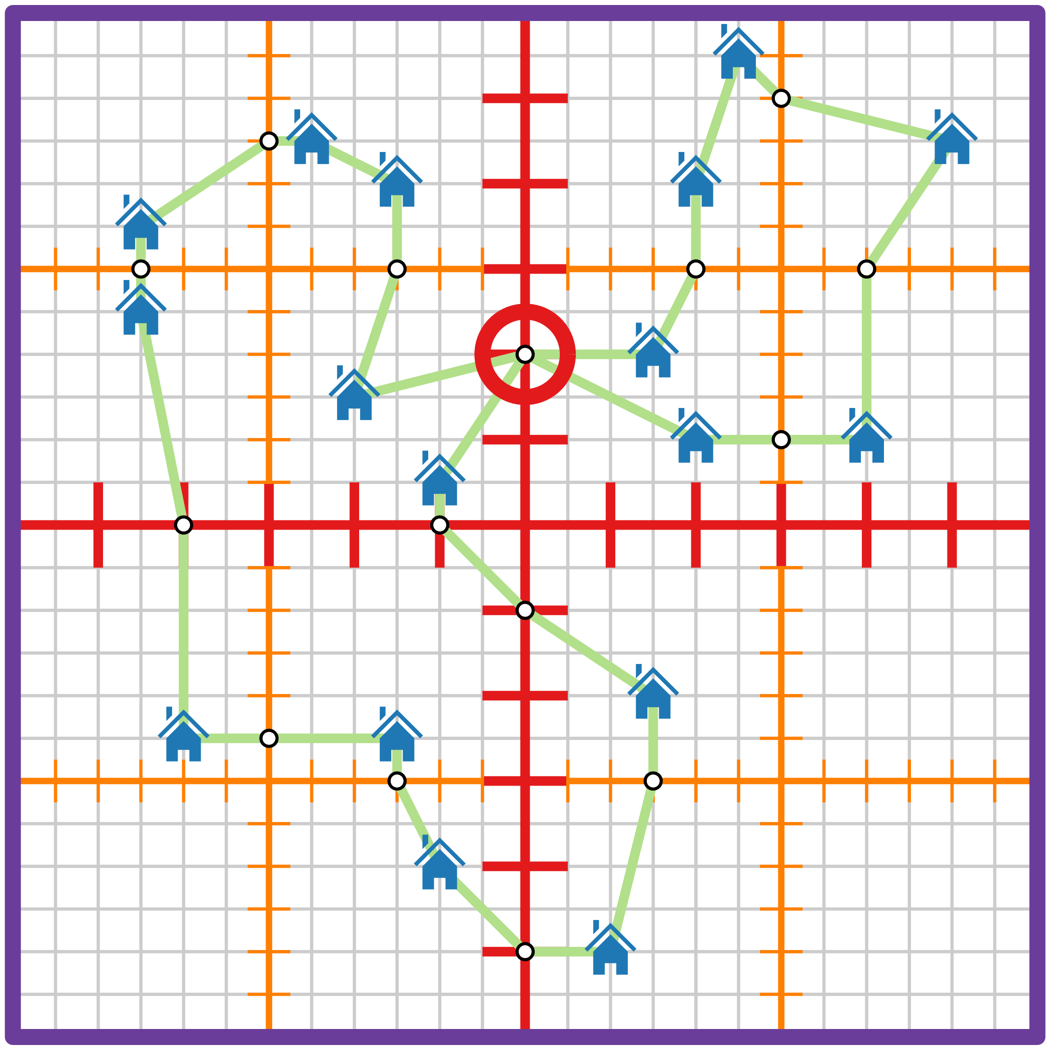
Well-Behaved Tours



A tour is *well-behaved* if

- it involves all houses and a subset of the portals,
- no edge of the tour crosses a line of the basic dissection,
- it is crossing-free.

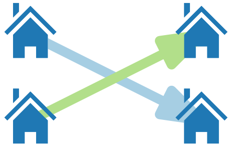
Well-Behaved Tours



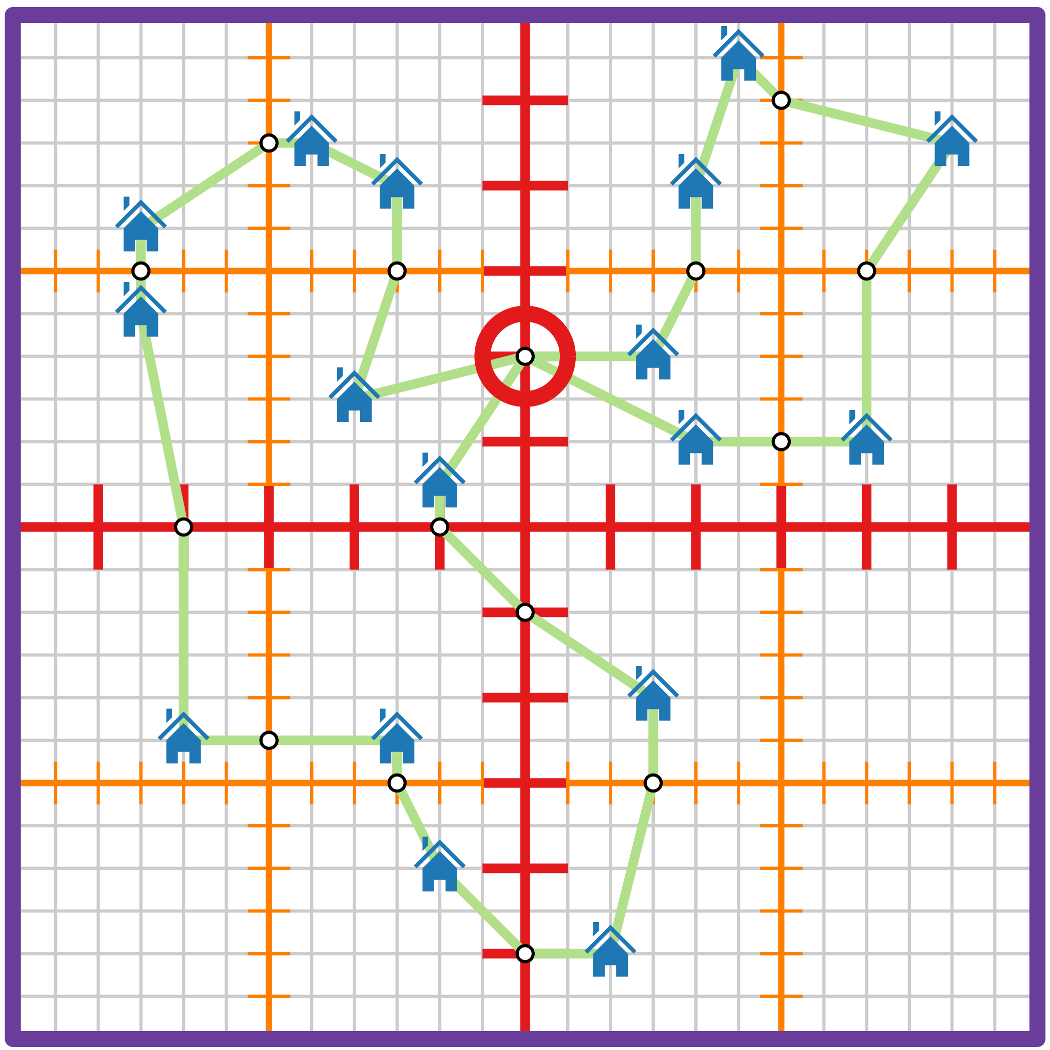
A tour is *well-behaved* if

- it involves all houses and a subset of the portals,
- no edge of the tour crosses a line of the basic dissection,
- it is crossing-free.

Crossing



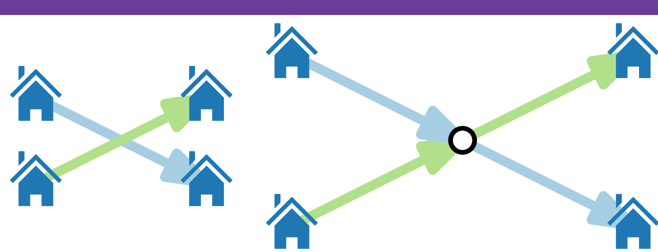
Well-Behaved Tours



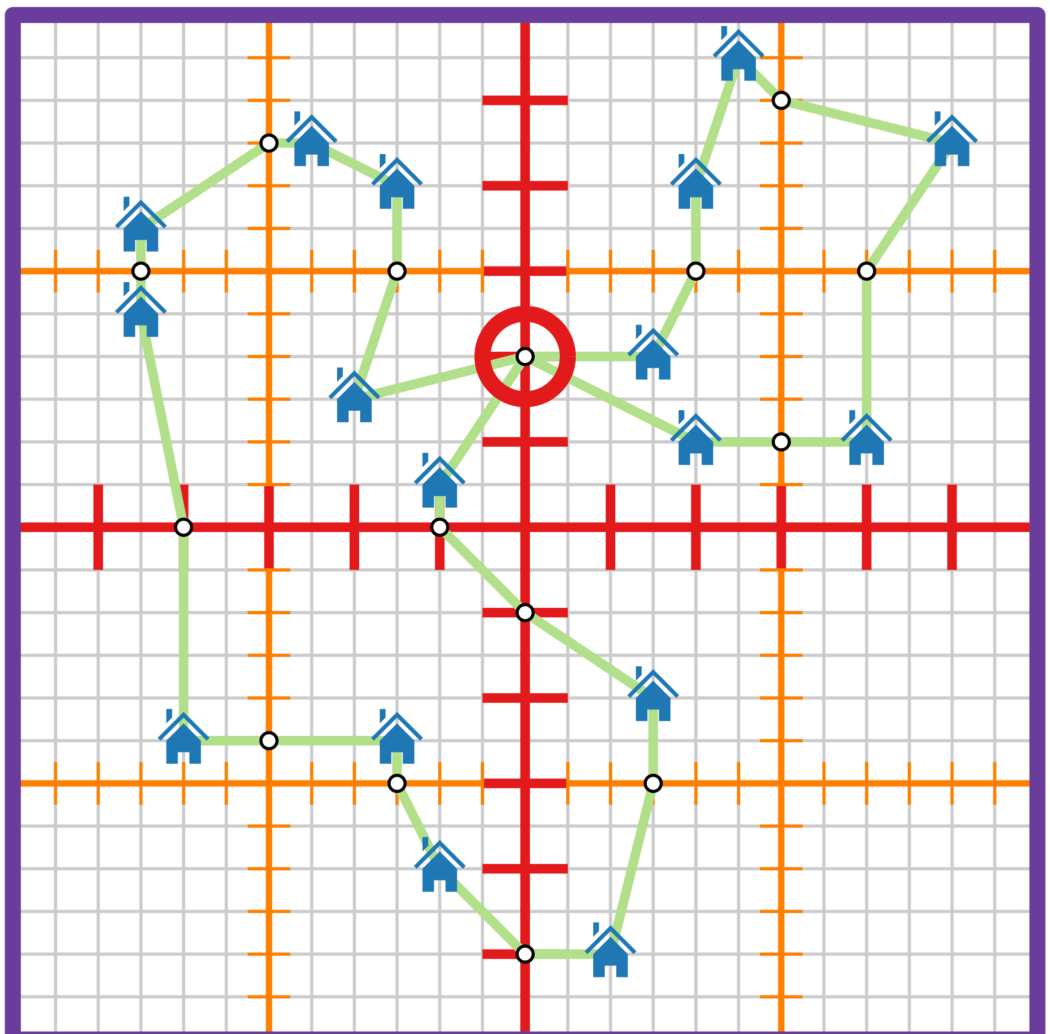
A tour is *well-behaved* if

- it involves all houses and a subset of the portals,
- no edge of the tour crosses a line of the basic dissection,
- it is crossing-free.

Crossing



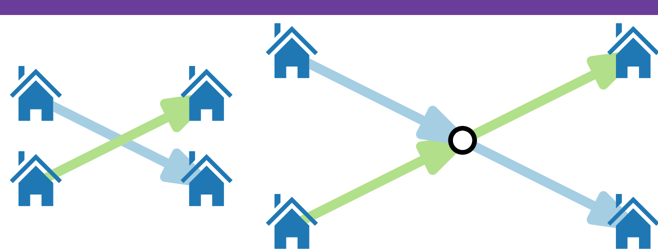
Well-Behaved Tours



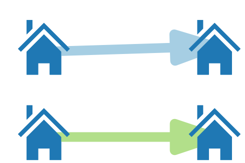
A tour is *well-behaved* if

- it involves all houses and a subset of the portals,
- no edge of the tour crosses a line of the basic dissection,
- it is crossing-free.

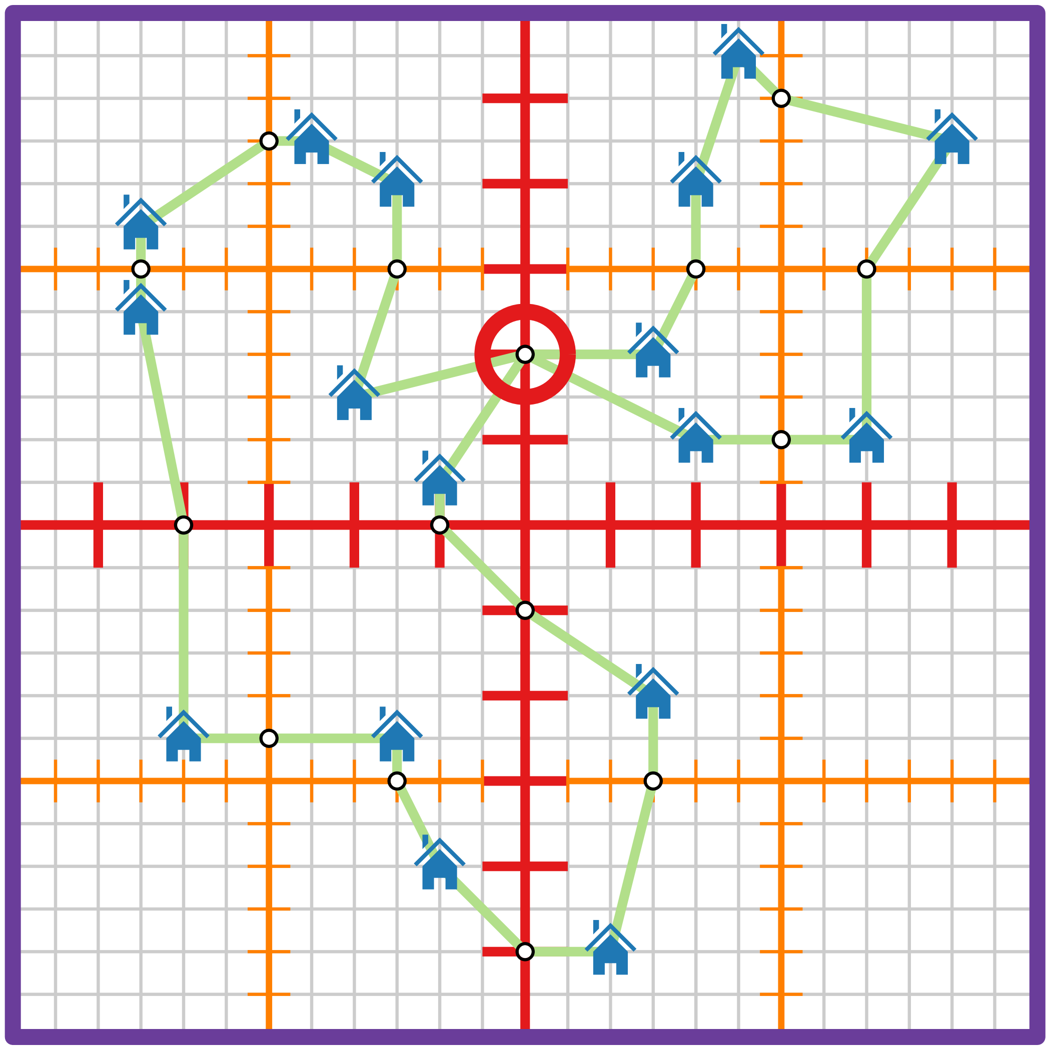
Crossing



No crossing



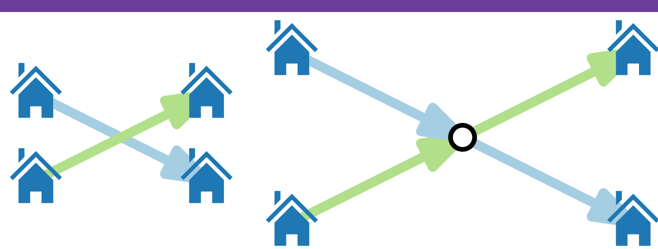
Well-Behaved Tours



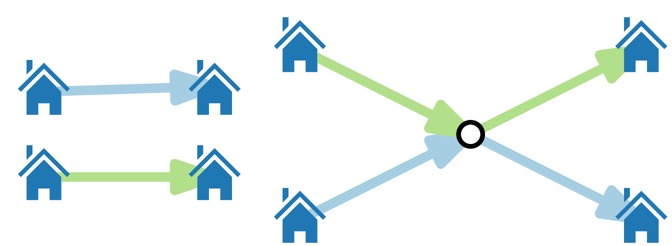
A tour is *well-behaved* if

- it involves all houses and a subset of the portals,
- no edge of the tour crosses a line of the basic dissection,
- it is crossing-free.

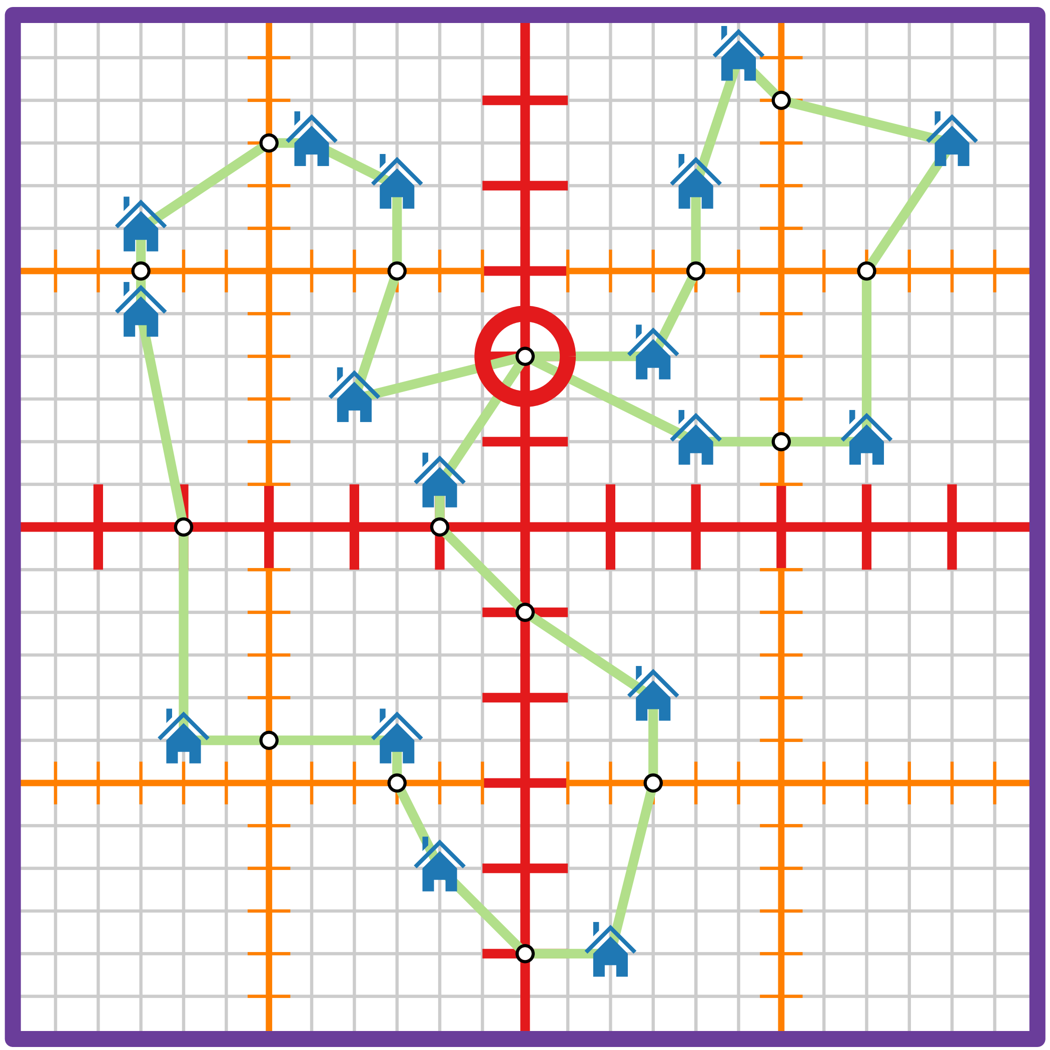
Crossing



No crossing



Well-Behaved Tours

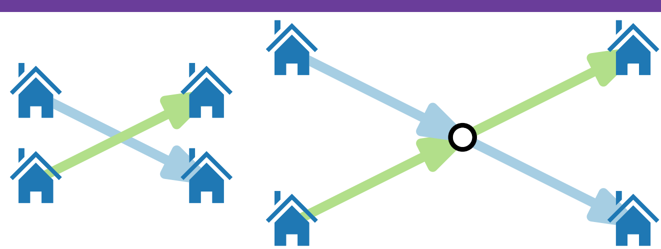


A tour is *well-behaved* if

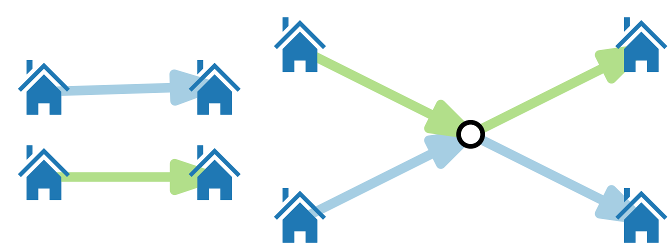
- it involves all houses and a subset of the portals,
- no edge of the tour crosses a line of the basic dissection,
- it is crossing-free.

W.l.o.g. (homework):
No portal visited more than twice

Crossing



No crossing



Computing a Well-Behaved Tour

Computing a Well-Behaved Tour

Lemma. An optimal well-behaved tour can be computed in $2^{O(m)} = n^{O(1/\varepsilon)}$ time.

Computing a Well-Behaved Tour

Lemma. An optimal well-behaved tour can be computed in $2^{O(m)} = n^{O(1/\varepsilon)}$ time.

Sketch.

Computing a Well-Behaved Tour

Lemma. An optimal well-behaved tour can be computed in $2^{O(m)} = n^{O(1/\varepsilon)}$ time.

Sketch.

- Dynamic programming!

Computing a Well-Behaved Tour

Lemma. An optimal well-behaved tour can be computed in $2^{O(m)} = n^{O(1/\varepsilon)}$ time.

Sketch.

- Dynamic programming!
- Compute sub-structure of an optimal tour for each square in the dissection tree.

Computing a Well-Behaved Tour

Lemma. An optimal well-behaved tour can be computed in $2^{O(m)} = n^{O(1/\varepsilon)}$ time.

Sketch.

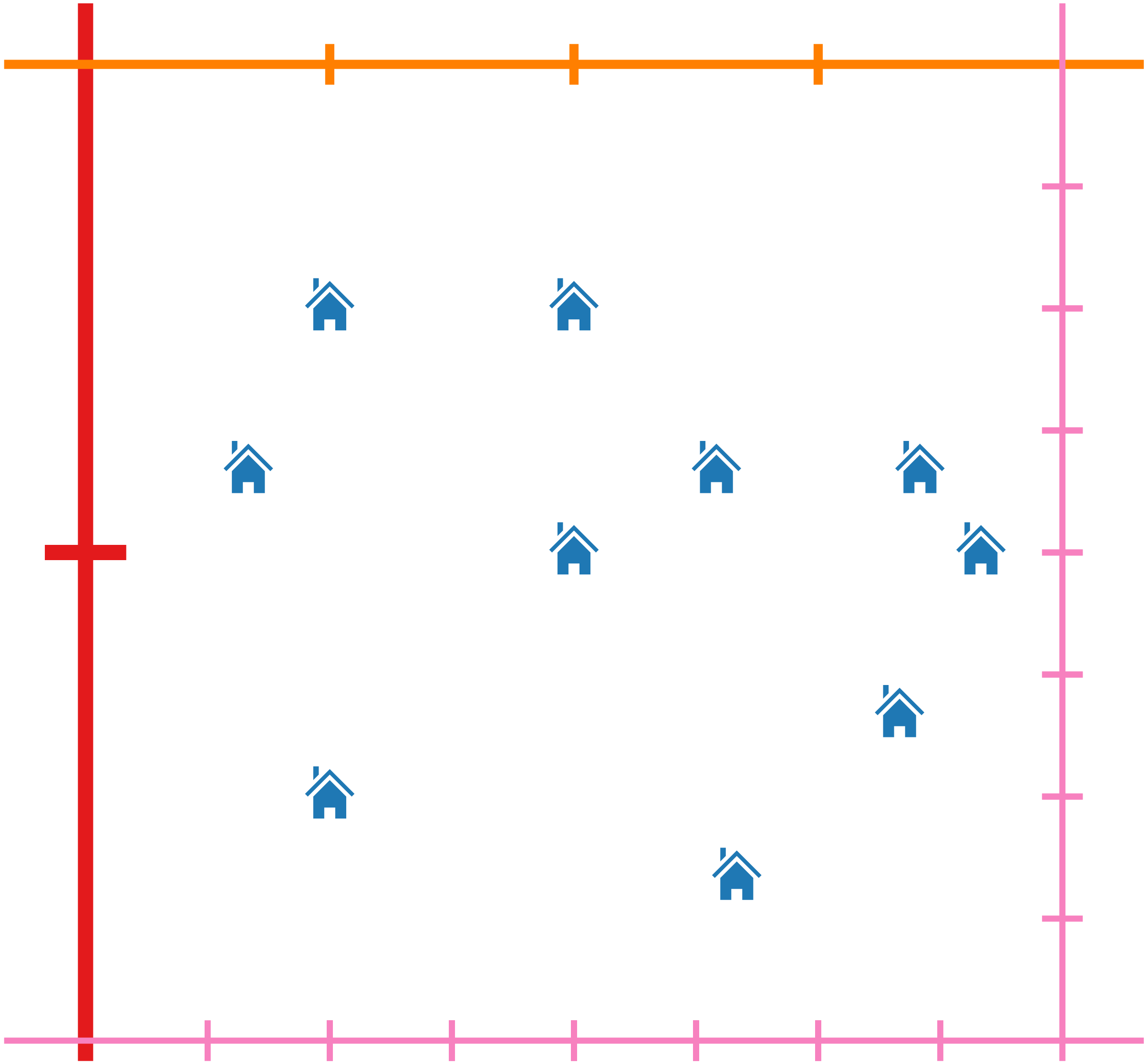
- Dynamic programming!
- Compute sub-structure of an optimal tour for each square in the dissection tree.
- These solutions can be efficiently propagated bottom-up through the dissection tree.

Approximation Algorithms

Lecture 9:
A PTAS for Euclidean TSP

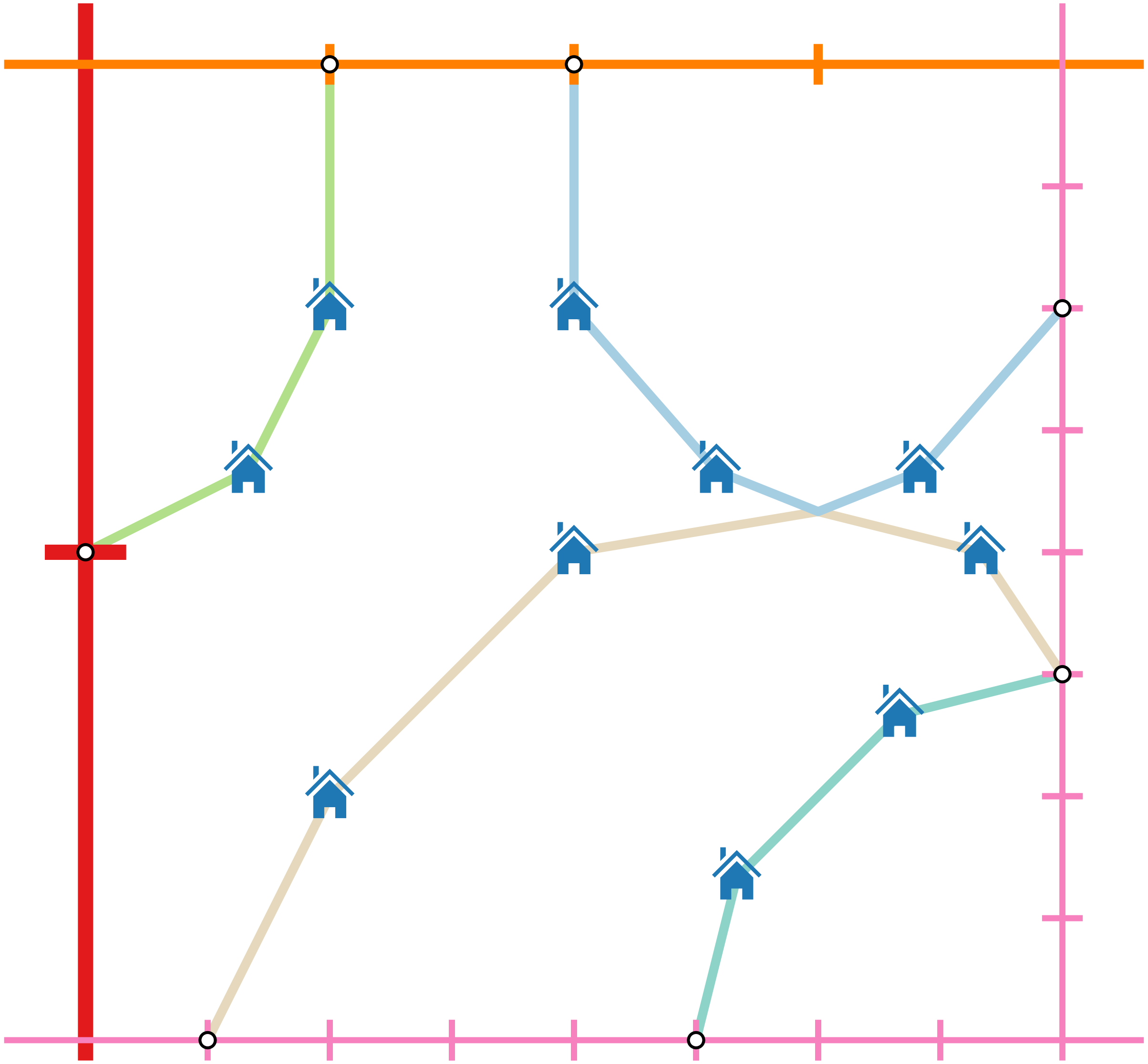
Part IV:
Dynamic Program

Dynamic Program (I)



Each well-behaved tour induces the following in each square Q of the dissection:

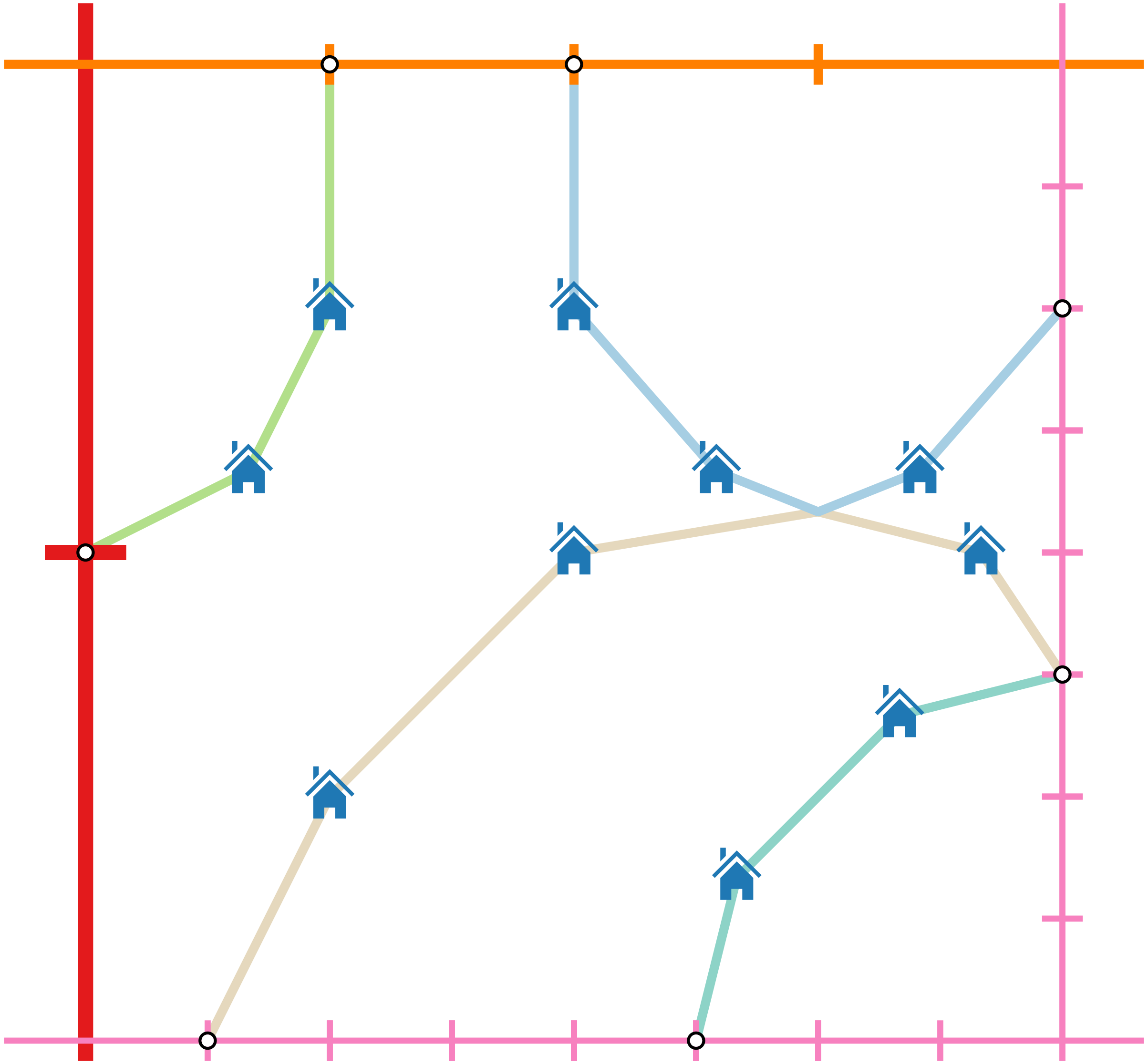
Dynamic Program (I)



Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,

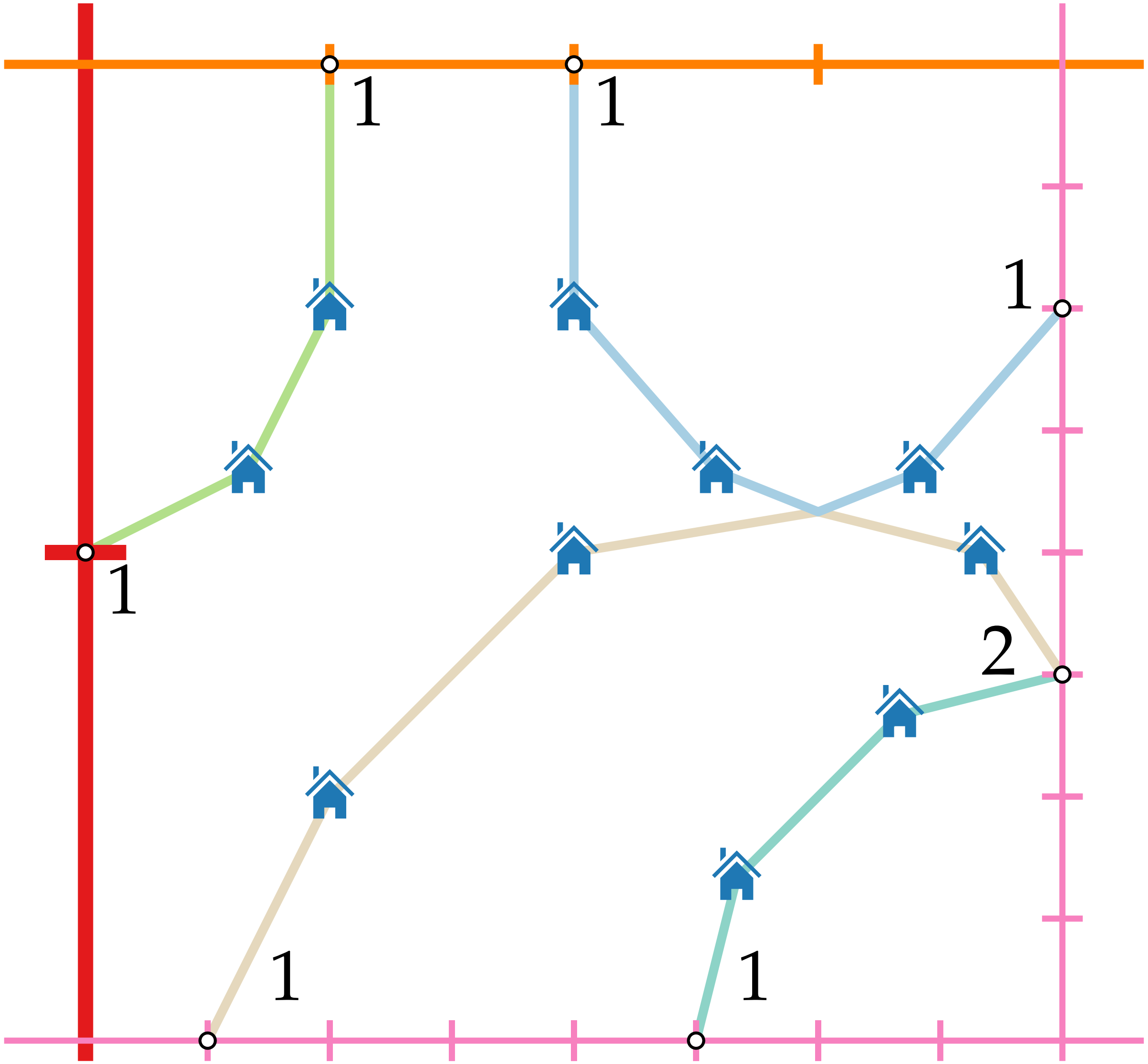
Dynamic Program (I)



Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,

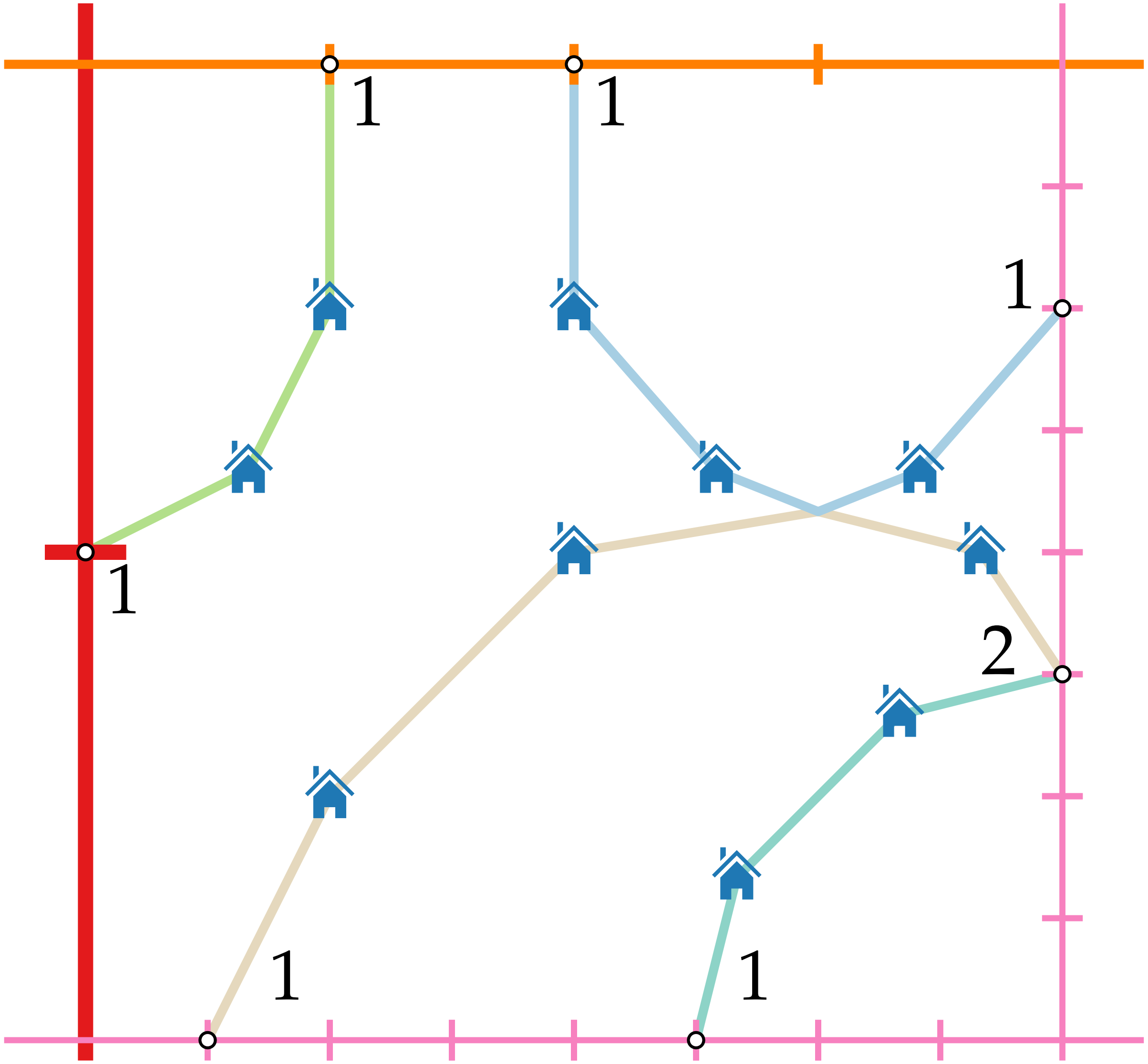
Dynamic Program (I)



Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,

Dynamic Program (I)

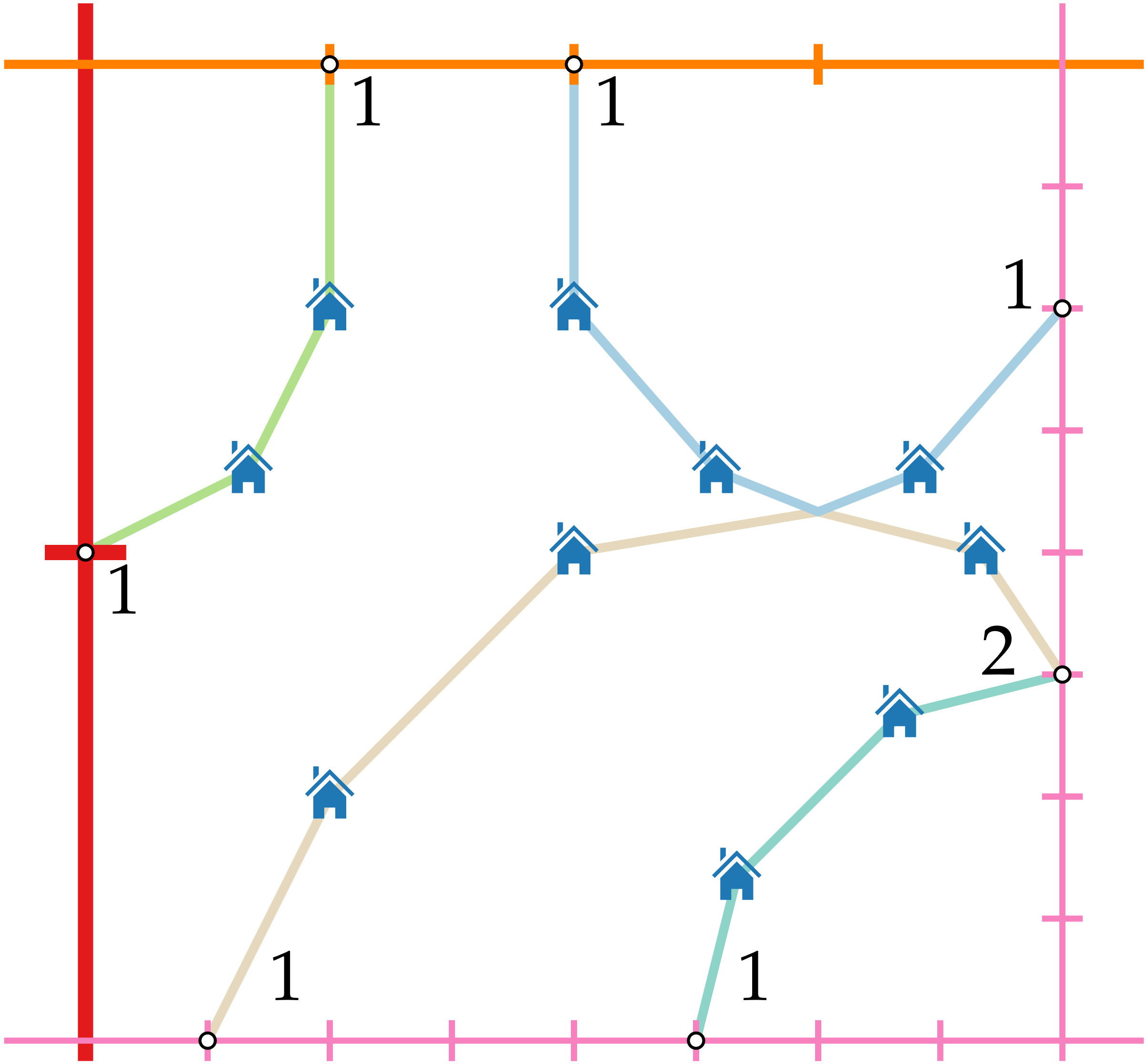


Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,

$\Rightarrow$ max. possibilities

Dynamic Program (I)

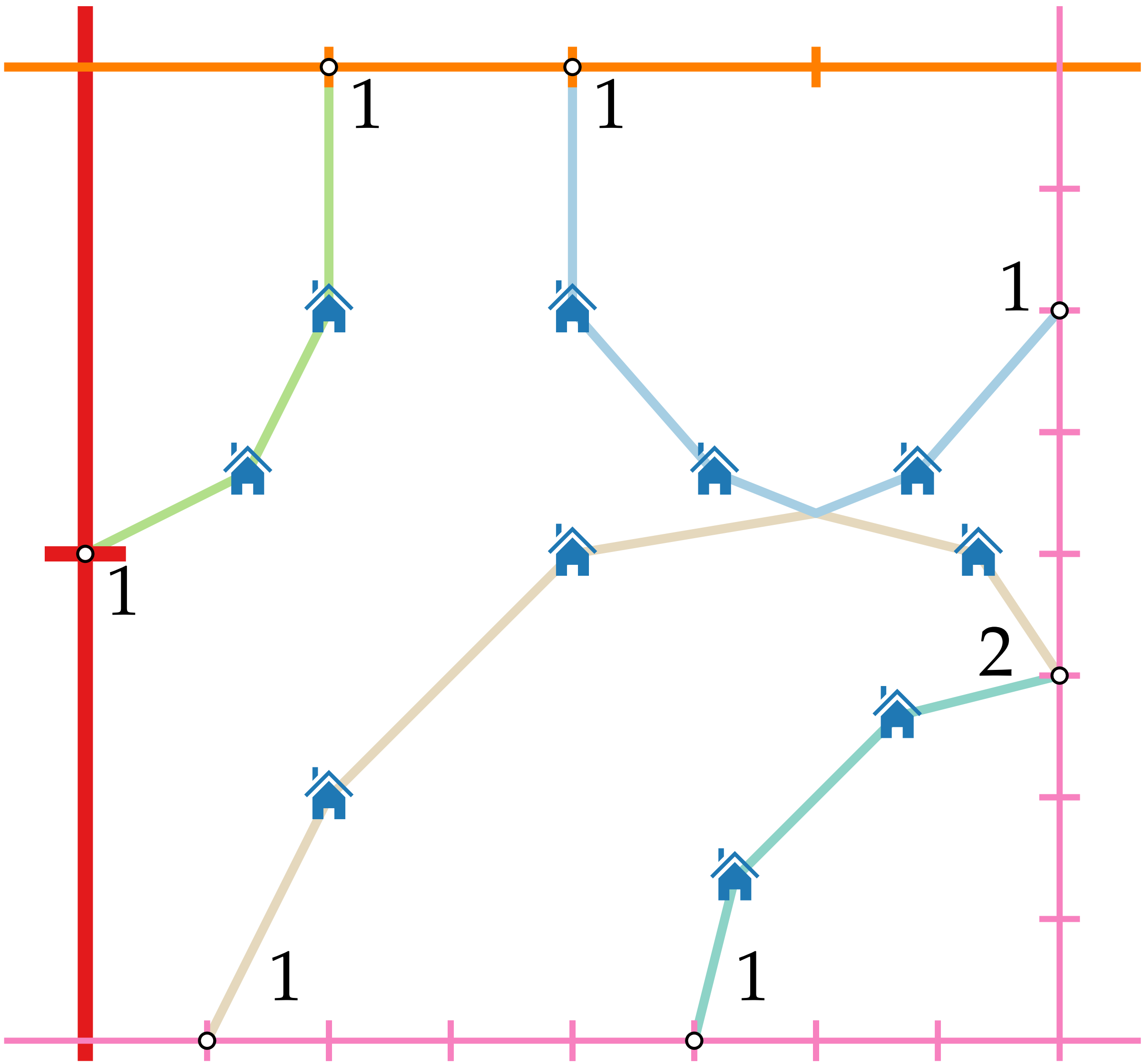


Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,

$\Rightarrow$ max. $3^{4m} \in$ possibilities

Dynamic Program (I)



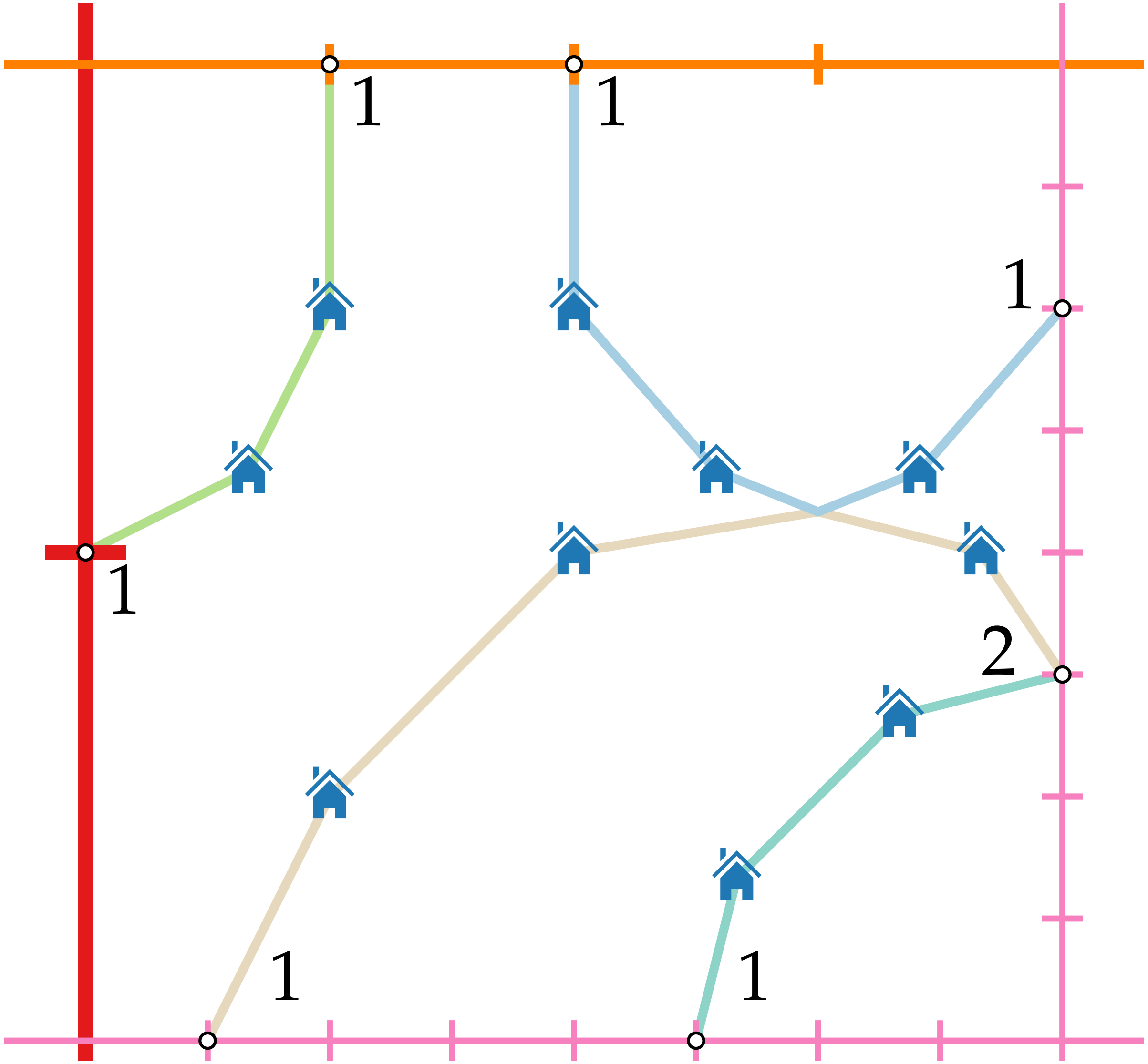
Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,

$\Rightarrow$ max. $3^{4m} \in$ possibilities

$m = O((\log n)/\varepsilon)$

Dynamic Program (I)



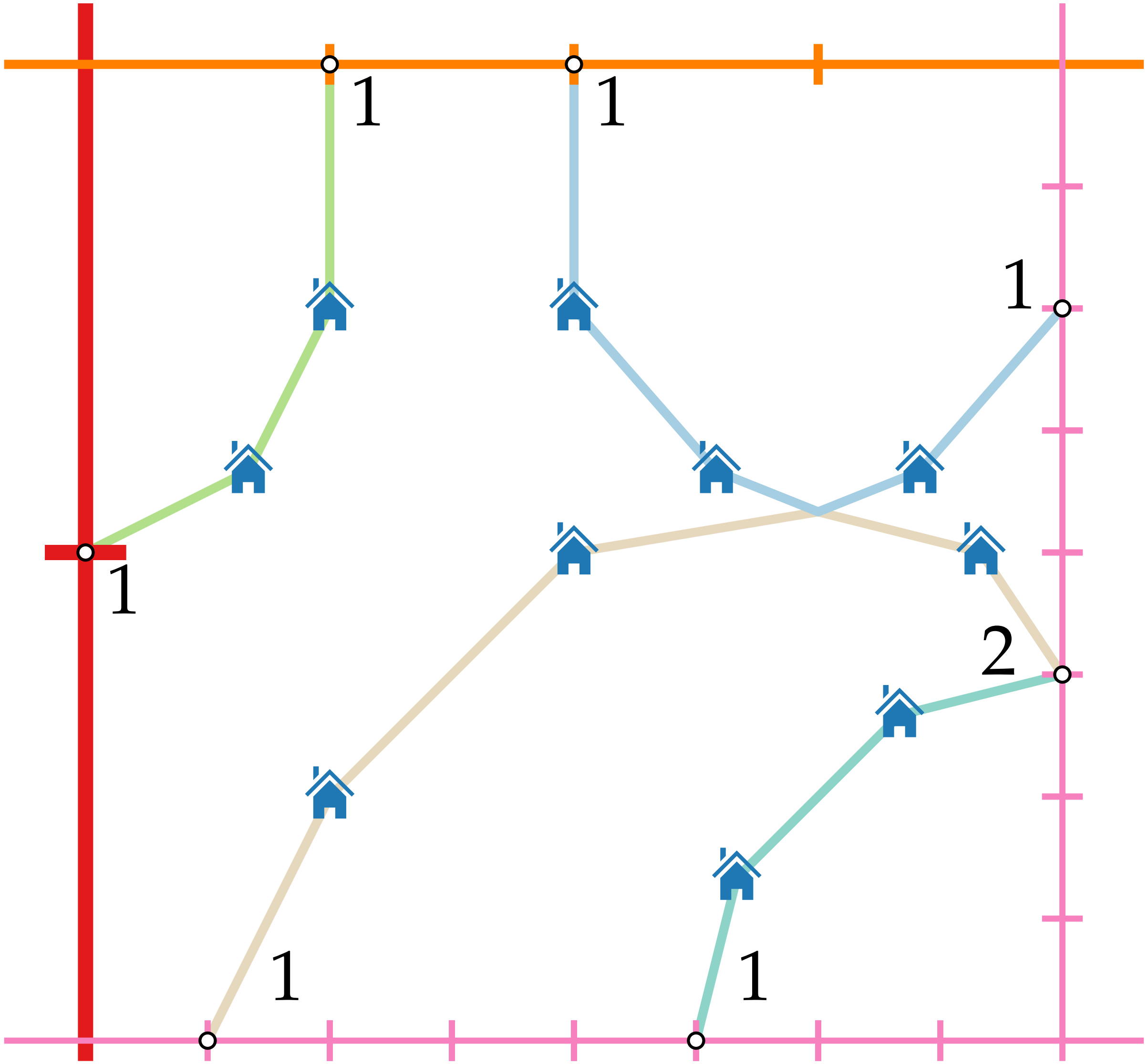
Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,

$\Rightarrow$ max. $3^{4m} \in 3^{O((\log n)/\varepsilon)}$ possibilities

$m = O((\log n)/\varepsilon)$

Dynamic Program (I)



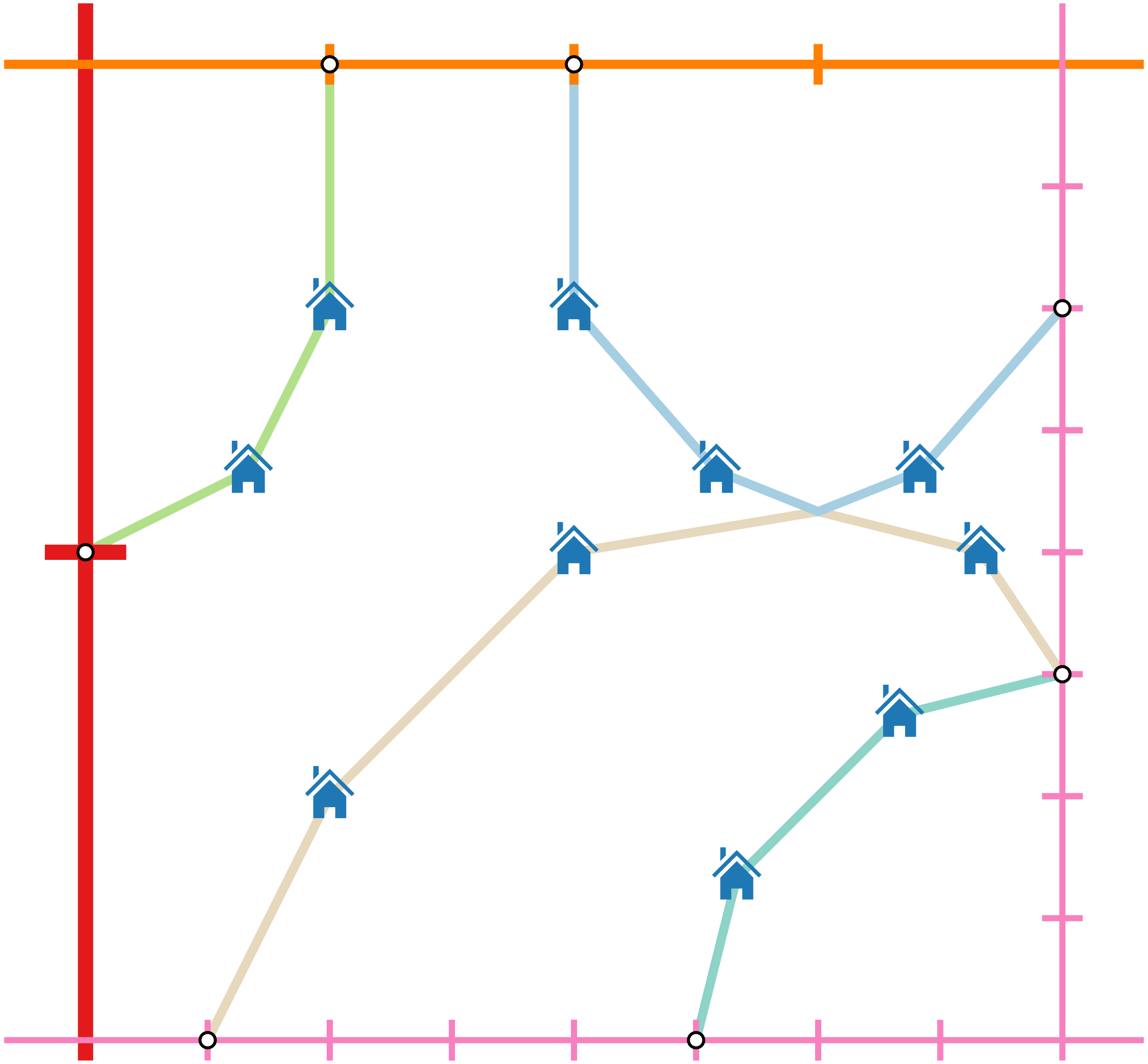
Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,

$\Rightarrow$ max. $3^{4m} \in 3^{O((\log n)/\varepsilon)} = n^{O(1/\varepsilon)}$ possibilities

$m = O((\log n)/\varepsilon)$

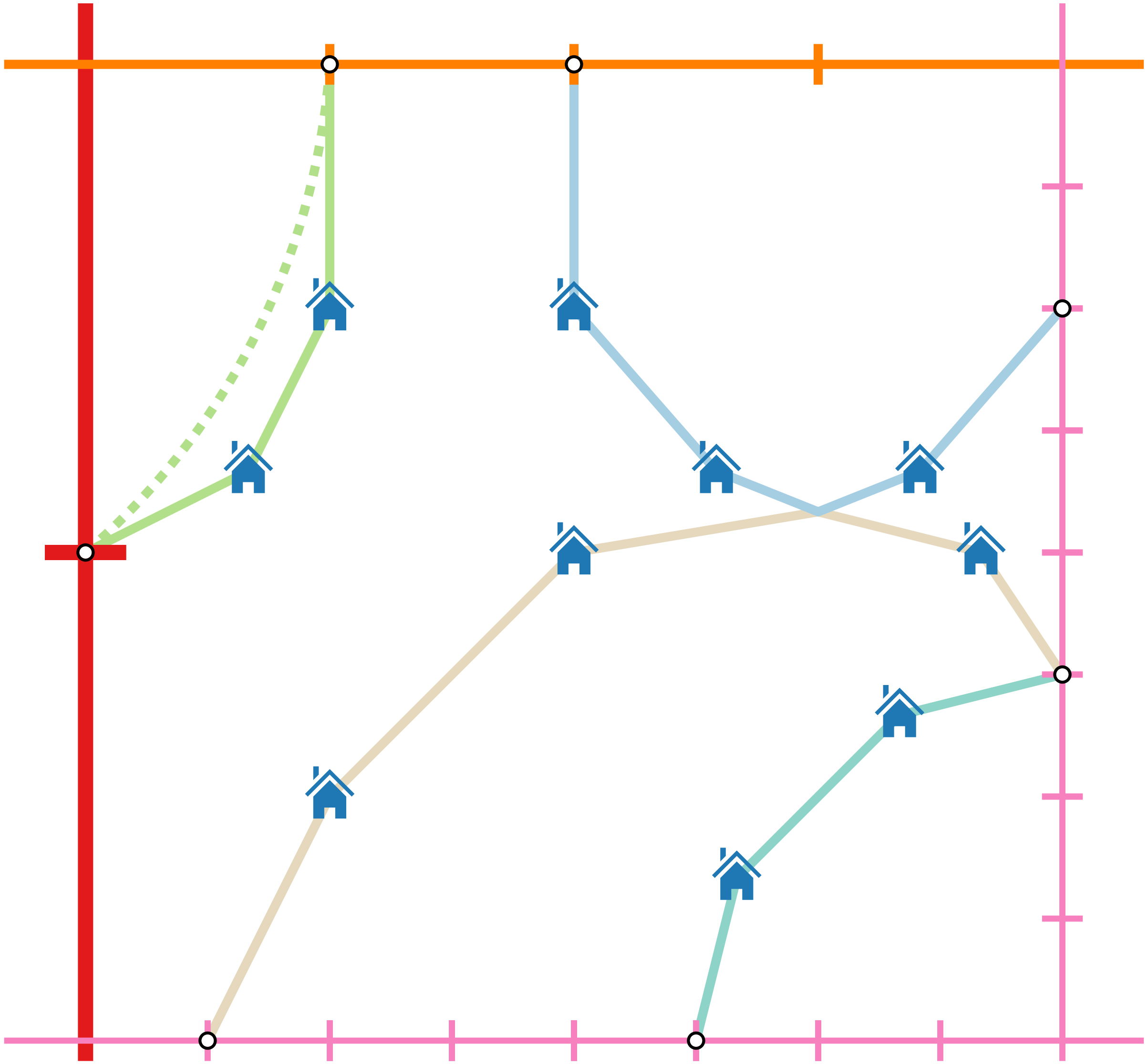
Dynamic Program (I)



Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.

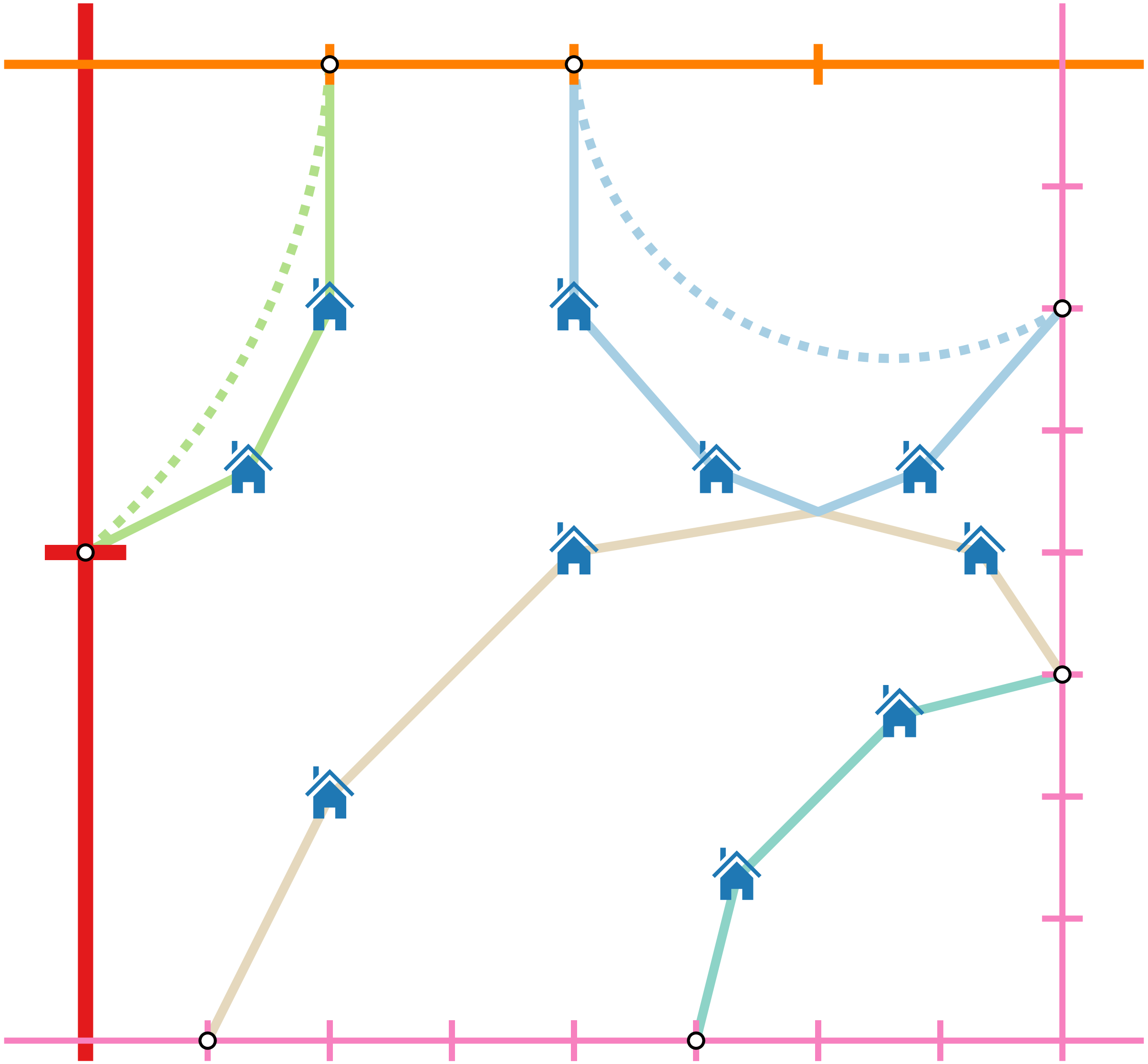
Dynamic Program (I)



Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.

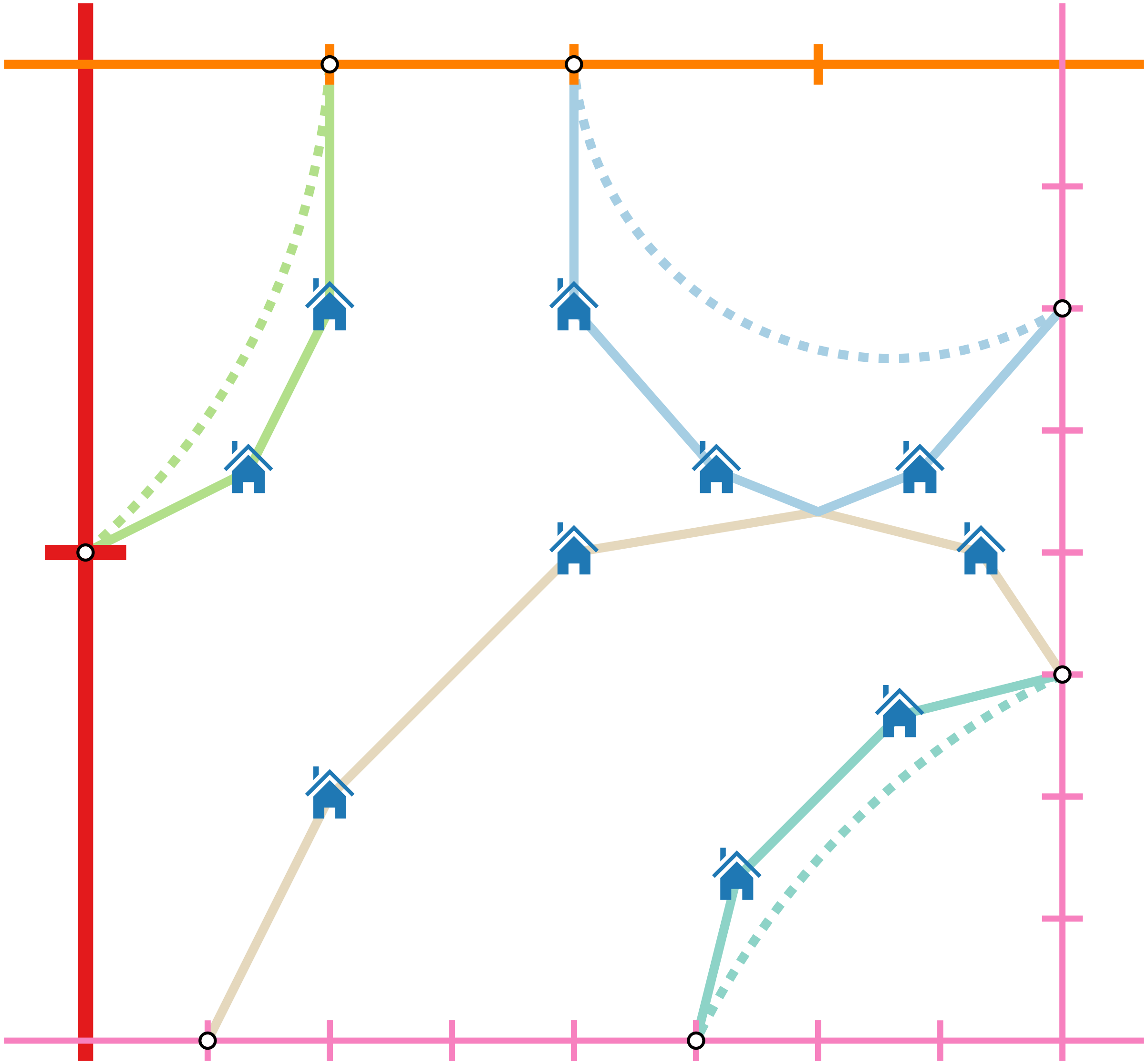
Dynamic Program (I)



Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.

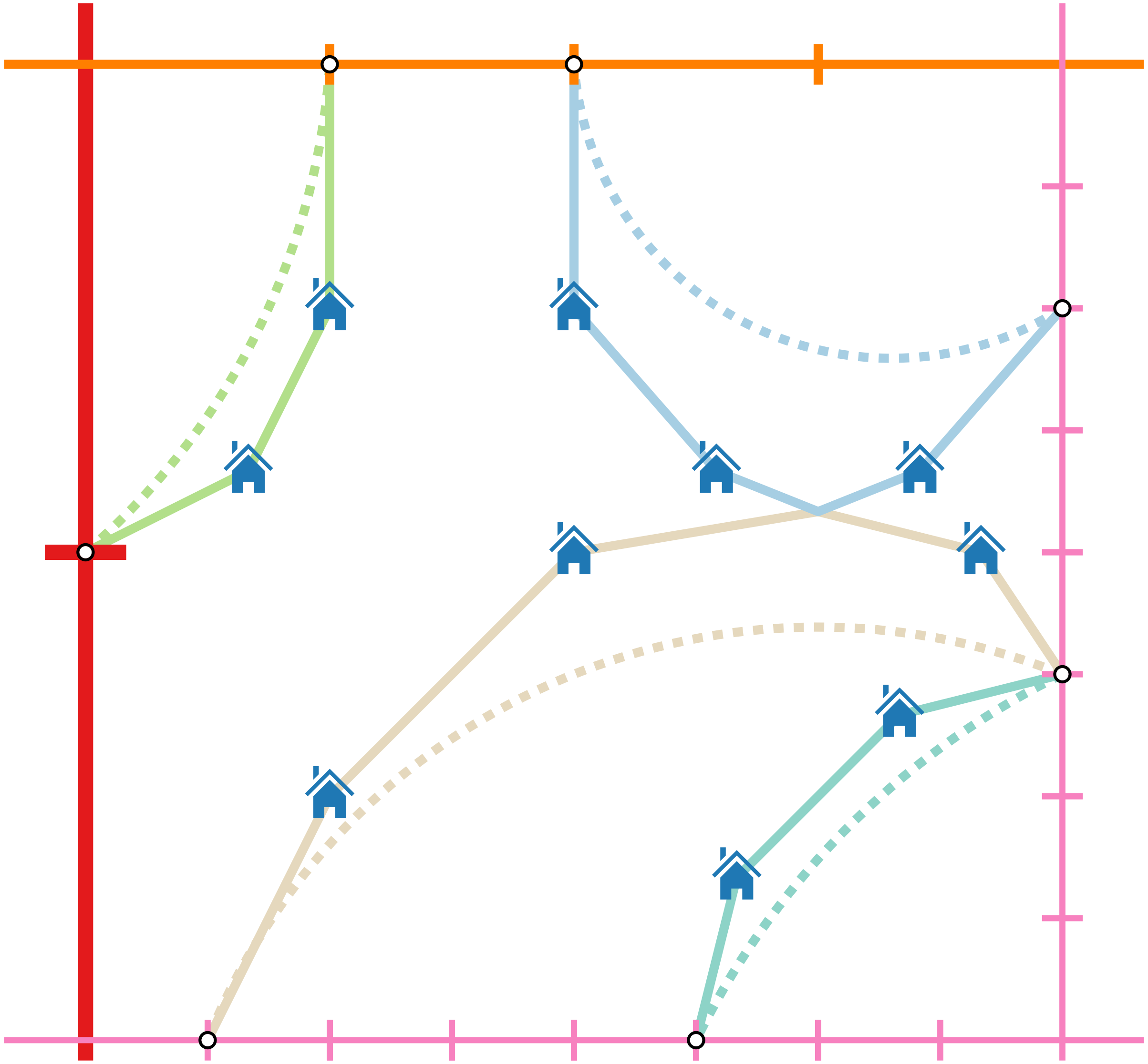
Dynamic Program (I)



Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.

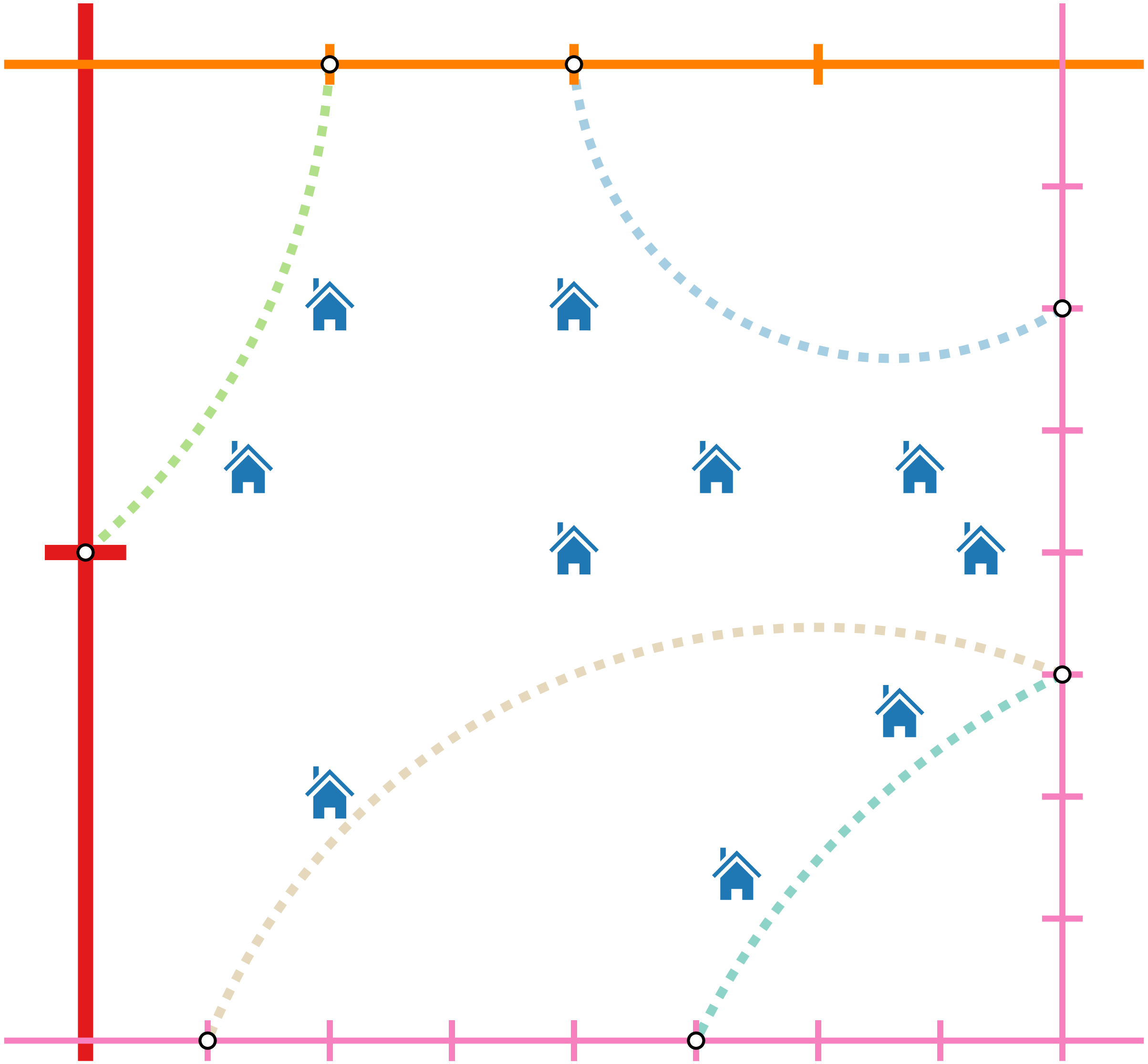
Dynamic Program (I)



Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.

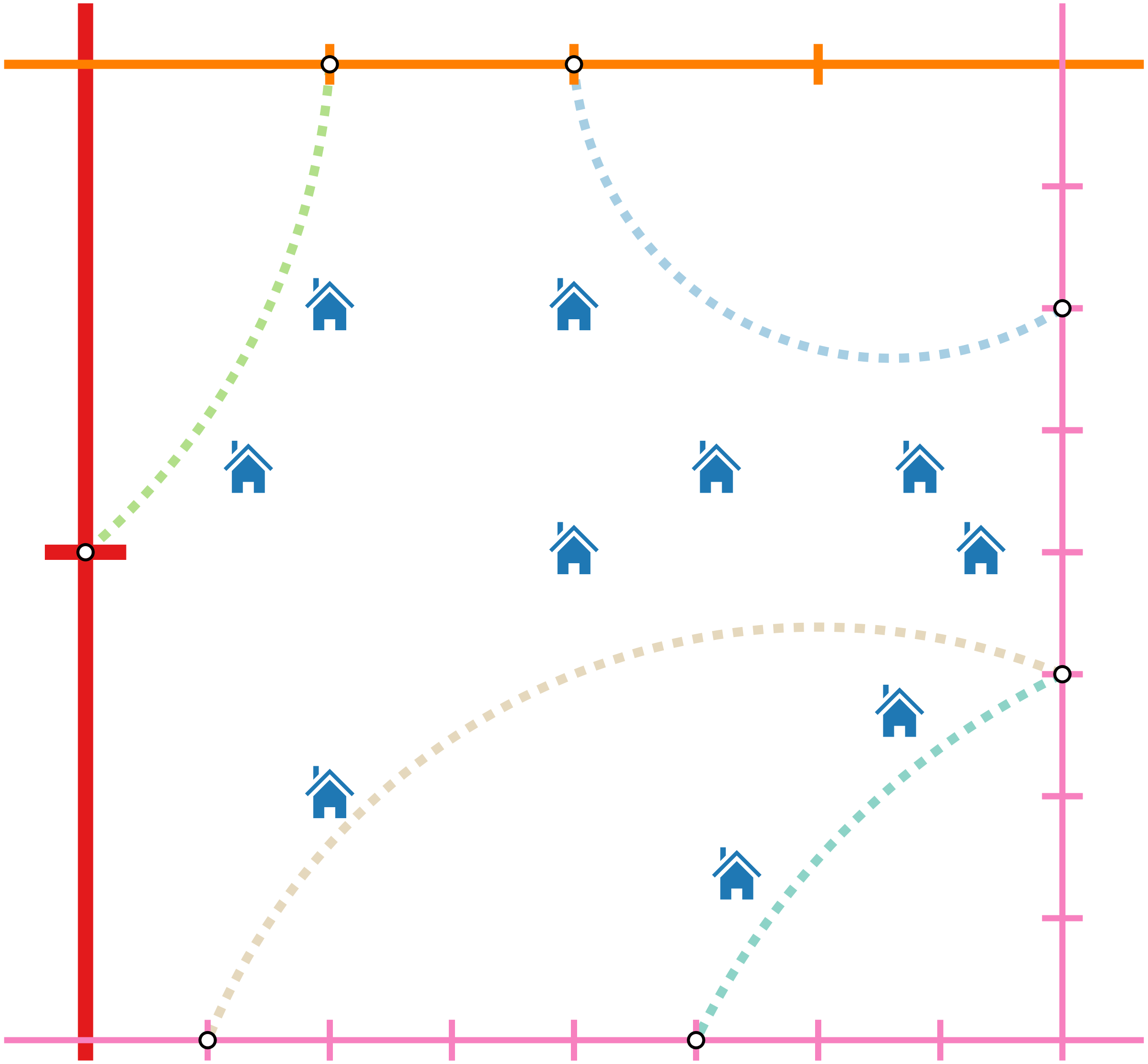
Dynamic Program (I)



Each well-behaved tour induces the following in each square Q of the dissection:

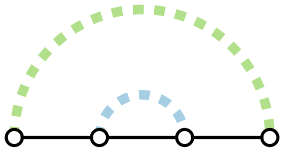
- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.

Dynamic Program (I)

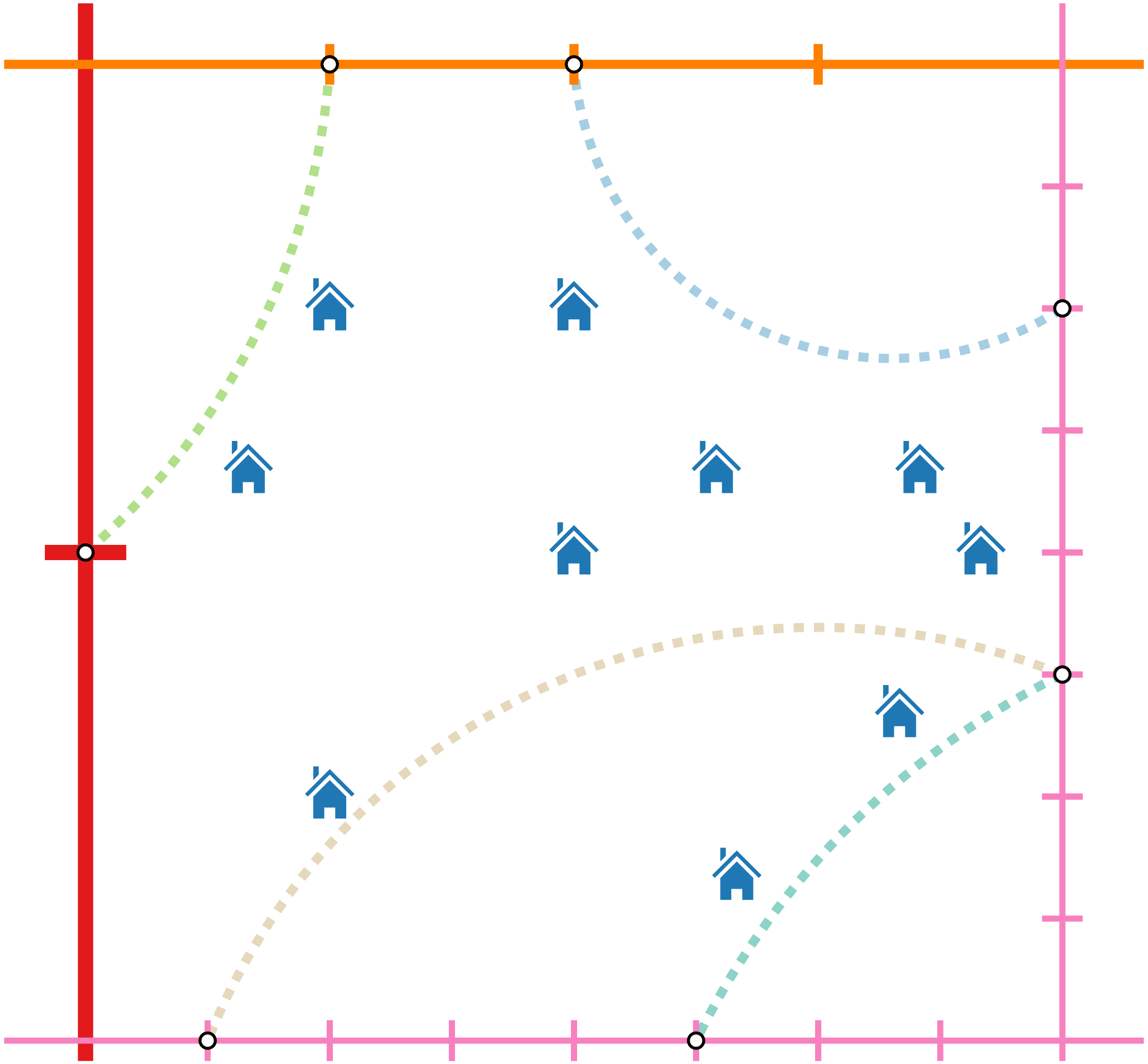


Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.

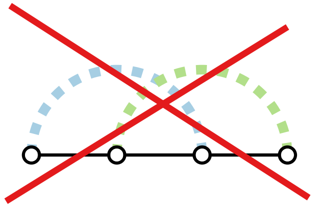


Dynamic Program (I)

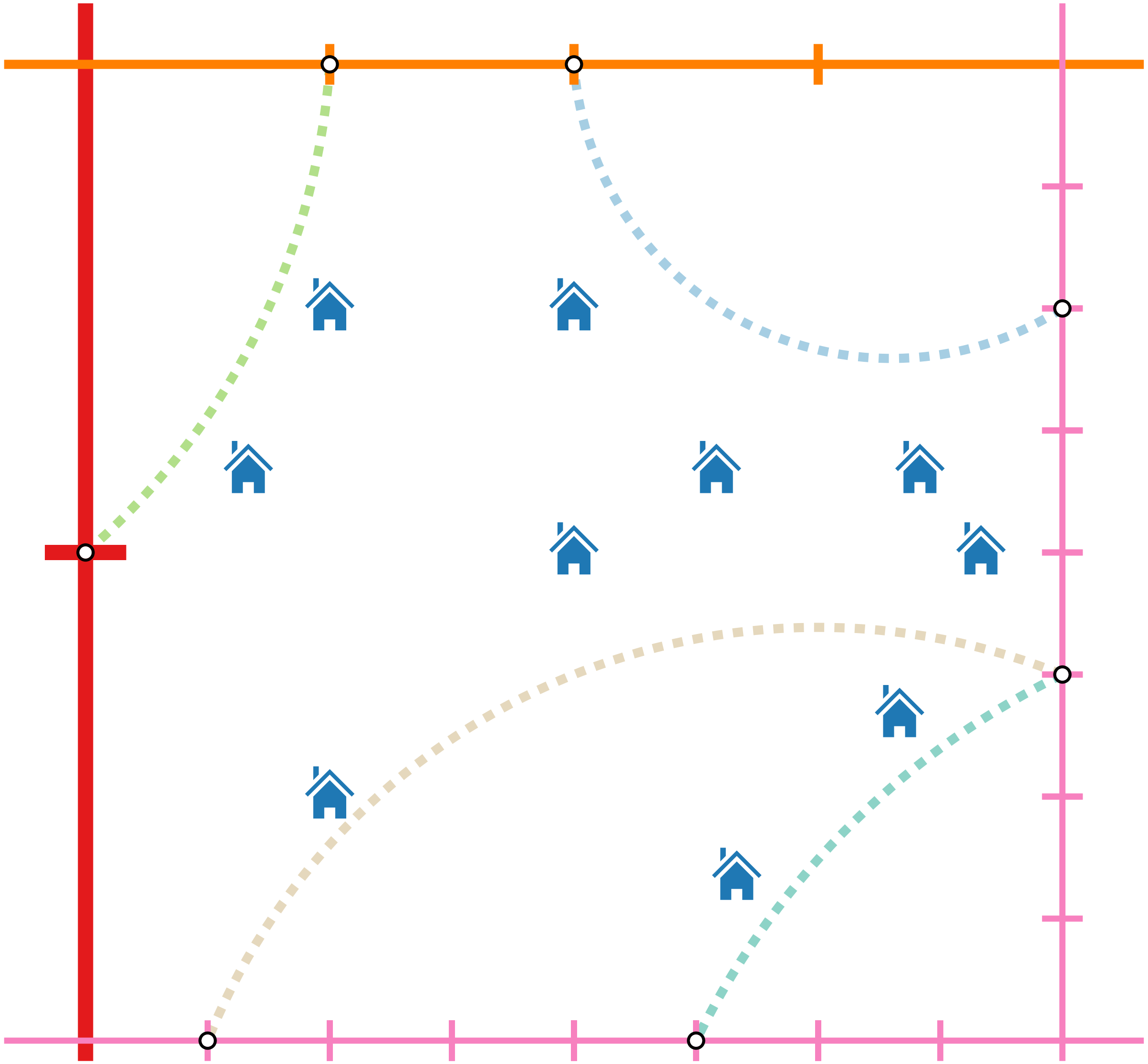


Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.



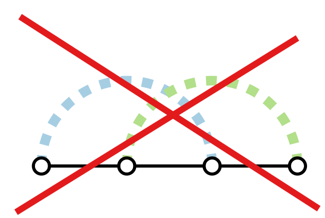
Dynamic Program (I)



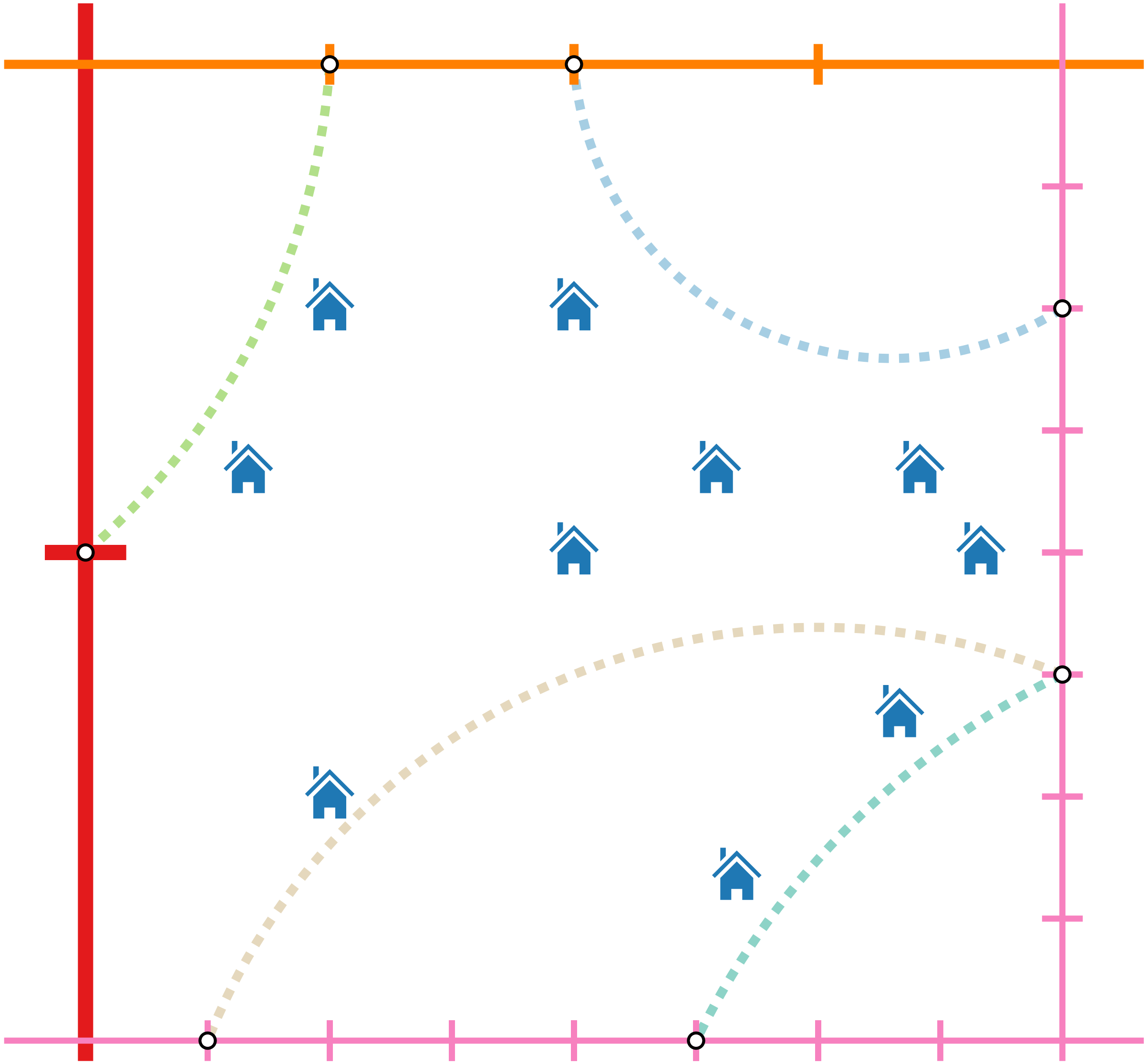
⇒ max.

Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.

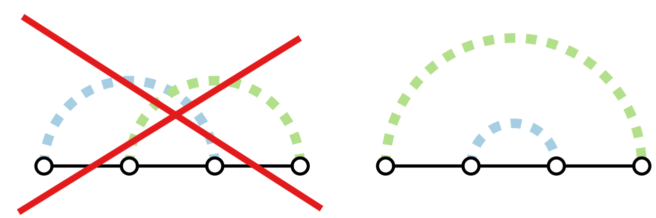


Dynamic Program (I)



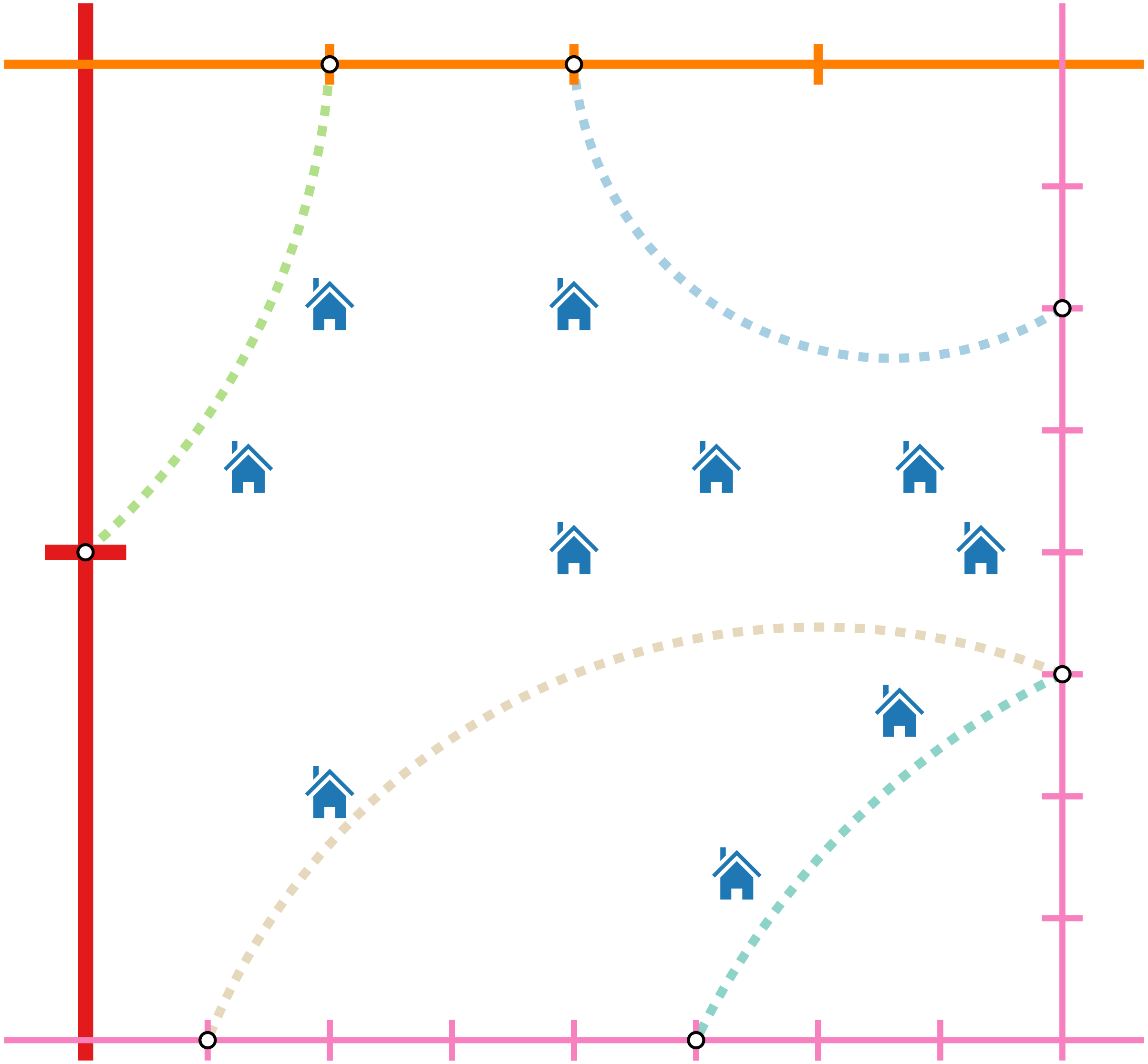
Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.



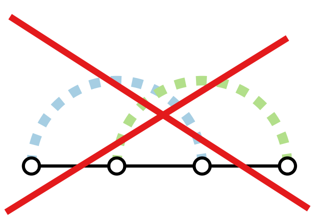
$\Rightarrow$ max. $\underbrace{}_{\text{\#visit vectors}}$

Dynamic Program (I)



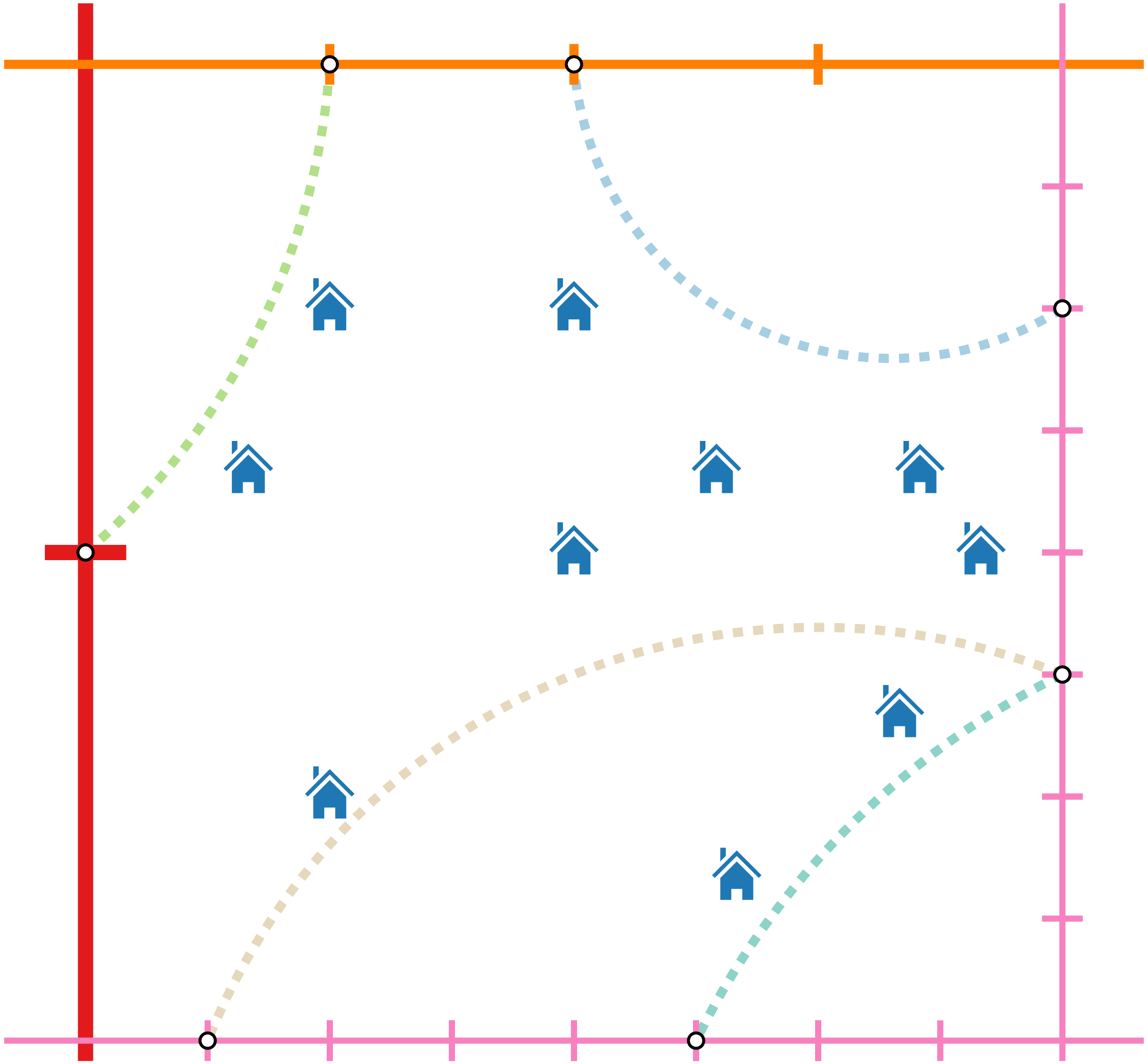
Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.



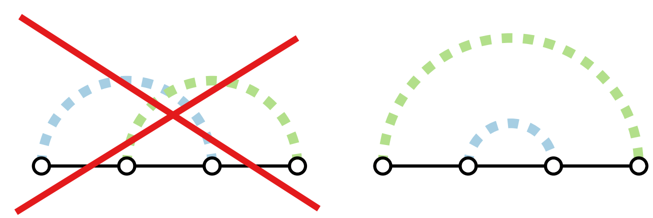
$\Rightarrow$ max. $\underbrace{\qquad}_{\#\text{visit vectors}} \times \underbrace{\qquad}_{\#\text{realizable pairings}}$

Dynamic Program (I)



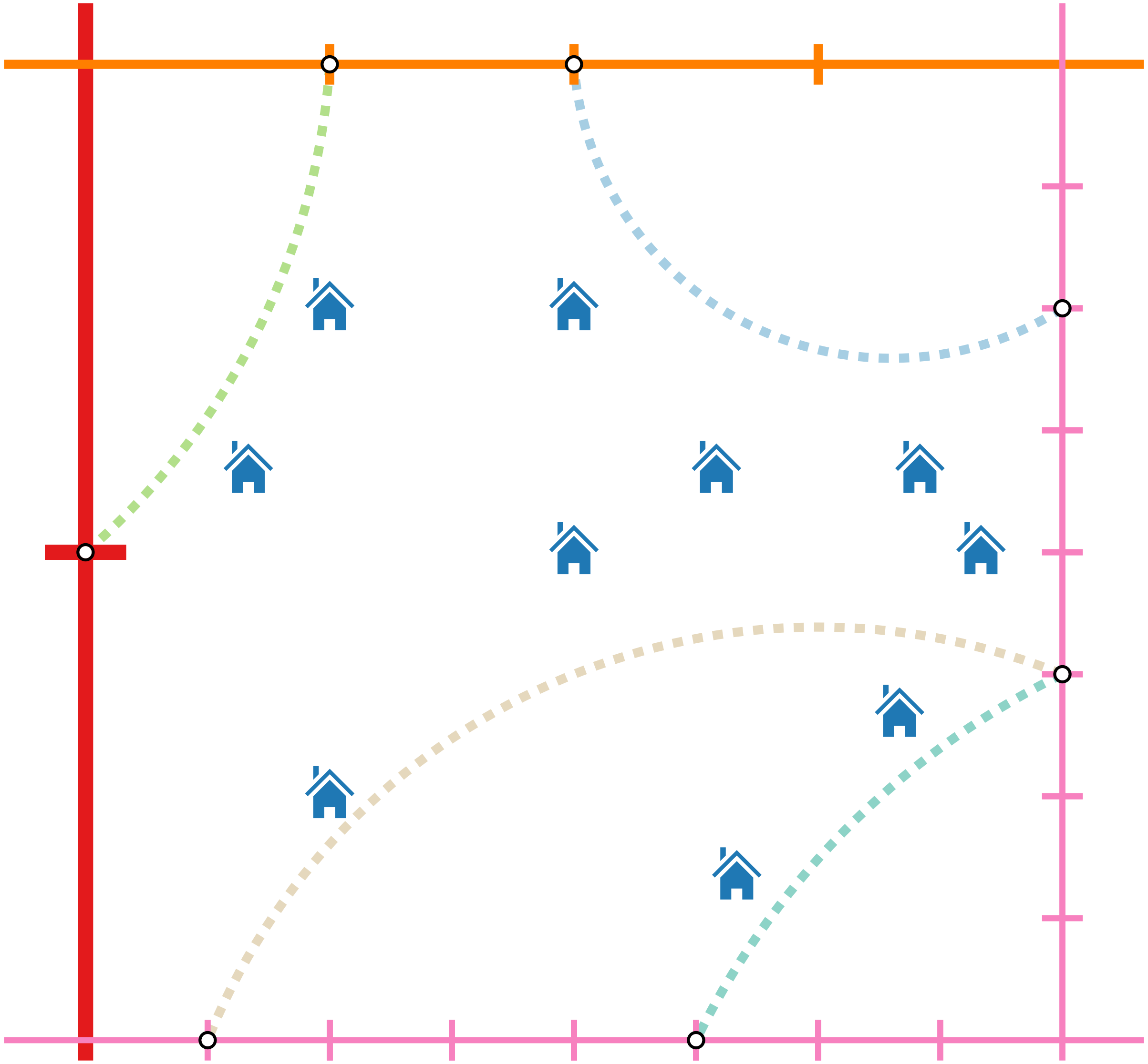
Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.



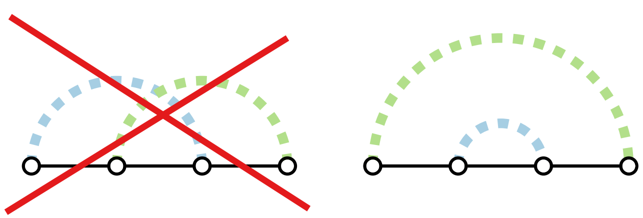
$\Rightarrow$ max. $\underbrace{n^{O(1/\varepsilon)}}_{\text{\#visit vectors}} \times \underbrace{\quad}_{\text{\#realizable pairings}}$

Dynamic Program (I)



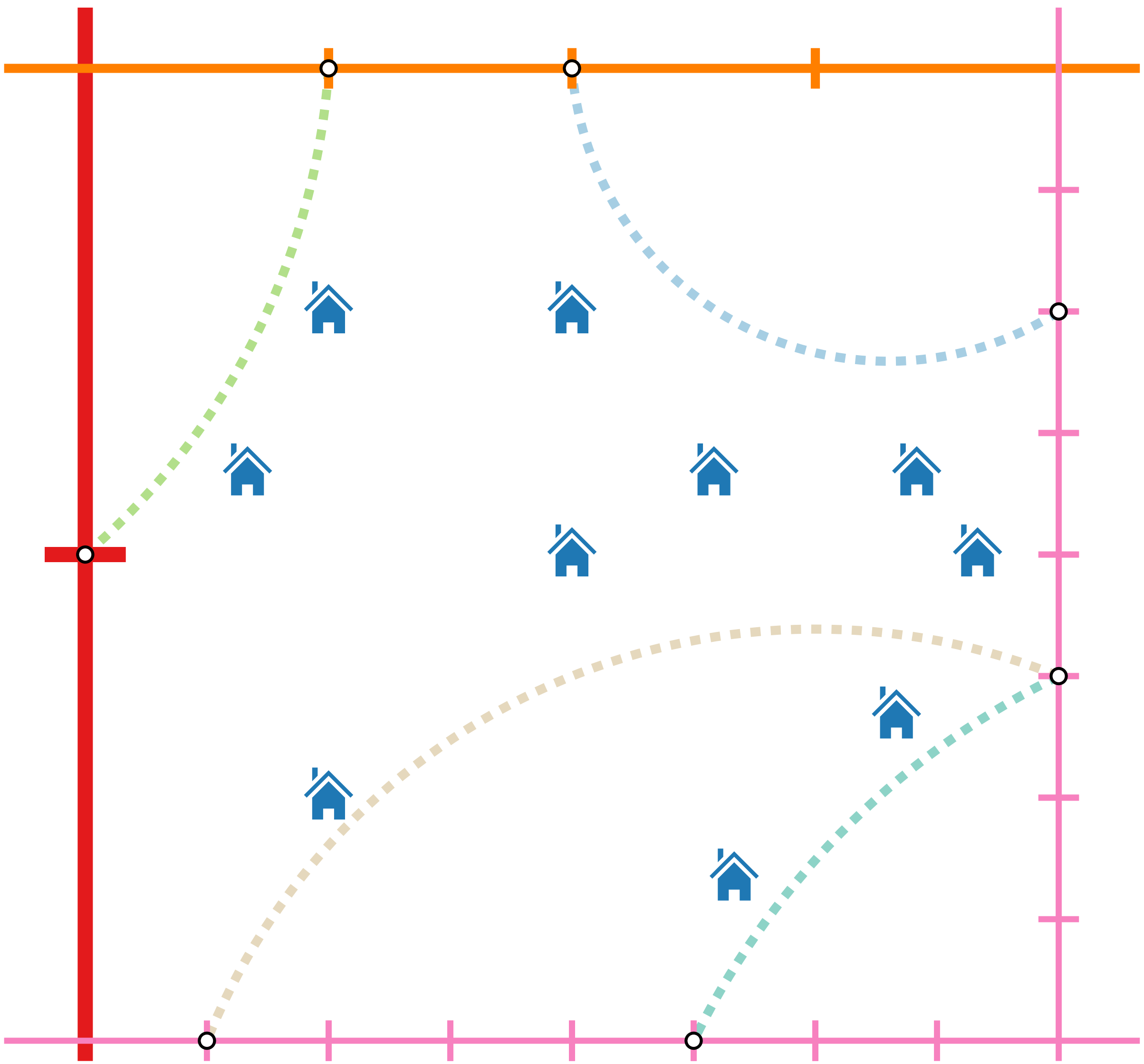
Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.



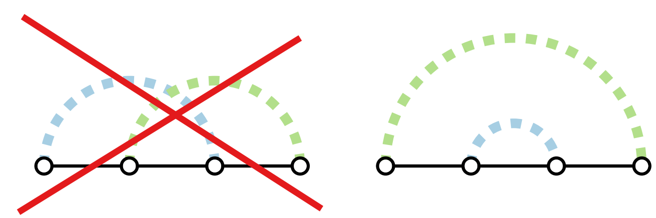
$$\Rightarrow \text{max. } \underbrace{n^{O(1/\varepsilon)}}_{\text{\#visit vectors}} \times \underbrace{2^{O(m)}}_{\text{\#realizable pairings}} =$$

Dynamic Program (I)



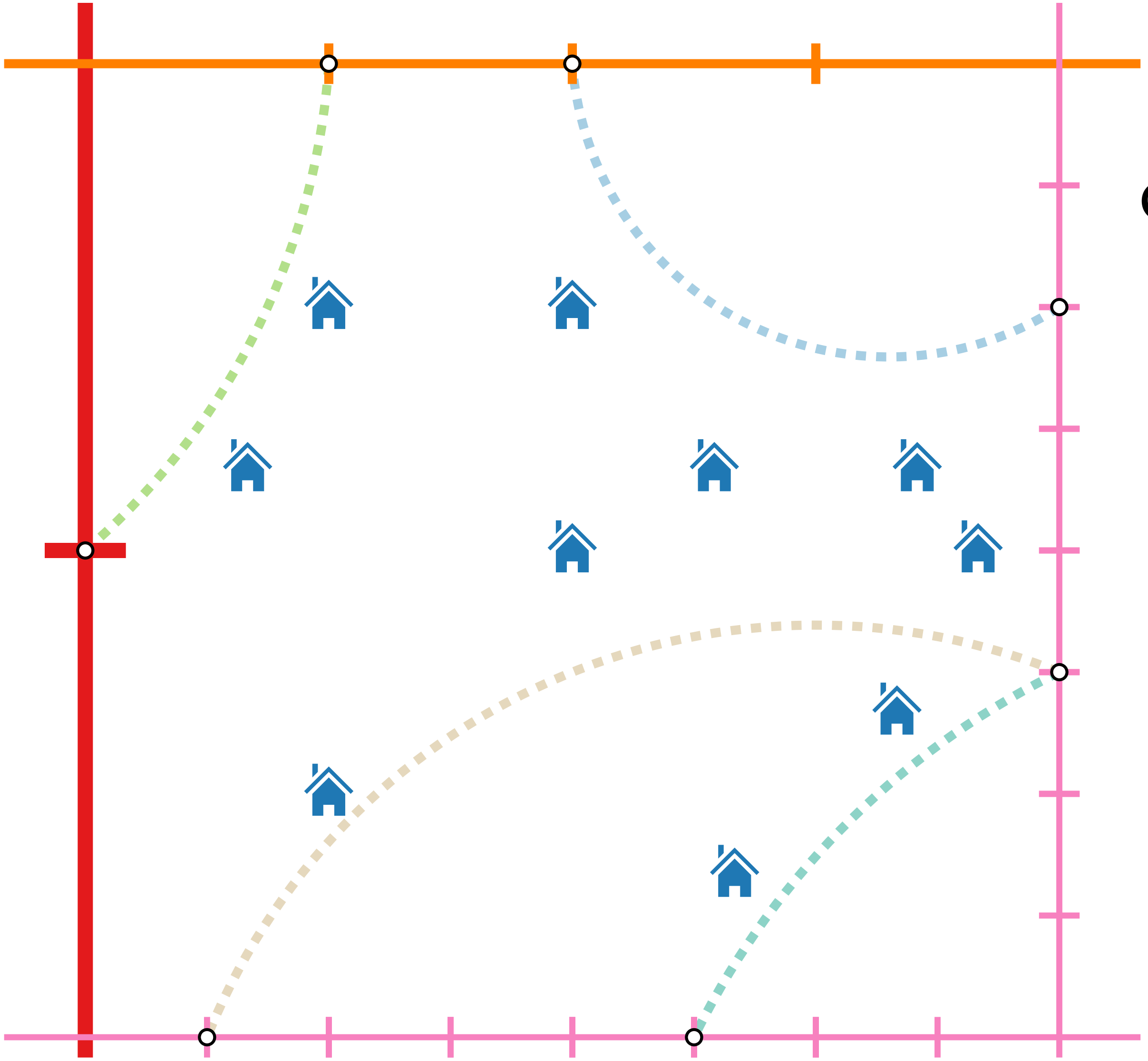
Each well-behaved tour induces the following in each square Q of the dissection:

- a path cover of the houses in Q,
- ...such that each portal of Q is visited 0, 1 or 2 times,
- a **crossing-free pairing** of the visited portals.



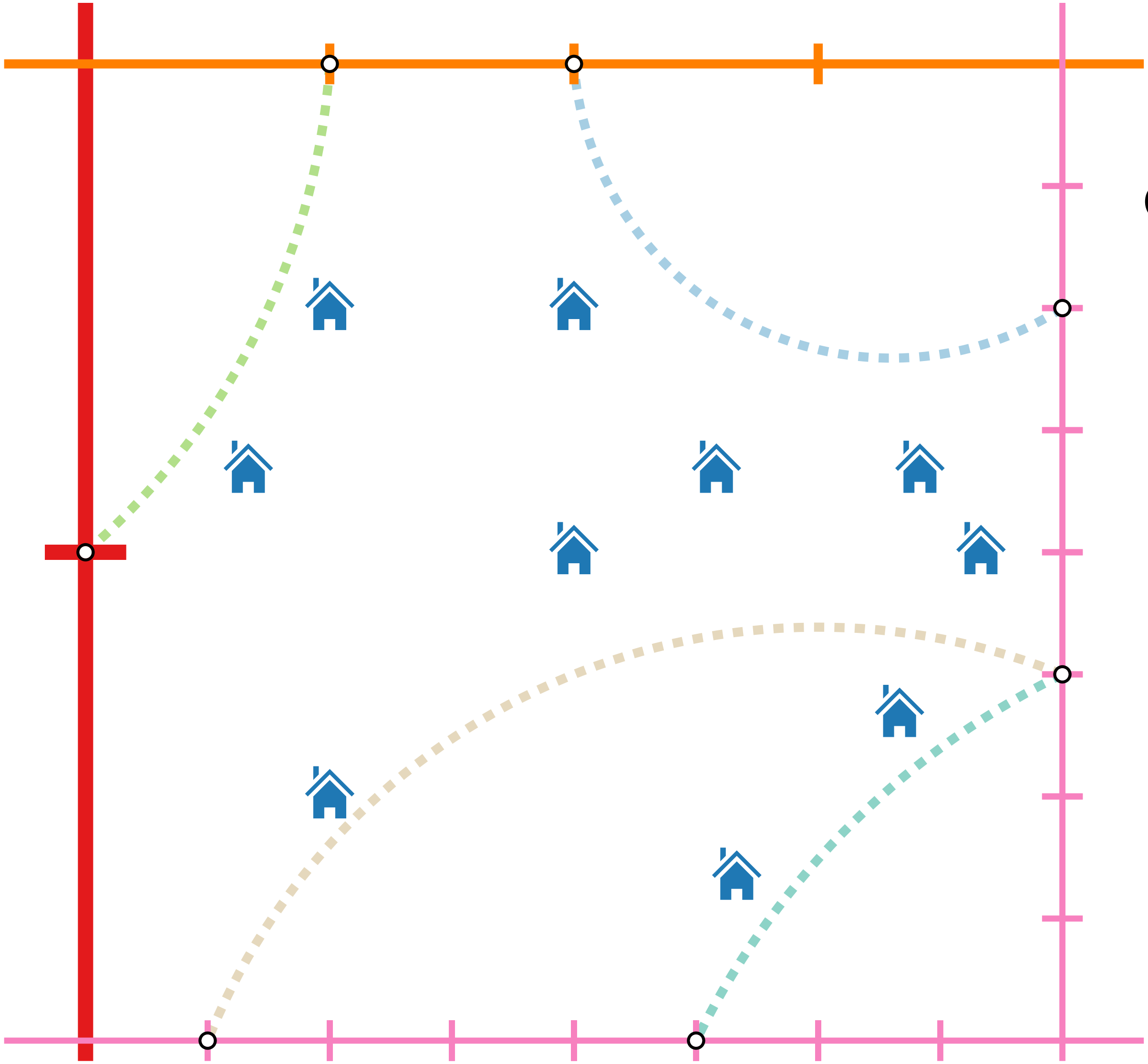
$$\Rightarrow \text{max. } \underbrace{n^{O(1/\varepsilon)}}_{\text{\#visit vectors}} \times \underbrace{2^{O(m)}}_{\text{\#realizable pairings}} = n^{O(1/\varepsilon)} \text{ crossing-free pairings}$$

Dynamic Program (II)



Compute

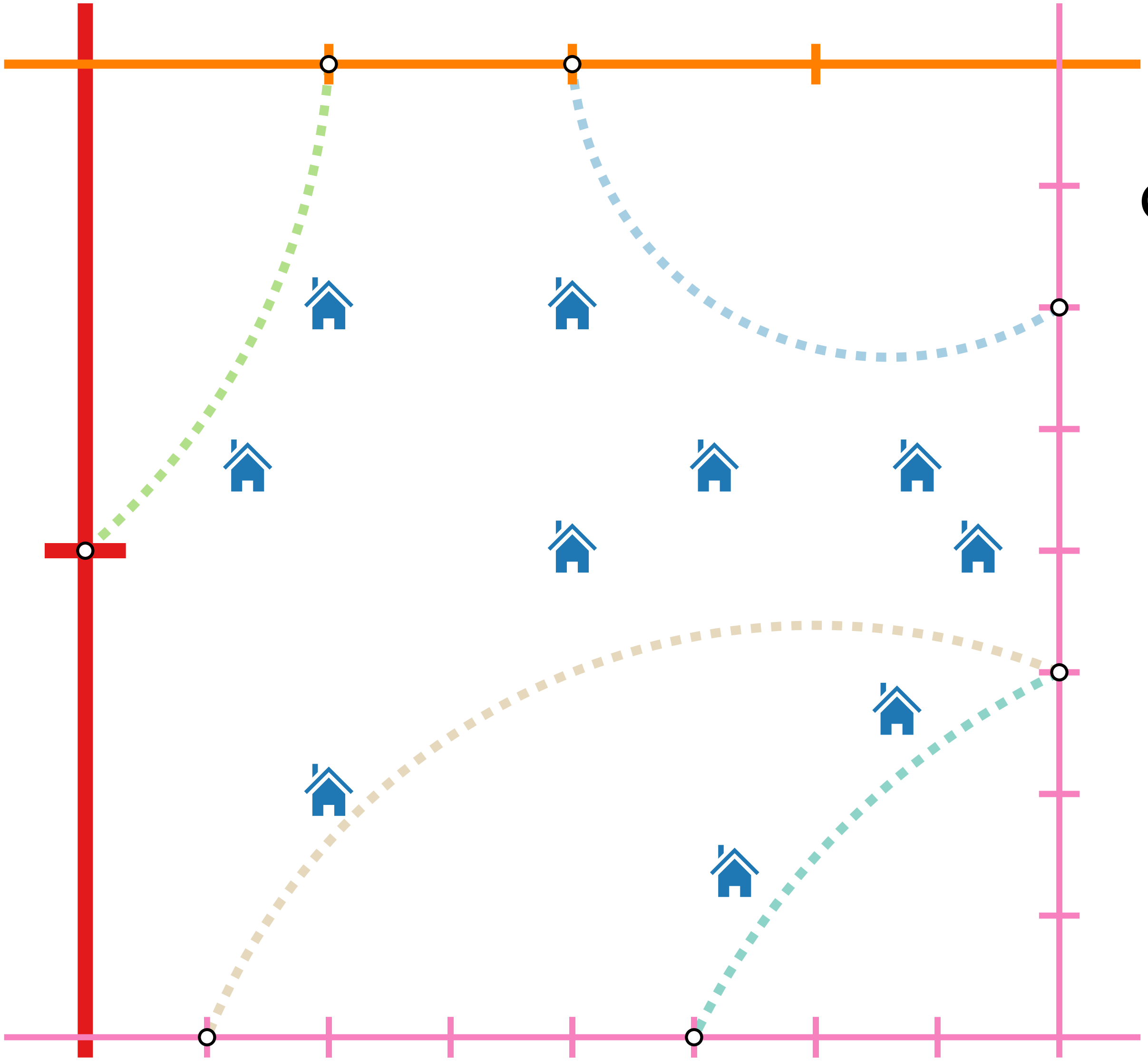
Dynamic Program (II)



Compute

- for each square Q in the dissection and

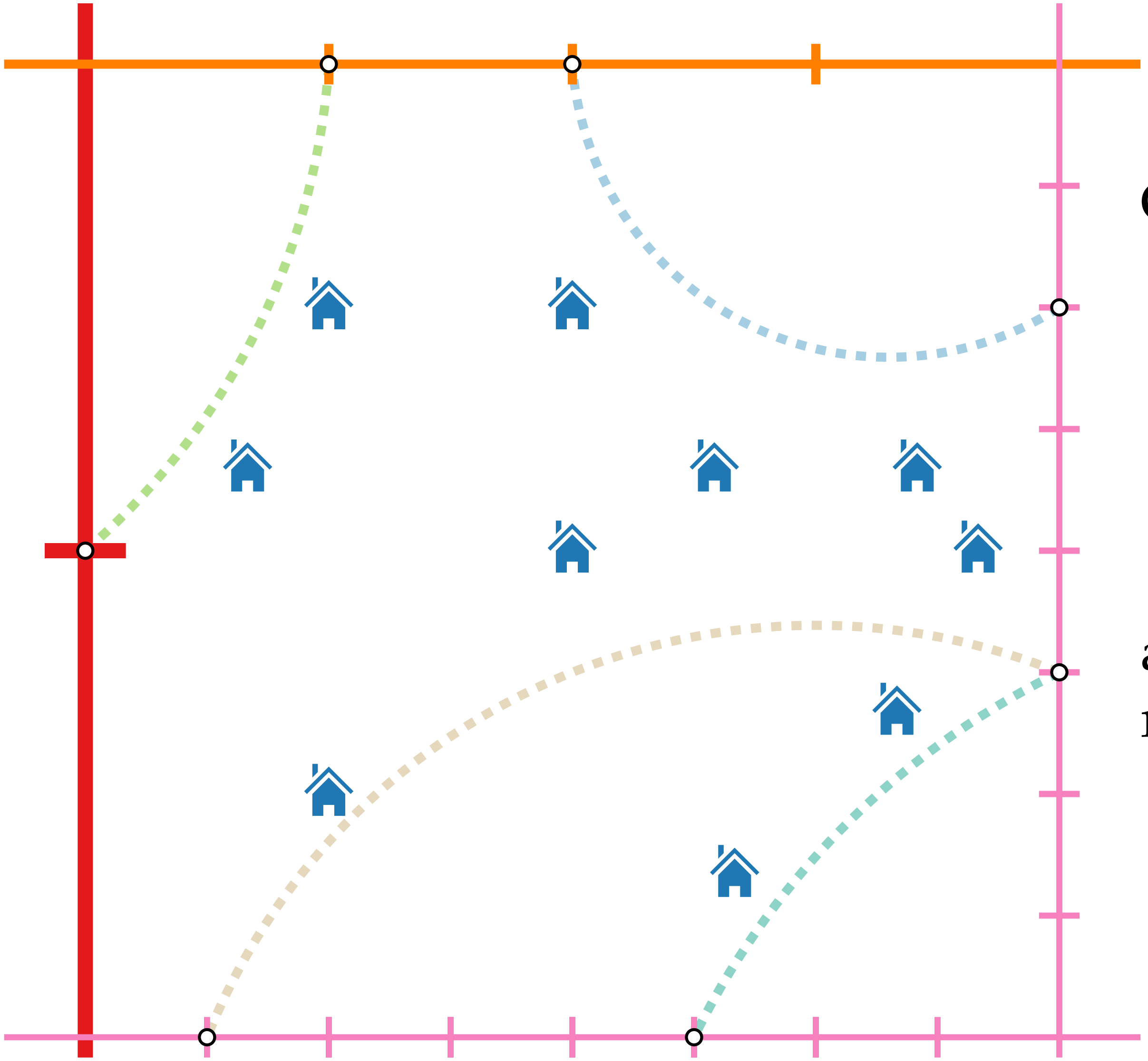
Dynamic Program (II)



Compute

- for each square Q in the dissection and
- for each crossing-free pairing P of Q,

Dynamic Program (II)

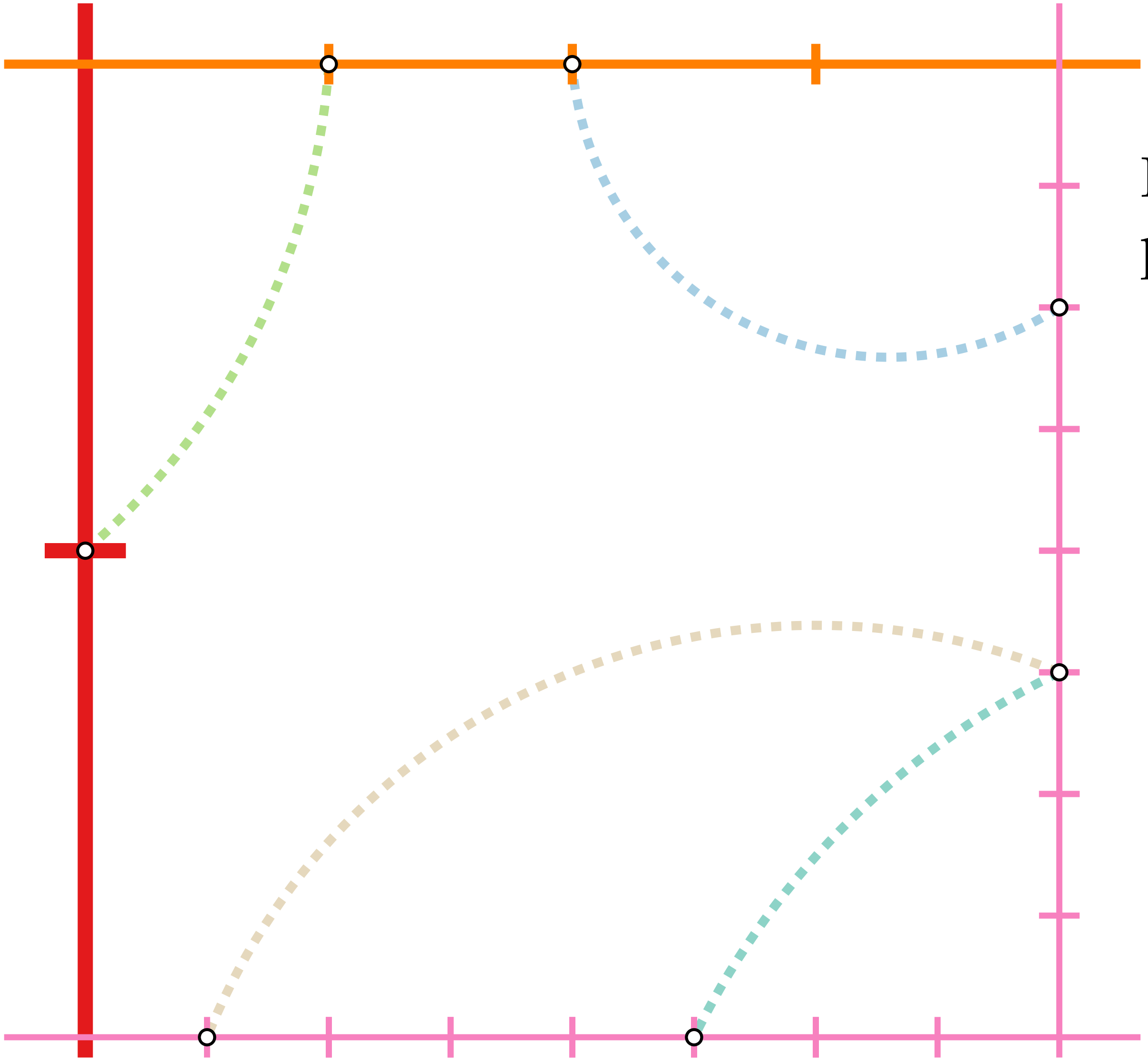


Compute

- for each square Q in the dissection and
- for each crossing-free pairing P of Q,

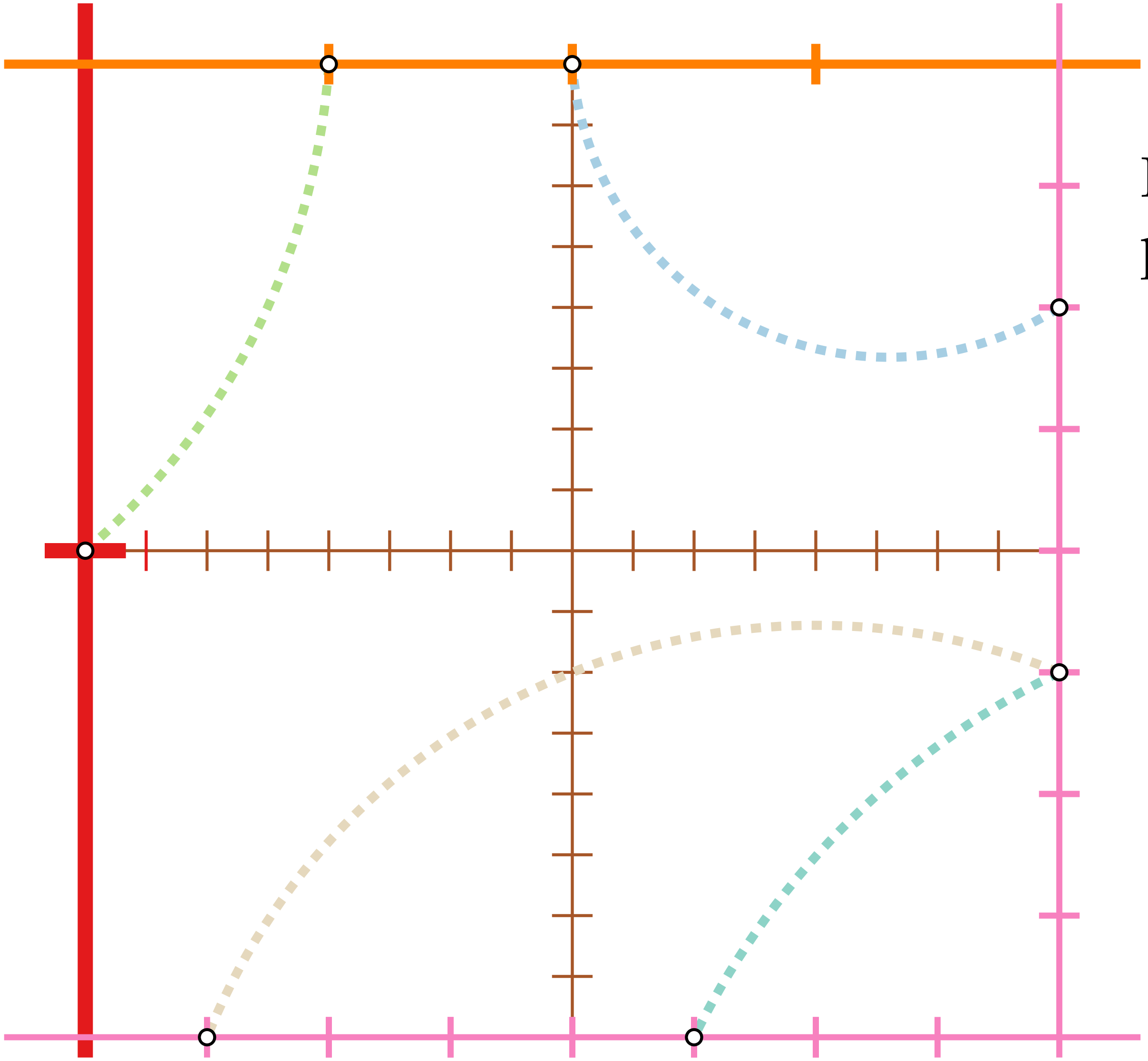
an optimal path cover that respects P.

Dynamic Program (III)



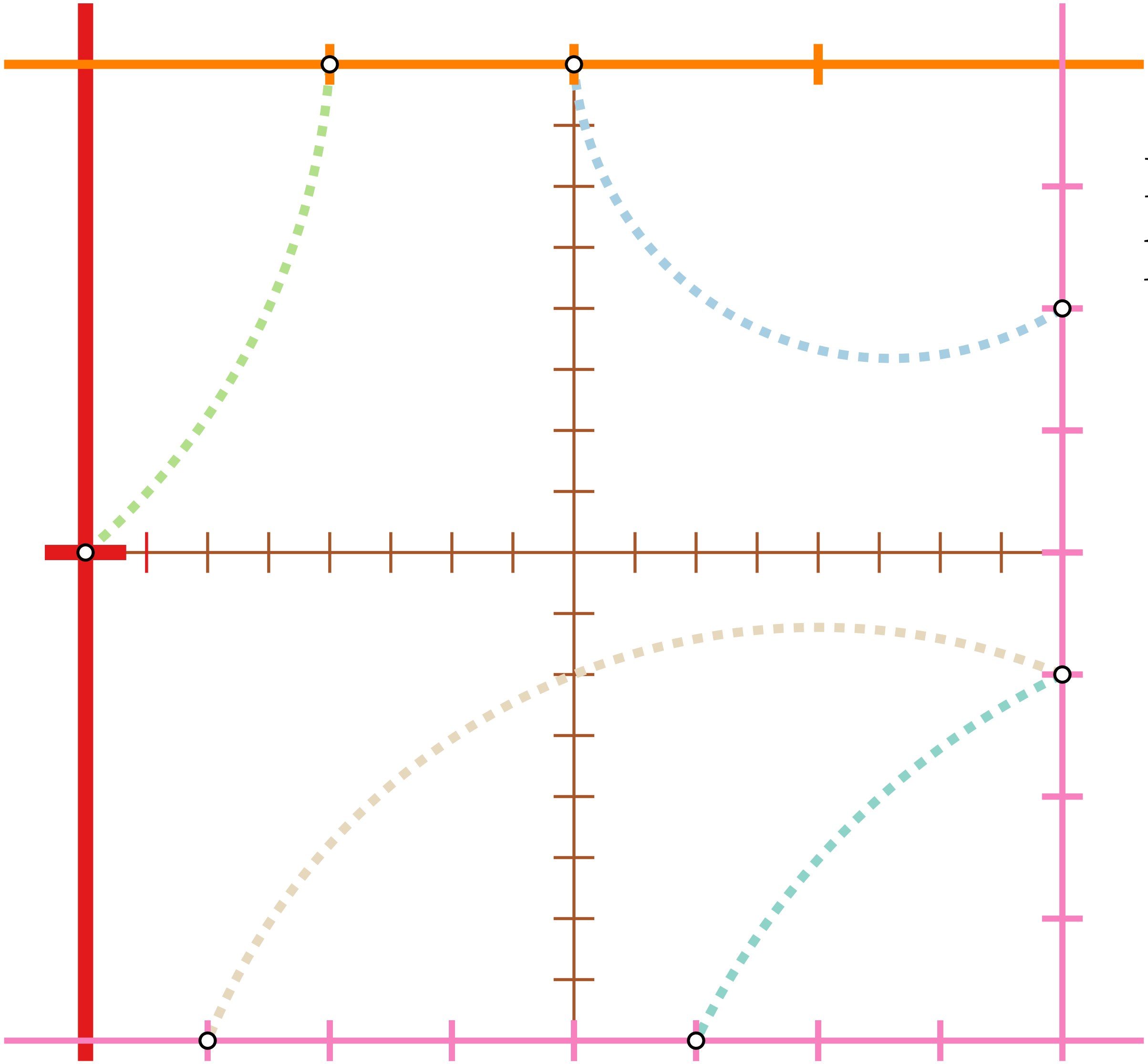
For a given square Q and pairing P:

Dynamic Program (III)



For a given square Q and pairing P:

Dynamic Program (III)

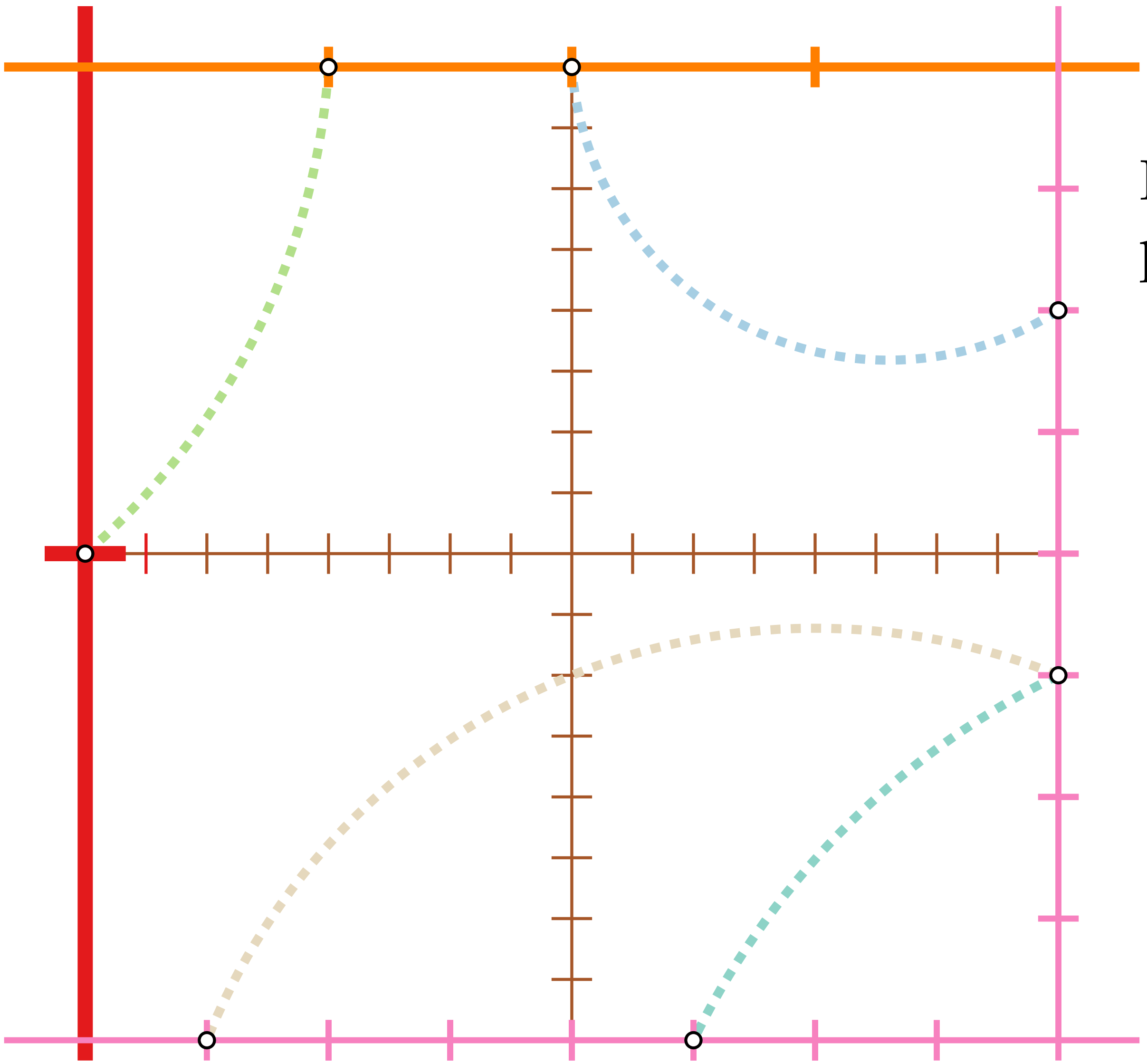


For a given square Q and pairing P:

- Iterate over all

  crossing-free pairings of the child squares.

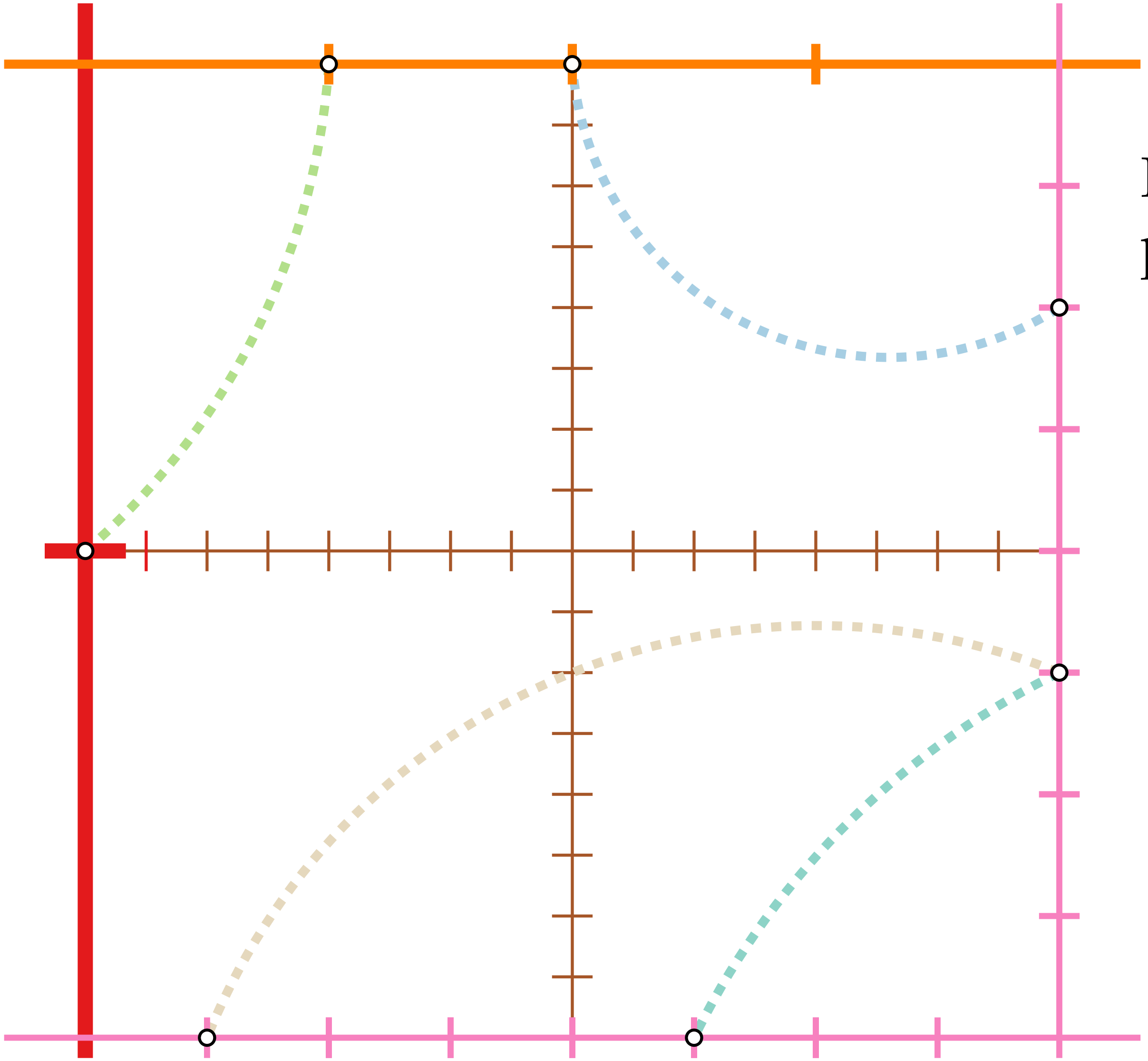
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 =$ crossing-free pairings of the child squares.

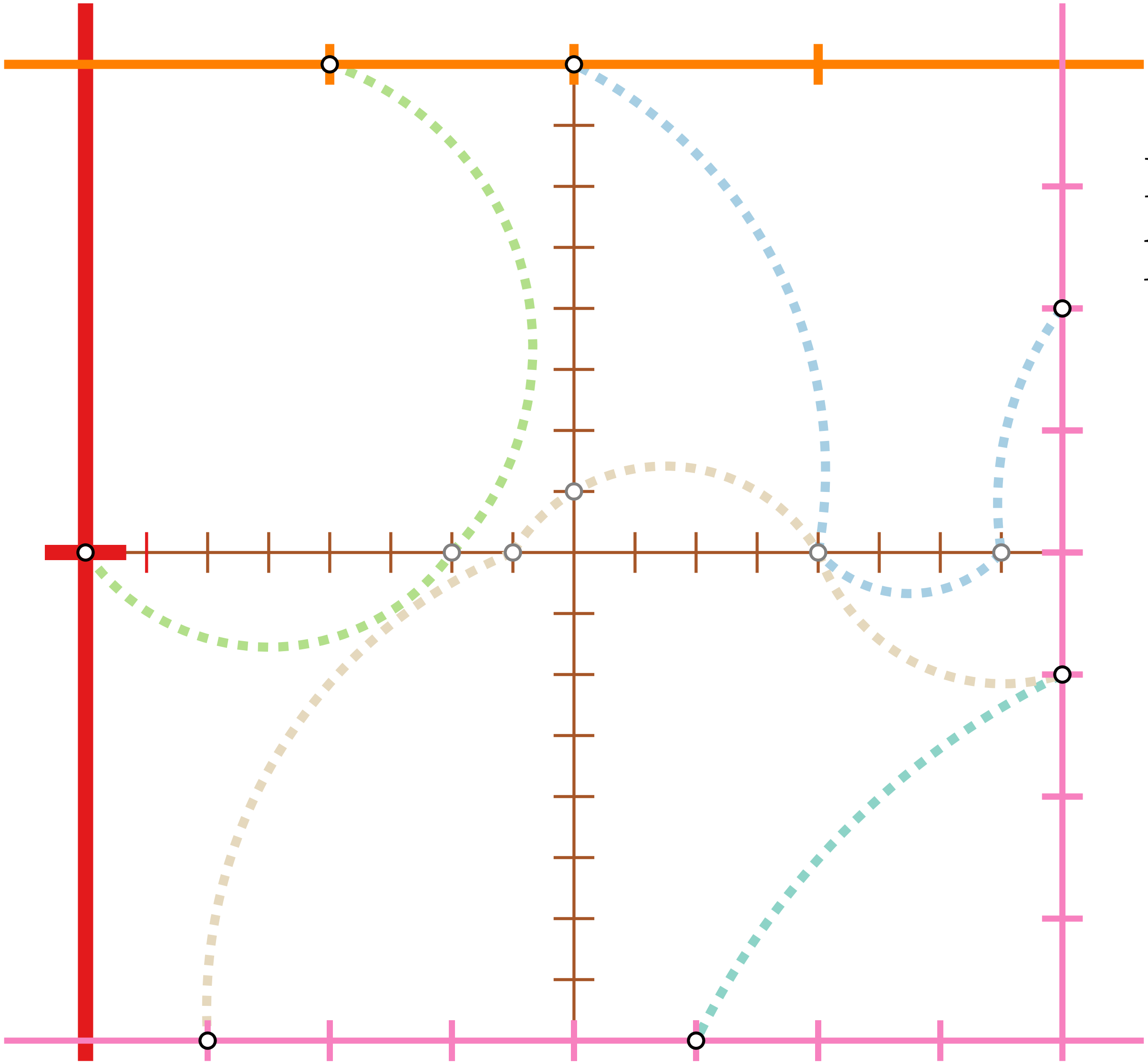
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.

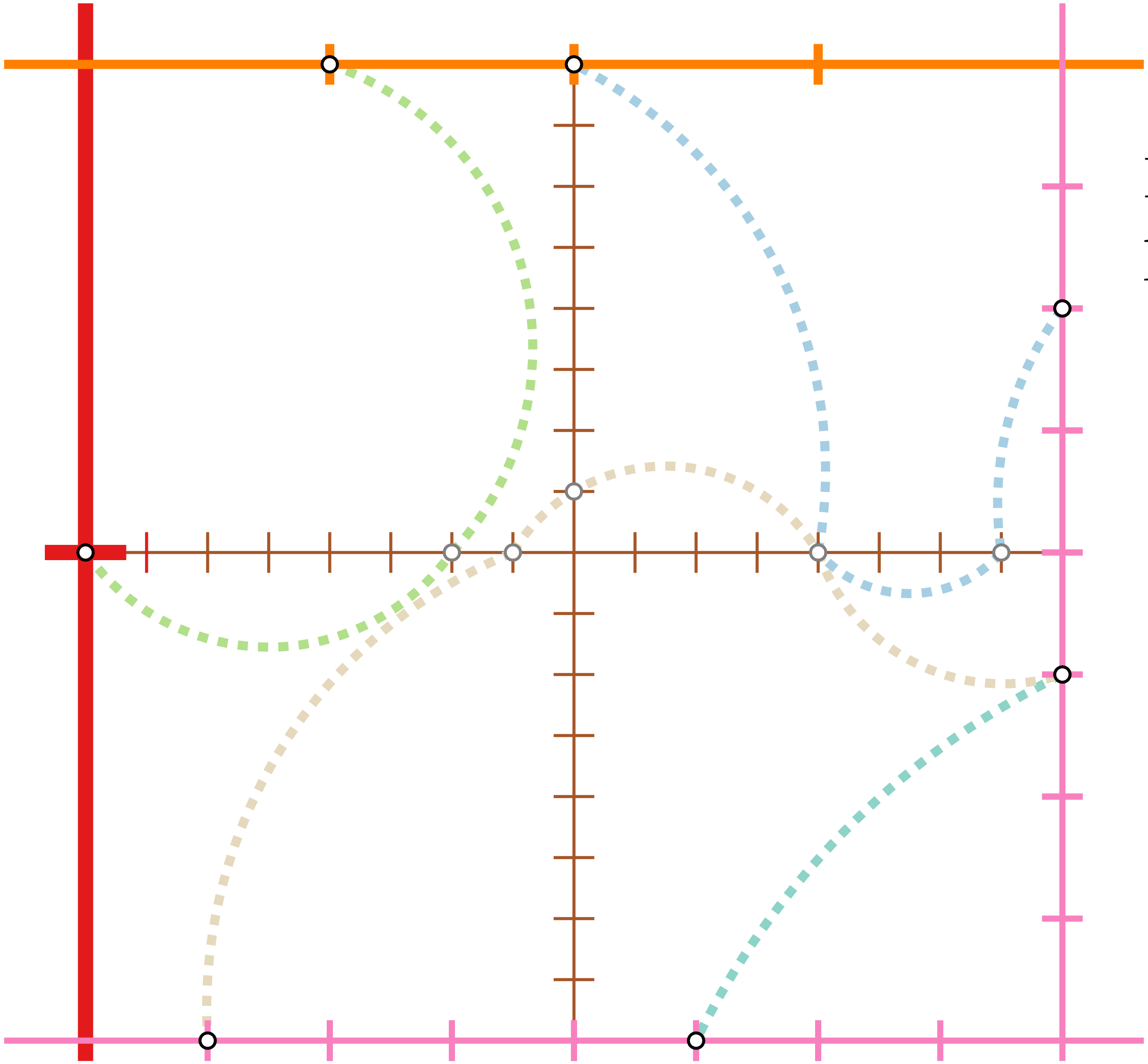
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.

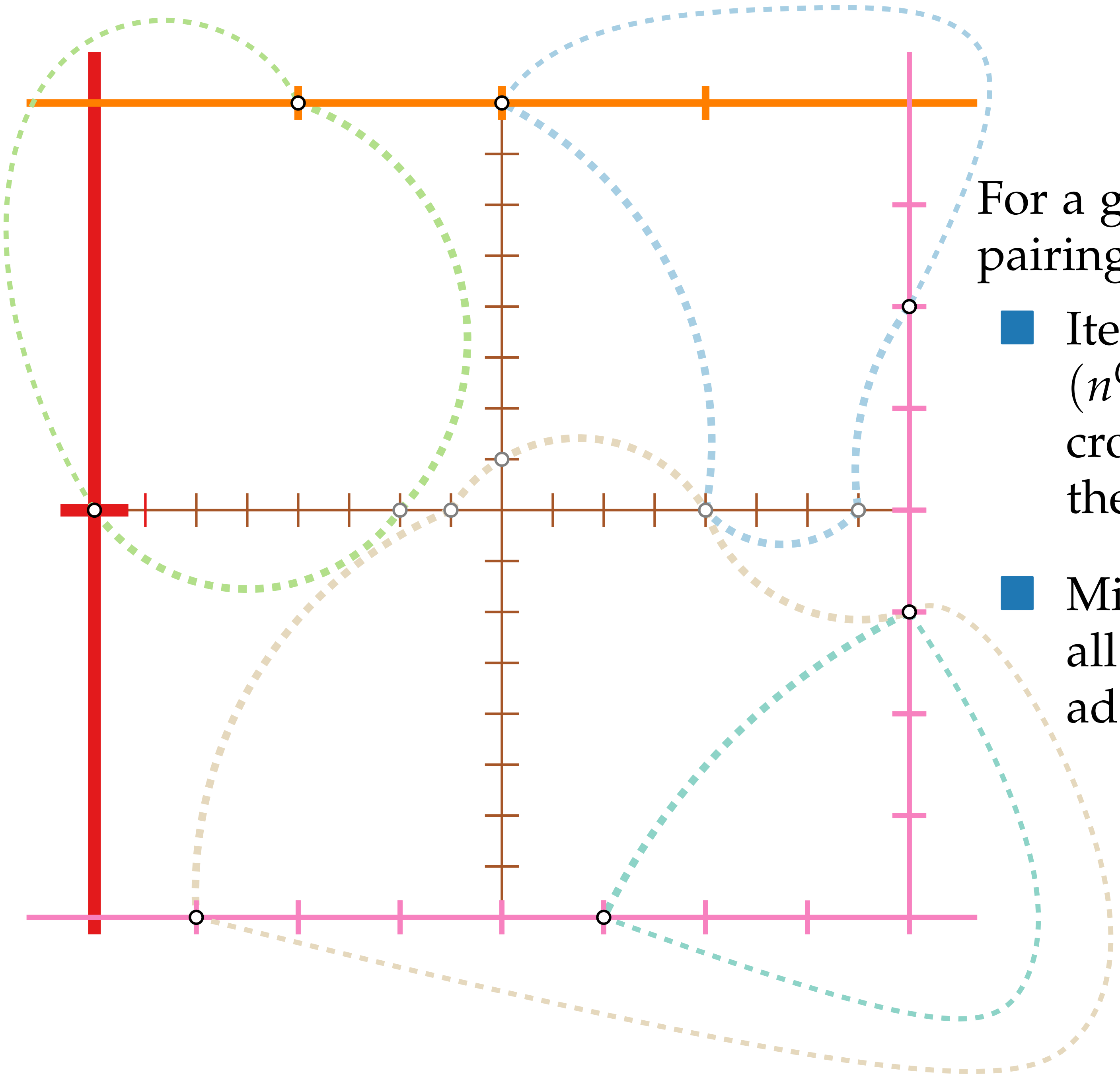
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.
- Minimize the cost over all such pairings that additionally respect P.

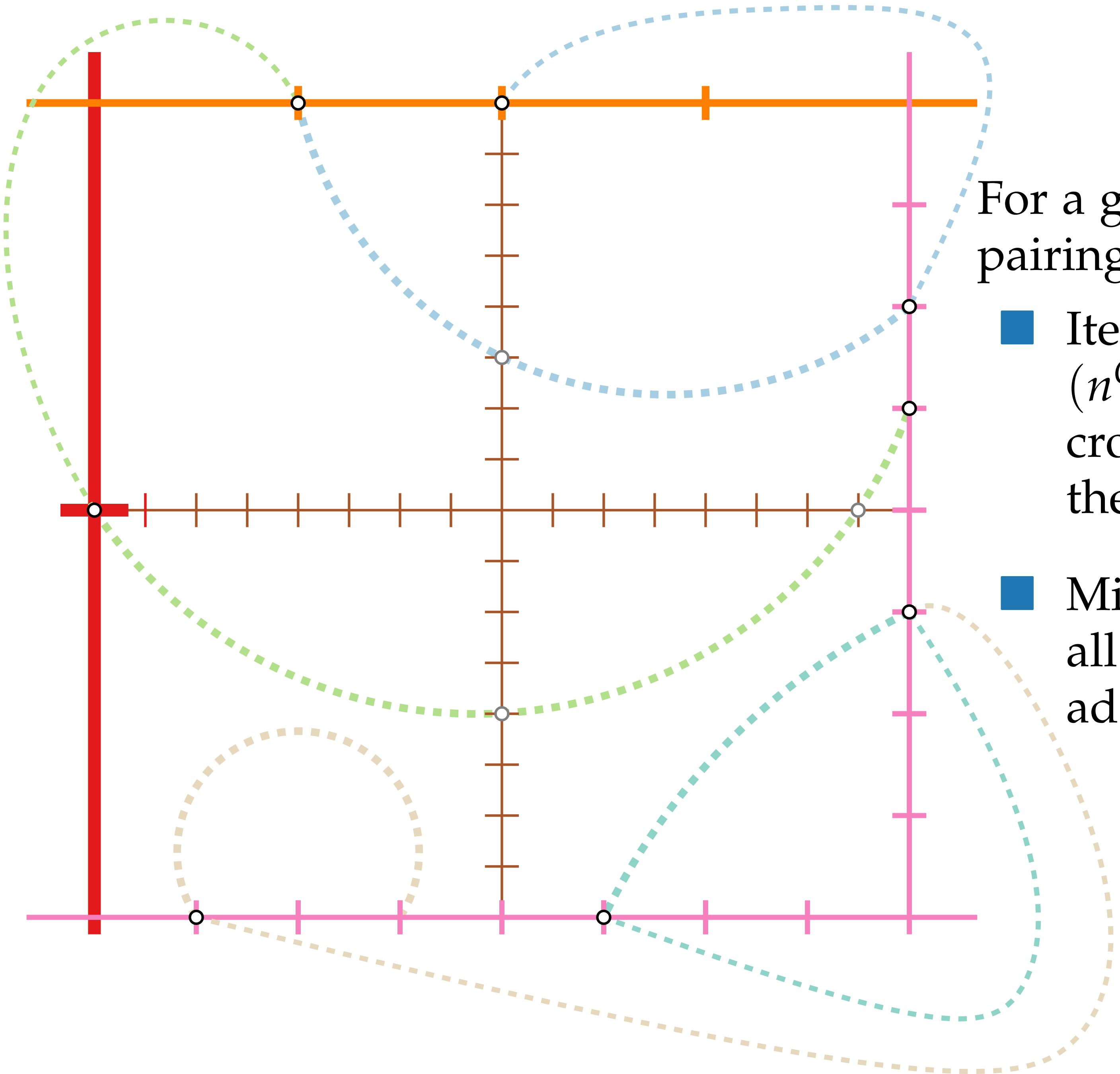
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.
- Minimize the cost over all such pairings that additionally respect P.

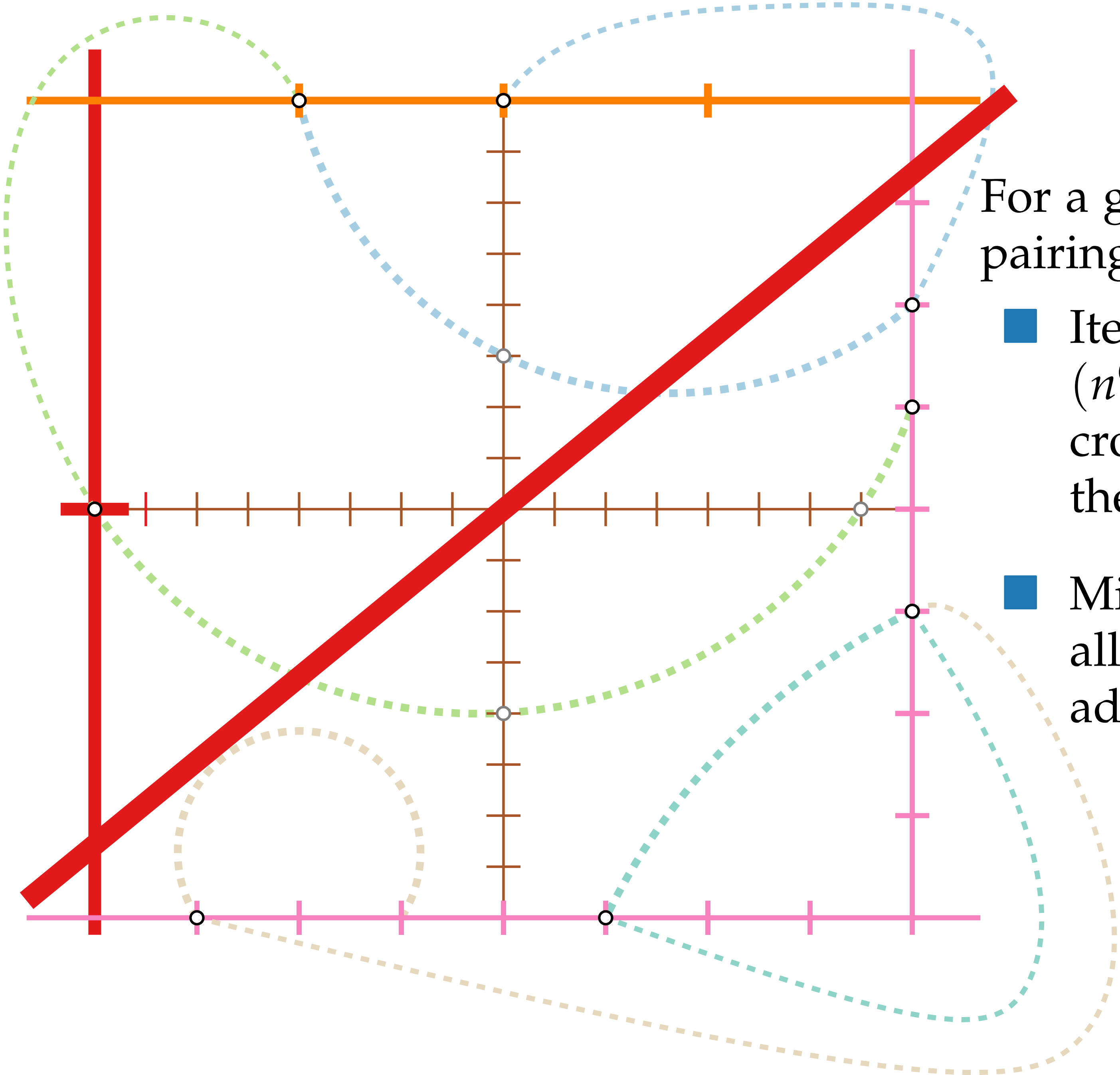
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.
- Minimize the cost over all such pairings that additionally respect P.

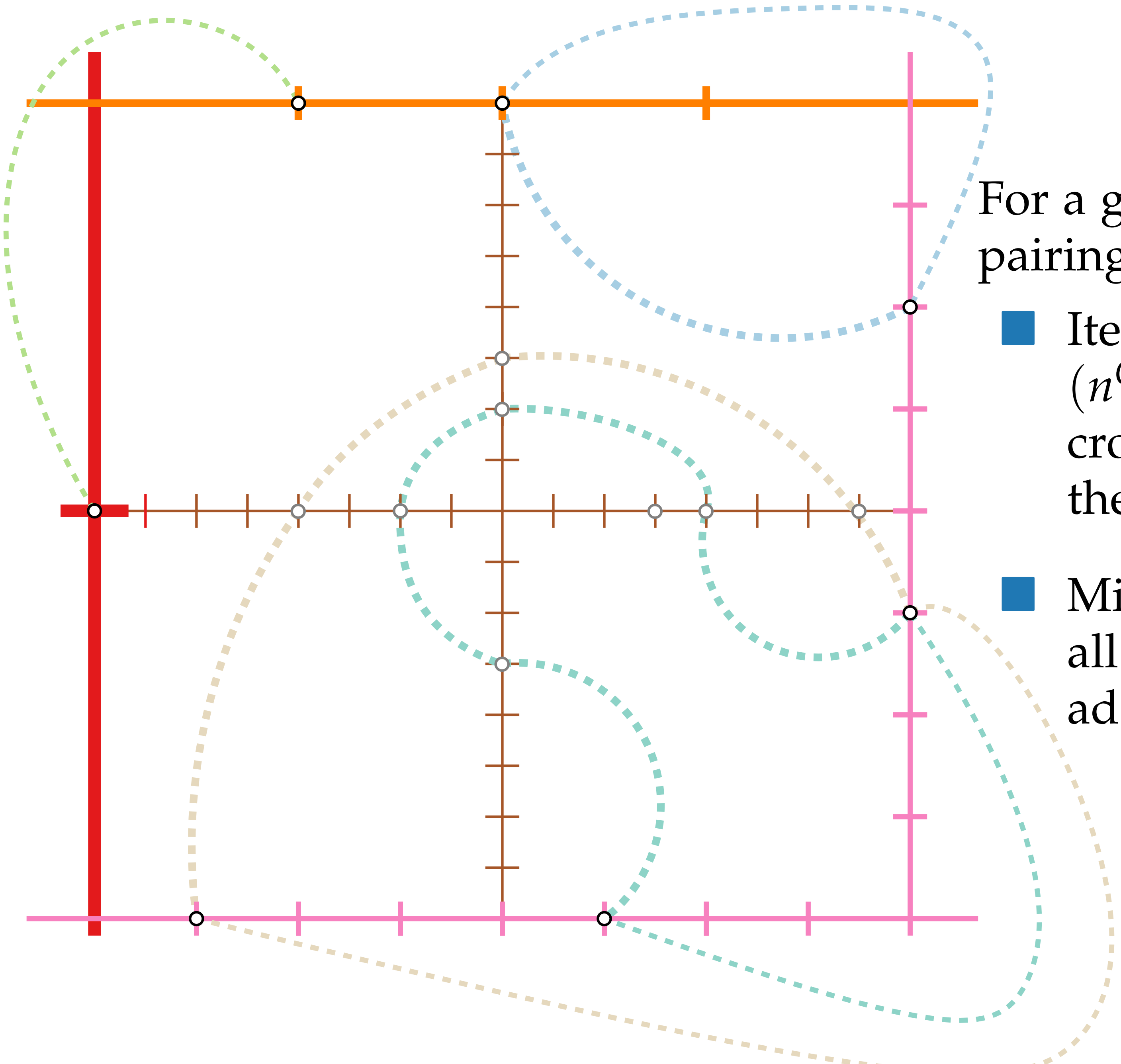
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.
- Minimize the cost over all such pairings that additionally respect P.

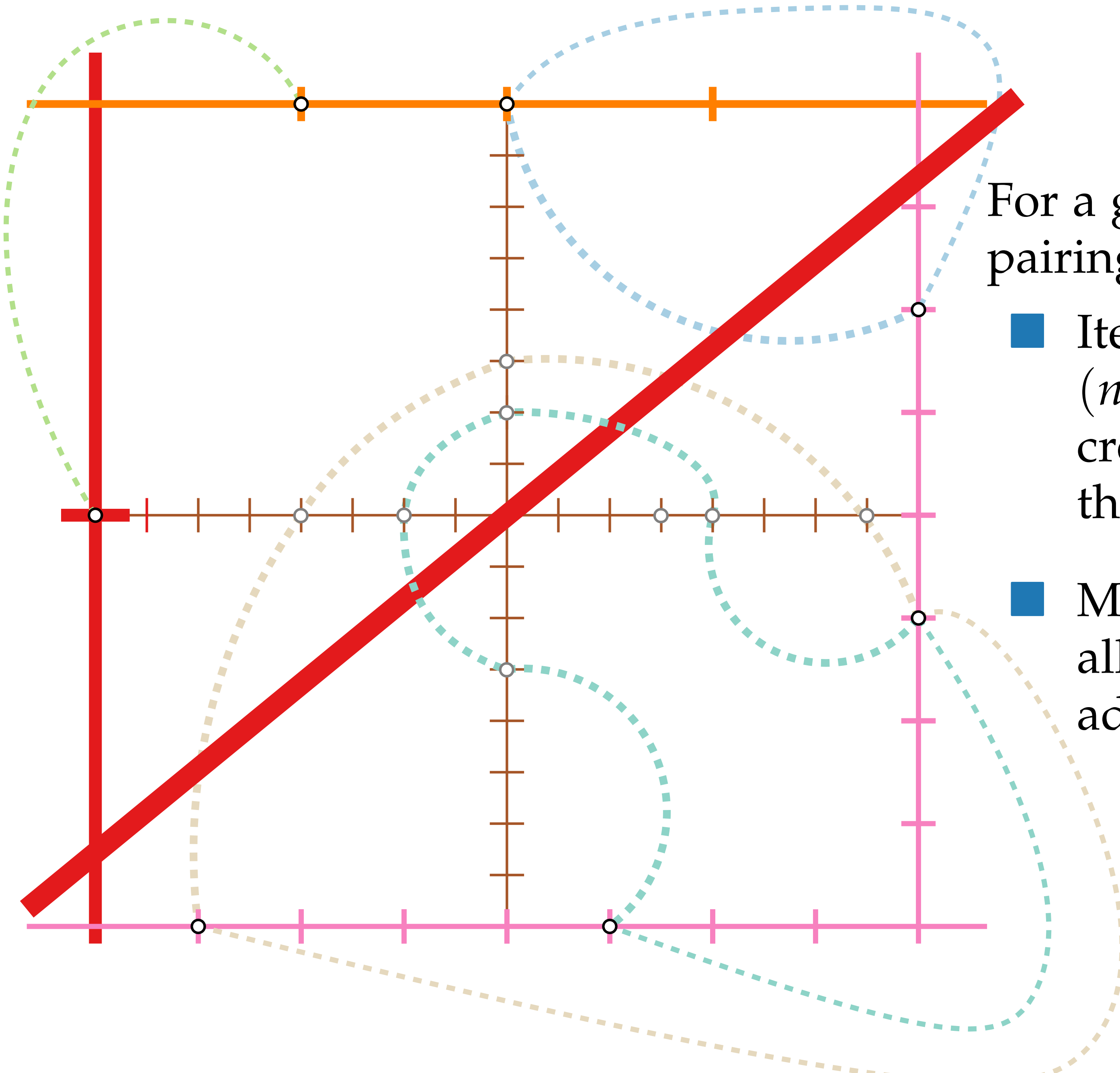
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.
- Minimize the cost over all such pairings that additionally respect P.

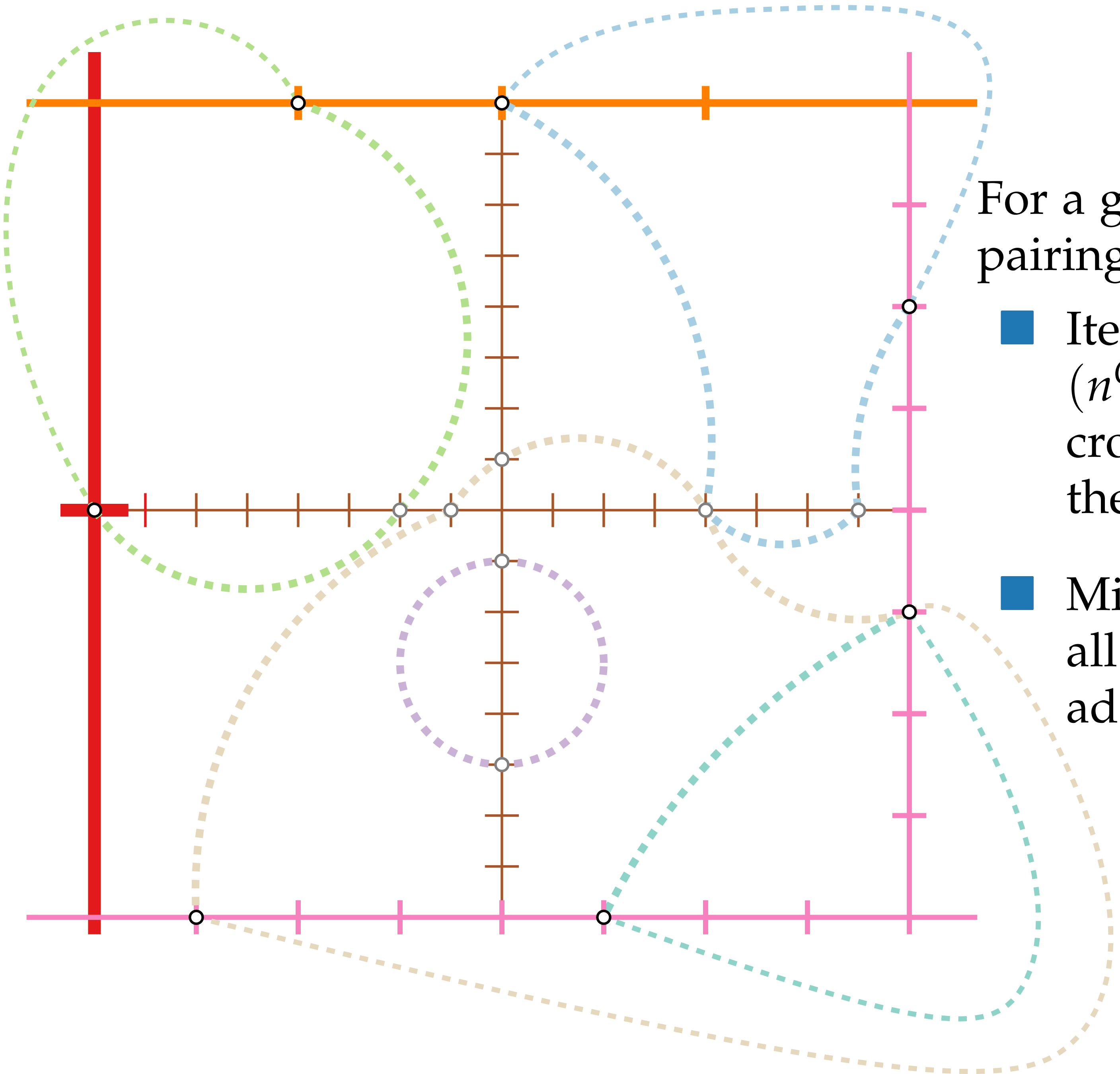
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.
- Minimize the cost over all such pairings that additionally respect P.

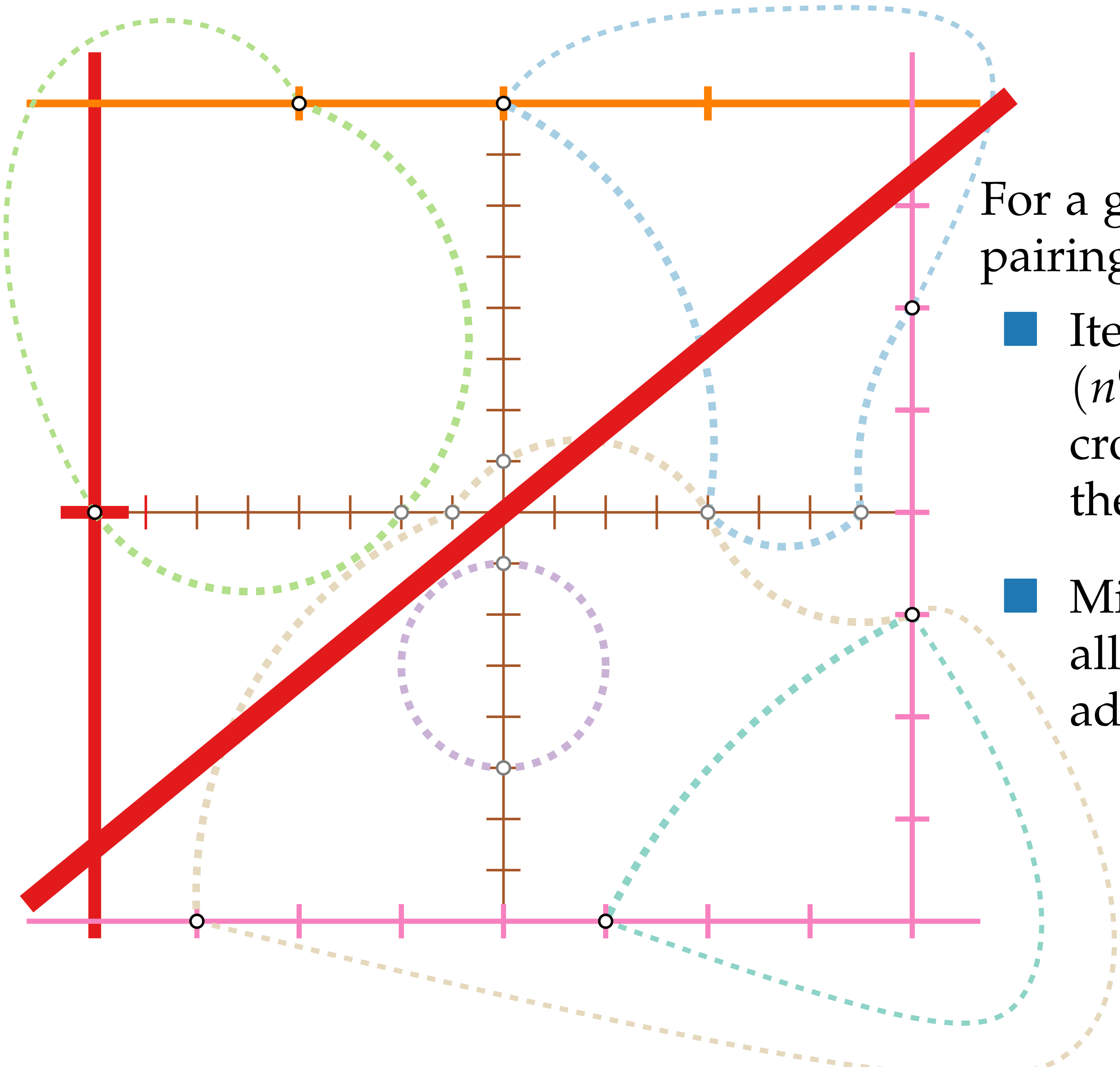
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.
- Minimize the cost over all such pairings that additionally respect P.

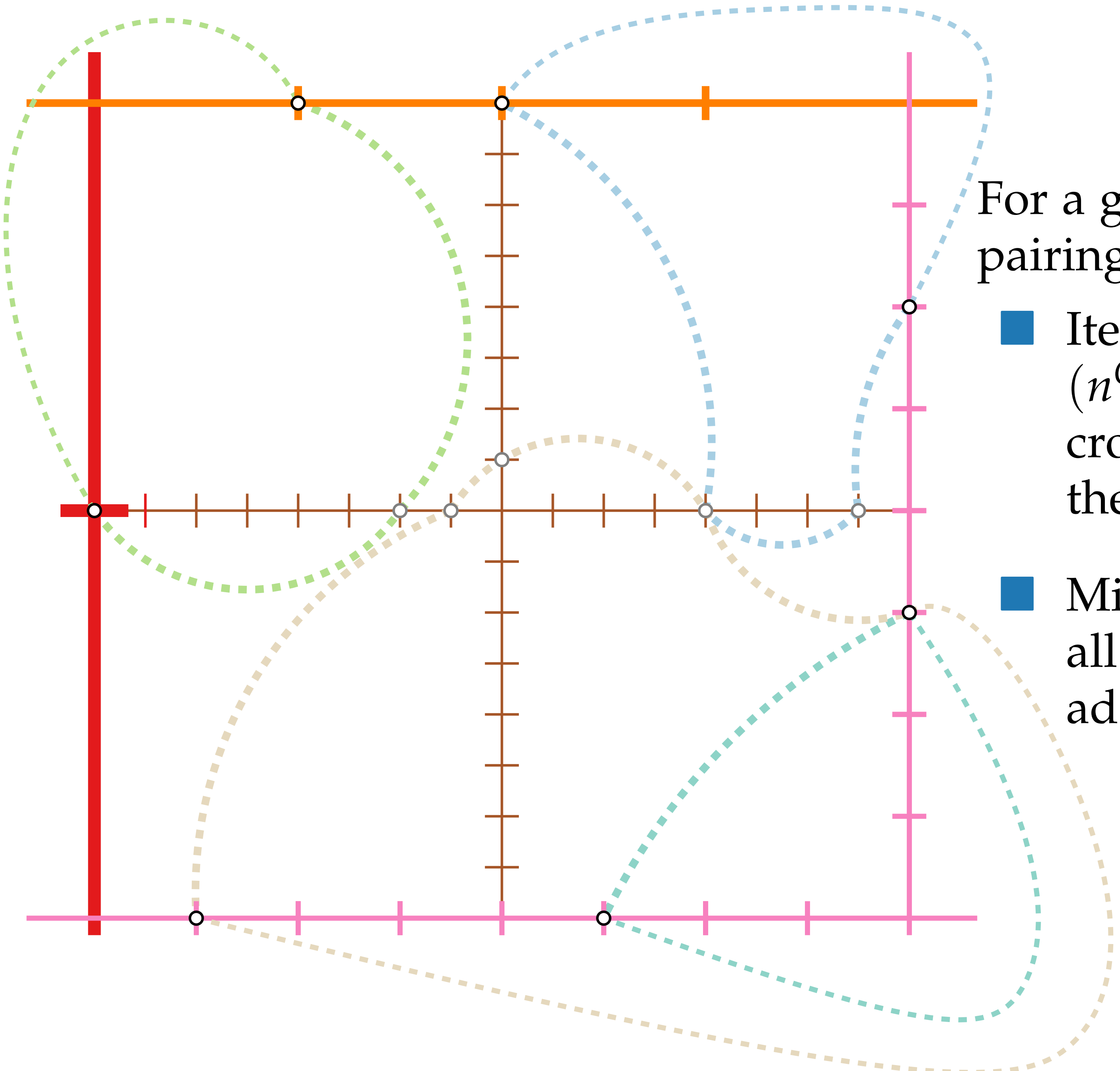
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.
- Minimize the cost over all such pairings that additionally respect P.

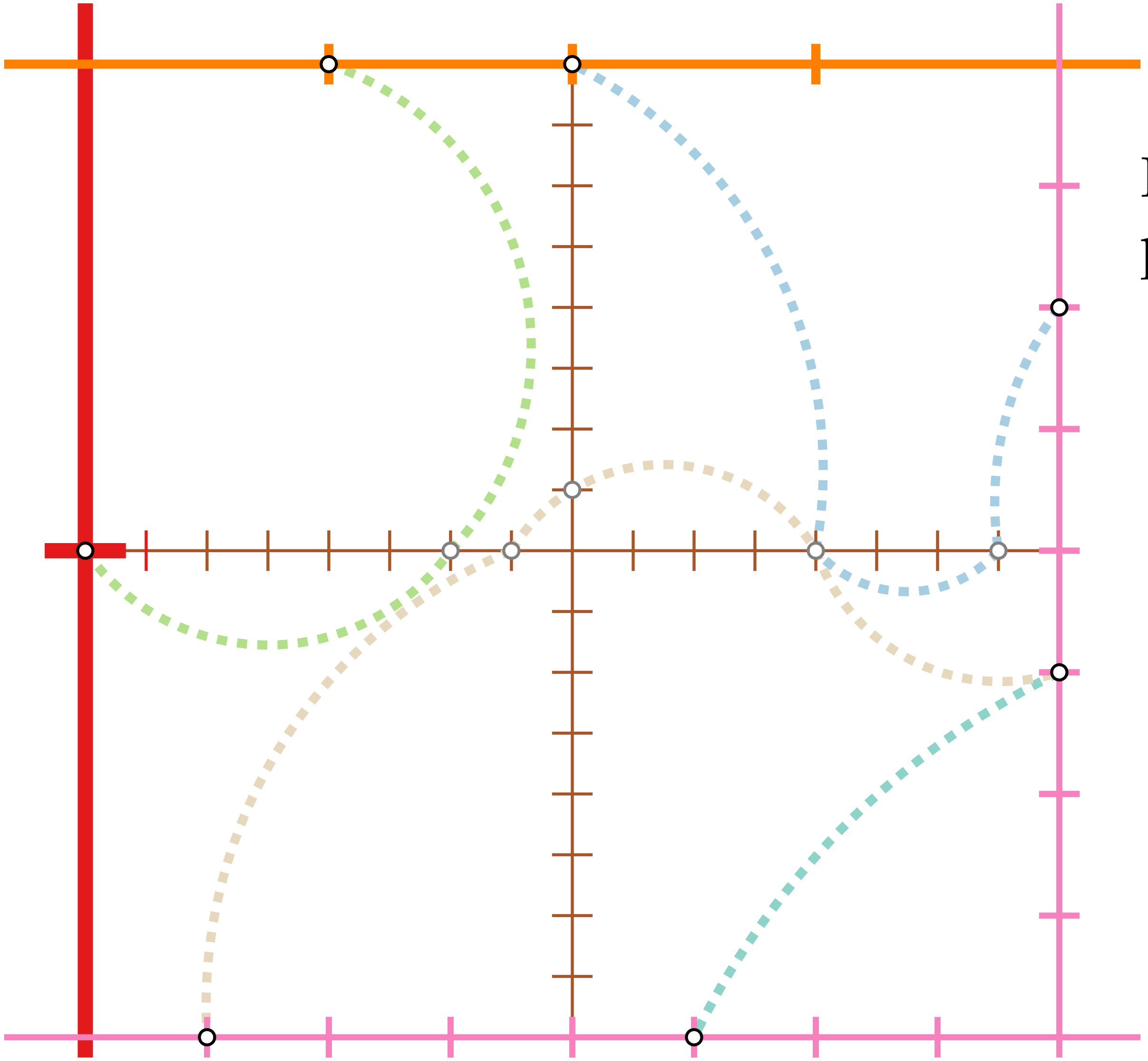
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.
- Minimize the cost over all such pairings that additionally respect P.

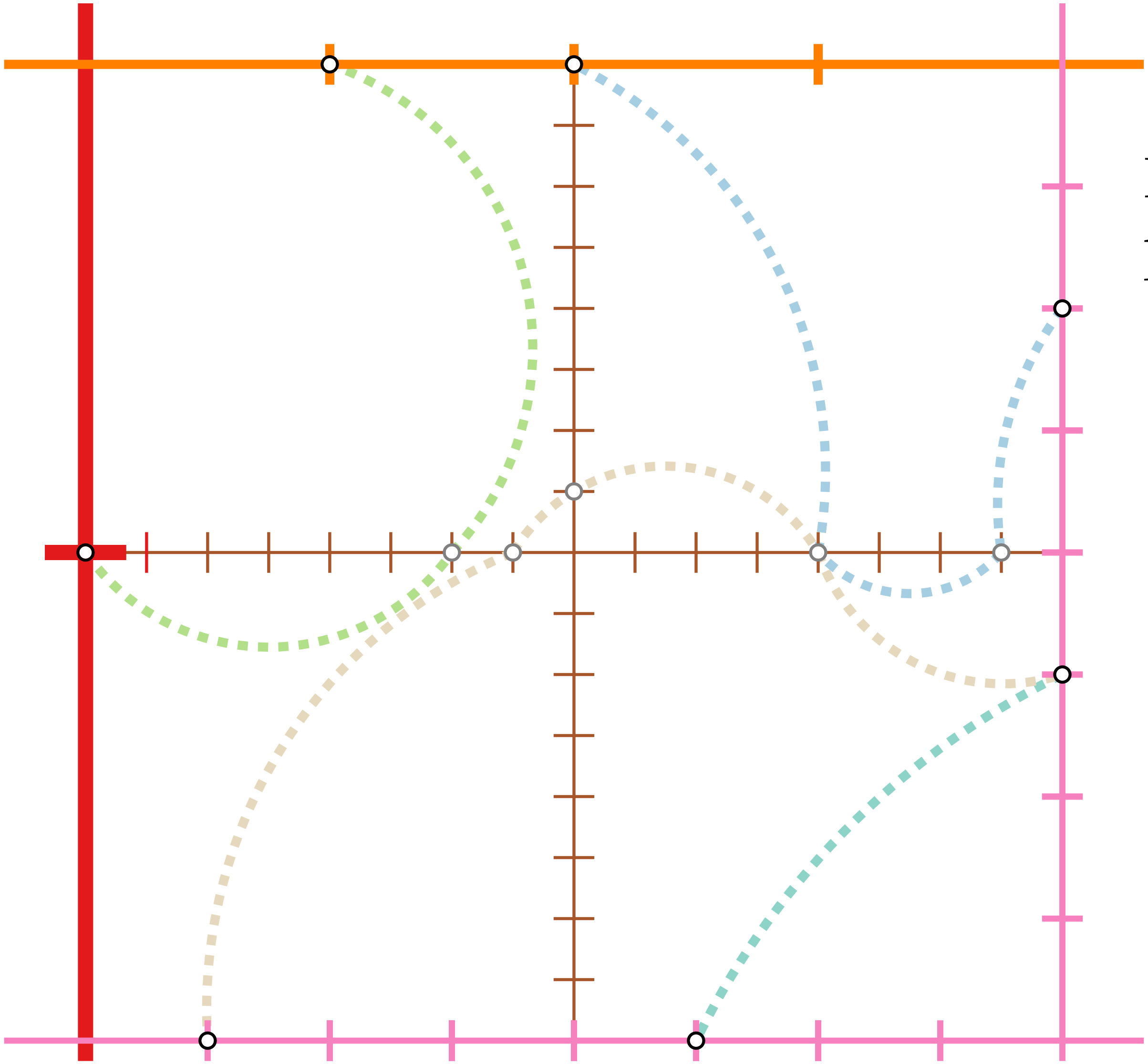
Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.
- Minimize the cost over all such pairings that additionally respect P.
- Correctness follows by induction.

Dynamic Program (III)



For a given square Q and pairing P:

- Iterate over all $(n^{O(1/\varepsilon)})^4 = n^{O(1/\varepsilon)}$ crossing-free pairings of the child squares.
- Minimize the cost over all such pairings that additionally respect P.
- Correctness follows by induction.

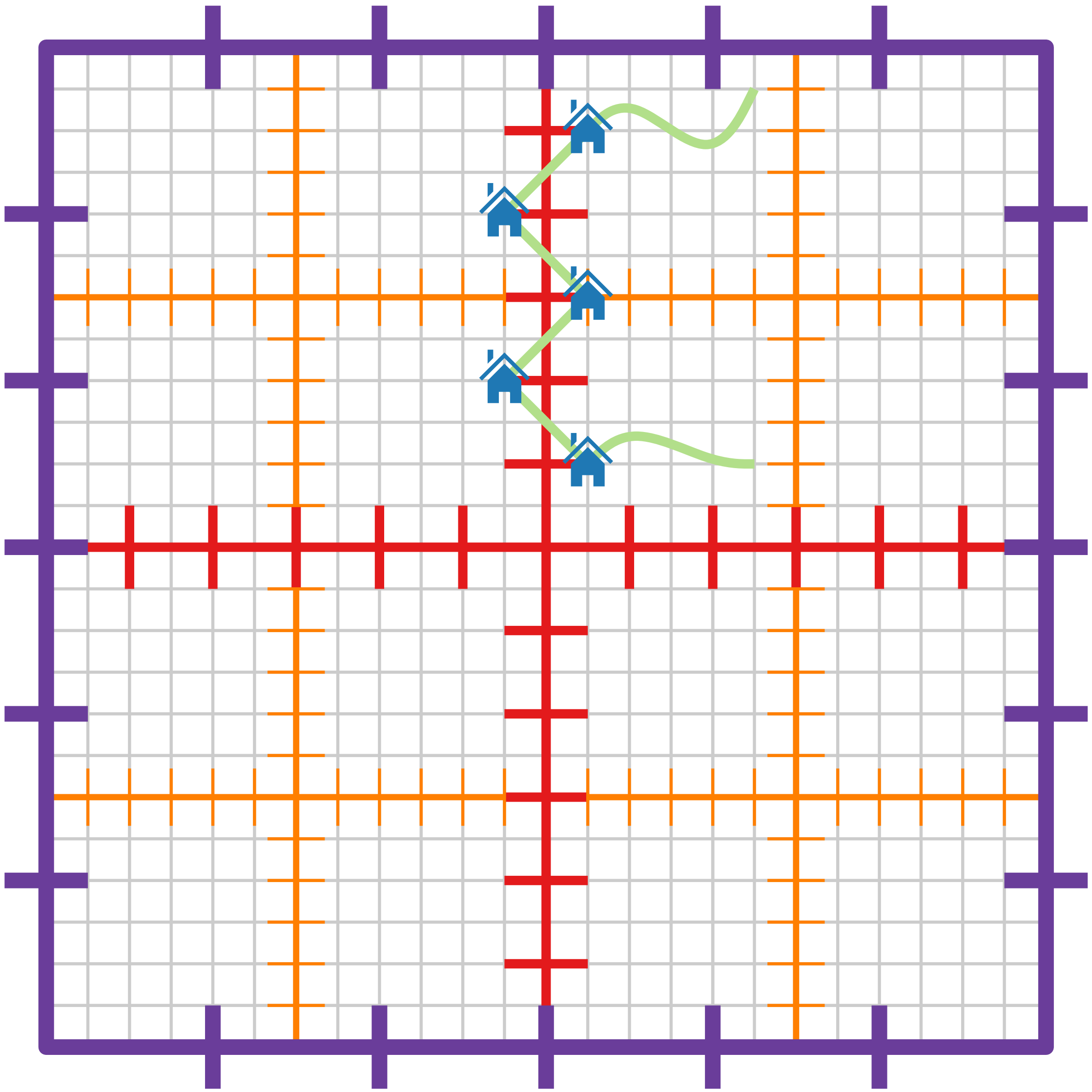
Lemma. An optimal well-behaved tour can be computed in $2^{O(m)} = n^{O(1/\varepsilon)}$ time.

Approximation Algorithms

Lecture 9:
A PTAS for Euclidean TSP

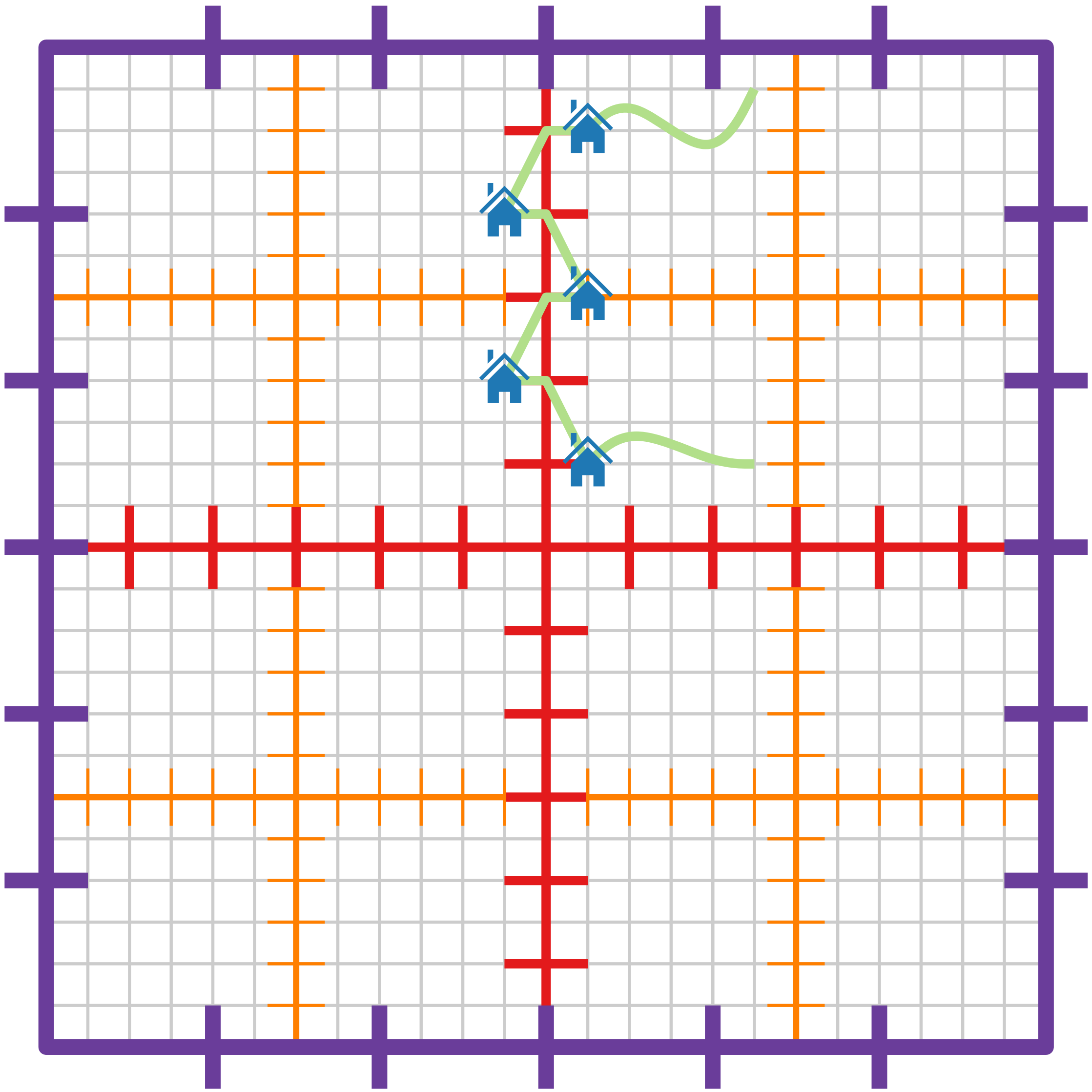
Part V:
Shifted Dissections

Shifted Dissections



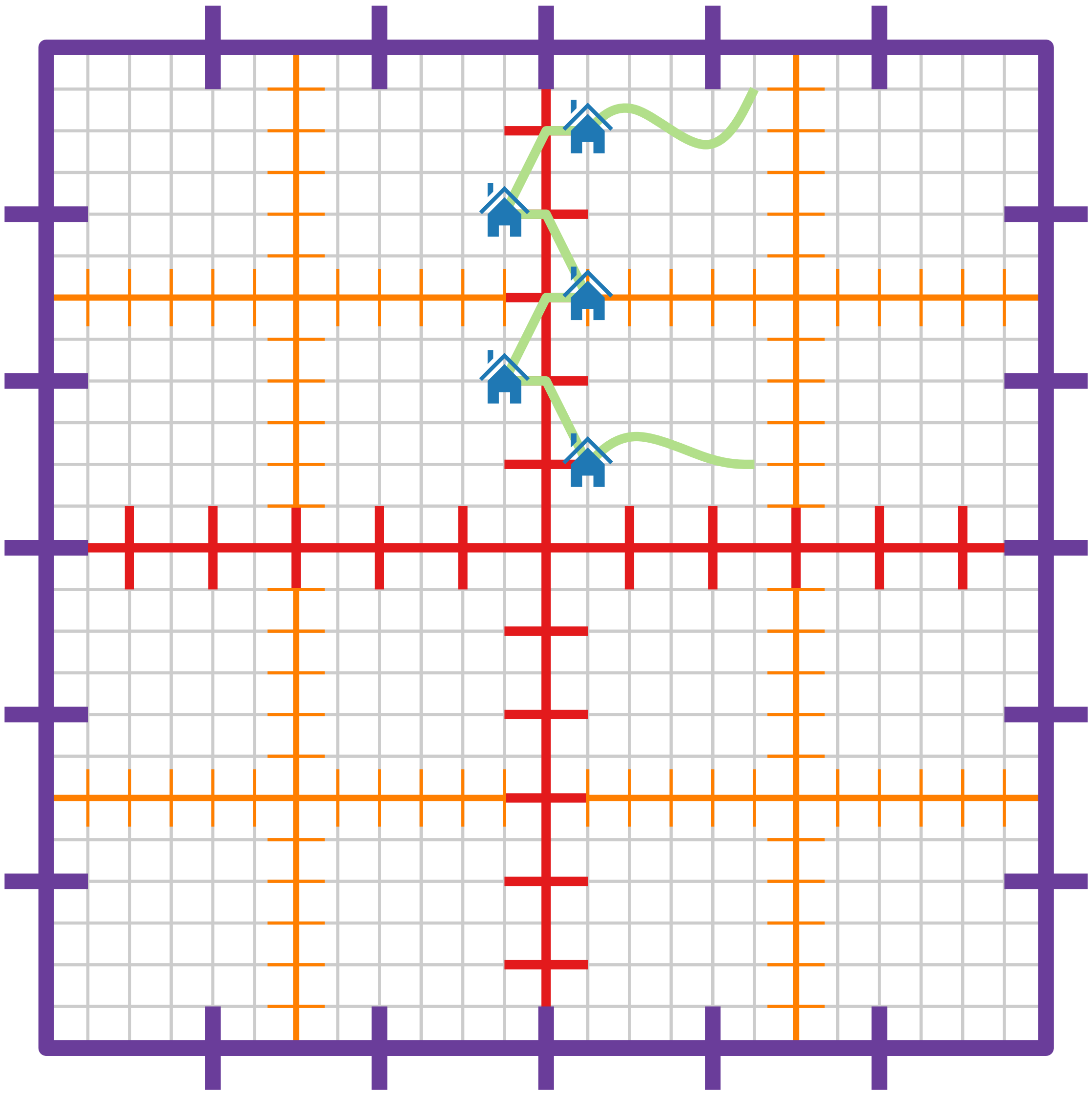
- The best well-behaved tour can be a bad approximation.

Shifted Dissections



- The best well-behaved tour can be a bad approximation.

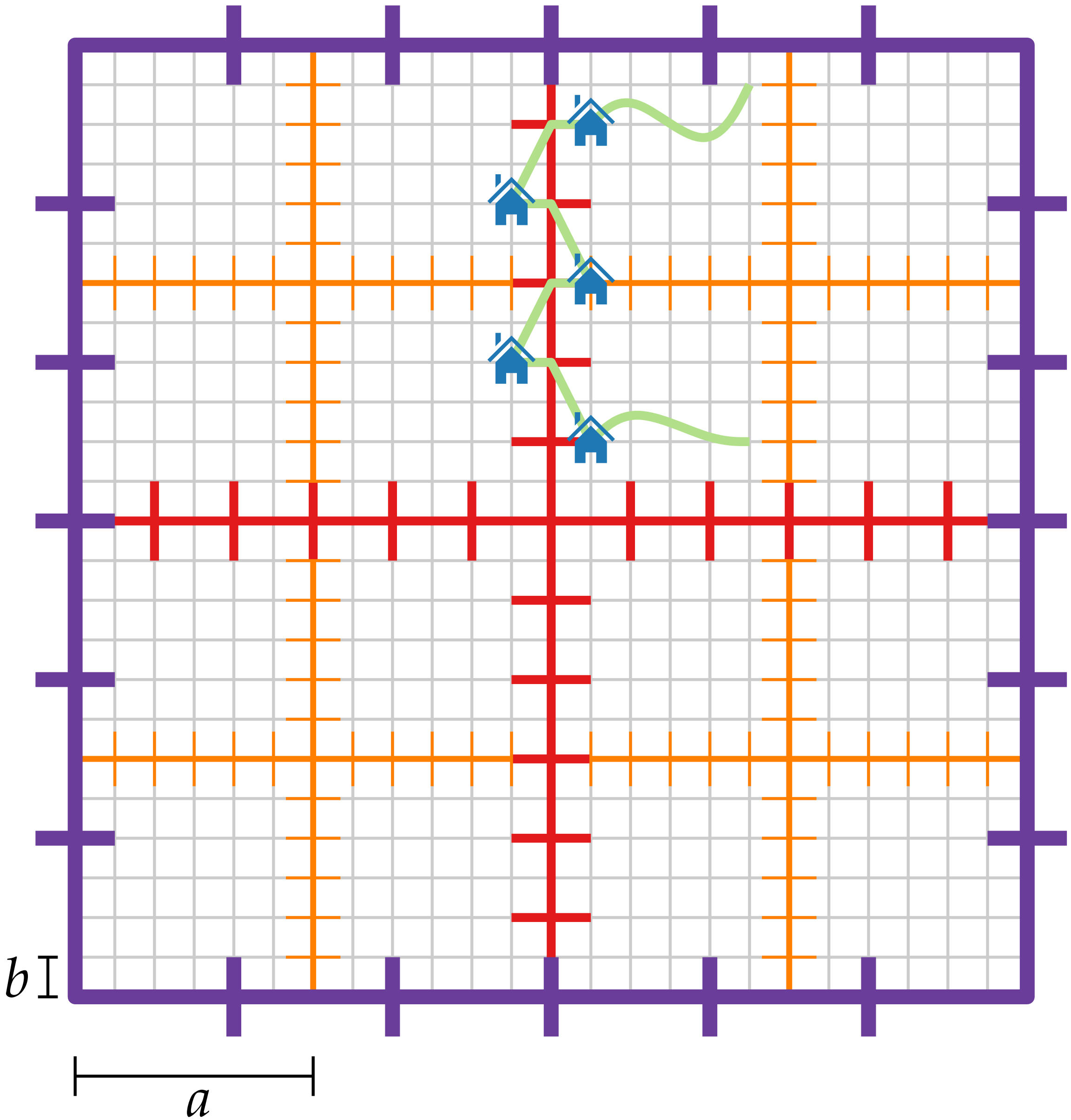
Shifted Dissections



- The best well-behaved tour can be a bad approximation.
- Consider an (a, b)-shifted dissection:

$$x \mapsto (x + a) \bmod L$$
$$y \mapsto (y + b) \bmod L$$

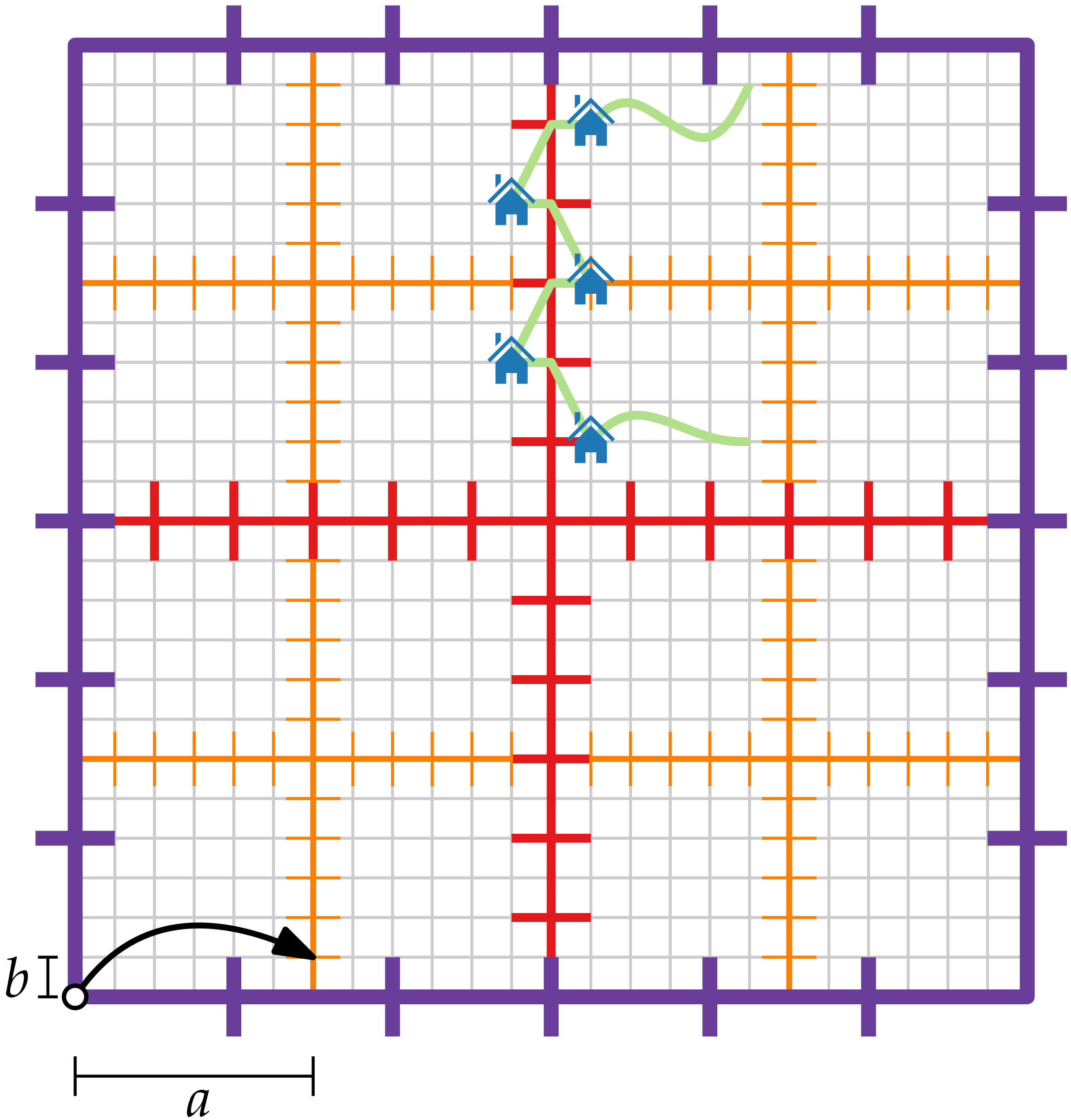
Shifted Dissections



- The best well-behaved tour can be a bad approximation.
- Consider an (a, b)-shifted dissection:

$$x \mapsto (x + a) \bmod L$$
$$y \mapsto (y + b) \bmod L$$

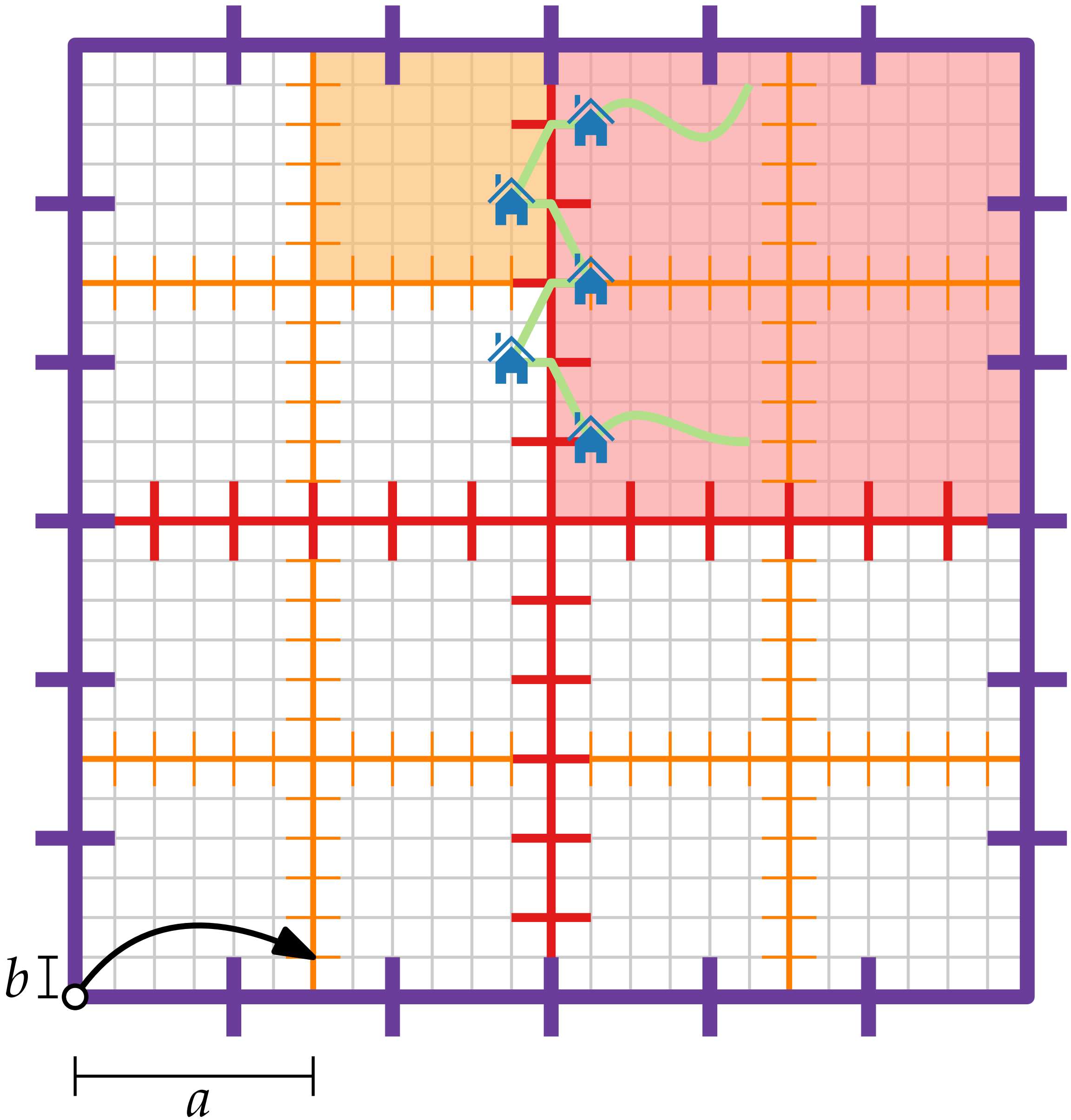
Shifted Dissections



- The best well-behaved tour can be a bad approximation.
- Consider an (a, b)-shifted dissection:

$$x \mapsto (x + a) \bmod L$$
$$y \mapsto (y + b) \bmod L$$

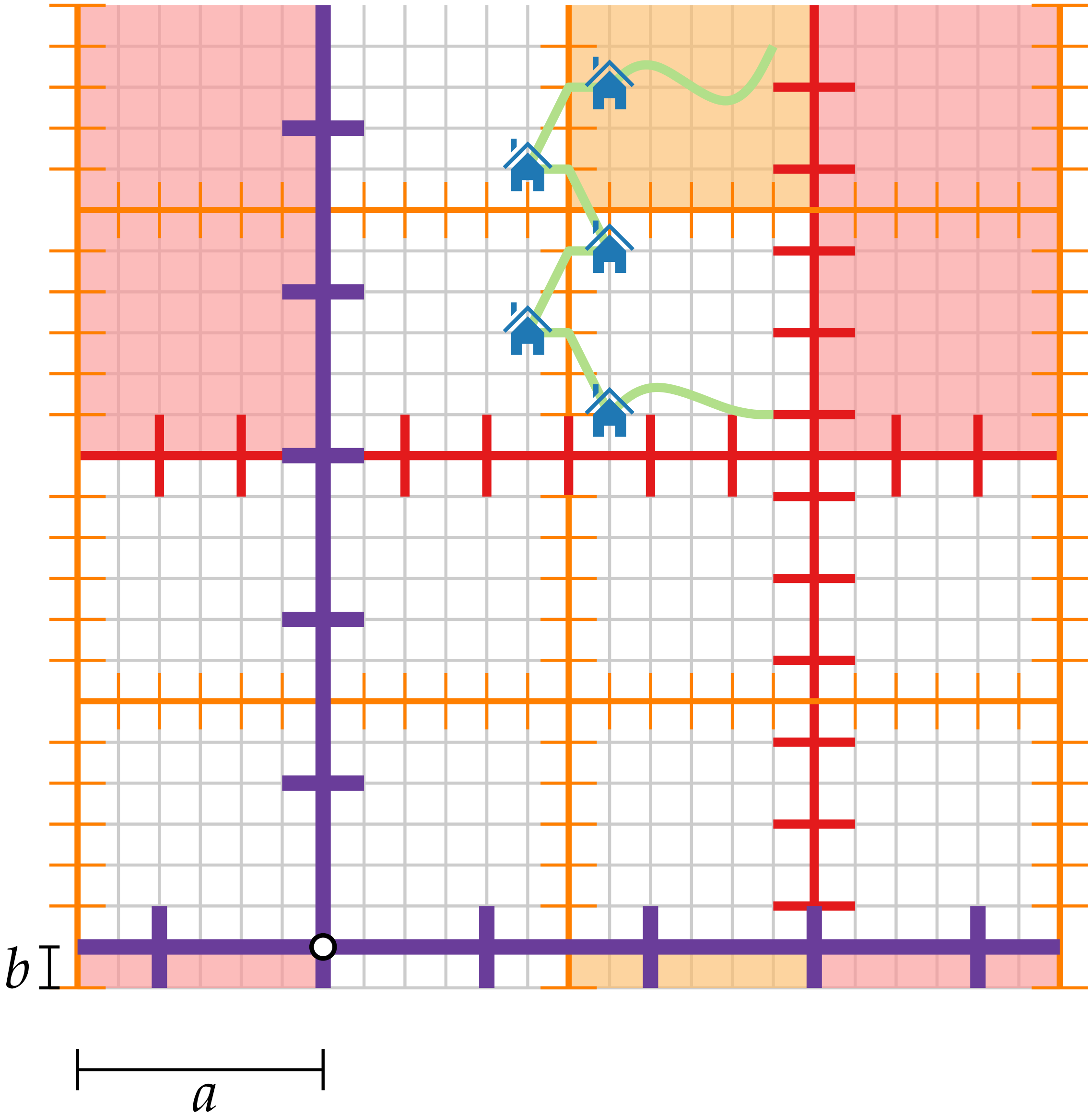
Shifted Dissections



- The best well-behaved tour can be a bad approximation.
- Consider an (a, b)-shifted dissection:

$$x \mapsto (x + a) \bmod L$$
$$y \mapsto (y + b) \bmod L$$

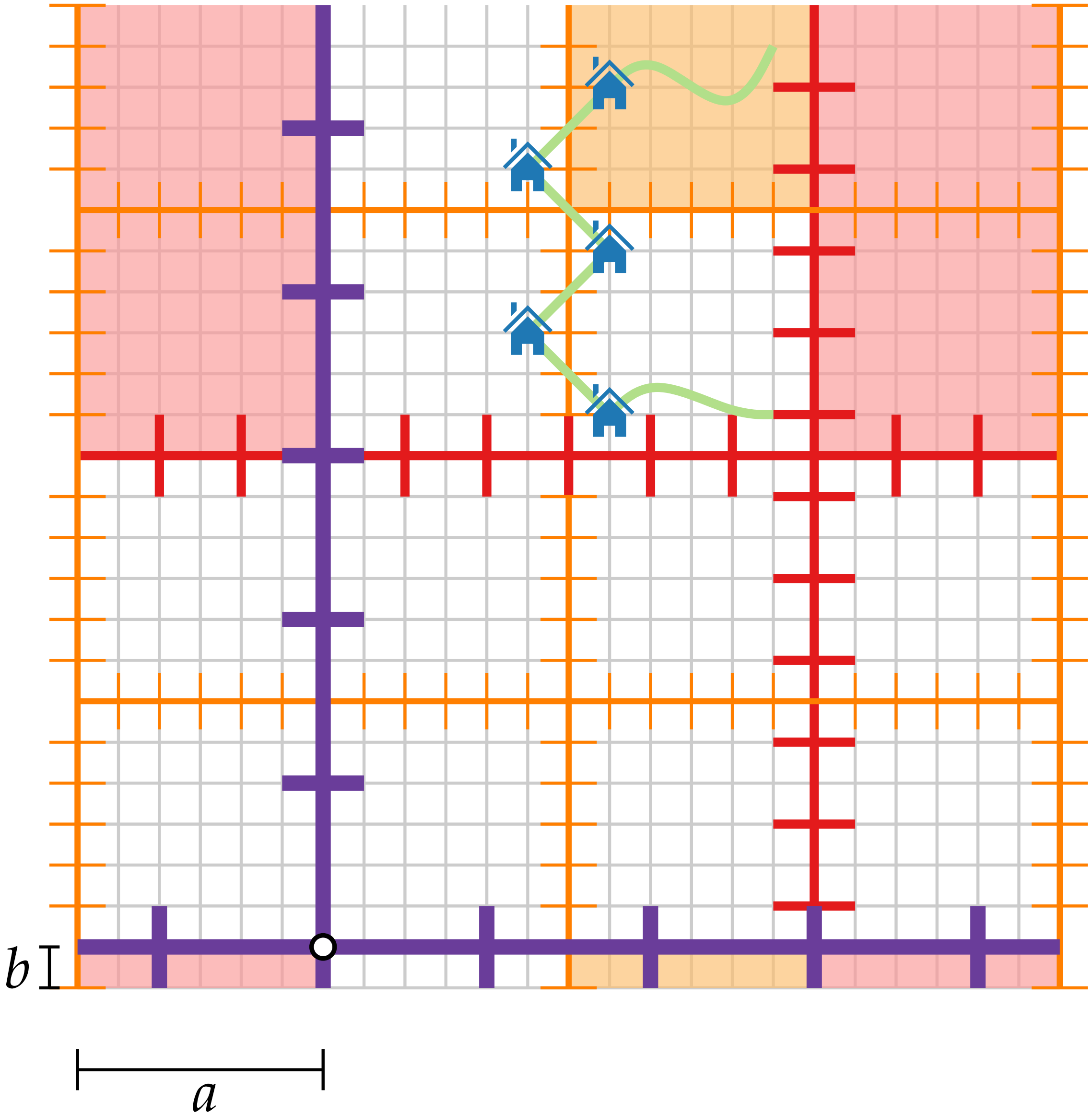
Shifted Dissections



- The best well-behaved tour can be a bad approximation.
- Consider an (a, b)-shifted dissection:

$$x \mapsto (x + a) \bmod L$$
$$y \mapsto (y + b) \bmod L$$

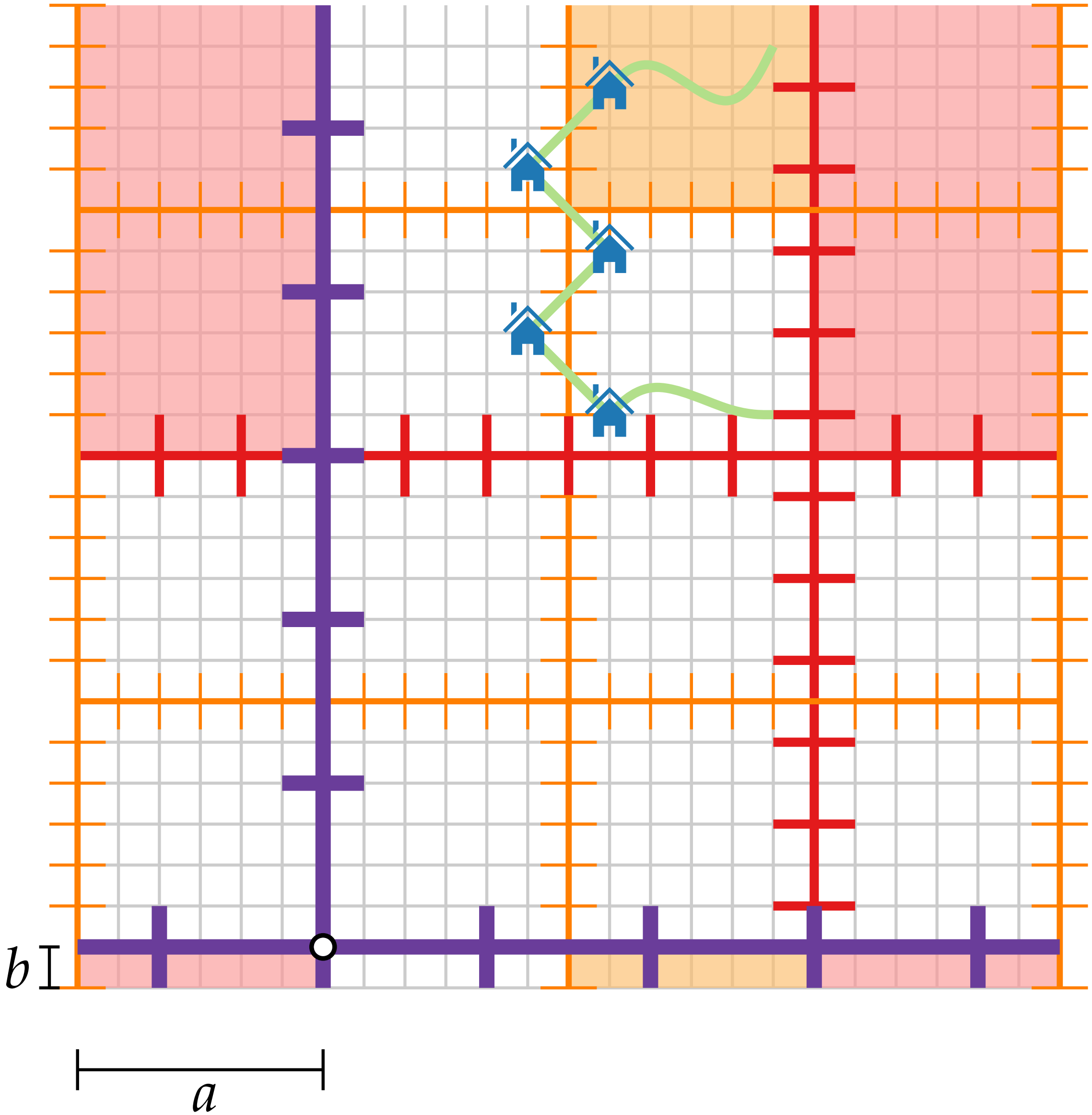
Shifted Dissections



- The best well-behaved tour can be a bad approximation.
- Consider an (a, b)-shifted dissection:

$$x \mapsto (x + a) \bmod L$$
$$y \mapsto (y + b) \bmod L$$

Shifted Dissections

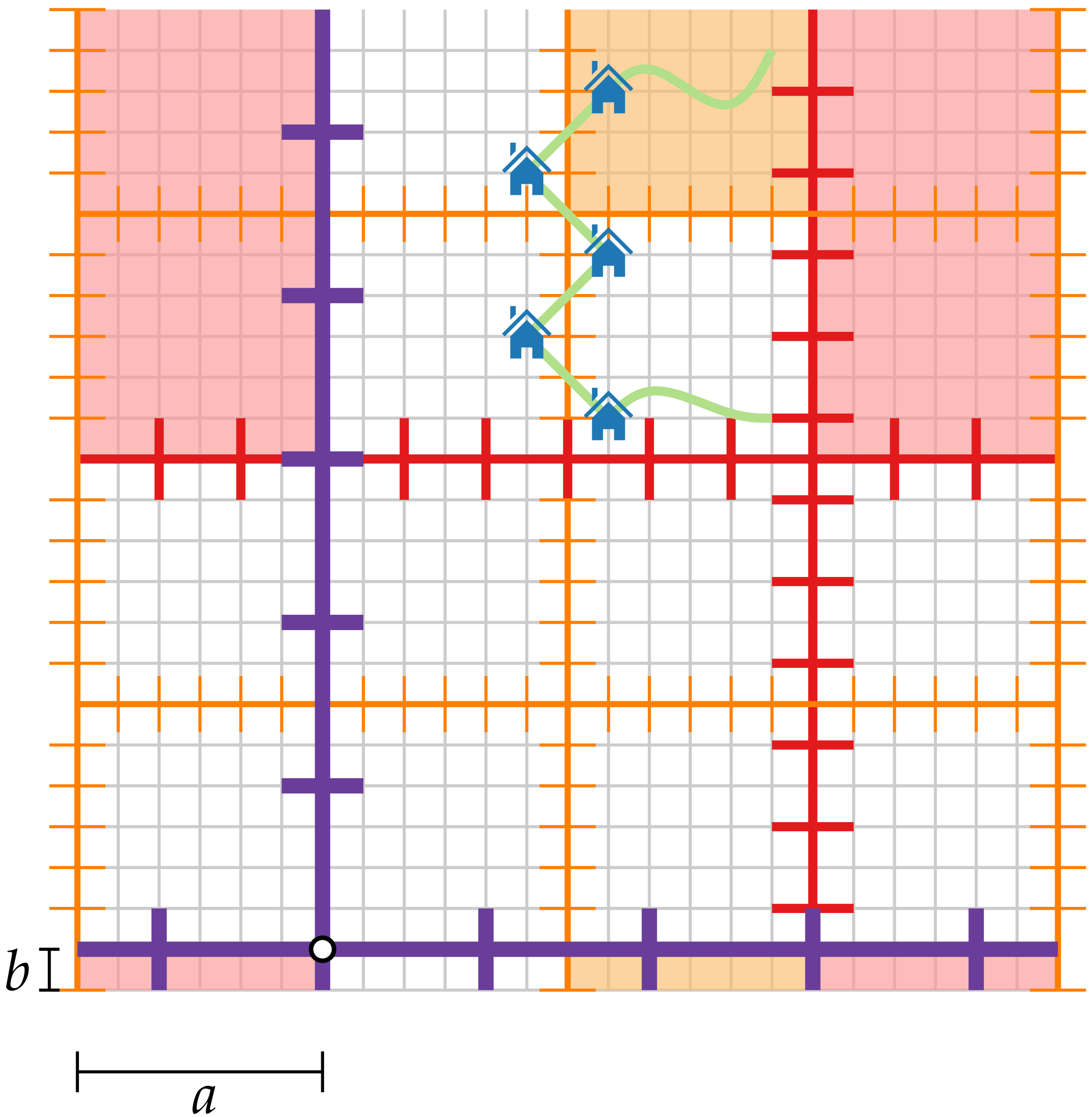


- The best well-behaved tour can be a bad approximation.
- Consider an (a, b)-shifted dissection:

$$x \mapsto (x + a) \bmod L$$
$$y \mapsto (y + b) \bmod L$$

- Squares in the dissection tree are “wrapped around”.

Shifted Dissections



- The best well-behaved tour can be a bad approximation.
- Consider an (a, b)-shifted dissection:

$$x \mapsto (x + a) \bmod L$$
$$y \mapsto (y + b) \bmod L$$

- Squares in the dissection tree are "wrapped around".
- Dynamic program must be modified accordingly.

Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid.

Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

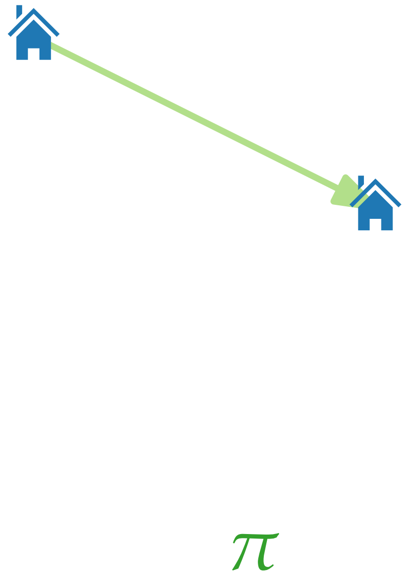
- Consider a tour as an ordered cyclic sequence.

Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.

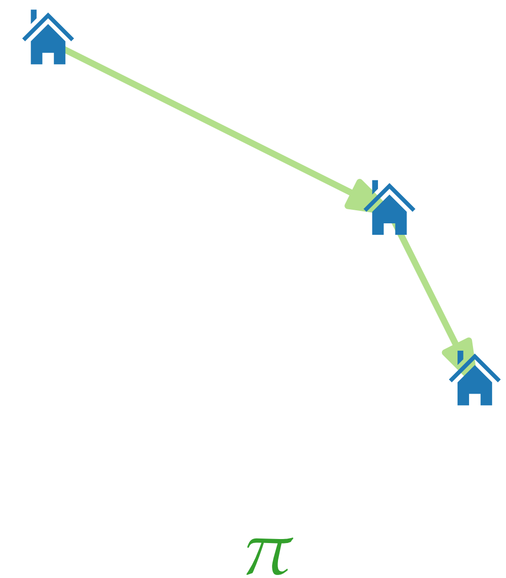


Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.

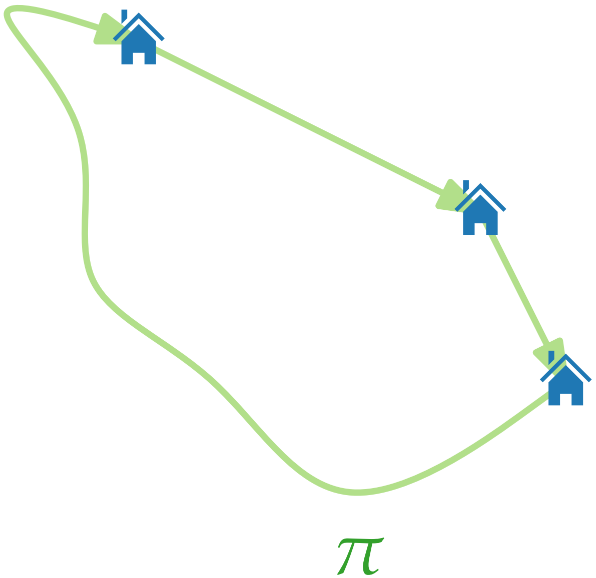


Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.

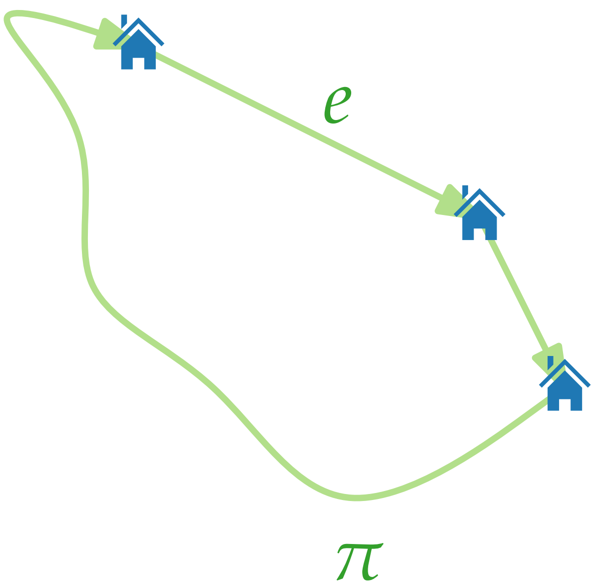


Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.

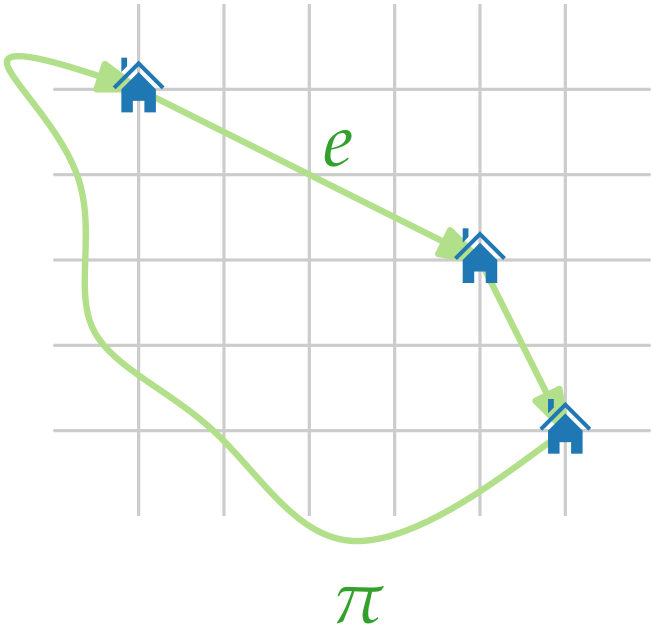


Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \text{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.

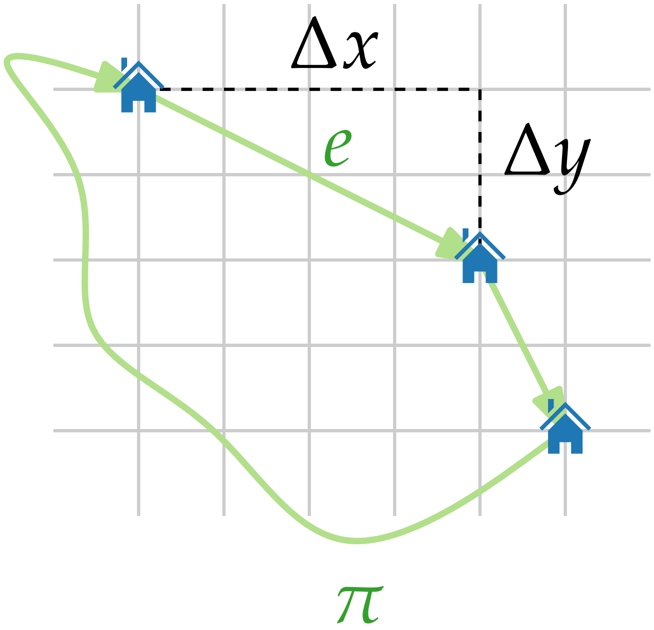


Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.

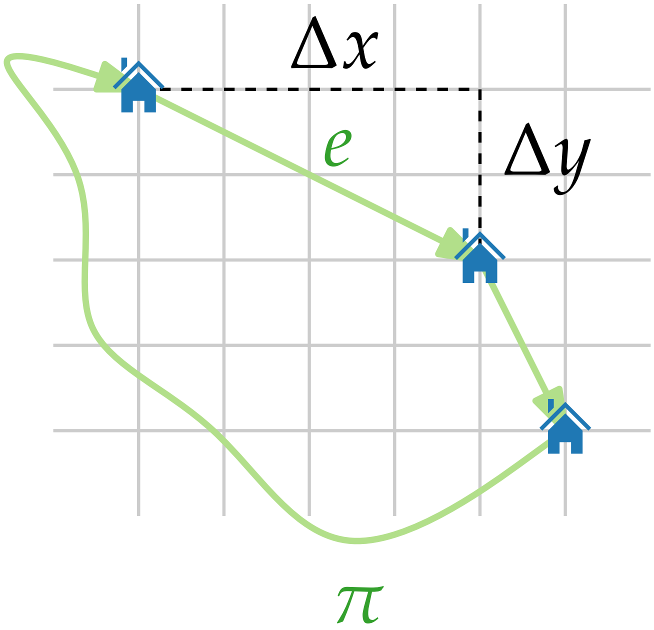


Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.
- Each edge e generates $N_e \leq \Delta x + \Delta y$ crossings.

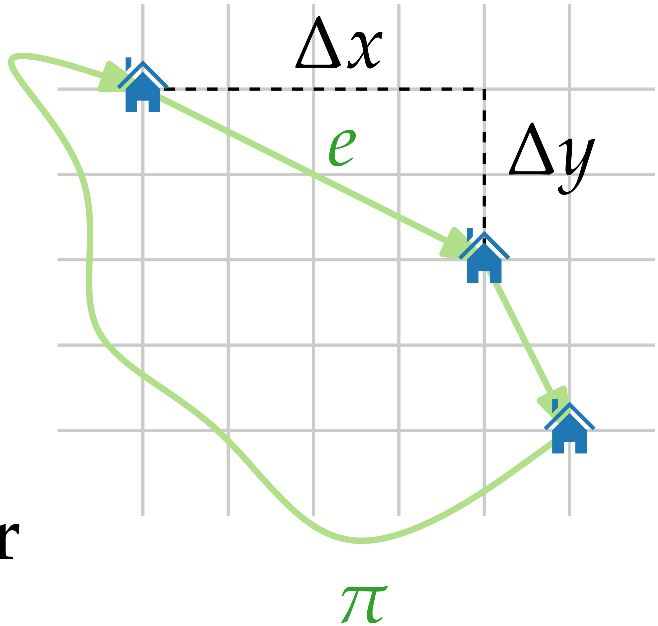


Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.
- Each edge e generates $N_e \leq \Delta x + \Delta y$ crossings.
- Crossings at the endpoint of an edge are counted for the next edge.

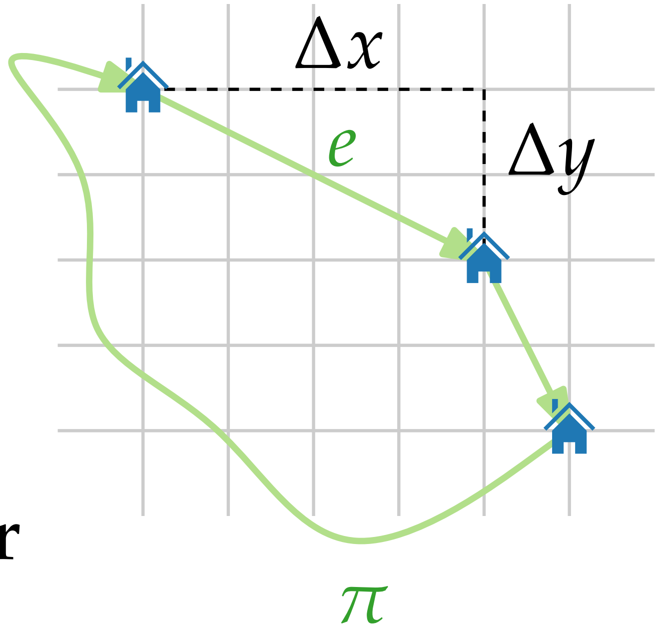


Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \text{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.
- Each edge e generates $N_e \leq \Delta x + \Delta y$ crossings.
- Crossings at the endpoint of an edge are counted for the next edge.
- ${N_e}^2 \leq$

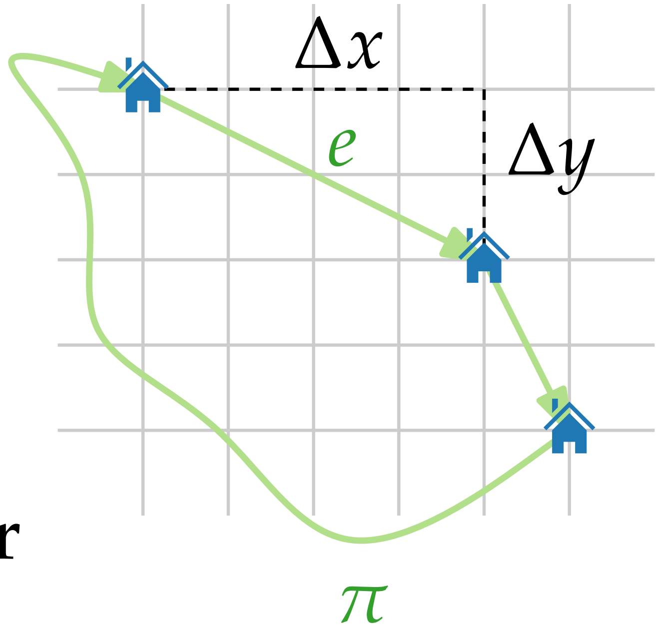


Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.
- Each edge e generates $N_e \leq \Delta x + \Delta y$ crossings.
- Crossings at the endpoint of an edge are counted for the next edge.
- $N_e{}^2 \leq (\Delta x + \Delta y)^2 \leq$

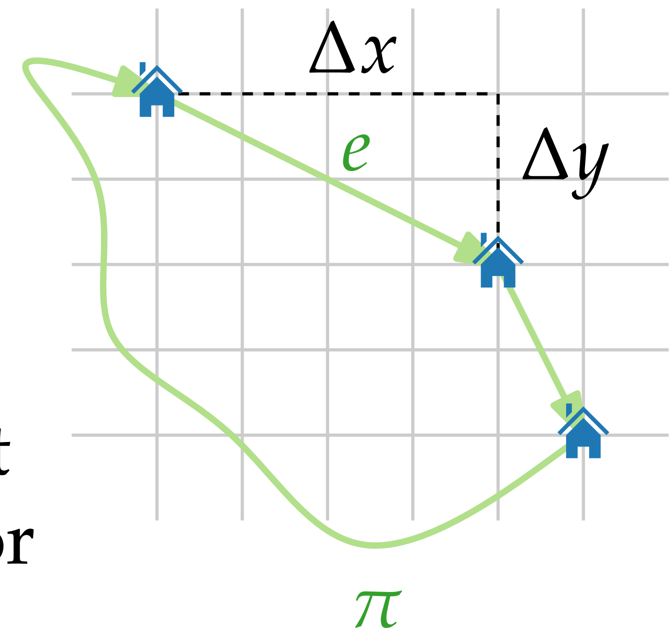


Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.
- Each edge e generates $N_e \leq \Delta x + \Delta y$ crossings.
- Crossings at the endpoint of an edge are counted for the next edge.



$0 \leq (\Delta x - \Delta y)^2$

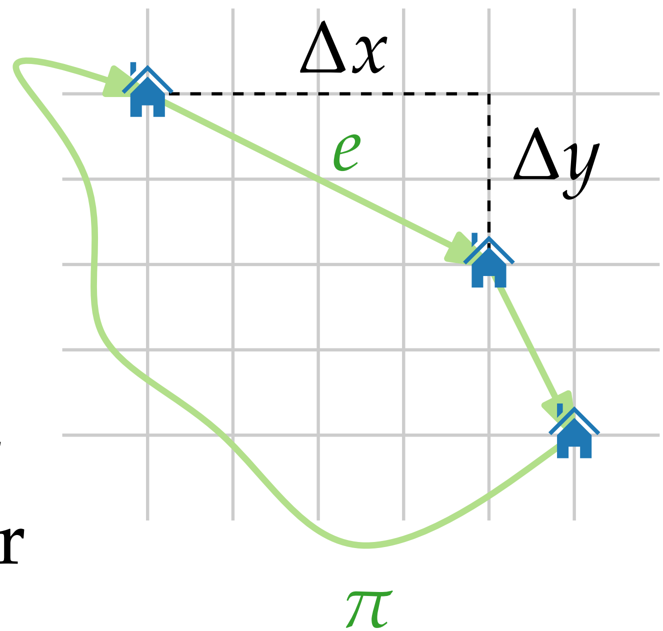
- ${N_e}^2 \leq (\Delta x + \Delta y)^2 \leq$

Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \le \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.
- Each edge e generates $N_e \le \Delta x + \Delta y$ crossings.
- Crossings at the endpoint of an edge are counted for the next edge.



$0 \le (\Delta x - \Delta y)^2$

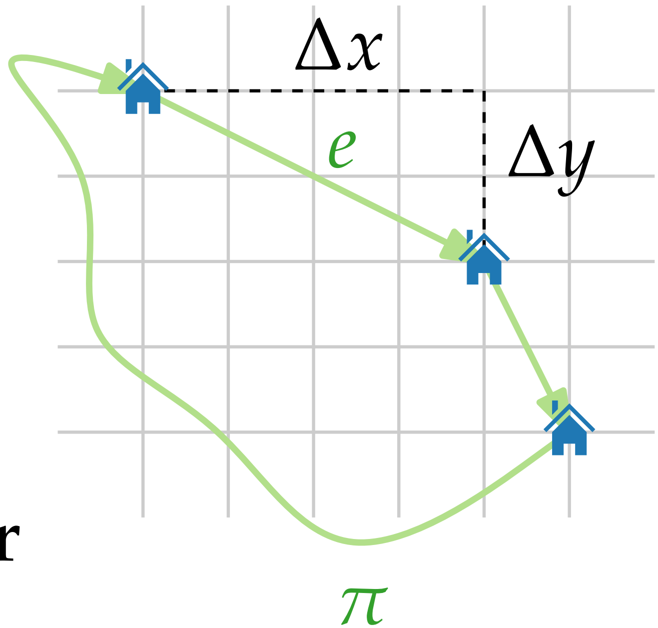
- $N_e{}^2 \le (\Delta x + \Delta y)^2 \le 2(\Delta x^2 + \Delta y^2) =$

Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \text{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.
- Each edge e generates $N_e \leq \Delta x + \Delta y$ crossings.
- Crossings at the endpoint of an edge are counted for the next edge.



$0 \leq (\Delta x - \Delta y)^2$

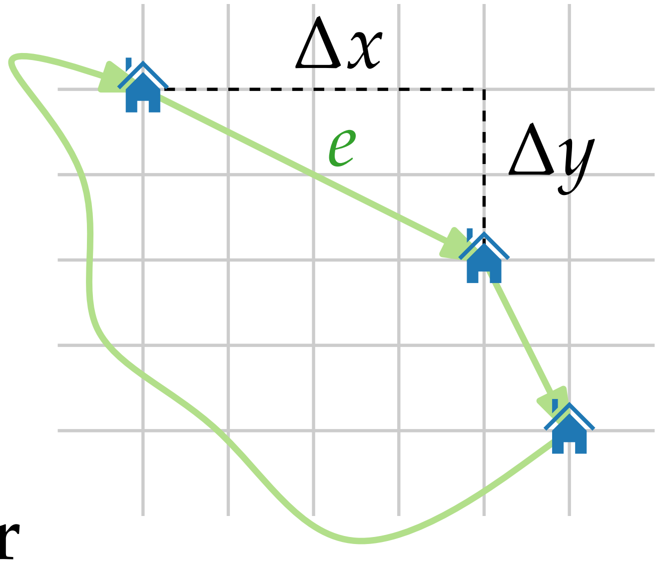
- $N_e{}^2 \leq (\Delta x + \Delta y)^2 \leq 2(\Delta x^2 + \Delta y^2) = 2|e|^2$.

Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.
- Each edge e generates $N_e \leq \Delta x + \Delta y$ crossings.
- Crossings at the endpoint of an edge are counted for the next edge.



$0 \leq (\Delta x - \Delta y)^2$

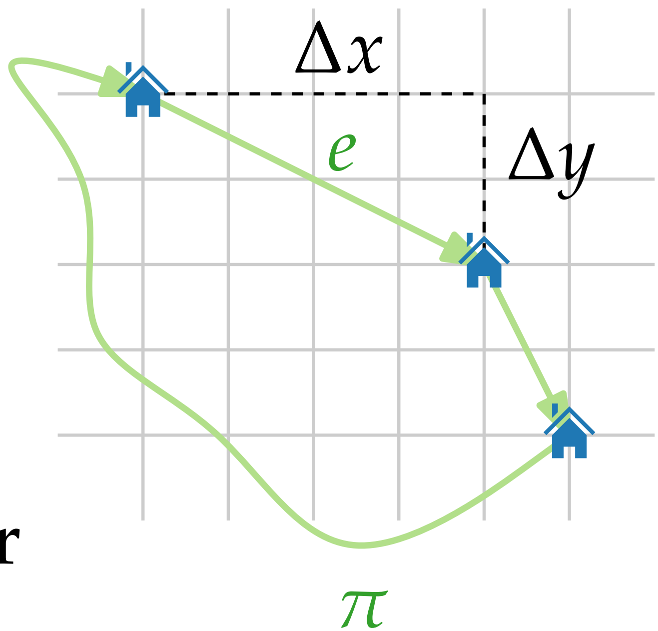
- $N_e{}^2 \leq (\Delta x + \Delta y)^2 \leq 2(\Delta x^2 + \Delta y^2) = 2|e|^2$.
- $N(\pi) =$

Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.
- Each edge e generates $N_e \leq \Delta x + \Delta y$ crossings.
- Crossings at the endpoint of an edge are counted for the next edge.



$0 \leq (\Delta x - \Delta y)^2$

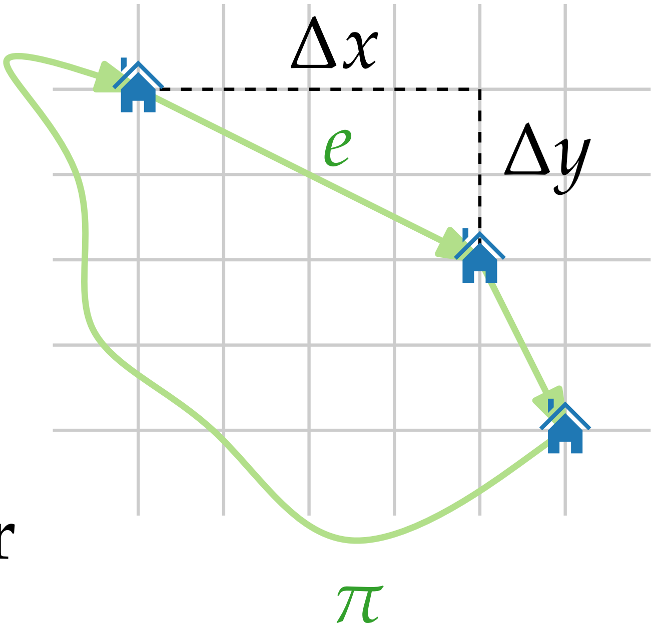
- $N_e{}^2 \leq (\Delta x + \Delta y)^2 \leq 2(\Delta x^2 + \Delta y^2) = 2|e|^2$.
- $N(\pi) = \sum_{e \in \pi} N_e \leq$

Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \text{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.
- Each edge e generates $N_e \leq \Delta x + \Delta y$ crossings.
- Crossings at the endpoint of an edge are counted for the next edge.



$0 \leq (\Delta x - \Delta y)^2$

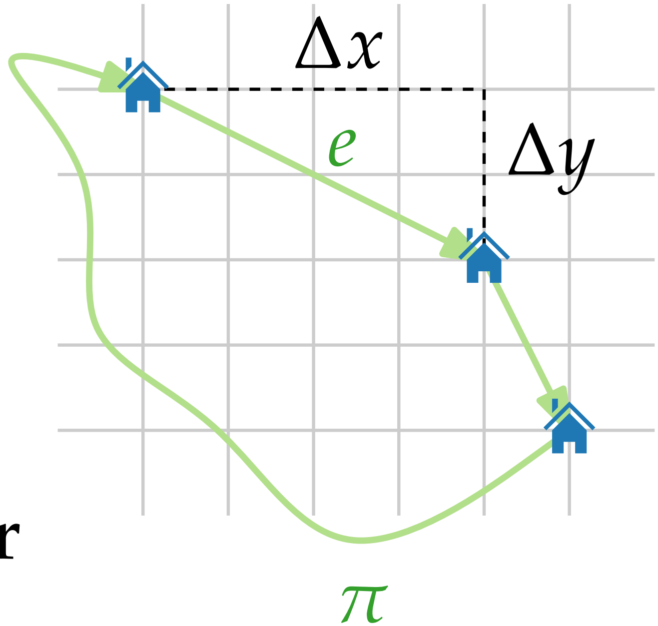
- $N_e{}^2 \leq (\Delta x + \Delta y)^2 \leq 2(\Delta x^2 + \Delta y^2) = 2|e|^2.$
- $N(\pi) = \sum_{e \in \pi} N_e \leq \sum_{e \in \pi} \sqrt{2|e|^2}$

Shifted Dissections (II)

Lemma. Let π be an optimal tour, and let $N(\pi)$ be the number of crossings of π with the lines of the $(L \times L)$-grid. Then we have $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$.

Proof.

- Consider a tour as an ordered cyclic sequence.
- Each edge e generates $N_e \leq \Delta x + \Delta y$ crossings.
- Crossings at the endpoint of an edge are counted for the next edge.



$0 \leq (\Delta x - \Delta y)^2$

- $N_e{}^2 \leq (\Delta x + \Delta y)^2 \leq 2(\Delta x^2 + \Delta y^2) = 2|e|^2$.
- $N(\pi) = \sum_{e \in \pi} N_e \leq \sum_{e \in \pi} \sqrt{2|e|^2} = \sqrt{2} \cdot \mathrm{OPT}$. □

Approximation Algorithms

Lecture 9:
A PTAS for Euclidean TSP

Part VI:
Approximation Factor

Shifted Dissections (III)

Theorem. Let $a, b \in [0, L-1]$ be chosen independently and uniformly at random.

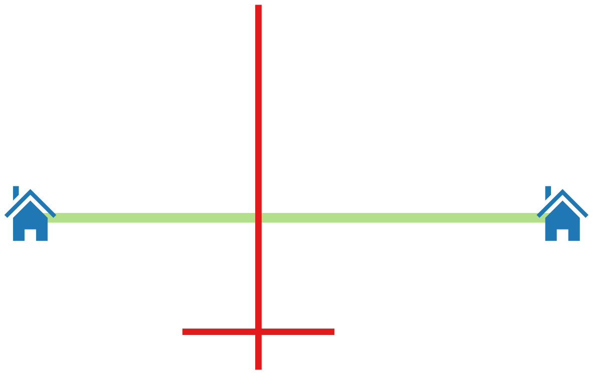
Shifted Dissections (III)

Theorem. Let $a, b \in [0, L-1]$ be chosen independently and uniformly at random. Then the expected cost of an optimal well-behaved tour with respect to the (a, b)-shifted dissection is $\leq (1 + 2\sqrt{2}\varepsilon)\mathrm{OPT}$.

Shifted Dissections (III)

Theorem. Let $a, b \in [0, L-1]$ be chosen independently and uniformly at random. Then the expected cost of an optimal well-behaved tour with respect to the (a, b)-shifted dissection is $\leq (1 + 2\sqrt{2}\varepsilon)\mathrm{OPT}$.

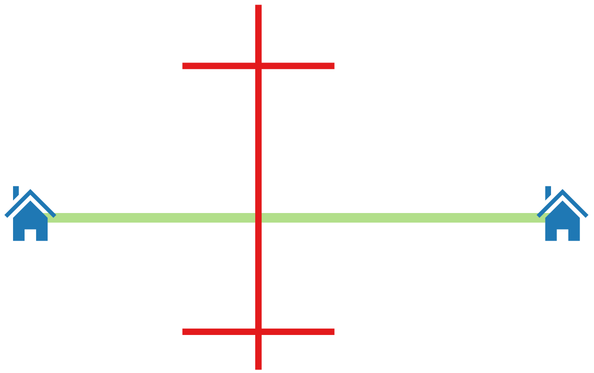
Proof. Consider optimal tour π.



Shifted Dissections (III)

Theorem. Let $a, b \in [0, L-1]$ be chosen independently and uniformly at random. Then the expected cost of an optimal well-behaved tour with respect to the (a, b)-shifted dissection is $\leq (1 + 2\sqrt{2}\varepsilon)\mathrm{OPT}$.

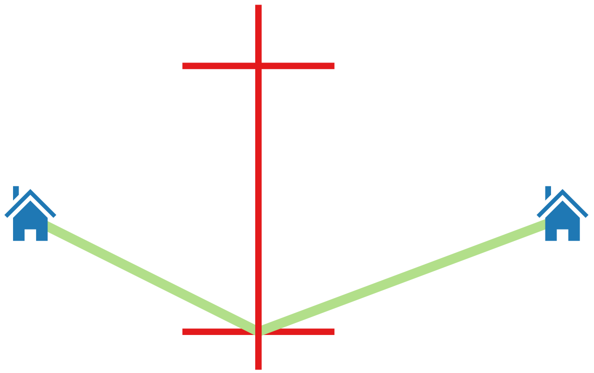
Proof. Consider optimal tour π. Make π well-behaved by moving each intersection point with the $(L \times L)$-grid to the nearest portal.



Shifted Dissections (III)

Theorem. Let $a, b \in [0, L-1]$ be chosen independently and uniformly at random. Then the expected cost of an optimal well-behaved tour with respect to the (a, b)-shifted dissection is $\leq (1 + 2\sqrt{2}\varepsilon)\mathrm{OPT}$.

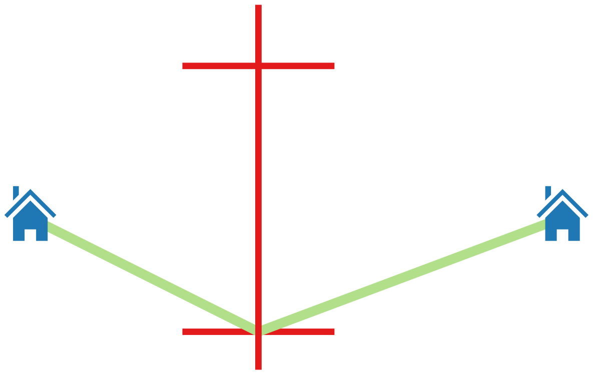
Proof. Consider optimal tour π. Make π well-behaved by moving each intersection point with the $(L \times L)$-grid to the nearest portal.



Shifted Dissections (III)

Theorem. Let $a, b \in [0, L-1]$ be chosen independently and uniformly at random. Then the expected cost of an optimal well-behaved tour with respect to the (a, b)-shifted dissection is $\leq (1 + 2\sqrt{2}\varepsilon)\mathrm{OPT}$.

Proof. Consider optimal tour π. Make π well-behaved by moving each intersection point with the $(L \times L)$-grid to the nearest portal.

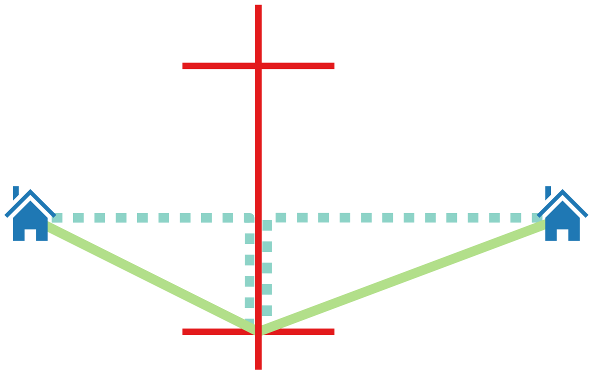


Detour per intersection $\leq$ inter-portal distance.

Shifted Dissections (III)

Theorem. Let $a, b \in [0, L-1]$ be chosen independently and uniformly at random. Then the expected cost of an optimal well-behaved tour with respect to the (a, b)-shifted dissection is $\leq (1 + 2\sqrt{2}\varepsilon)\mathrm{OPT}$.

Proof. Consider optimal tour π. Make π well-behaved by moving each intersection point with the $(L \times L)$-grid to the nearest portal.



Detour per intersection $\leq$ inter-portal distance.

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* , the line l is a level-i line.

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* $2^i/L$, the line l is a level-i line.

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* $2^i/L$, the line l is a level-i line.
  $\Rightarrow$ Increase in tour length $\leq$ (inter-portal distance).

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* $2^i/L$, the line l is a level-i line.
  $\Rightarrow$ Increase in tour length $\leq L/(2^i m)$ (inter-portal distance).

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* $2^i/L$, the line l is a level-i line.
$\Rightarrow$ Increase in tour length $\leq L/(2^i m)$ (inter-portal distance).

- Thus, the expected increase in tour length due to this intersection is at most:

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* $2^i/L$, the line l is a level-i line.
  $\Rightarrow$ Increase in tour length $\leq L/(2^i m)$ (inter-portal distance).

- Thus, the expected increase in tour length due to this intersection is at most:

$$\sum_{i=0}^{k}$$

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* $2^i/L$, the line l is a level-i line.
$\Rightarrow$ Increase in tour length $\leq L/(2^i m)$ (inter-portal distance).

- Thus, the expected increase in tour length due to this intersection is at most:

$$\sum_{i=0}^{k} \frac{2^i}{L} \cdot$$

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* $2^i/L$, the line l is a level-i line.
  $\Rightarrow$ Increase in tour length $\leq L/(2^i m)$ (inter-portal distance).

- Thus, the expected increase in tour length due to this intersection is at most:

$$\sum_{i=0}^{k} \frac{2^i}{L} \cdot \frac{L}{2^i m} \leq$$

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* $2^i/L$, the line l is a level-i line.
  $\Rightarrow$ Increase in tour length $\leq L/(2^i m)$ (inter-portal distance).

- Thus, the expected increase in tour length due to this intersection is at most:

$$\sum_{i=0}^{k} \frac{2^i}{L} \cdot \frac{L}{2^i m} \leq \frac{k+1}{m} \leq$$

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* $2^i/L$, the line l is a level-i line.
  $\Rightarrow$ Increase in tour length $\leq L/(2^i m)$ (inter-portal distance).

- Thus, the expected increase in tour length due to this intersection is at most: $m \in [k/\varepsilon, 2k/\varepsilon]$

$$\sum_{i=0}^{k} \frac{2^i}{L} \cdot \frac{L}{2^i m} \leq \frac{k+1}{m} \leq$$

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* $2^i/L$, the line l is a level-i line.
  $\Rightarrow$ Increase in tour length $\leq L/(2^i m)$ (inter-portal distance).

- Thus, the expected increase in tour length due to this intersection is at most: $m \in [k/\varepsilon, 2k/\varepsilon]$

$$\sum_{i=0}^{k} \frac{2^i}{L} \cdot \frac{L}{2^i m} \leq \frac{k+1}{m} \leq 2\varepsilon.$$

Shifted Dissections (III)

- Consider an intersection point between π and a line l of the $(L \times L)$-grid.

- With probability *at most* $2^i/L$, the line l is a level-i line.
$\Rightarrow$ Increase in tour length $\leq L/(2^i m)$ (inter-portal distance).

- Thus, the expected increase in tour length due to this intersection is at most: $m \in [k/\varepsilon, 2k/\varepsilon]$

$$\sum_{i=0}^{k} \frac{2^i}{L} \cdot \frac{L}{2^i m} \leq \frac{k+1}{m} \leq 2\varepsilon.$$

- Summing over all $N(\pi) \leq \sqrt{2} \cdot \mathrm{OPT}$ intersection points and applying linearity of expectation yields the claim.

Polynomial-Time Approximation Scheme

Theorem. Let $a, b \in [0, L-1]$ be chosen independently and uniformly at random. Then the expected cost of an optimal well-behaved tour with respect to the (a, b)-shifted dissection is $\leq (1 + 2\sqrt{2}\varepsilon)\mathrm{OPT}$.

Polynomial-Time Approximation Scheme

Theorem. Let $a, b \in [0, L-1]$ be chosen independently and uniformly at random. Then the expected cost of an optimal well-behaved tour with respect to the (a, b)-shifted dissection is $\leq (1 + 2\sqrt{2}\varepsilon)\mathrm{OPT}$.

Theorem. There is a *deterministic* algorithm (PTAS) for Euclidean TSP that provides, for every $\varepsilon > 0$, a $(1+\varepsilon)$-approximation in $n^{O(1/\varepsilon)}$ time.

Polynomial-Time Approximation Scheme

Theorem. Let $a, b \in [0, L-1]$ be chosen independently and uniformly at random. Then the expected cost of an optimal well-behaved tour with respect to the (a, b)-shifted dissection is $\leq (1 + 2\sqrt{2}\varepsilon)\mathrm{OPT}$.

Theorem. There is a *deterministic* algorithm (PTAS) for Euclidean TSP that provides, for every $\varepsilon > 0$, a $(1+\varepsilon)$-approximation in $n^{O(1/\varepsilon)}$ time.

Proof. Try all L^2 many (a, b)-shifted dissections.

Polynomial-Time Approximation Scheme

Theorem. Let $a, b \in [0, L-1]$ be chosen independently and uniformly at random. Then the expected cost of an optimal well-behaved tour with respect to the (a, b)-shifted dissection is $\leq (1 + 2\sqrt{2}\varepsilon)\mathrm{OPT}$.

Theorem. There is a *deterministic* algorithm (PTAS) for Euclidean TSP that provides, for every $\varepsilon > 0$, a $(1+\varepsilon)$-approximation in $n^{O(1/\varepsilon)}$ time.

Proof. Try all L^2 many (a, b)-shifted dissections. By the previous theorem and the pigeon-hole principle, one of them is good enough. $\square$

Literature

- William J. Cook: Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation.
  Princeton University Press, 2011.

Literature

- William J. Cook: Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation. Princeton University Press, 2011.

- Sanjeev Arora: Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and other Geometric Problems. J. ACM, 45(5):753–782, 1998.

- Joseph S. B. Mitchell: Guillotine Subdivisions Approximate Polygonal Subdivisions: A Simple Polynomial-Time Approximation Scheme for Geometric TSP, *k*-MST, and Related Problems. SIAM J. Comput., 28(4):1298–1309, 1999.

Literature

- William J. Cook: Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation. Princeton University Press, 2011.
- Sanjeev Arora: Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and other Geometric Problems. J. ACM, 45(5):753–782, 1998.
- Joseph S. B. Mitchell: Guillotine Subdivisions Approximate Polygonal Subdivisions: A Simple Polynomial-Time Approximation Scheme for Geometric TSP, k-MST, and Related Problems. SIAM J. Comput., 28(4):1298–1309, 1999.
- Sanjeev Arora: Nearly linear time approximation schemes for Euclidean TSP and other geometric problems. Network Design 1–2, 1997

Literature

- William J. Cook: Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation. Princeton University Press, 2011.

- Sanjeev Arora: Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and other Geometric Problems. J. ACM, 45(5):753–782, 1998.

- Joseph S. B. Mitchell: Guillotine Subdivisions Approximate Polygonal Subdivisions: A Simple Polynomial-Time Approximation Scheme for Geometric TSP, k-MST, and Related Problems. SIAM J. Comput., 28(4):1298–1309, 1999.

- Sanjeev Arora: Nearly linear time approximation schemes for Euclidean TSP and other geometric problems. Network Design 1–2, 1997

Randomized, $O\left(n(\log n)^{O(1/\varepsilon)}\right)$ time.

Literature (cont'd)

- Sanjeev Arora, Michelangelo Grigni, David Karger, Philip Klein, Andrzej Woloszyn: Polynomial time approximation scheme for Weighted Planar Graph TSP. Proc. SIAM-ACM SODA, p. 33–41, 1998.

Literature (cont'd)

- Sanjeev Arora, Michelangelo Grigni, David Karger, Philip Klein, Andrzej Woloszyn: Polynomial time approximation scheme for Weighted Planar Graph TSP. Proc. SIAM-ACM SODA, p. 33–41, 1998.

Runtime $O\left(n^{O(1/\varepsilon^2)}\right)$