Stack-sorting and preimages of mesh patterns

Hjalti Magnússon (hjaltim07@ru.is)
Henning Arnór Úlfarsson (henningu@ru.is)

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

School of Computer Science

Permutation Patterns
June 2012

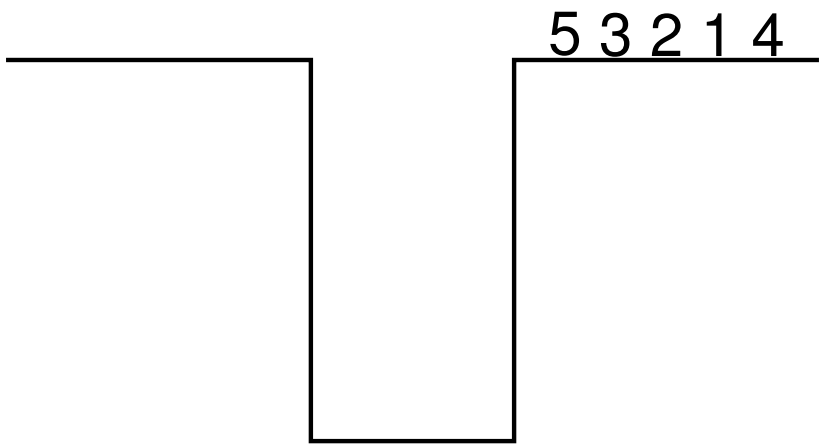
Outline

Outline

Outline

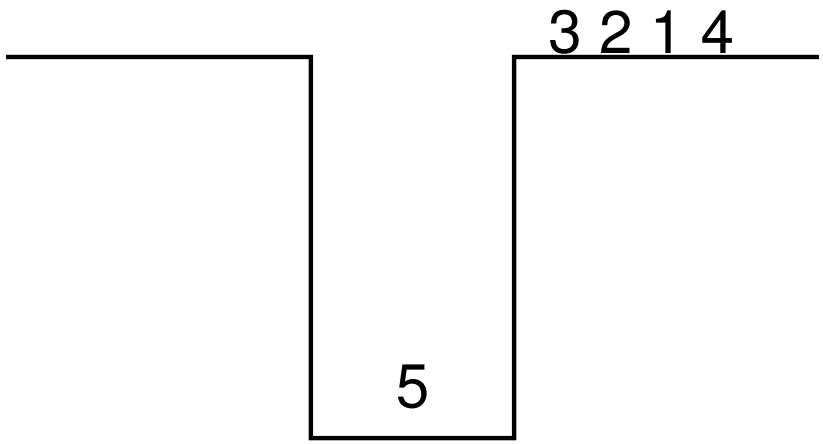
An Example

Let's stack-sort 53214



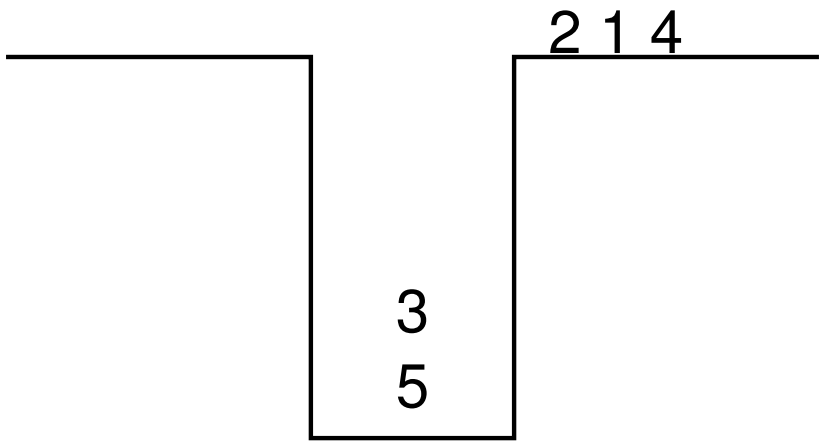
An Example

Let's stack-sort 53214



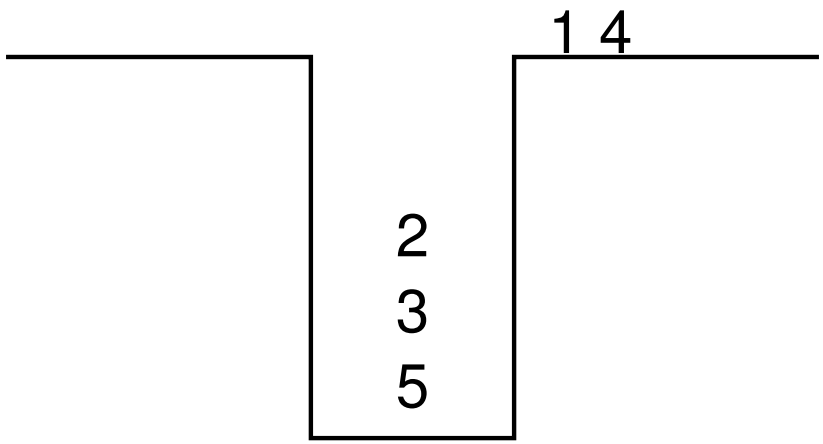
An Example

Let's stack-sort 53214



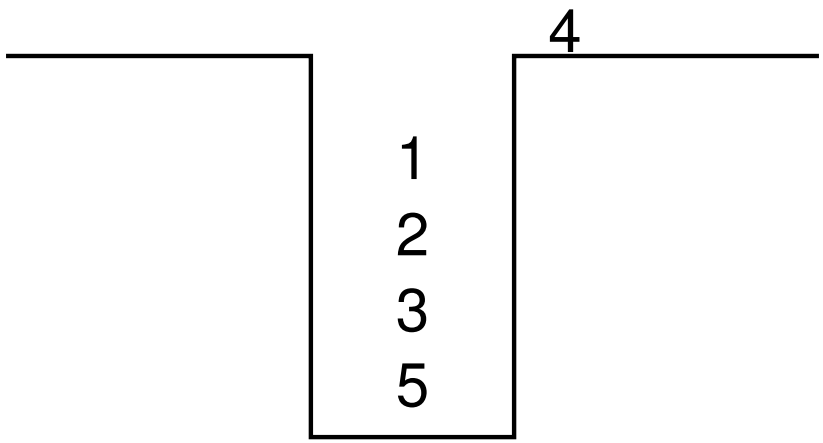
An Example

Let's stack-sort 53214



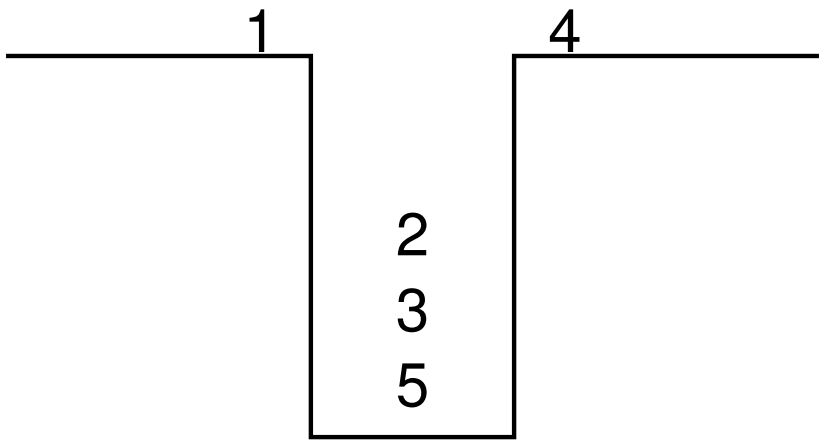
An Example

Let's stack-sort 53214



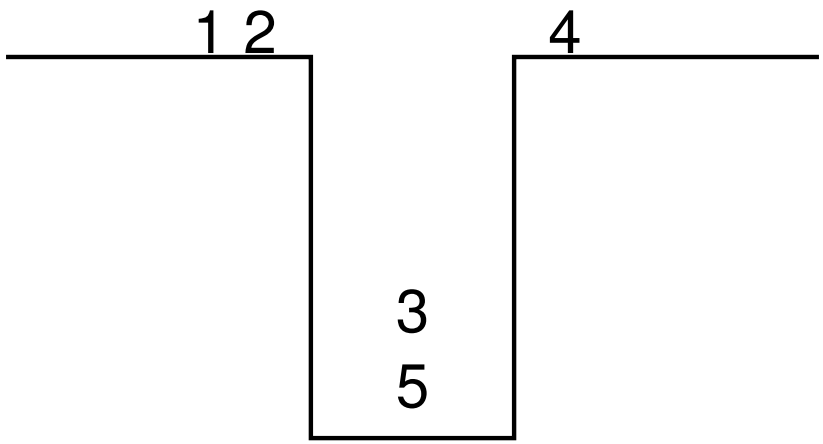
An Example

Let's stack-sort 53214



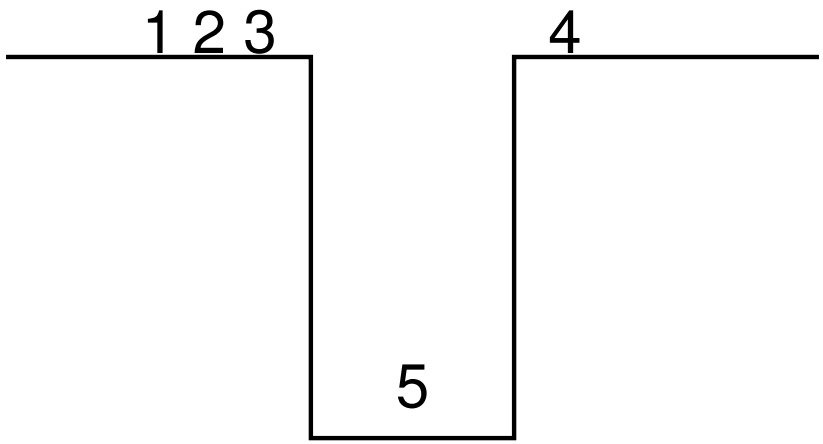
An Example

Let's stack-sort 53214



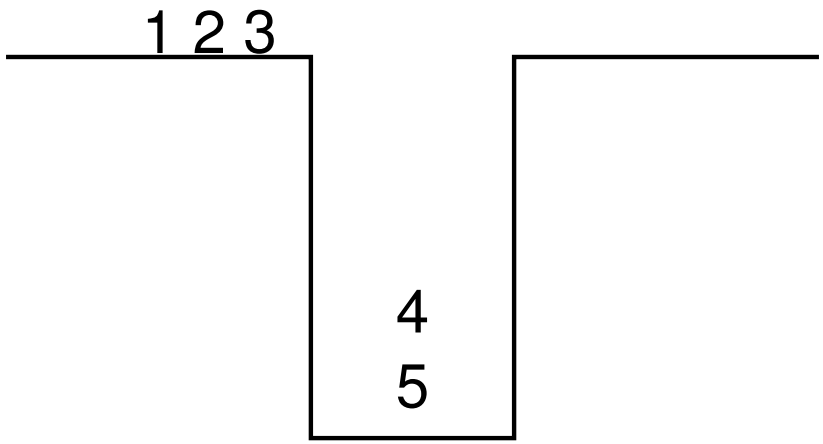
An Example

Let's stack-sort 53214



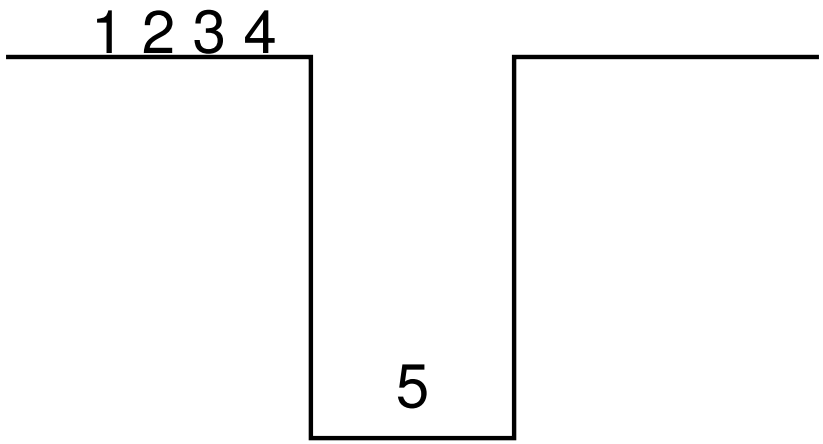
An Example

Let's stack-sort 53214



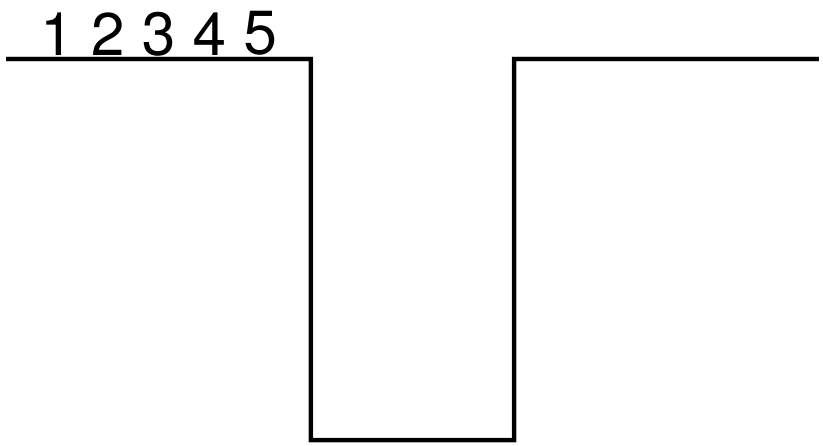
An Example

Let's stack-sort 53214



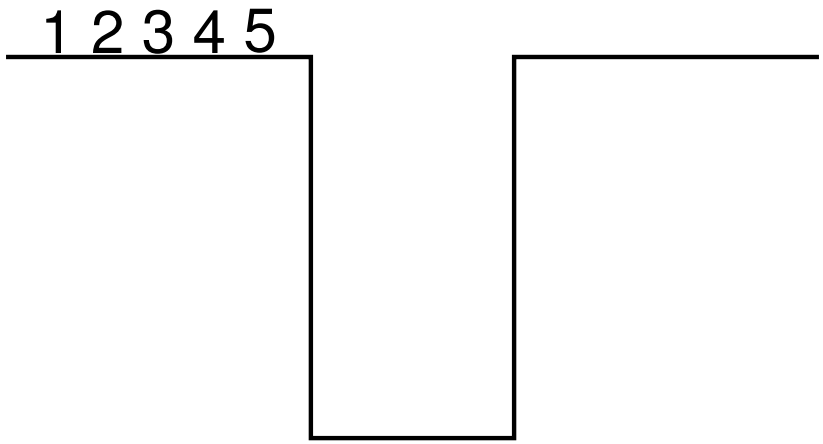
An Example

Let's stack-sort 53214



An Example

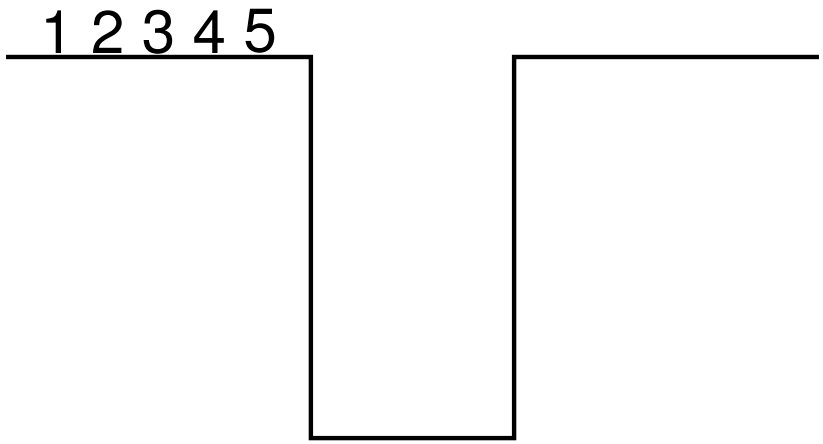
Let's stack-sort 53214



So $S(53214) = 12345$

An Example

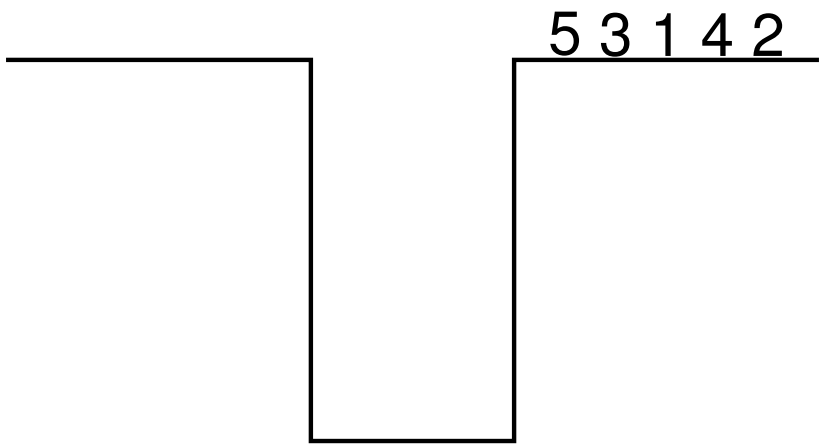
Let's stack-sort 53214



So $S(53214) = 12345$, which is great!

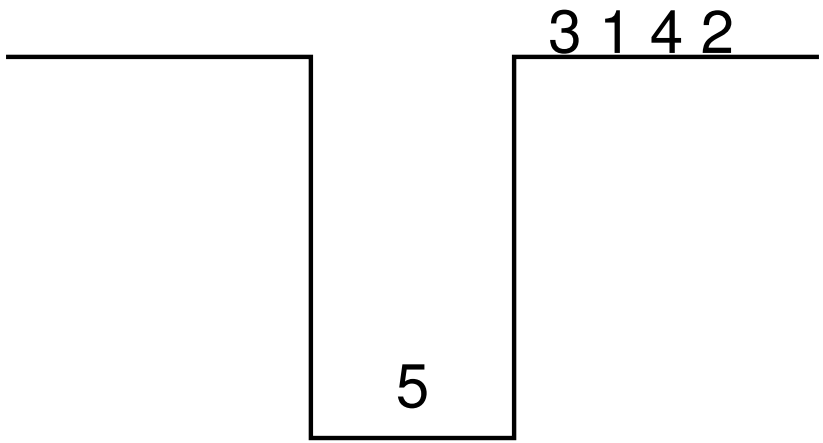
Another Example

What about 53142?



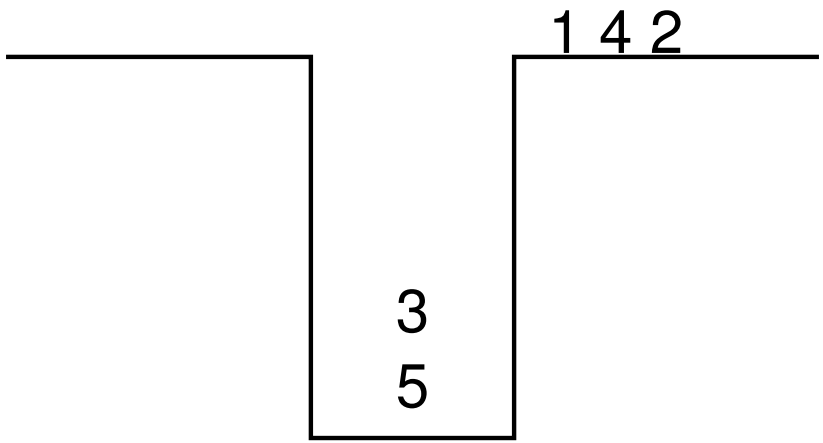
Another Example

What about 53142?



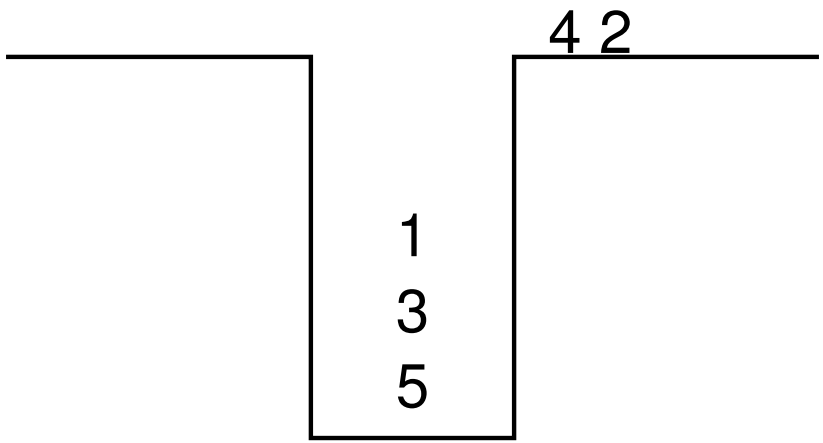
Another Example

What about 53142?



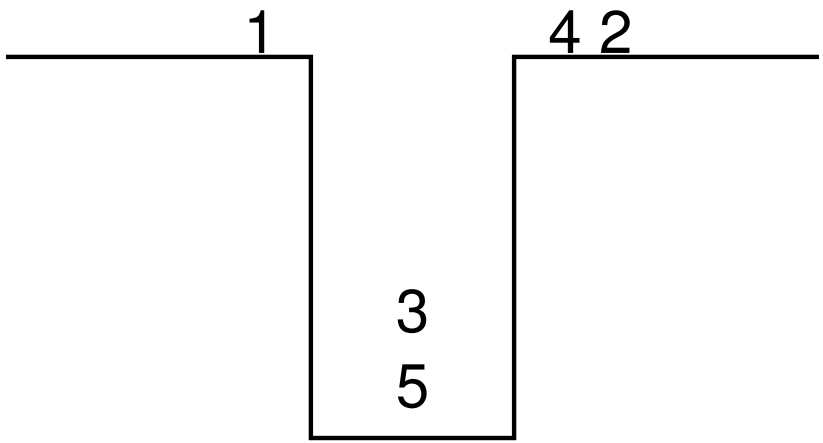
Another Example

What about 53142?



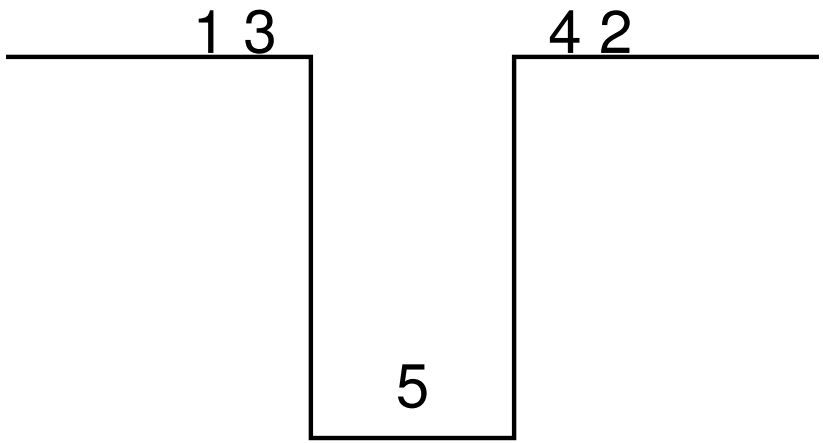
Another Example

What about 53142?



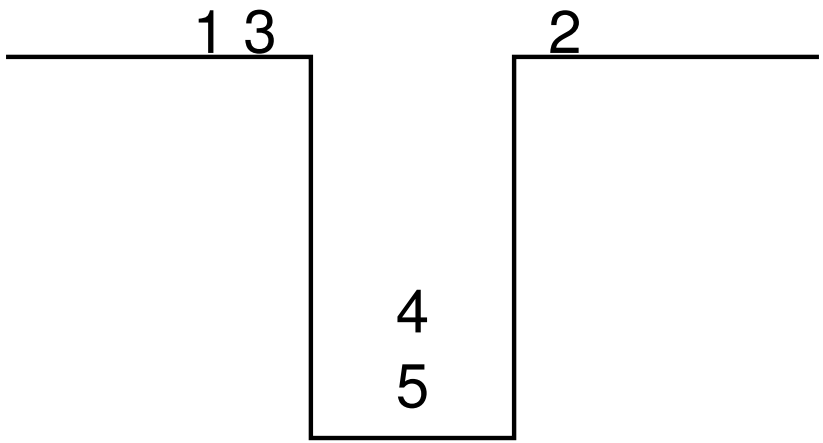
Another Example

What about 53142?



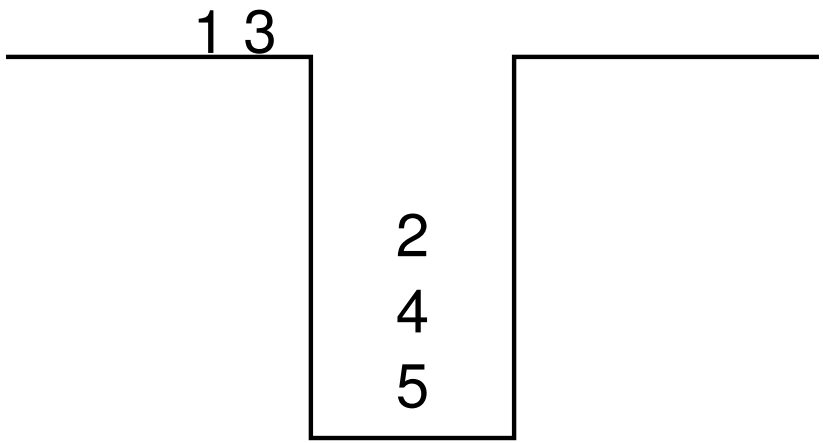
Another Example

What about 53142?



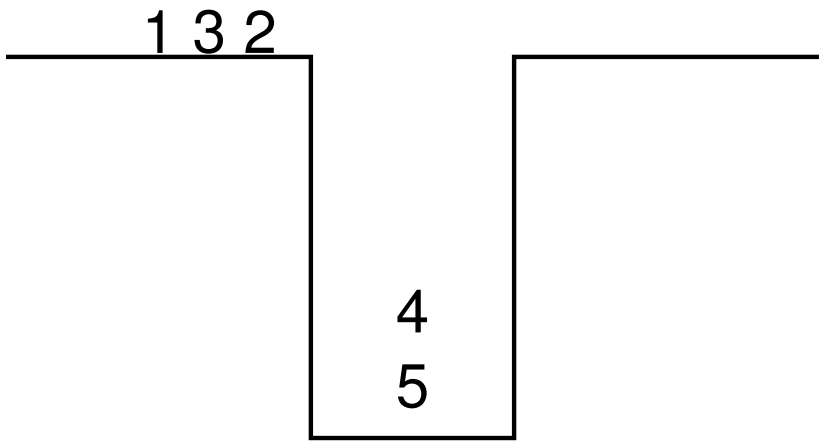
Another Example

What about 53142?



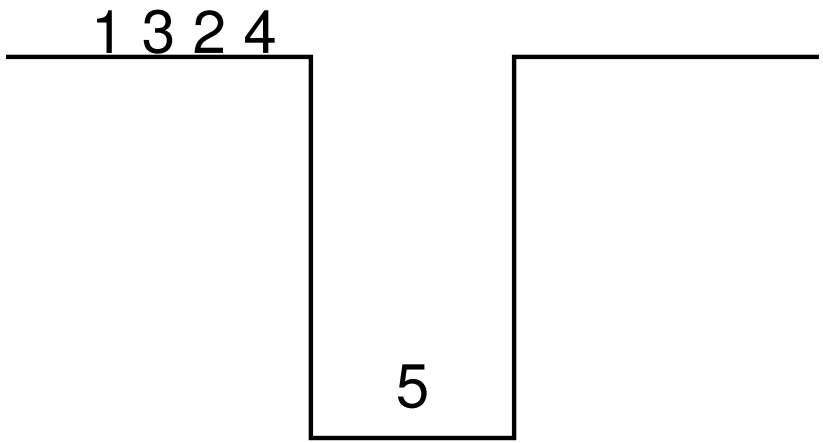
Another Example

What about 53142?



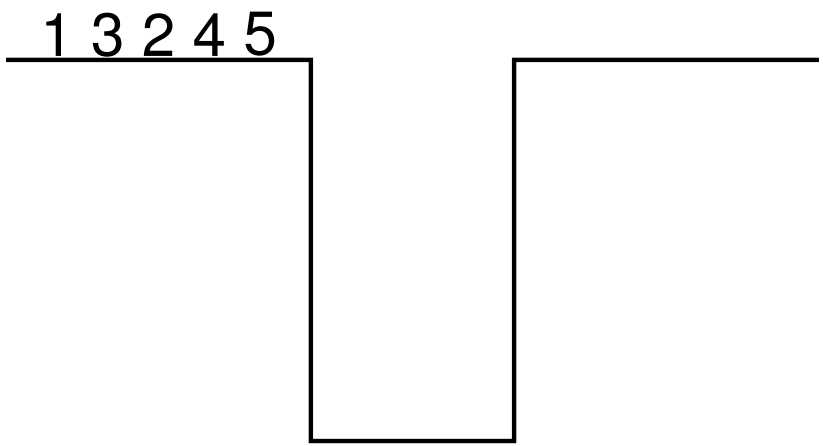
Another Example

What about 53142?



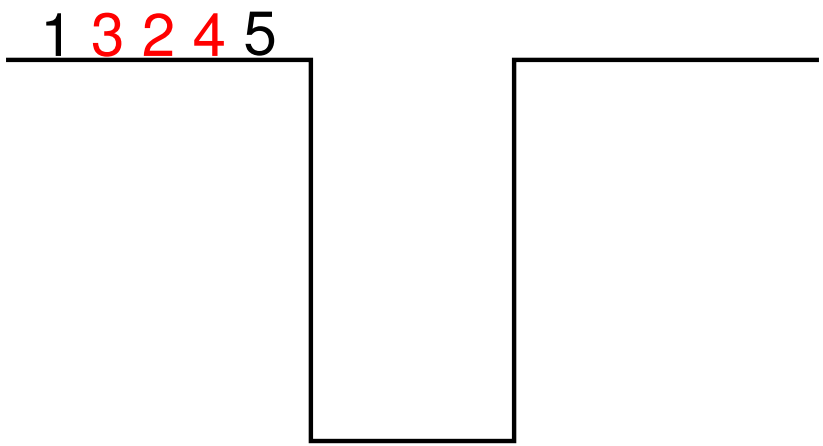
Another Example

What about 53142?



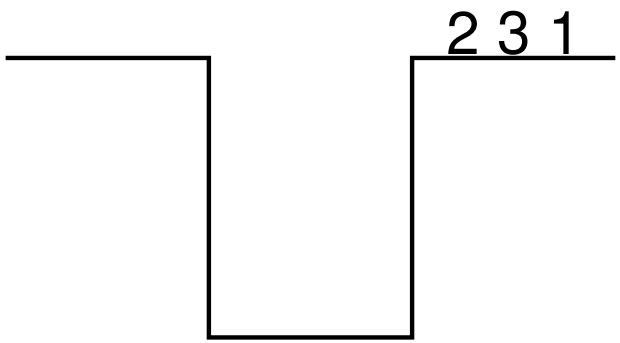
Another Example

What about 53142?



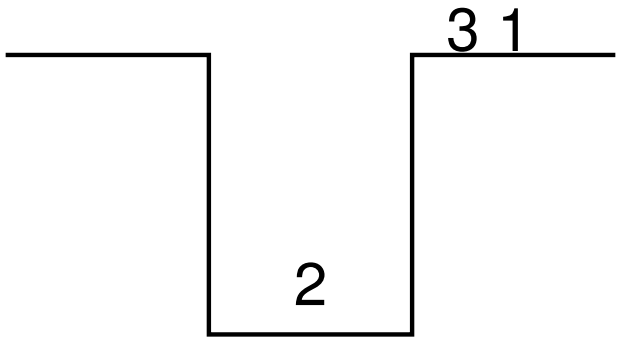
The Culprit

The culprit is the pattern 231



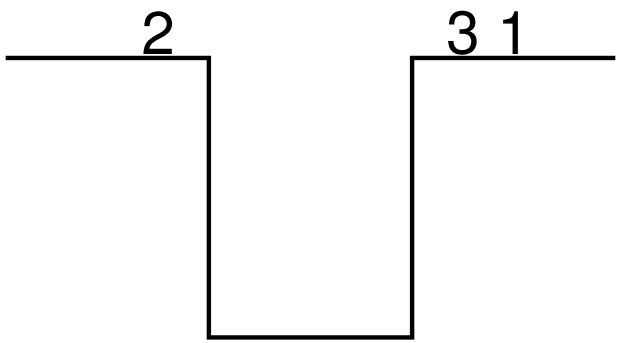
The Culprit

The culprit is the pattern 231



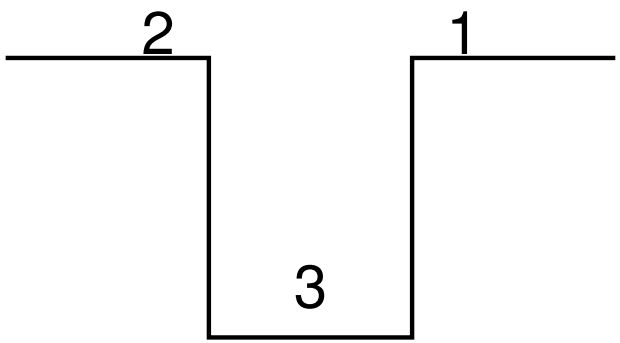
The Culprit

The culprit is the pattern 231



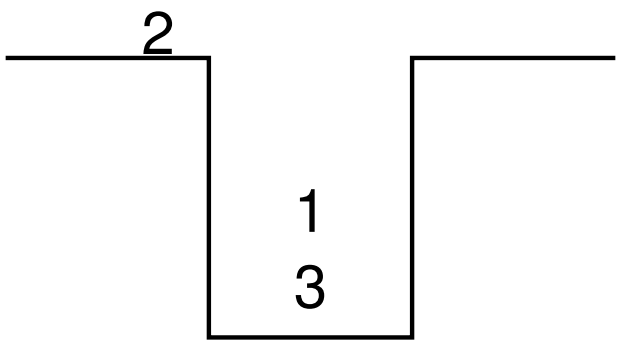
The Culprit

The culprit is the pattern 231



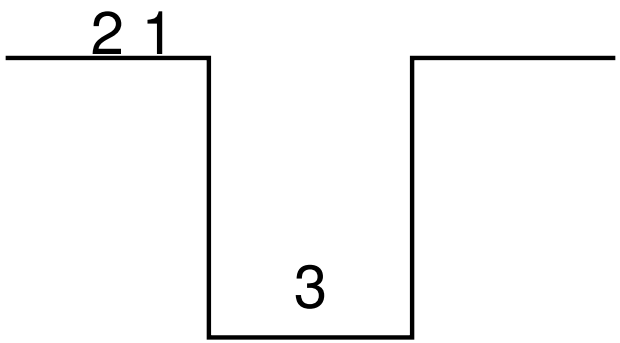
The Culprit

The culprit is the pattern 231



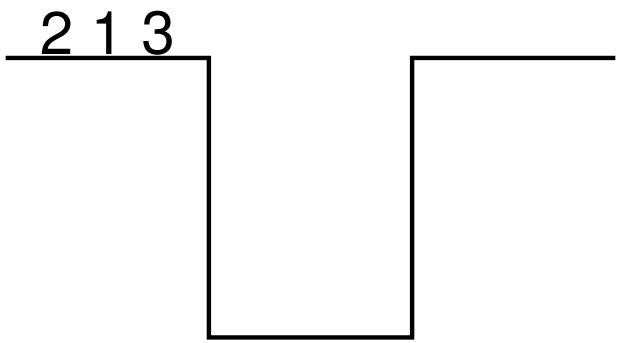
The Culprit

The culprit is the pattern 231



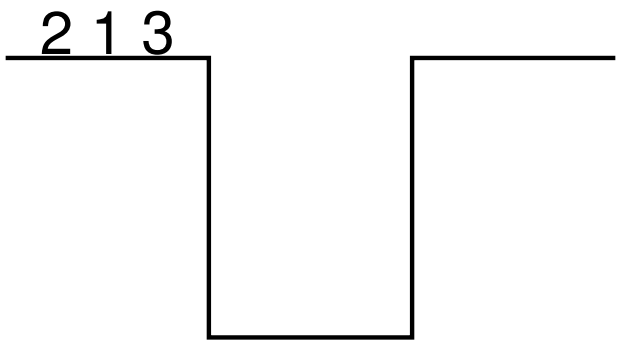
The Culprit

The culprit is the pattern 231



The Culprit

The culprit is the pattern 231



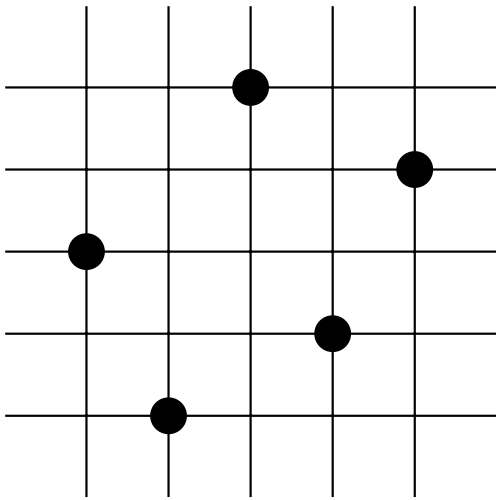
Theorem (Knuth, 1973)

A permutation is stack-sortable if and only if it avoids the classical pattern 231

Outline

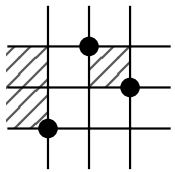
Patterns as Diagrams

- We view patterns (and permutations) as points in a Cartesian coordinate system
- The pattern 31524 is depicted as



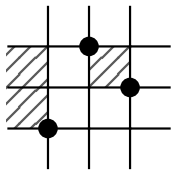
Mesh Patterns

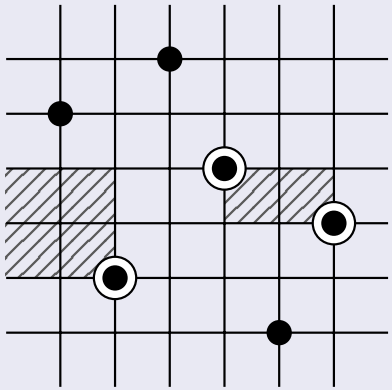
Mesh patterns (Brändèn and Claesson, 2010) may forbid certain boxes from containing elements.

The pattern  occurs in the permutation 526413.

Mesh Patterns

Mesh patterns (Brändèn and Claesson, 2010) may forbid certain boxes from containing elements.

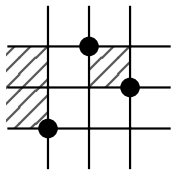
The pattern  occurs in the permutation 526413.

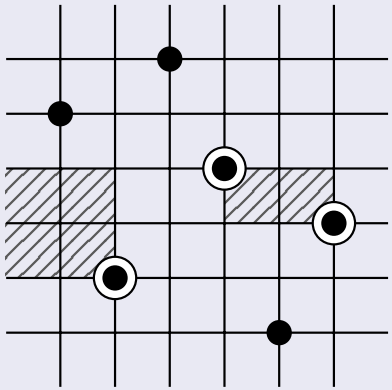


This is an occurrence

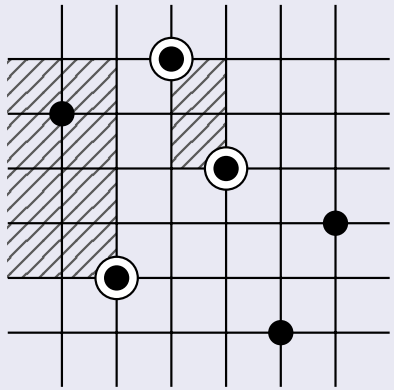
Mesh Patterns

Mesh patterns (Brändèn and Claesson, 2010) may forbid certain boxes from containing elements.

The pattern  occurs in the permutation 526413.



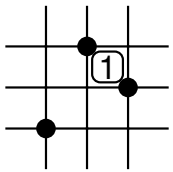
This is an occurrence



This is not an occurrence

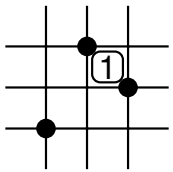
Marked Mesh Patterns

Marked mesh patterns (Úlfarsson, 2011) allow finer control of the number of elements in a region.

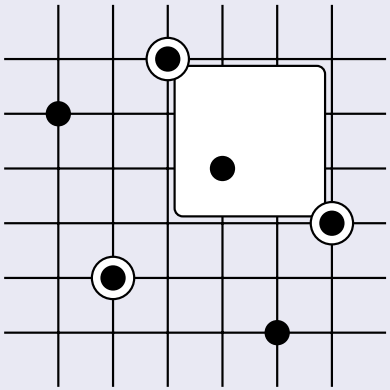
The pattern  occurs in the permutation 526413.

Marked Mesh Patterns

Marked mesh patterns (Úlfarsson, 2011) allow finer control of the number of elements in a region.

The pattern 

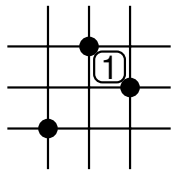
occurs in the permutation 526413.

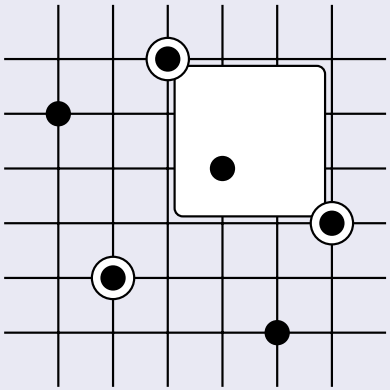


This is an occurrence

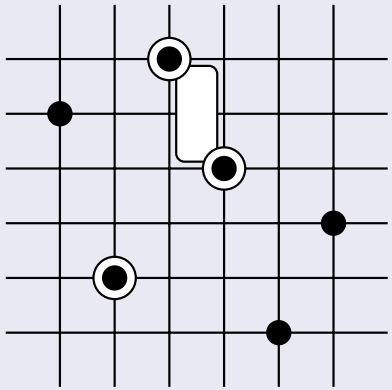
Marked Mesh Patterns

Marked mesh patterns (Úlfarsson, 2011) allow finer control of the number of elements in a region.

The pattern  occurs in the permutation 526413.



This is an occurrence

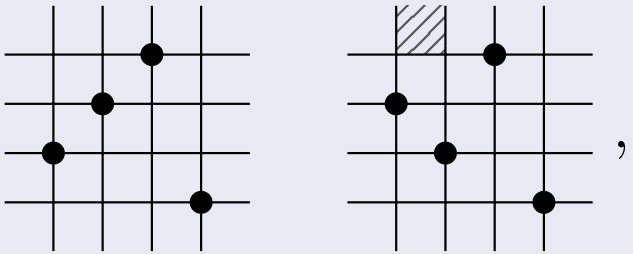


This is not an occurrence

West-2-Stack-Sortable

Theorem (West, 1993)

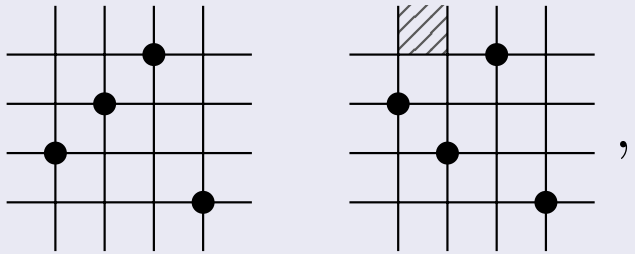
A permutation is sortable with two passes through a stack if and only if it avoids the following two mesh patterns



West-2-Stack-Sortable

Theorem (West, 1993)

A permutation is sortable with two passes through a stack if and only if it avoids the following two mesh patterns



or using Knuth's result

$$S^{-1}(\mathrm{Av}(231)) = \mathrm{Av}\left(\ \ ,\ \ \right)$$

Outline

An Algorithm to Find Preimages

- In 2012 Claesson and Úlfarsson introduced an algorithm, `ClassicalPreim`, to find preimages of avoidance classes under the stack-sort operator.
- Given a classical pattern p (the target), the algorithm finds a set M of marked mesh patterns such that

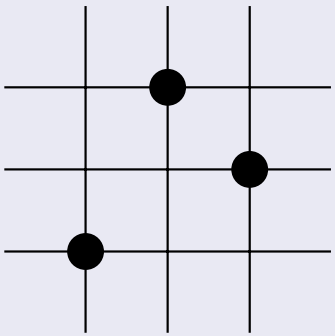
$$S^{-1}(\mathrm{Av}(p)) = \mathrm{Av}(M)$$

- The algorithm proceeds in two steps
  1. Generate possible classical pattern *candidates*
  2. Add appropriate shadings and markings to each candidate, if possible

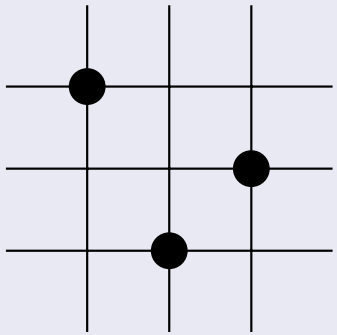
Candidates

Identify patterns that can possibly become the target pattern after stack sorting

Target



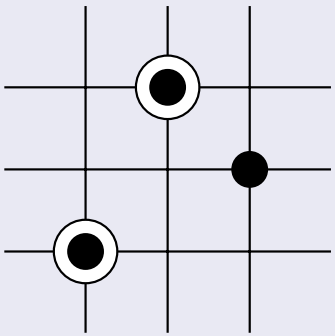
Candidate



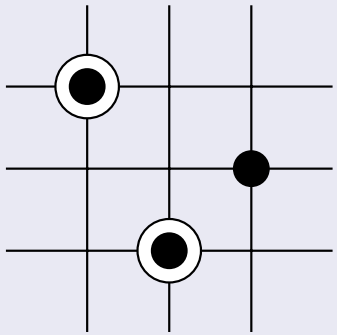
Candidates

Identify patterns that can possibly become the target pattern after stack sorting

Target



Candidate

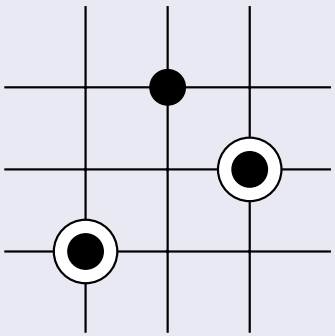


Inversions can become non-inversions

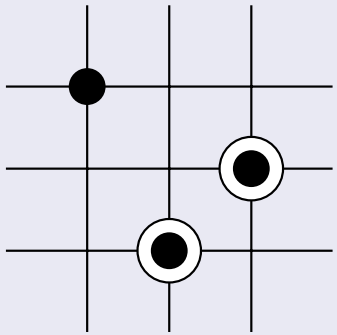
Candidates

Identify patterns that can possibly become the target pattern after stack sorting

Target



Candidate

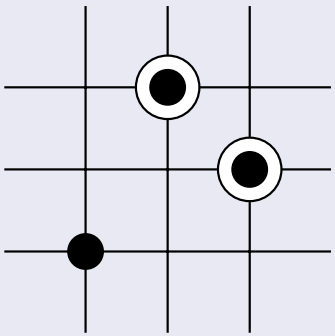


Non-inversions must remain non-inversions

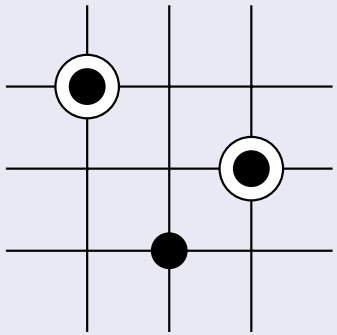
Candidates

Identify patterns that can possibly become the target pattern after stack sorting

Target



Candidate

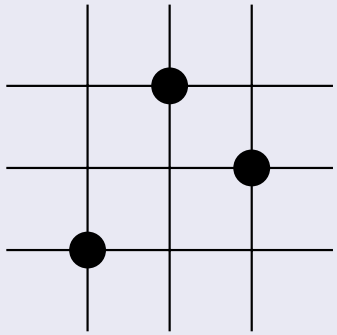


Inversions can stay inverted

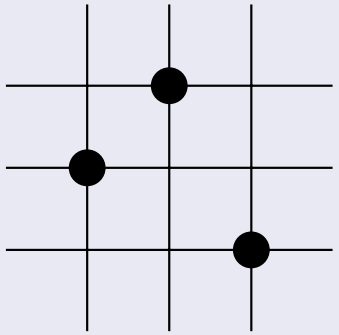
Candidates

Identify patterns that can possibly become the target pattern after stack sorting

Target



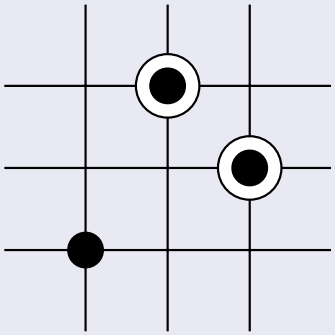
Not a candidate



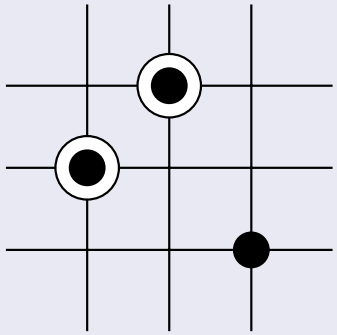
Candidates

Identify patterns that can possibly become the target pattern after stack sorting

Target



Not a candidate

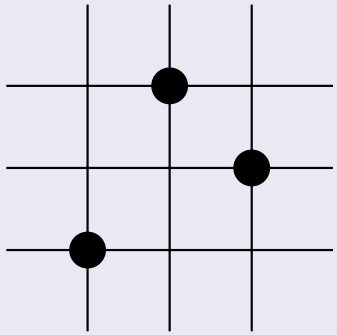


Non-inversons cannot become inversions

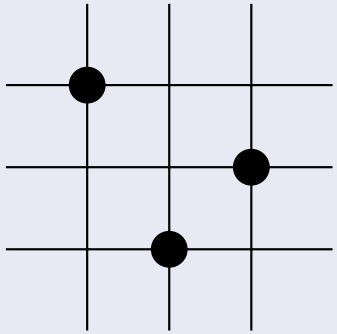
Shading and Marking

Try to add shadings and markings to the candidate so that it becomes the target after stack sorting

Target



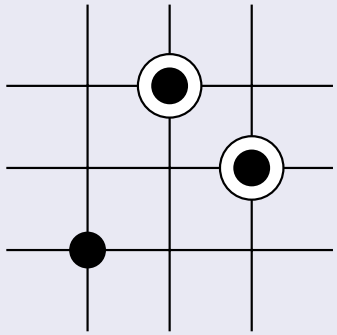
Candidate



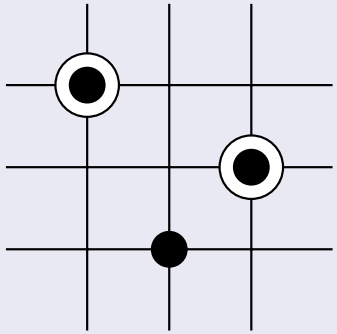
Shading and Marking

Try to add shadings and markings to the candidate so that it becomes the target after stack sorting

Target



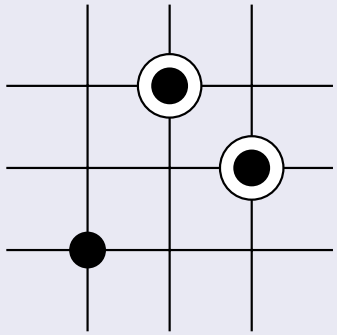
Candidate



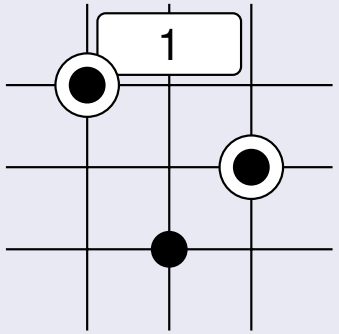
Shading and Marking

Try to add shadings and markings to the candidate so that it becomes the target after stack sorting

Target



Candidate

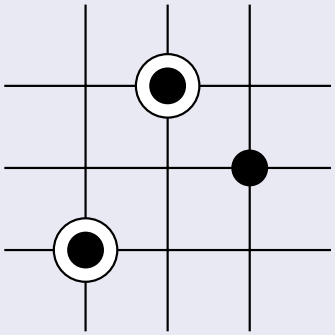


Inversion must be maintained

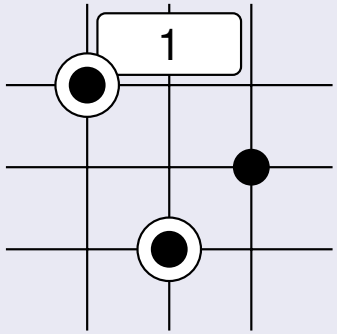
Shading and Marking

Try to add shadings and markings to the candidate so that it becomes the target after stack sorting

Target



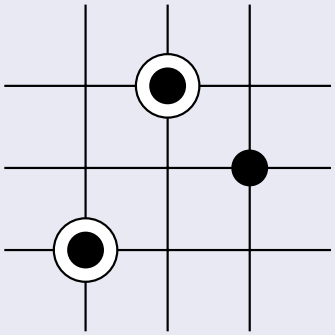
Candidate



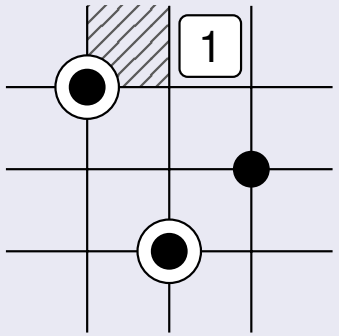
Shading and Marking

Try to add shadings and markings to the candidate so that it becomes the target after stack sorting

Target



Candidate

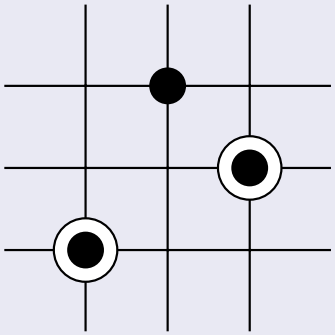


Inversion must be fixed

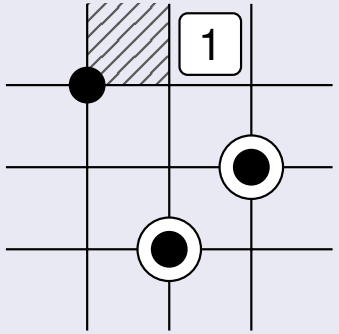
Shading and Marking

Try to add shadings and markings to the candidate so that it becomes the target after stack sorting

Target



Candidate

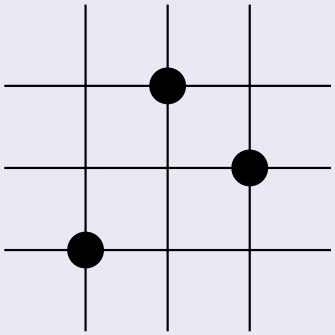


Non-inversion stays the same

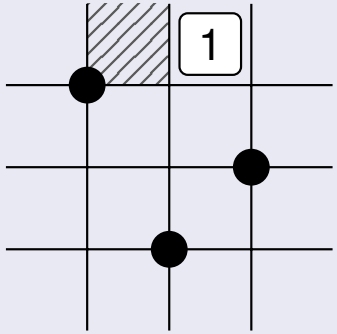
Shading and Marking

Try to add shadings and markings to the candidate so that it becomes the target after stack sorting

Target

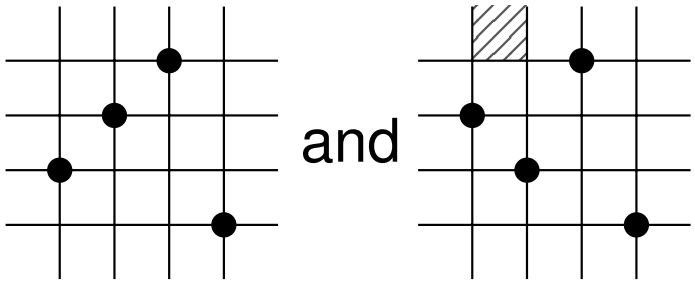


Candidate



West's Results

- By finding the permutations that avoid 231 after a single pass through stack sort we obtain the West-2-stack-sortable permutations
- This algorithm can therefore be used to reproduce West's results. By setting the target as 231 we obtain:



- The algorithm cannot be applied to these results to obtain a description of West-3-stack-sortable permutations

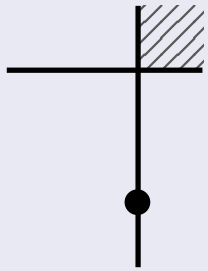
Outline

Extension

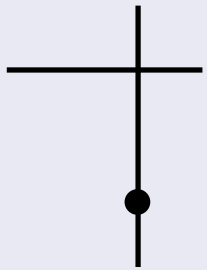
- We extend ClassicalPreim to handle mesh pattern targets with a single shaded rectangular region, with some additional constraints
- The extended algorithm, MeshPreim, proceeds as follows
  1. Apply ClassicalPreim to the underlying classical pattern
  2. For each pattern obtained from step 1, add appropriate shadings, markings and *decorations*

MeshPreim

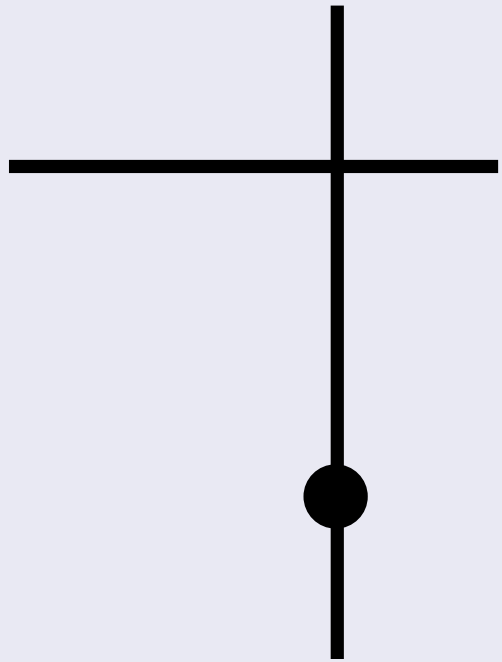
Target



Output from ClassicalPreim

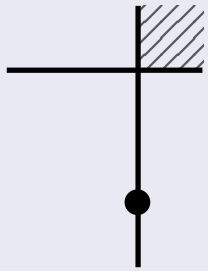


MeshPreim

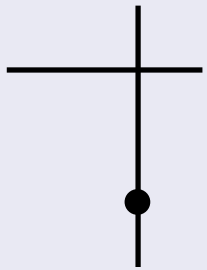


MeshPreim

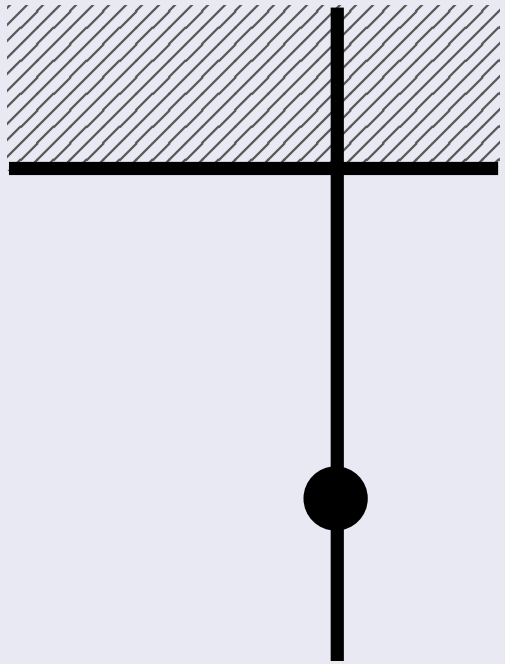
Target



Output from ClassicalPreim

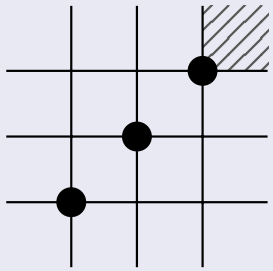


MeshPreim

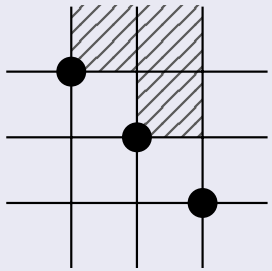


Example

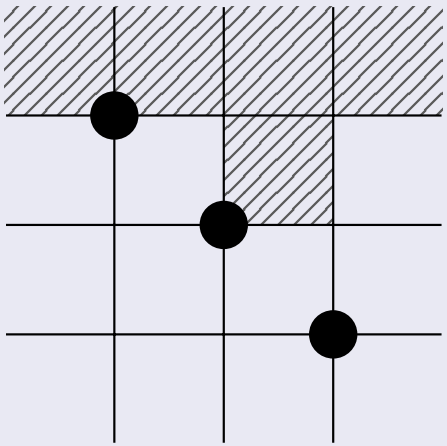
Target



Output from ClassicalPreim

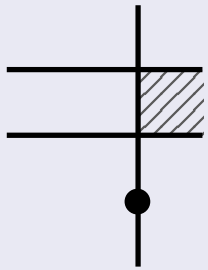


MeshPreim

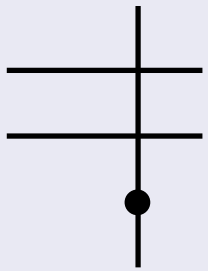


MeshPreim

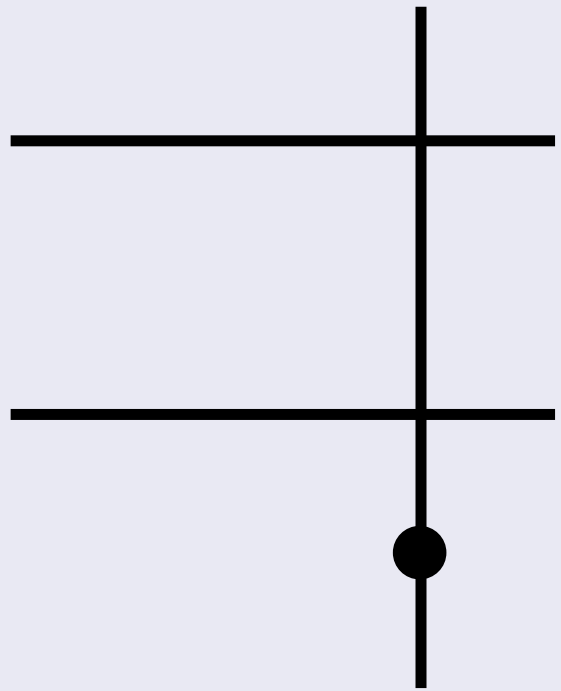
Target



Output from ClassicalPreim

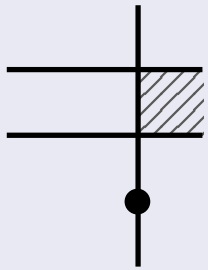


MeshPreim - Case 1

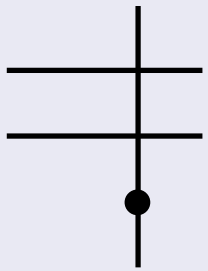


MeshPreim

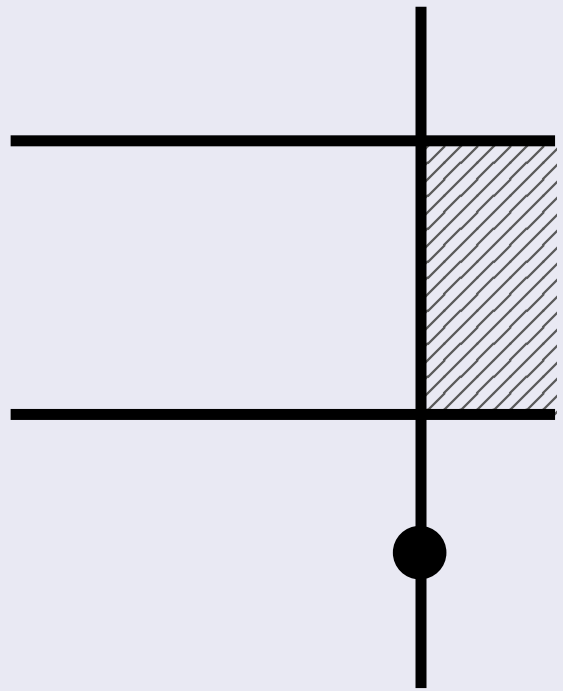
Target



Output from ClassicalPreim

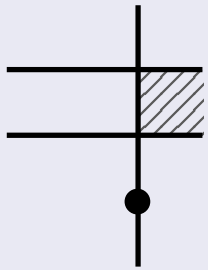


MeshPreim - Case 1

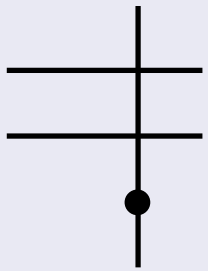


MeshPreim

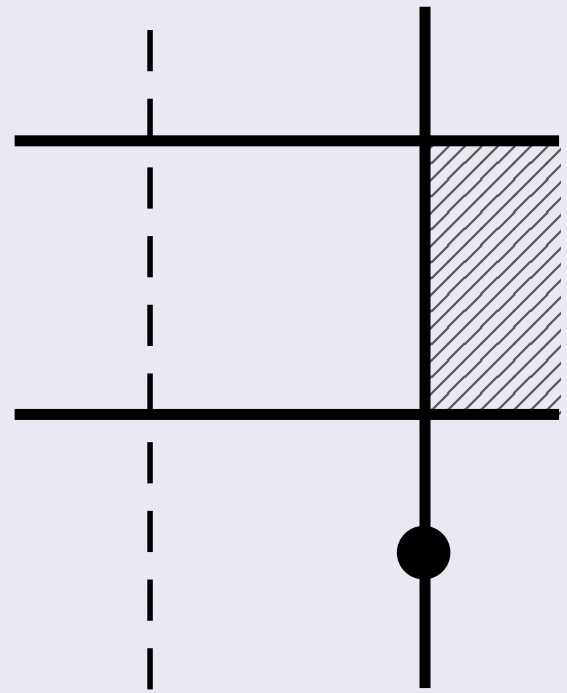
Target



Output from ClassicalPreim

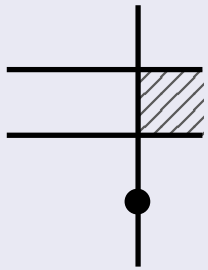


MeshPreim - Case 1

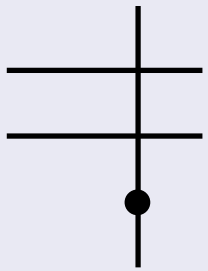


MeshPreim

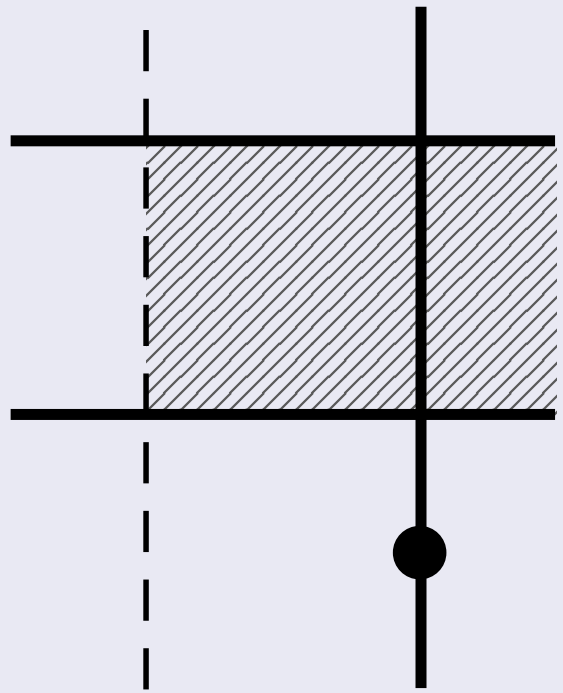
Target



Output from ClassicalPreim

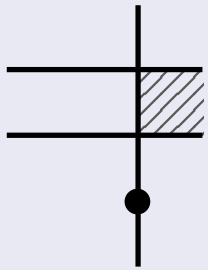


MeshPreim - Case 1

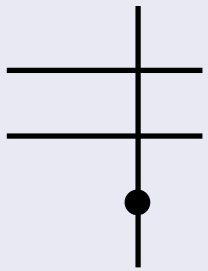


MeshPreim

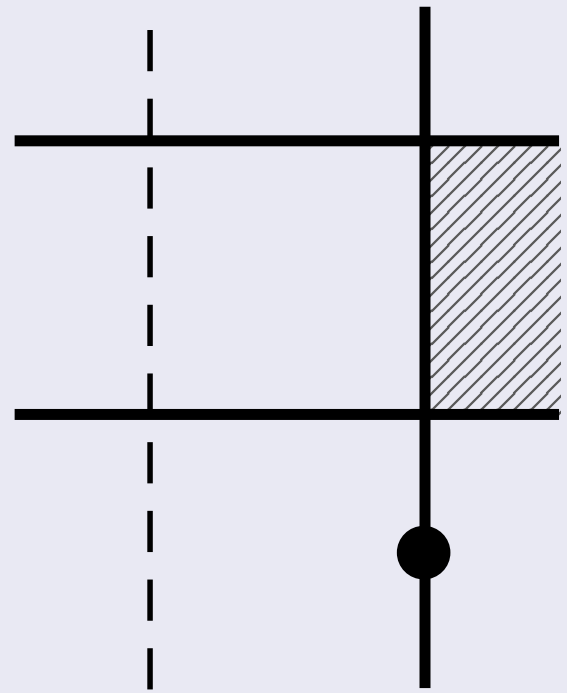
Target



Output from ClassicalPreim

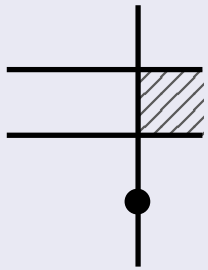


MeshPreim - Case 2

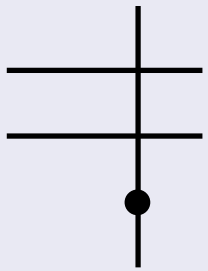


MeshPreim

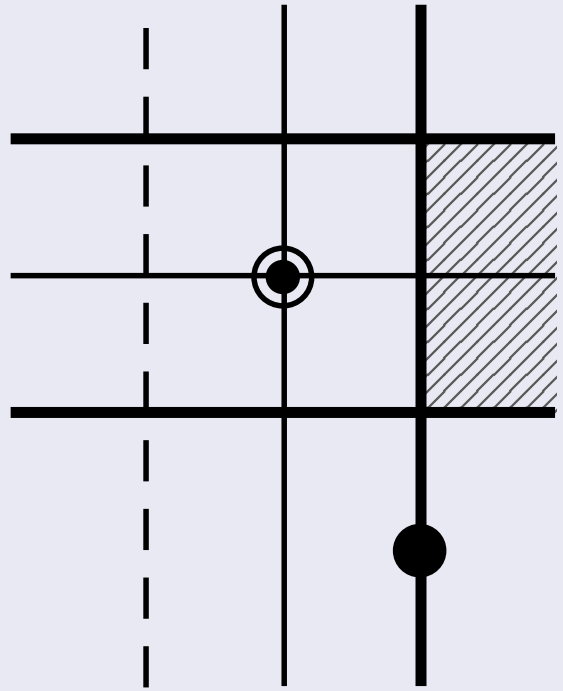
Target



Output from ClassicalPreim

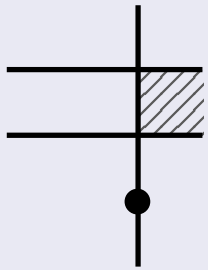


MeshPreim - Case 2

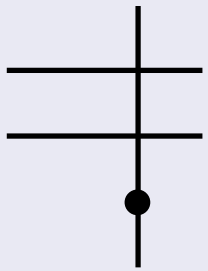


MeshPreim

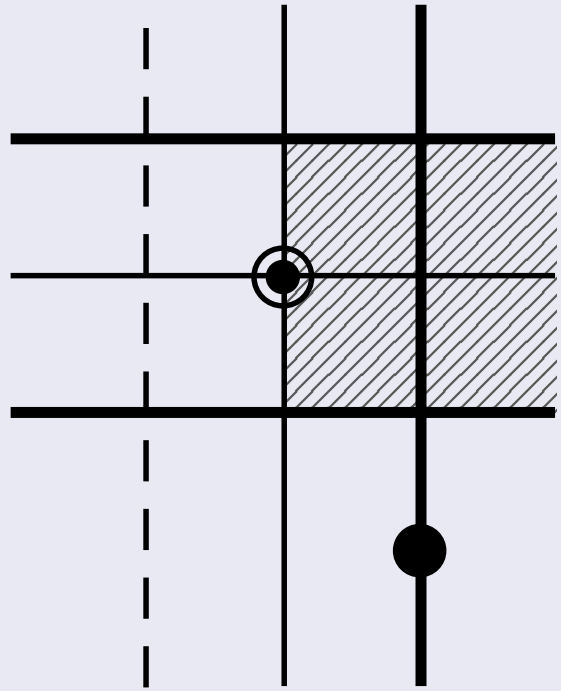
Target



Output from ClassicalPreim

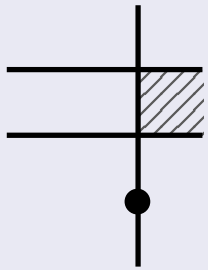


MeshPreim - Case 2

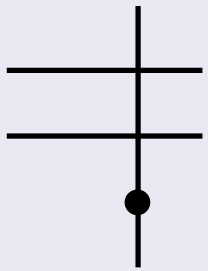


MeshPreim

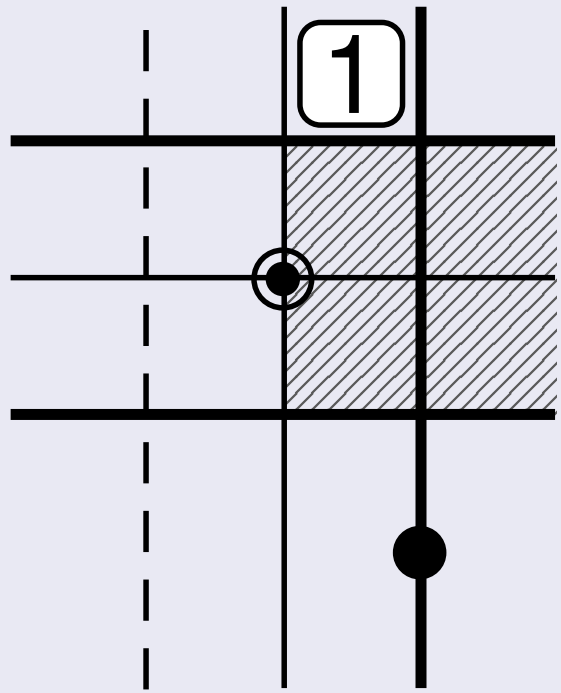
Target



Output from ClassicalPreim

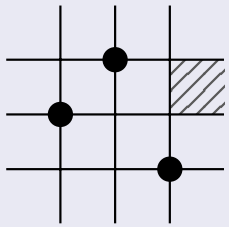


MeshPreim - Case 2

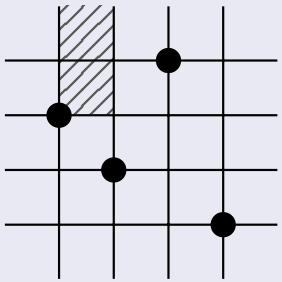


Example

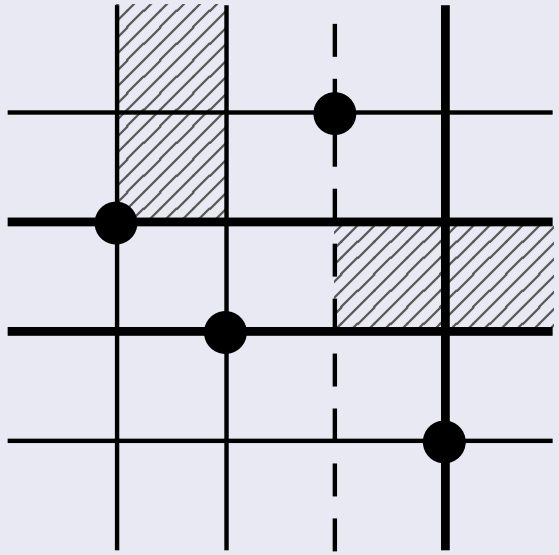
Target



Output from ClassicalPreim

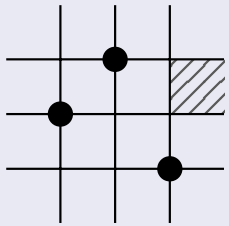


MeshPreim - Case 1

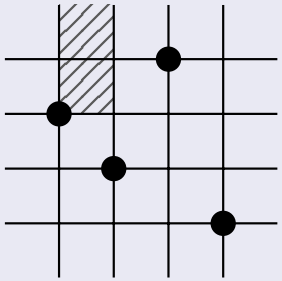


Example

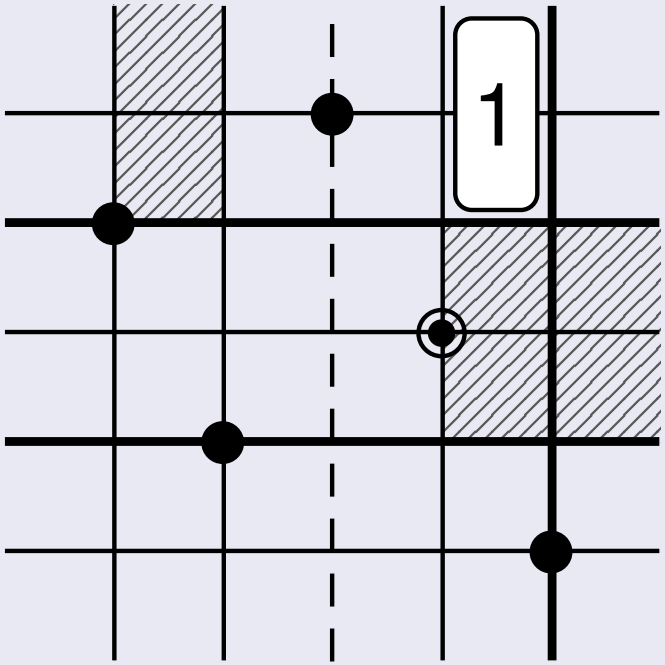
Target



Output from ClassicalPreim

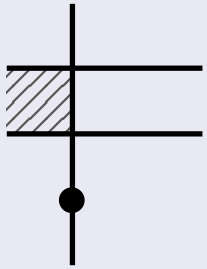


MeshPreim - Case 2

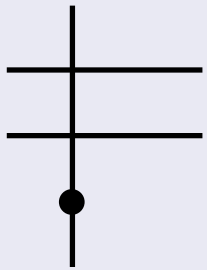


MeshPreim

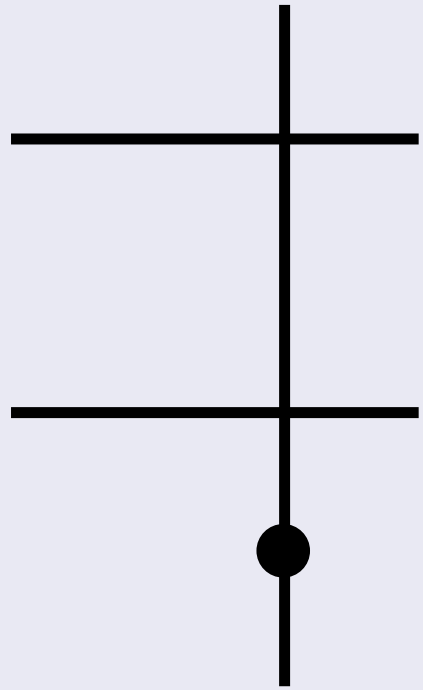
Target



Output from ClassicalPreim

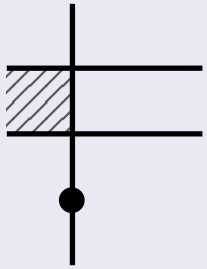


MeshPreim - Case 1

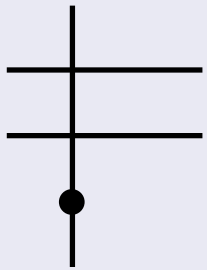


MeshPreim

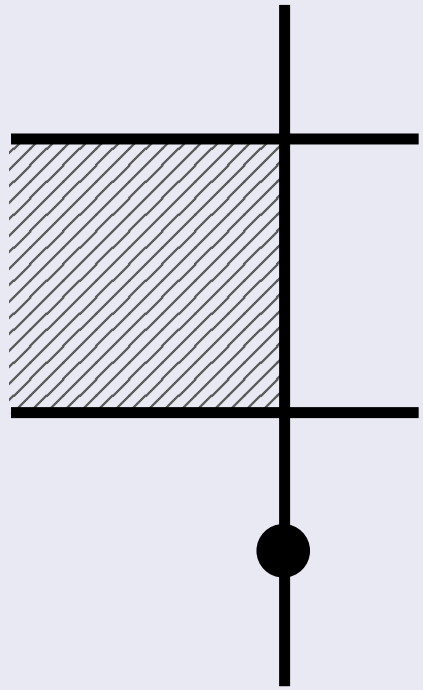
Target



Output from ClassicalPreim

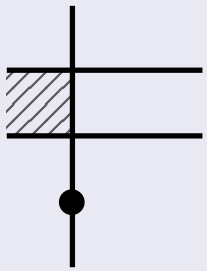


MeshPreim - Case 1

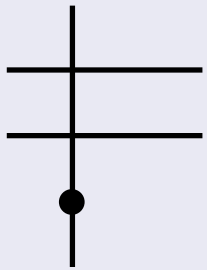


MeshPreim

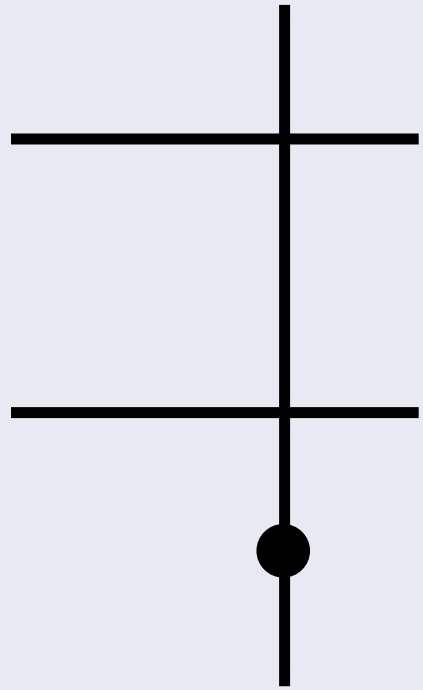
Target



Output from ClassicalPreim

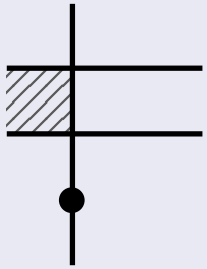


MeshPreim - Case 2

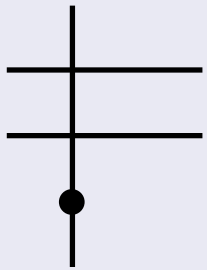


MeshPreim

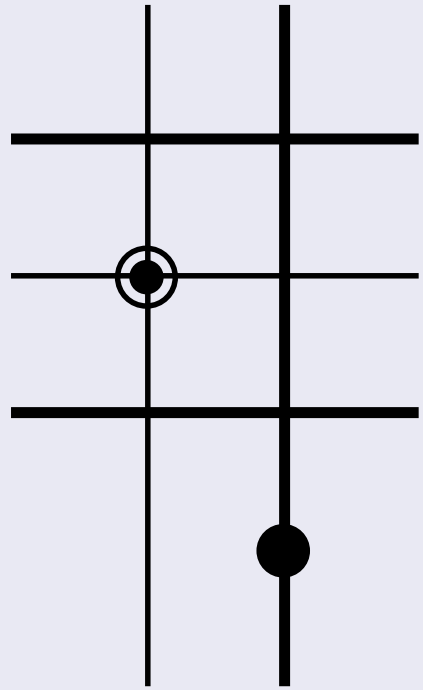
Target



Output from ClassicalPreim

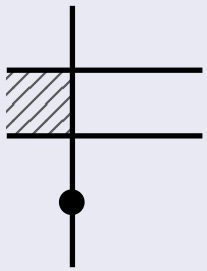


MeshPreim - Case 2

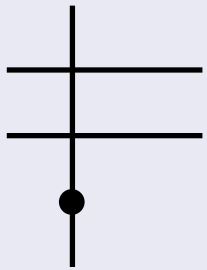


MeshPreim

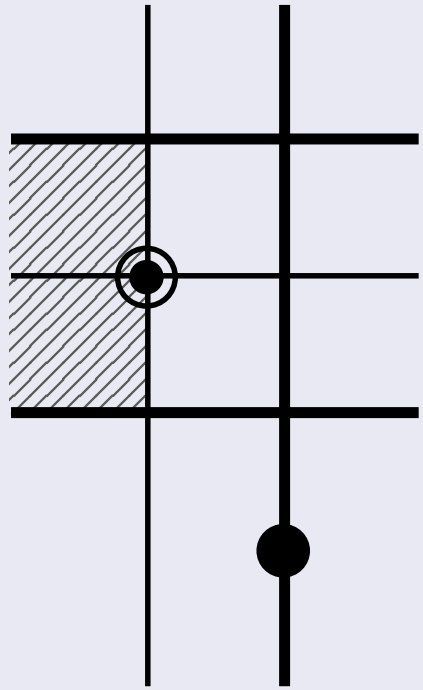
Target



Output from ClassicalPreim

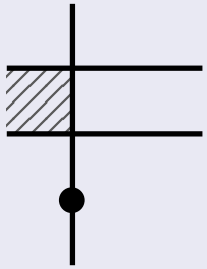


MeshPreim - Case 2

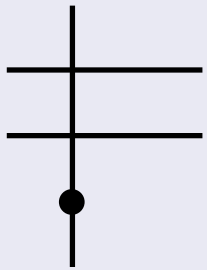


MeshPreim

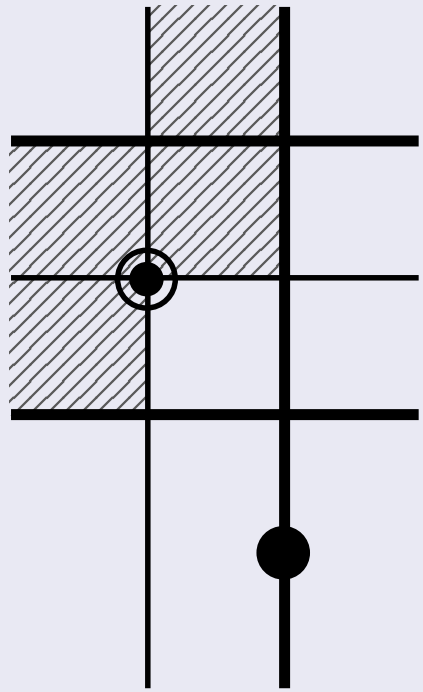
Target



Output from ClassicalPreim

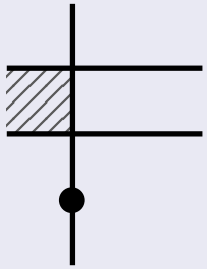


MeshPreim - Case 2

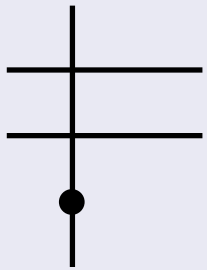


MeshPreim

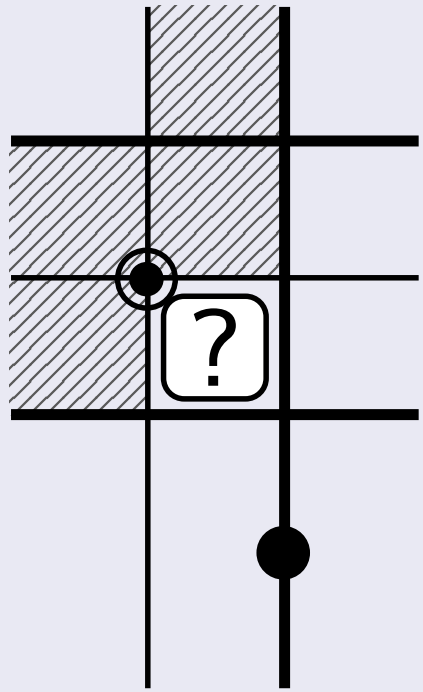
Target



Output from ClassicalPreim

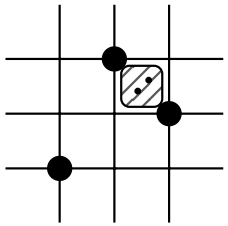


MeshPreim - Case 2



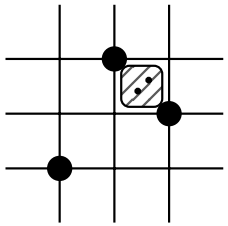
Decorated Patterns

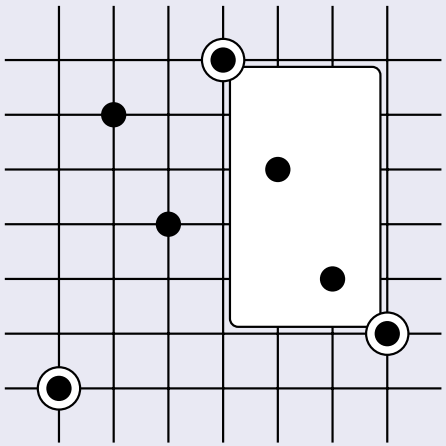
Decorated patterns (Úlfarsson, 2012) allow us to forbid occurrences of patterns inside boxes

The pattern  occurs in the permutation 1647532.

Decorated Patterns

Decorated patterns (Úlfarsson, 2012) allow us to forbid occurrences of patterns inside boxes

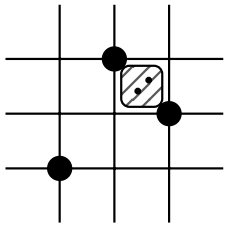
The pattern  occurs in the permutation 1647532.

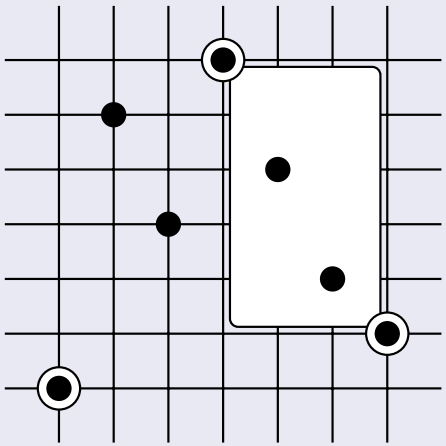


This is an occurrence

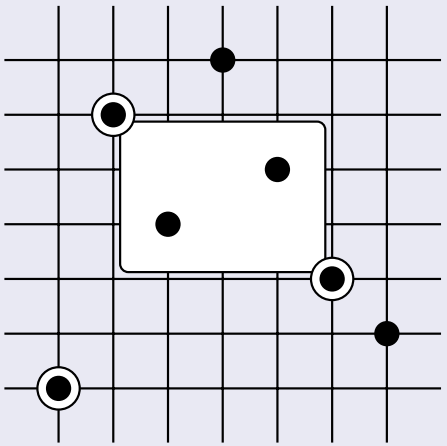
Decorated Patterns

Decorated patterns (Úlfarsson, 2012) allow us to forbid occurrences of patterns inside boxes

The pattern  occurs in the permutation 1647532.



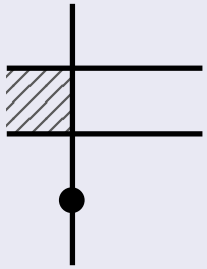
This is an occurrence



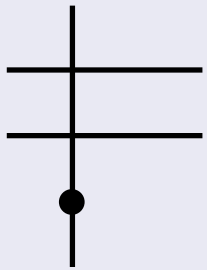
This is not an occurrence

MeshPreim

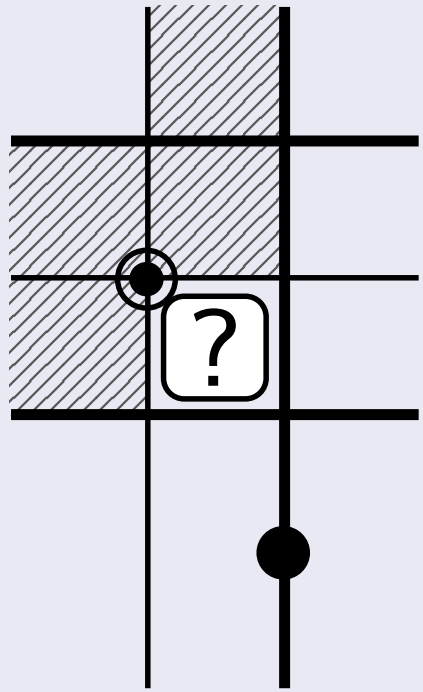
Target



Output from ClassicalPreim

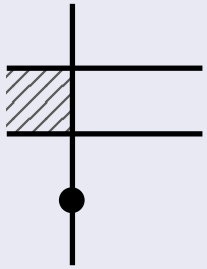


MeshPreim - Case 2

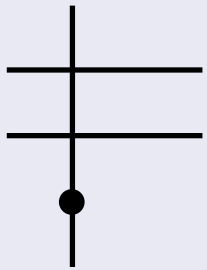


MeshPreim

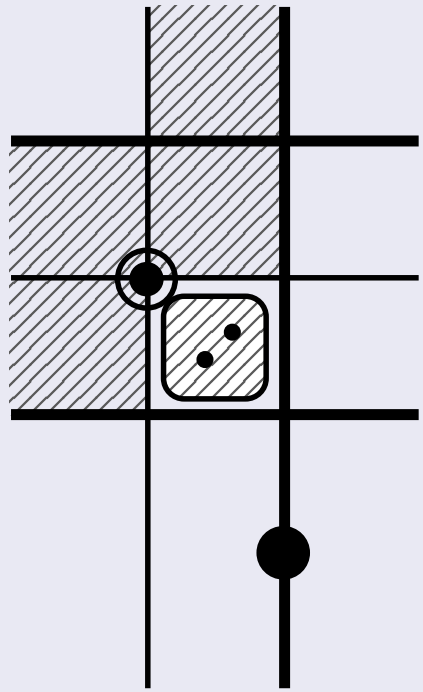
Target



Output from ClassicalPreim



MeshPreim - Case 2



The Extended Algorithm

- In 2012, Úlfarsson described West-3-stack-sortable permutations (by hand)
- By applying MeshPreim to West's results, we obtain a description of West-3-stack-sortable permutations
- By using ClassicalPreim and MeshPreim we can therefore fully automate the process of describing stack sortable, West-2-stack-sortable and West-3-stack-sortable permutations.

Outline

Queue sort

Definition

Let $R(\pi)$ denote the sequence of all left to right maxima of π. We then define queue sort as

$$Q(\pi) = M(R(\pi), \pi \setminus R(\pi)),$$

where $M(\varepsilon, \alpha) = M(\alpha, \varepsilon) = \alpha$ and

$$M(k\alpha, m\beta) = \begin{cases} kM(\alpha, m\beta) & \text{if } k < m, \\ mM(k\alpha, \beta) & \text{otherwise} \end{cases}$$

A Linear Time Algorithm for Avoidance

Theorem (Yours truly & Úlfarsson, 2012)

A permutation π avoids 4312 *if and only if*

$$(S \circ r \circ c \circ Q)(\pi) = \mathrm{id}$$

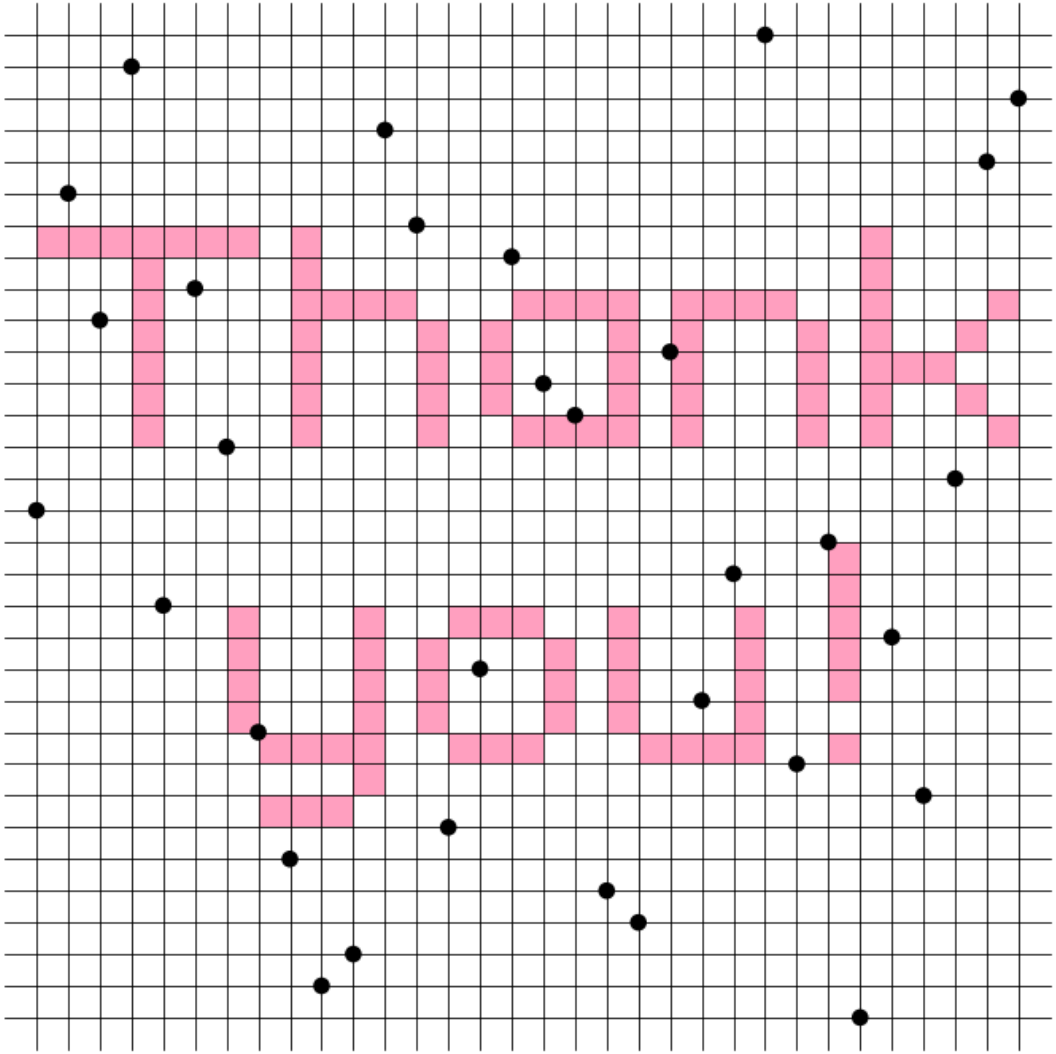
A Linear Time Algorithm for Avoidance

Theorem (Yours truly & Úlfarsson, 2012)

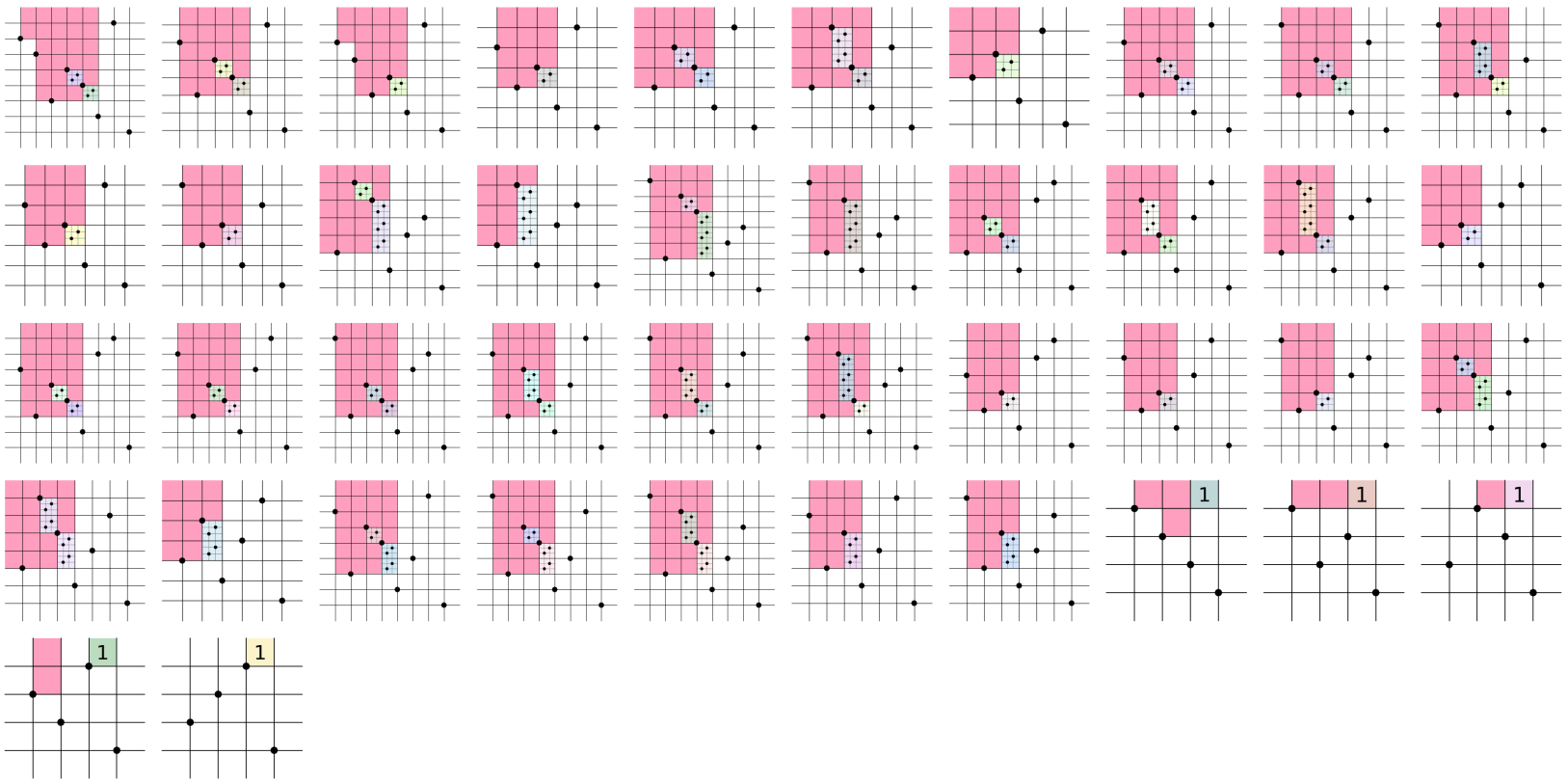
A permutation π avoids 4312 *if and only if*

$$(S \circ r \circ c \circ Q)(\pi) = \mathrm{id}$$

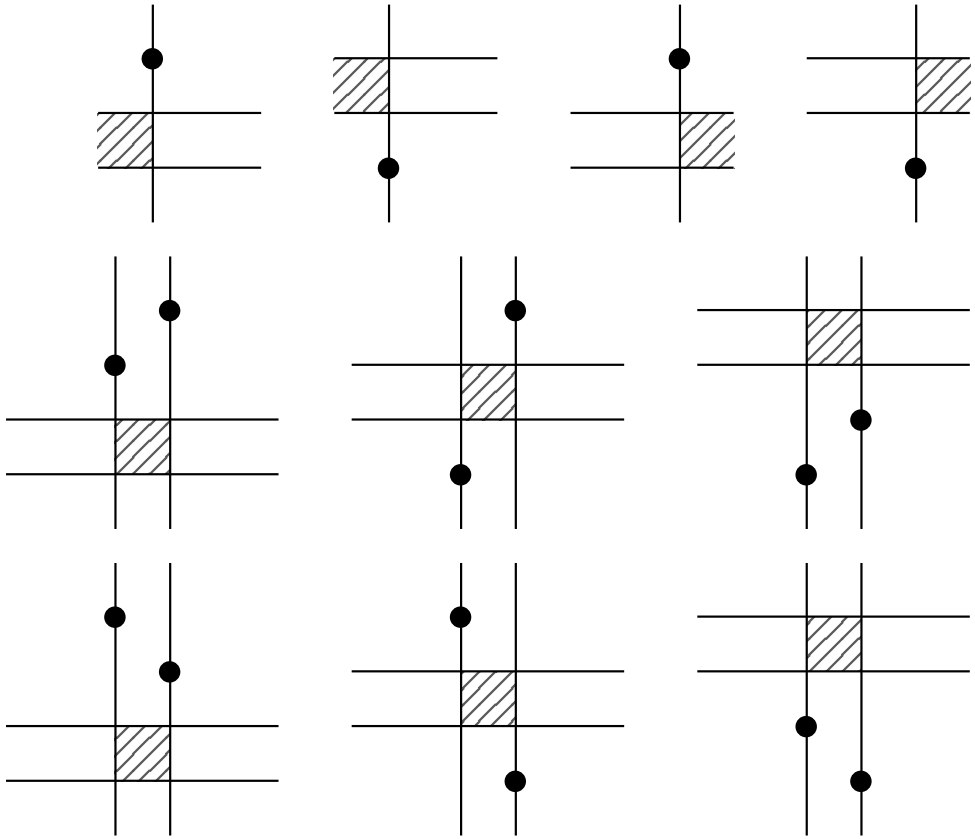
This theorem gives a linear time algorithm to check if 4312 (or any of its symmetries) occurs in a permutation.



West-3-Stack-Sortable

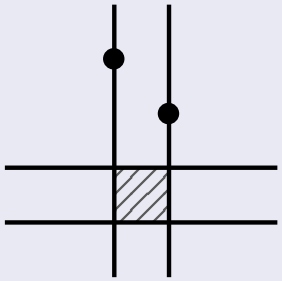


The Constraints

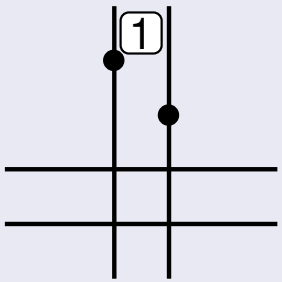


MeshPreim

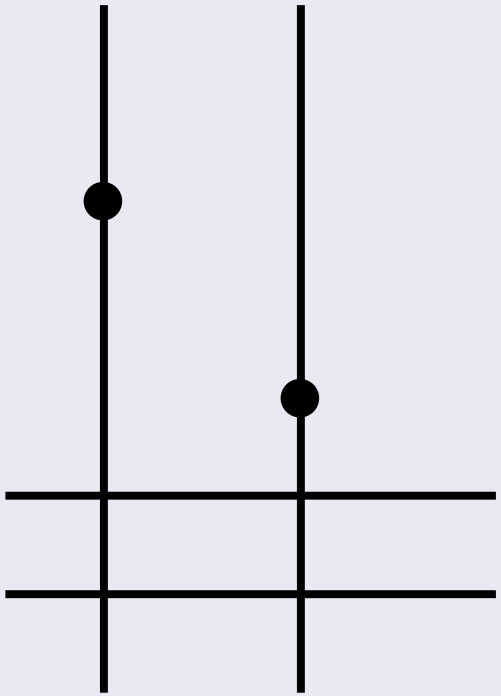
Target



Output from ClassicalPreim

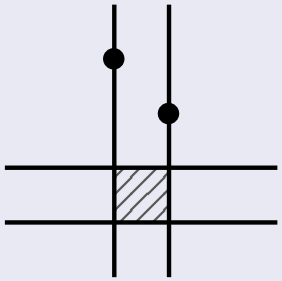


MeshPreim - Case 1

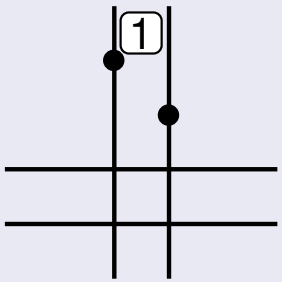


MeshPreim

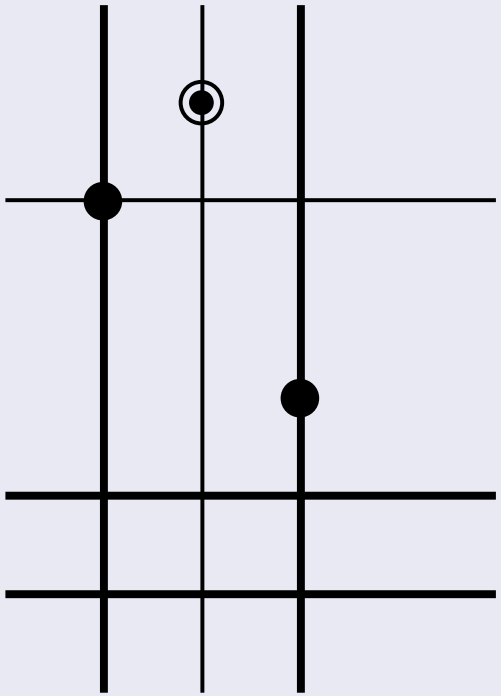
Target



Output from ClassicalPreim

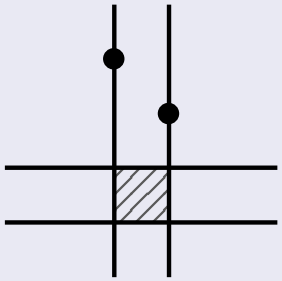


MeshPreim - Case 1

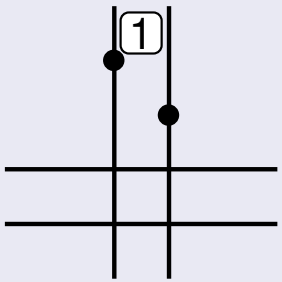


MeshPreim

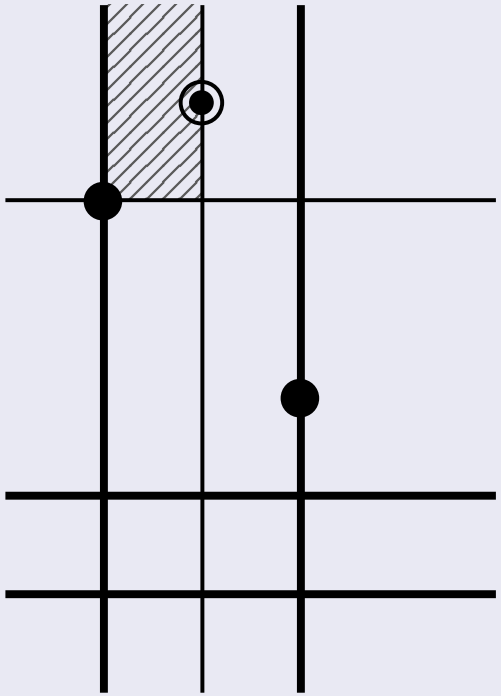
Target



Output from ClassicalPreim

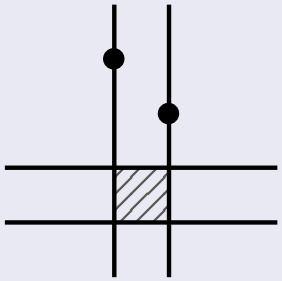


MeshPreim - Case 1

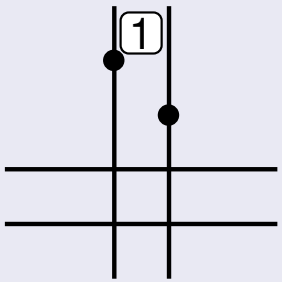


MeshPreim

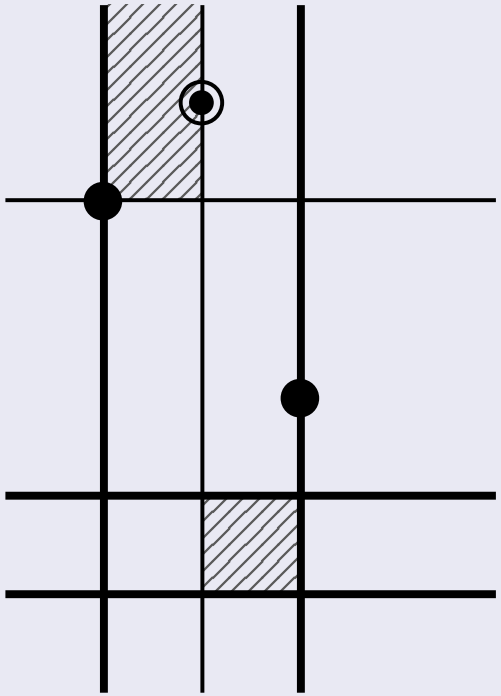
Target



Output from ClassicalPreim

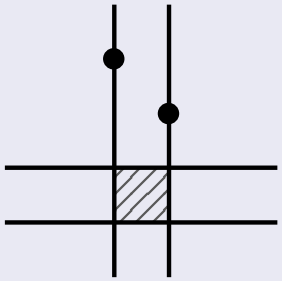


MeshPreim - Case 1

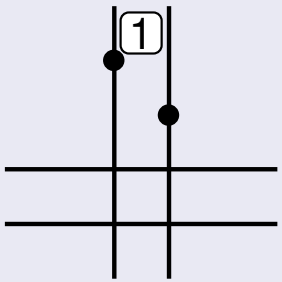


MeshPreim

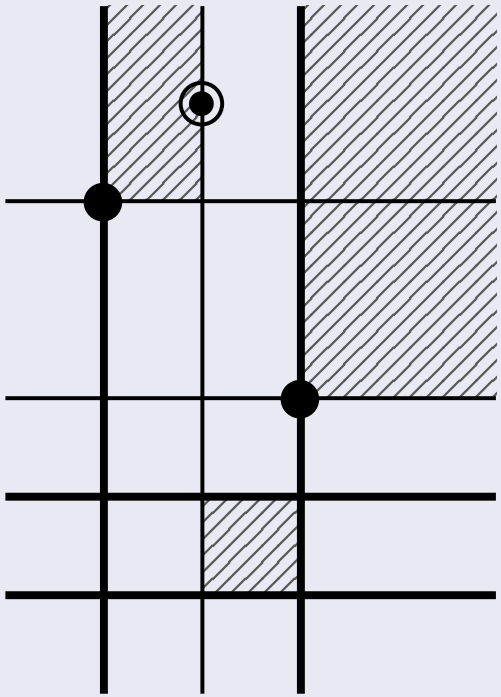
Target



Output from ClassicalPreim

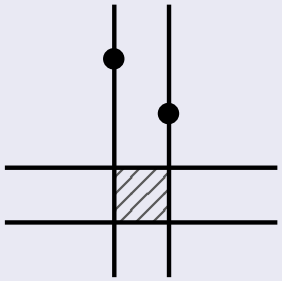


MeshPreim - Case 1

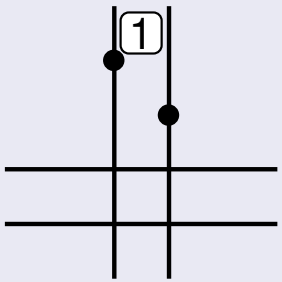


MeshPreim

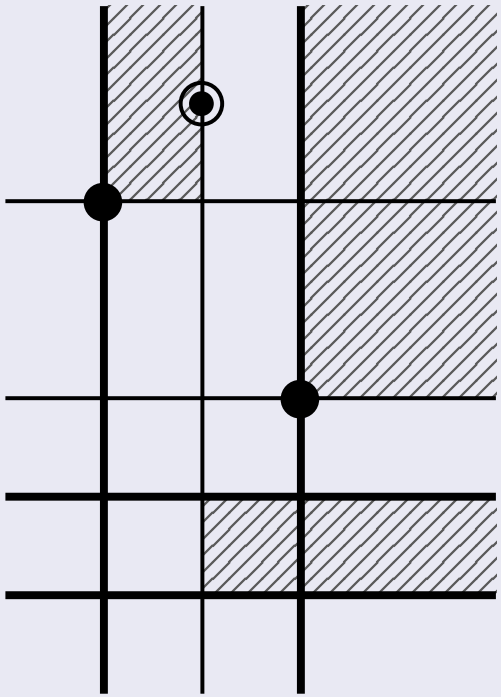
Target



Output from ClassicalPreim

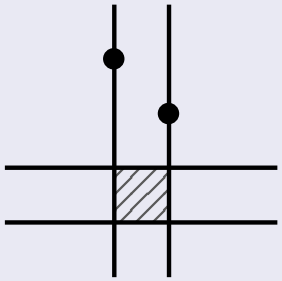


MeshPreim - Case 1

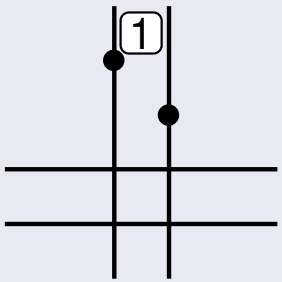


MeshPreim

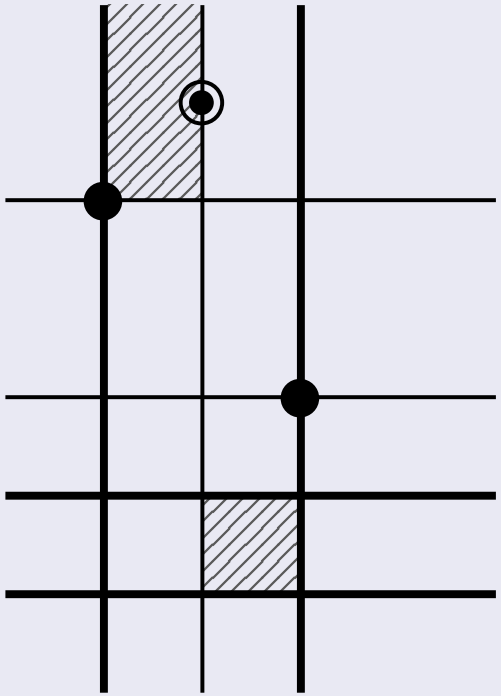
Target



Output from ClassicalPreim

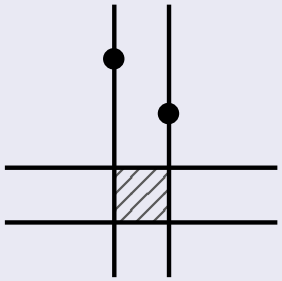


MeshPreim - Case 2

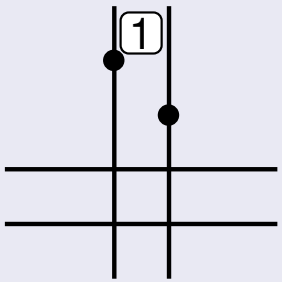


MeshPreim

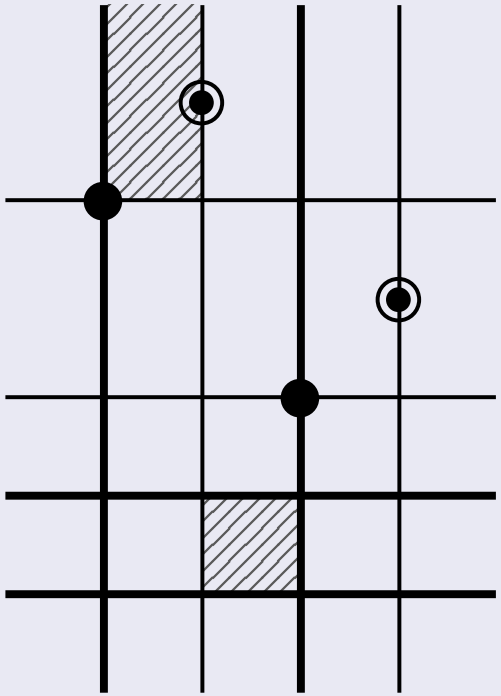
Target



Output from ClassicalPreim

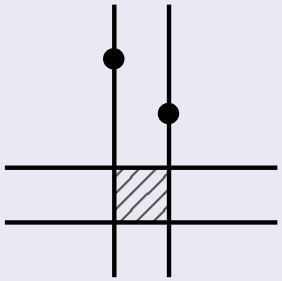


MeshPreim - Case 2

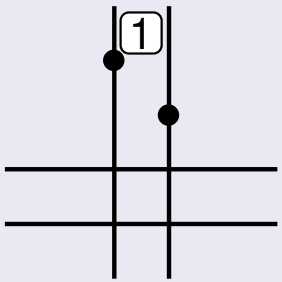


MeshPreim

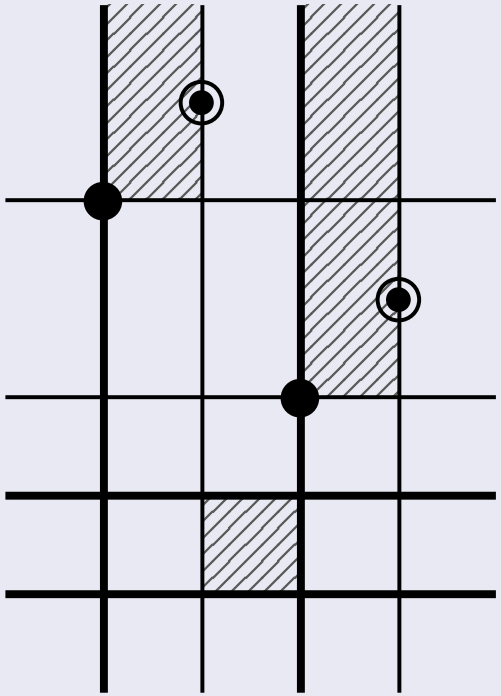
Target



Output from ClassicalPreim

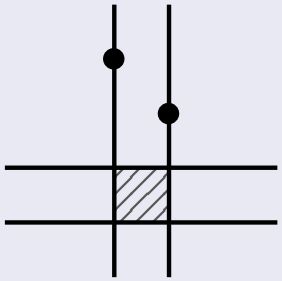


MeshPreim - Case 2

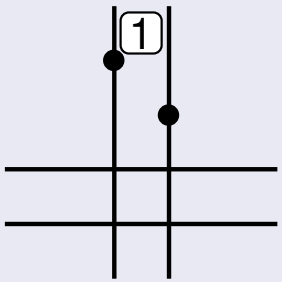


MeshPreim

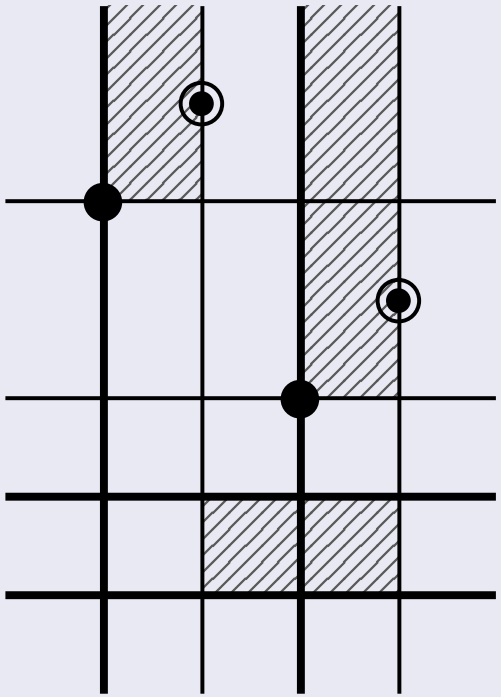
Target



Output from ClassicalPreim

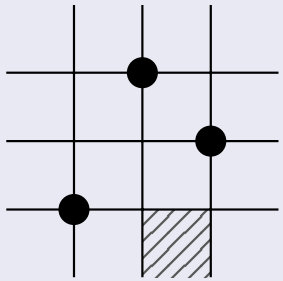


MeshPreim - Case 2

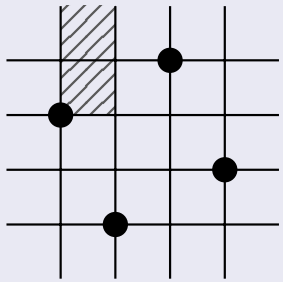


Example

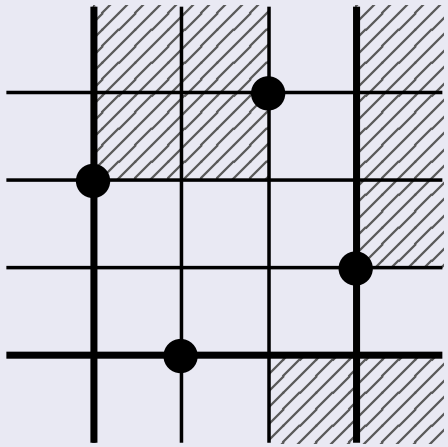
Target



Output from ClassicalPreim

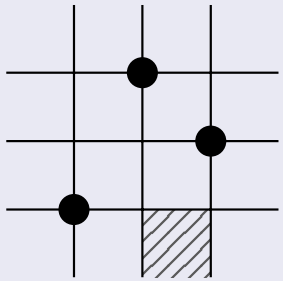


MeshPreim

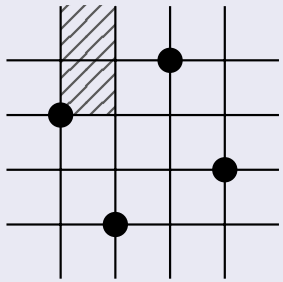


Example

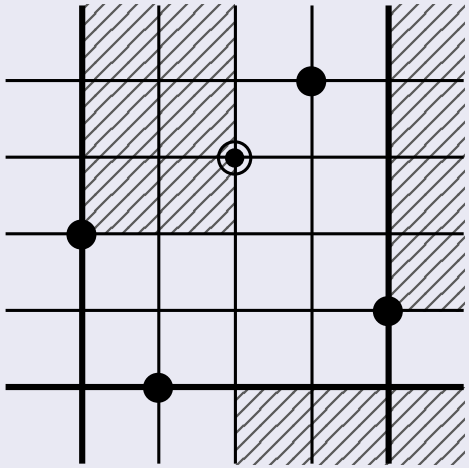
Target

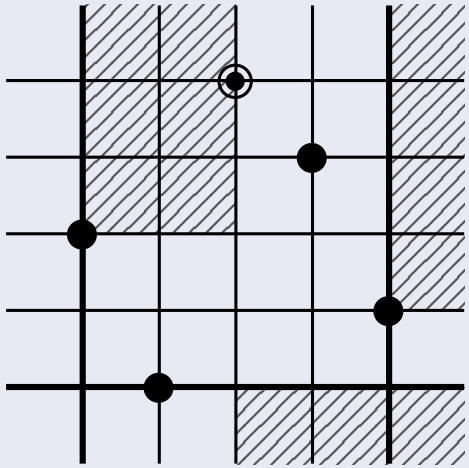


Output from ClassicalPreim



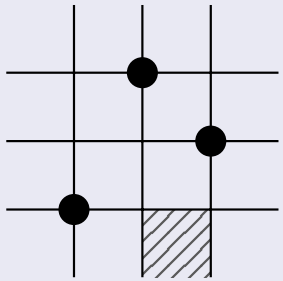
MeshPreim



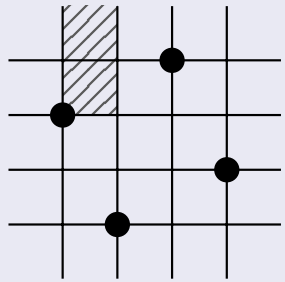


Example

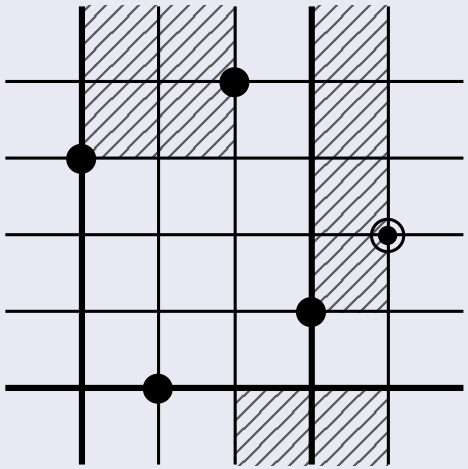
Target

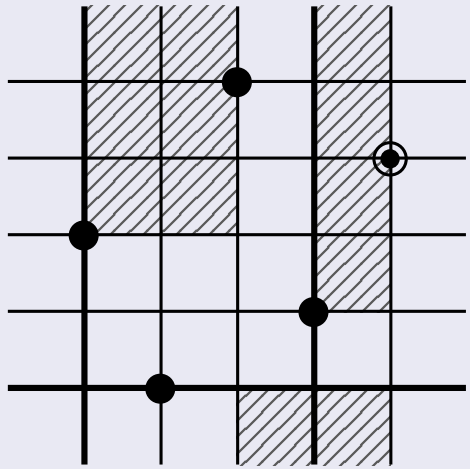


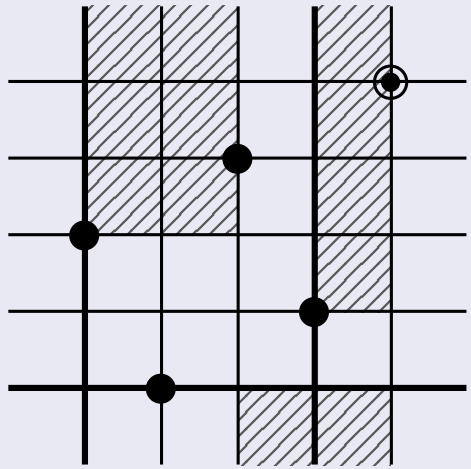
Output from ClassicalPreim



MeshPreim

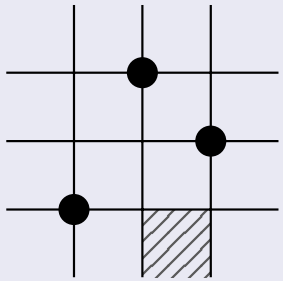




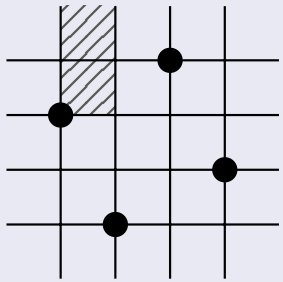


Example

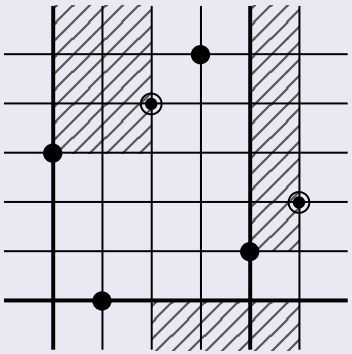
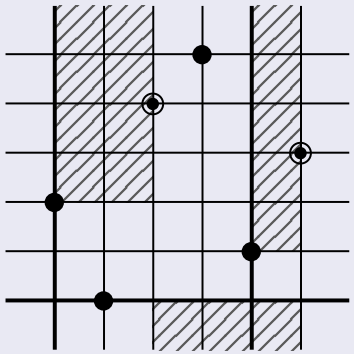
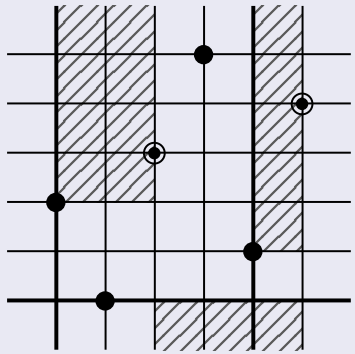
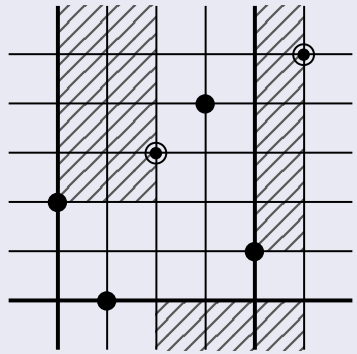
Target



Output from ClassicalPreim

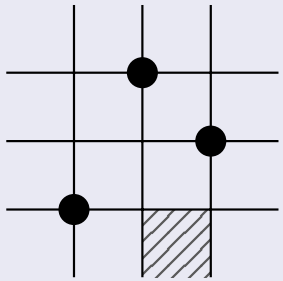


MeshPreim

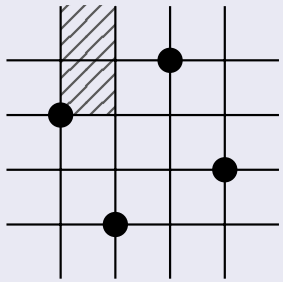





Example

Target



Output from ClassicalPreim



MeshPreim