

*A reduction algorithm for the NPMLE for the distribution
function of bivariate interval censored data*

Marloes H. Maathuis

Advisors: Piet Groeneboom and Jon A. Wellner

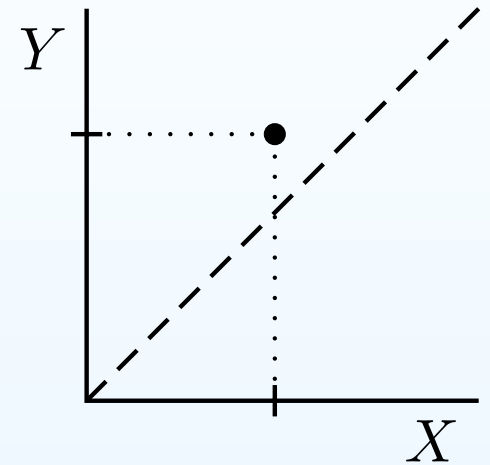
Department of Statistics, University of Washington

marloes@stat.washington.edu

www.stat.washington.edu/marloes

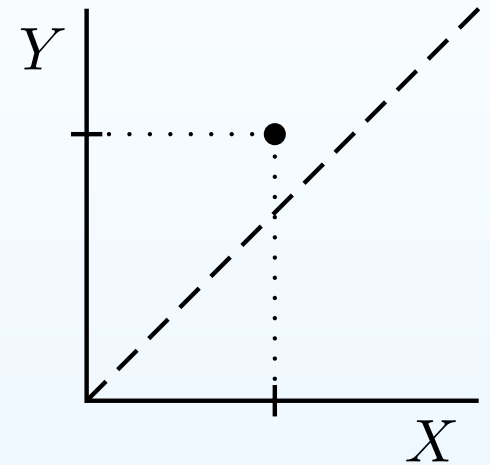
Bivariate interval censored data: an example

- We want to estimate the joint distribution function of (X, Y) , where:
 - X : time of HIV infection
 - Y : time of onset of AIDS



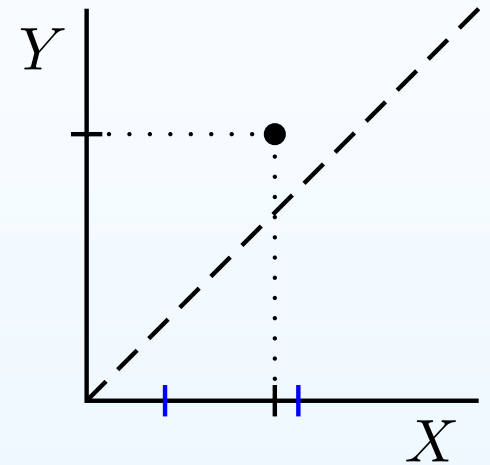
Bivariate interval censored data: an example

- We want to estimate the joint distribution function of (X, Y) , where:
 - X : time of HIV infection
 - Y : time of onset of AIDS
- X and Y can be interval censored.



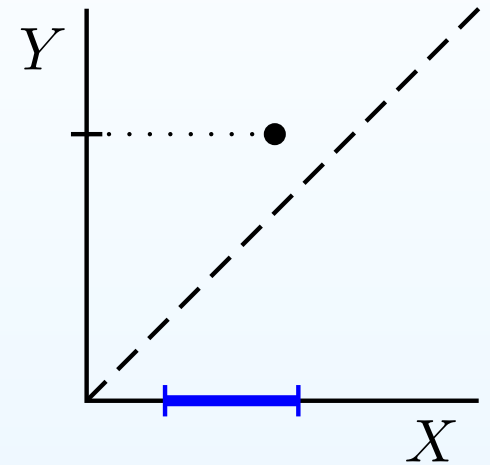
Bivariate interval censored data: an example

- We want to estimate the joint distribution function of (X, Y) , where:
 - X : time of HIV infection
 - Y : time of onset of AIDS
- X and Y can be interval censored.



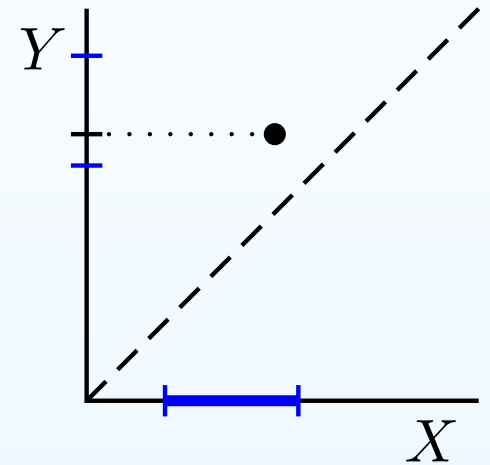
Bivariate interval censored data: an example

- We want to estimate the joint distribution function of (X, Y) , where:
 - X : time of HIV infection
 - Y : time of onset of AIDS
- X and Y can be interval censored.



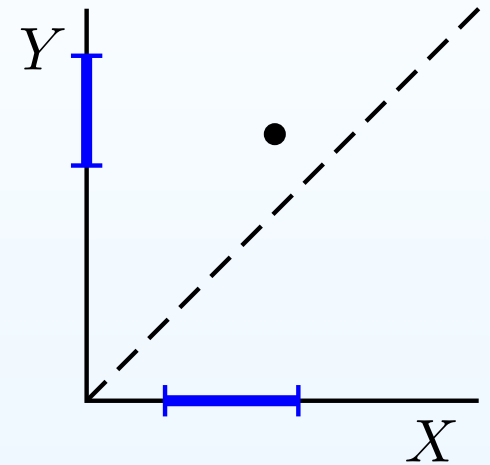
Bivariate interval censored data: an example

- We want to estimate the joint distribution function of (X, Y) , where:
 - X : time of HIV infection
 - Y : time of onset of AIDS
- X and Y can be interval censored.



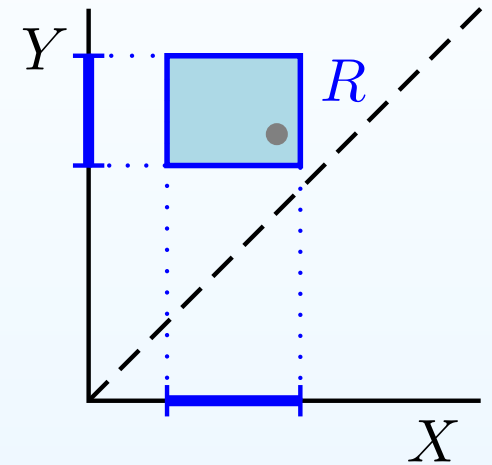
Bivariate interval censored data: an example

- We want to estimate the joint distribution function of (X, Y) , where:
 - X : time of HIV infection
 - Y : time of onset of AIDS
- X and Y can be interval censored.



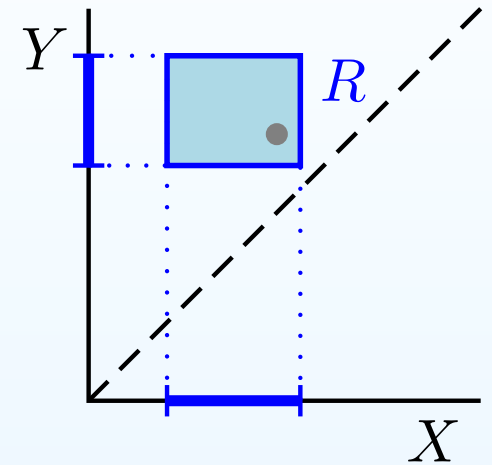
Bivariate interval censored data: an example

- We want to estimate the joint distribution function of (X, Y) , where:
 - X : time of HIV infection
 - Y : time of onset of AIDS
- X and Y can be interval censored. Instead of a realization (x, y) , we observe an *observation rectangle* R that is known to contain (x, y) .



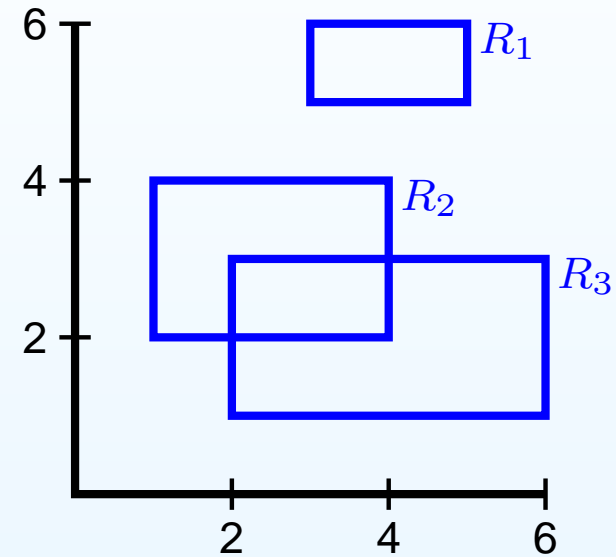
Bivariate interval censored data: an example

- We want to estimate the joint distribution function of (X, Y) , where:
 - X : time of HIV infection
 - Y : time of onset of AIDS
- X and Y can be interval censored. Instead of a realization (x, y) , we observe an *observation rectangle* R that is known to contain (x, y) .
- Goal: based on n i.i.d observation rectangles R_1, \dots, R_n we want to compute the NPMLE for the joint distribution function of (X, Y) .



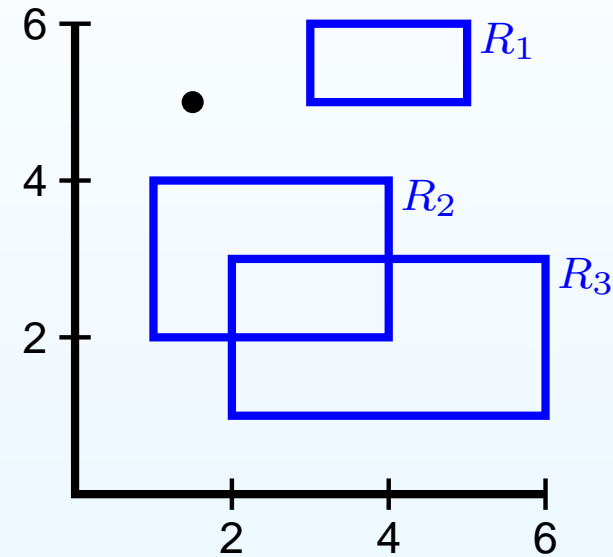
Computing the NPMLE

$$\max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i))$$



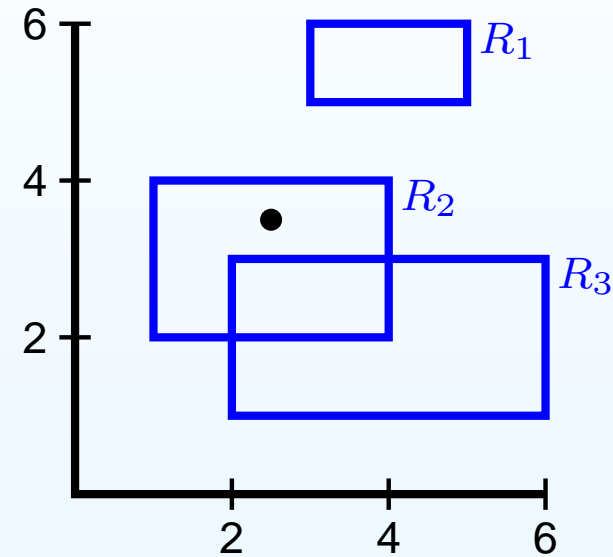
Computing the NPMLE

$$\max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i))$$



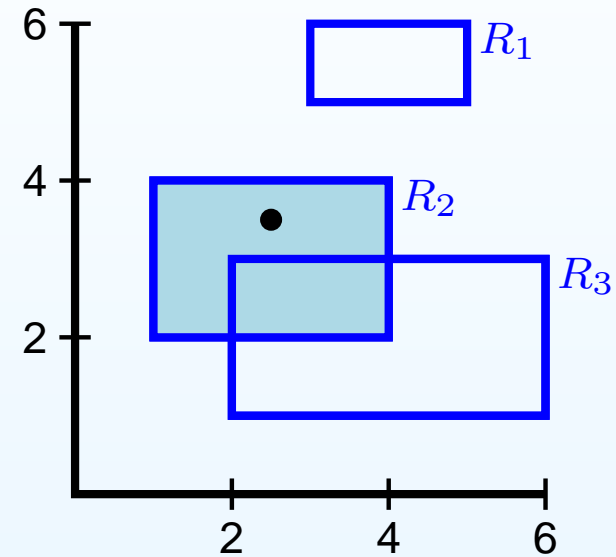
Computing the NPMLE

$$\max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i))$$



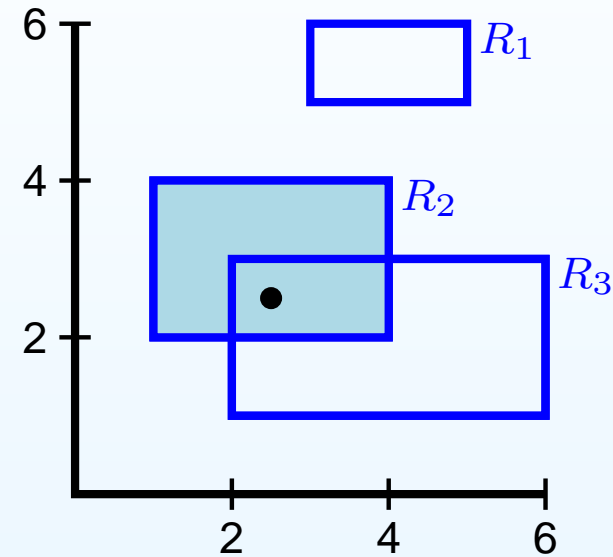
Computing the NPMLE

$$\max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i))$$



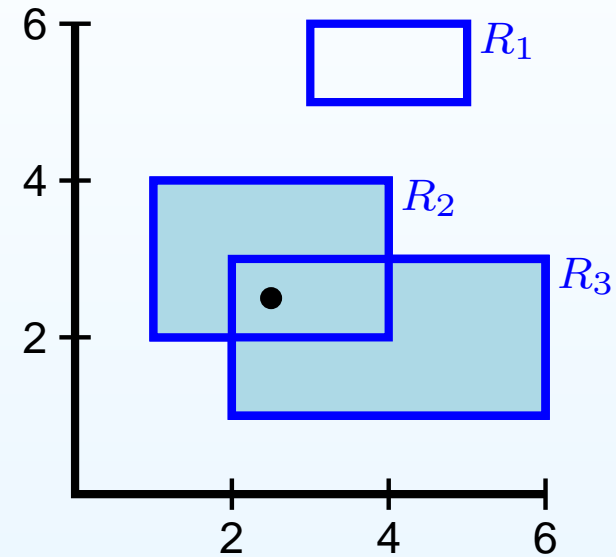
Computing the NPMLE

$$\max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i))$$



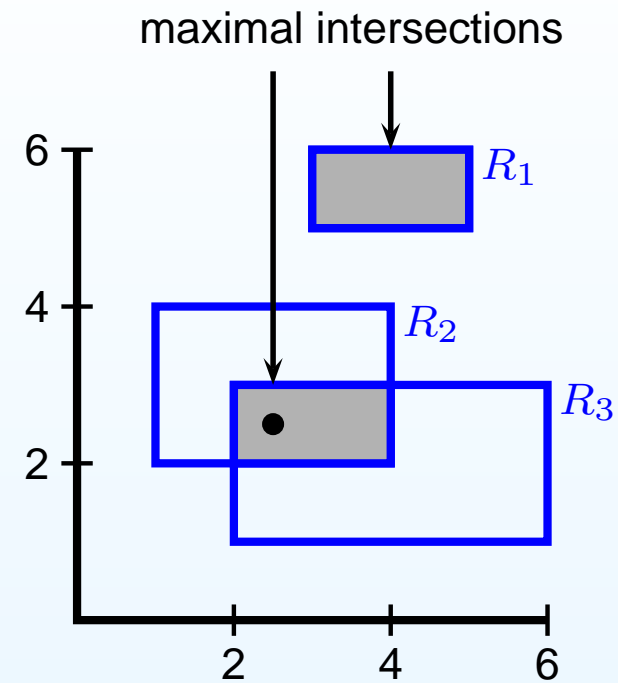
Computing the NPMLE

$$\max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i))$$



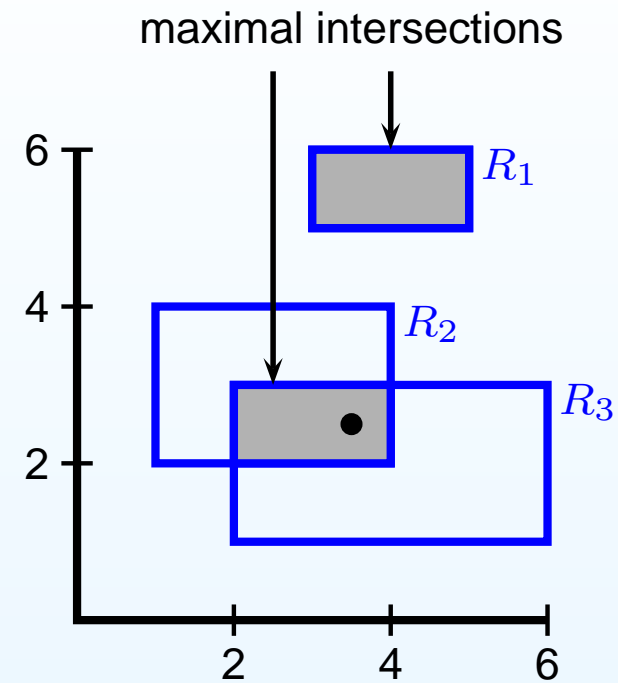
Computing the NPMLE

$$\max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i))$$



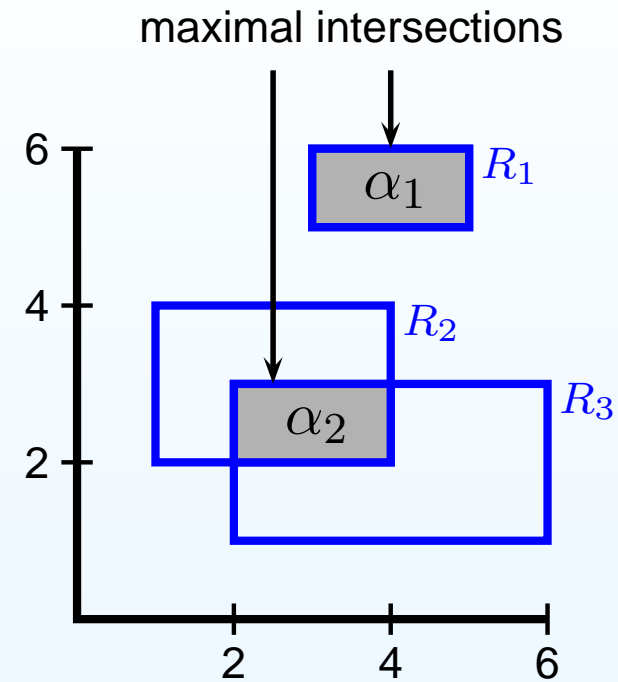
Computing the NPMLE

$$\max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i))$$



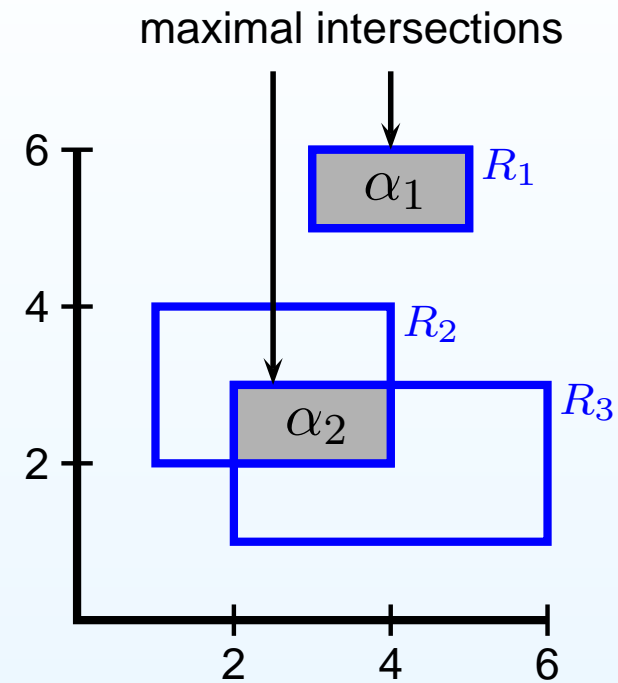
Computing the NPMLE

$$\max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i))$$



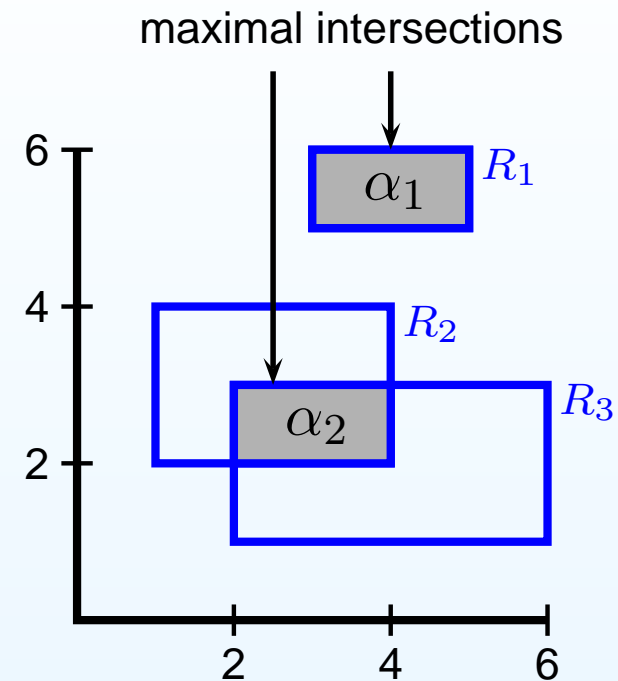
Computing the NPMLE

$$\begin{aligned} & \max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i)) \\ & = \max_{\alpha} \log(\alpha_1) + 2 \log(\alpha_2) \end{aligned}$$



Computing the NPMLE

$$\begin{aligned} & \max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i)) \\ & = \max_{\alpha} \log(\alpha_1) + 2 \log(\alpha_2) \\ & \Rightarrow \alpha_1 = 1/3, \quad \alpha_2 = 2/3 \end{aligned}$$

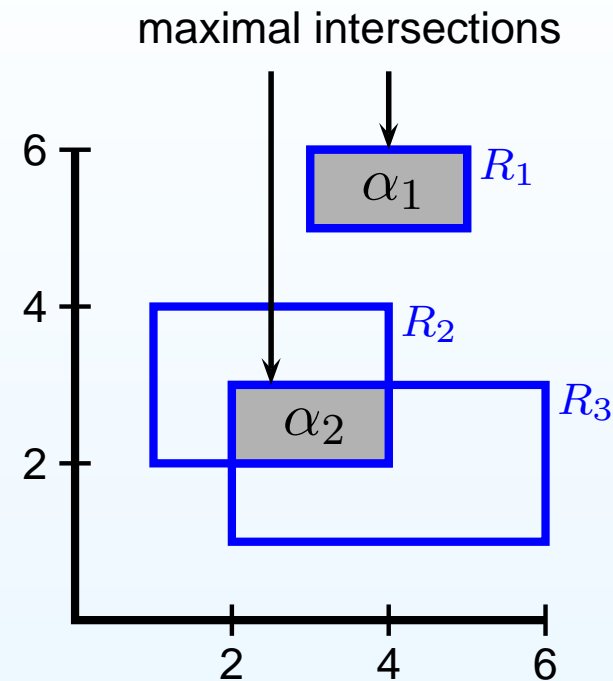


Computing the NPMLE

$$\begin{aligned} & \max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i)) \\ & = \max_{\alpha} \log(\alpha_1) + 2 \log(\alpha_2) \\ & \Rightarrow \alpha_1 = 1/3, \quad \alpha_2 = 2/3 \end{aligned}$$

Computation of the NPMLE:

- Reduction step: find maximal intersections
- Optimization step: solve optimization problem in α



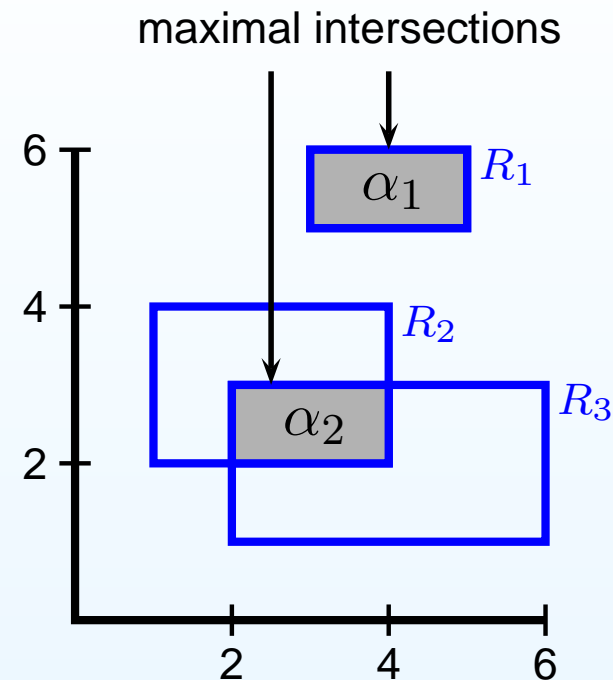
Computing the NPMLE

$$\begin{aligned} & \max_{F \in \mathcal{F}} \sum_{i=1}^n \log(P_F(R_i)) \\ & = \max_{\alpha} \log(\alpha_1) + 2 \log(\alpha_2) \\ & \Rightarrow \alpha_1 = 1/3, \quad \alpha_2 = 2/3 \end{aligned}$$

Computation of the NPMLE:

- Reduction step: find maximal intersections
- Optimization step: solve optimization problem in α

The reduction step used to be a bottleneck. We worked on improving this.



Previous work on the reduction step and our work

Reduction algorithms:

- Betensky and Finkelstein (1999)

Previous work on the reduction step and our work

Reduction algorithms:

- Betensky and Finkelstein (1999)
- Song (2001)

Previous work on the reduction step and our work

Reduction algorithms:

- Betensky and Finkelstein (1999)
- Song (2001)
- Gentleman and Vandal (2001), time complexity $O(n^5)$

Previous work on the reduction step and our work

Reduction algorithms:

- Betensky and Finkelstein (1999)
- Song (2001)
- Gentleman and Vandal (2001), time complexity $O(n^5)$
- Bogaerts and Lesaffre (2004), time complexity $O(n^3)$

Previous work on the reduction step and our work

Reduction algorithms:

- Betensky and Finkelstein (1999)
- Song (2001)
- Gentleman and Vandal (2001), time complexity $O(n^5)$
- Bogaerts and Lesaffre (2004), time complexity $O(n^3)$

Related algorithm: finding the maximum number of rectangles having a non-empty intersection:

- Lee (1983), time complexity $O(n \log n)$

Previous work on the reduction step and our work

Reduction algorithms:

- Betensky and Finkelstein (1999)
- Song (2001)
- Gentleman and Vandal (2001), time complexity $O(n^5)$
- Bogaerts and Lesaffre (2004), time complexity $O(n^3)$

Related algorithm: finding the maximum number of rectangles having a non-empty intersection:

- Lee (1983), time complexity $O(n \log n)$

Our new reduction algorithms:

- Tree algorithm, based on Lee (1983)

Previous work on the reduction step and our work

Reduction algorithms:

- Betensky and Finkelstein (1999)
- Song (2001)
- Gentleman and Vandal (2001), time complexity $O(n^5)$
- Bogaerts and Lesaffre (2004), time complexity $O(n^3)$

Related algorithm: finding the maximum number of rectangles having a non-empty intersection:

- Lee (1983), time complexity $O(n \log n)$

Our new reduction algorithms:

- Tree algorithm, based on Lee (1983)
- HeightMap algorithm, time complexity $O(n^2)$

Outline

- Introduction: data, goal, previous work

Outline

- Introduction: data, goal, previous work
- HeightMap algorithm
 - the algorithm
 - time complexity
 - extension to d -dimensional interval censored data

Outline

- Introduction: data, goal, previous work
- HeightMap algorithm
 - the algorithm
 - time complexity
 - extension to d -dimensional interval censored data
- Simulation study

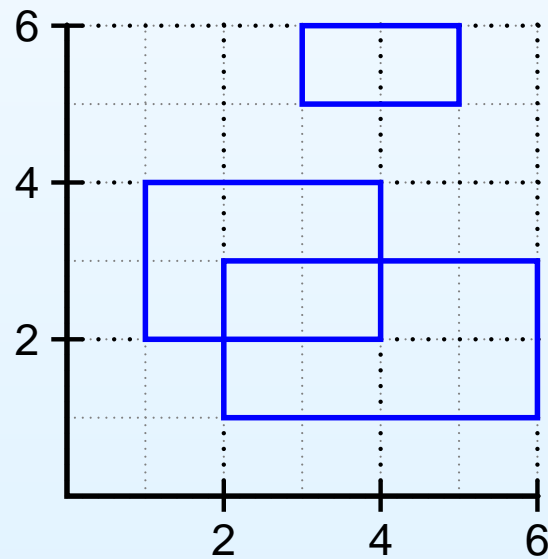
Outline

- Introduction: data, goal, previous work
- HeightMap algorithm
 - the algorithm
 - time complexity
 - extension to d -dimensional interval censored data
- Simulation study
- Future work

HeightMap algorithm

Basic idea:

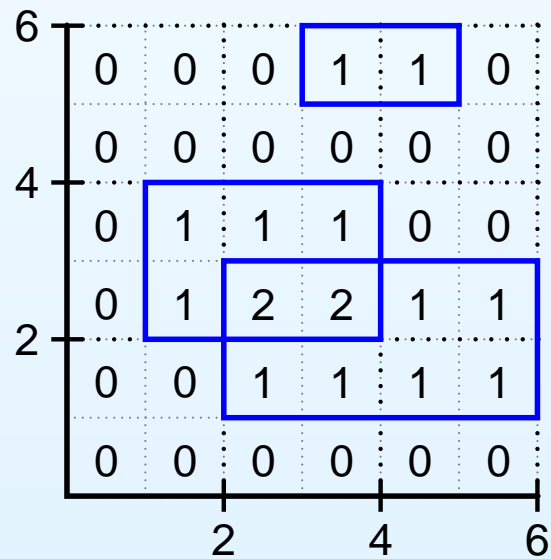
- Define a height map of the observation rectangles



HeightMap algorithm

Basic idea:

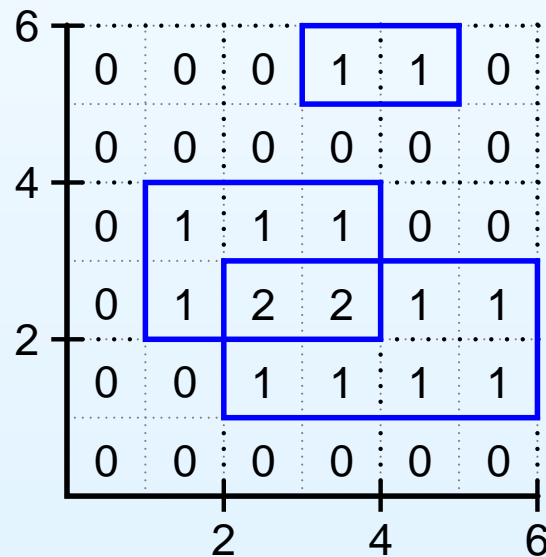
- Define a height map of the observation rectangles



HeightMap algorithm

Basic idea:

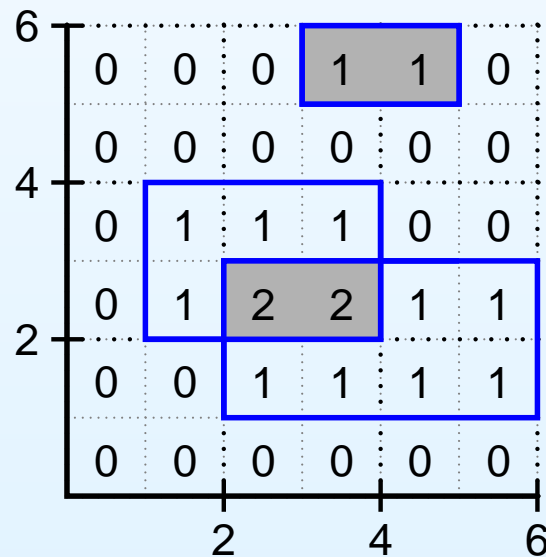
- Define a height map of the observation rectangles
- For any set of observation rectangles without ties, the maximal intersections are the local maxima of the height map



HeightMap algorithm

Basic idea:

- Define a height map of the observation rectangles
- For any set of observation rectangles without ties, the maximal intersections are the local maxima of the height map



HeightMap algorithm

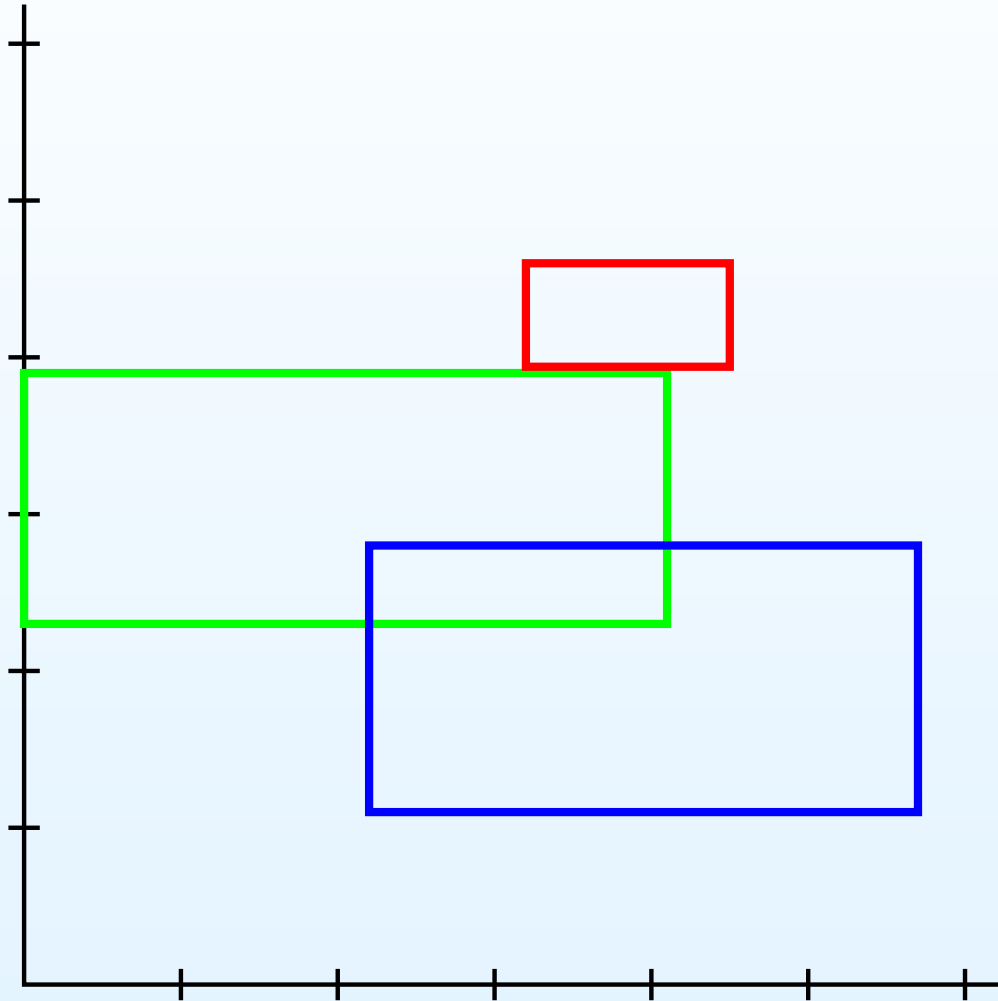
Basic idea:

- Define a height map of the observation rectangles
- For any set of observation rectangles without ties, the maximal intersections are the local maxima of the height map

Algorithm:

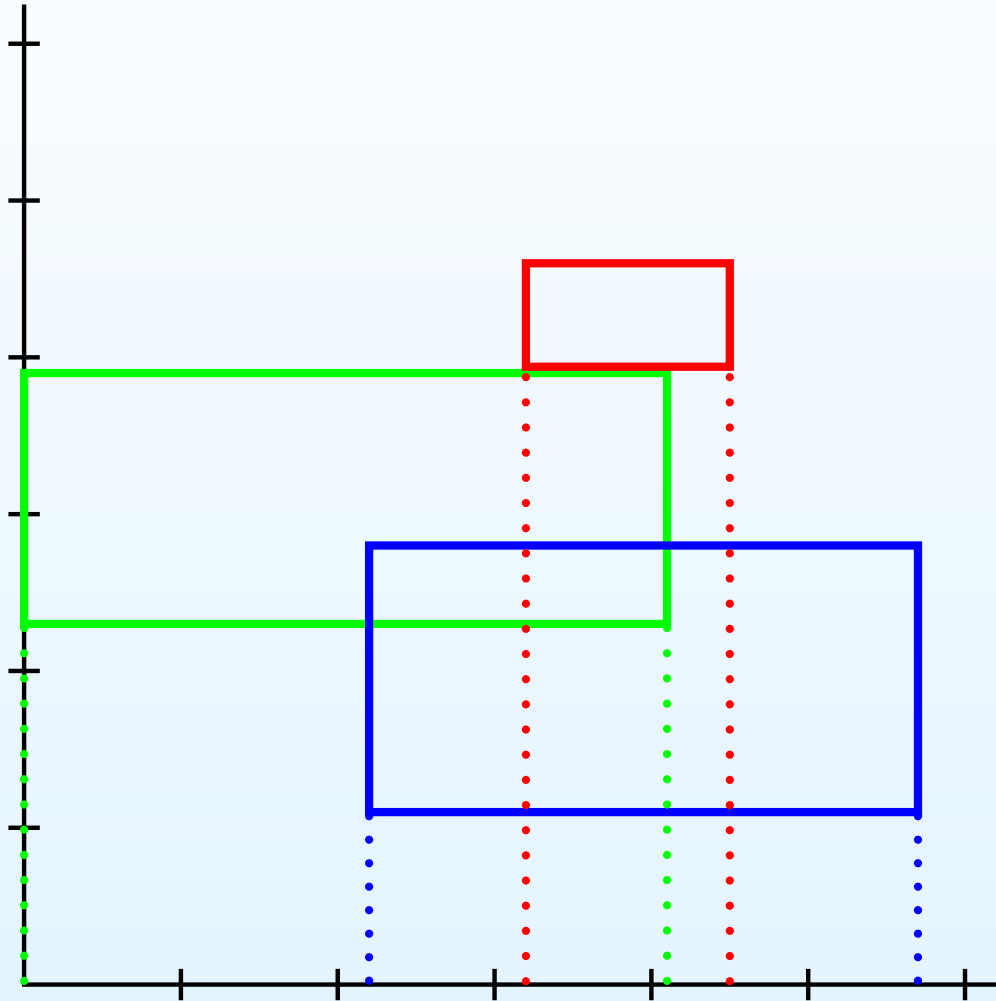
- Transform observation rectangles into canonical rectangles
- Find local maxima of the height map of the canonical rectangles (by sweeping)
- Transform local maxima back to original coordinates

Transform rectangles into canonical rectangles

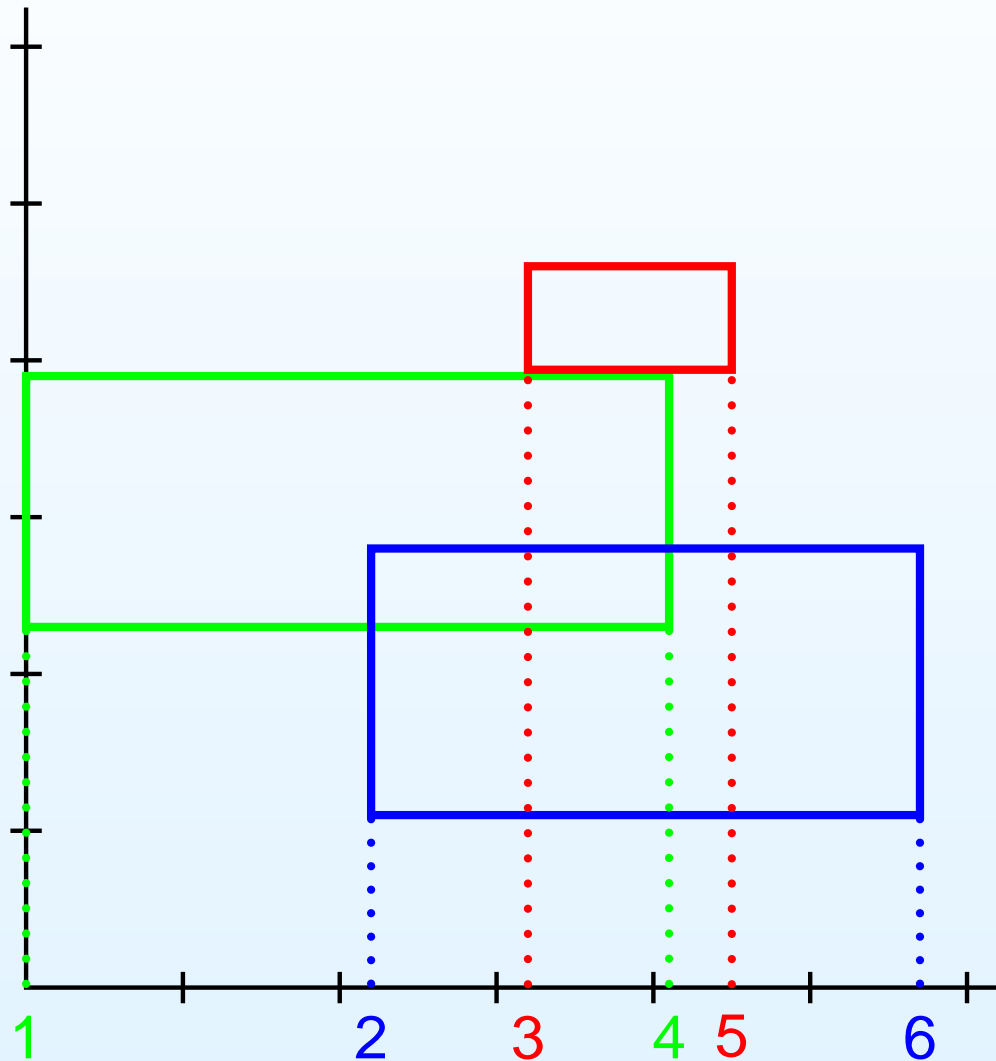


Transform rectangles into canonical rectangles

- Replace x -coordinates by their order statistics.

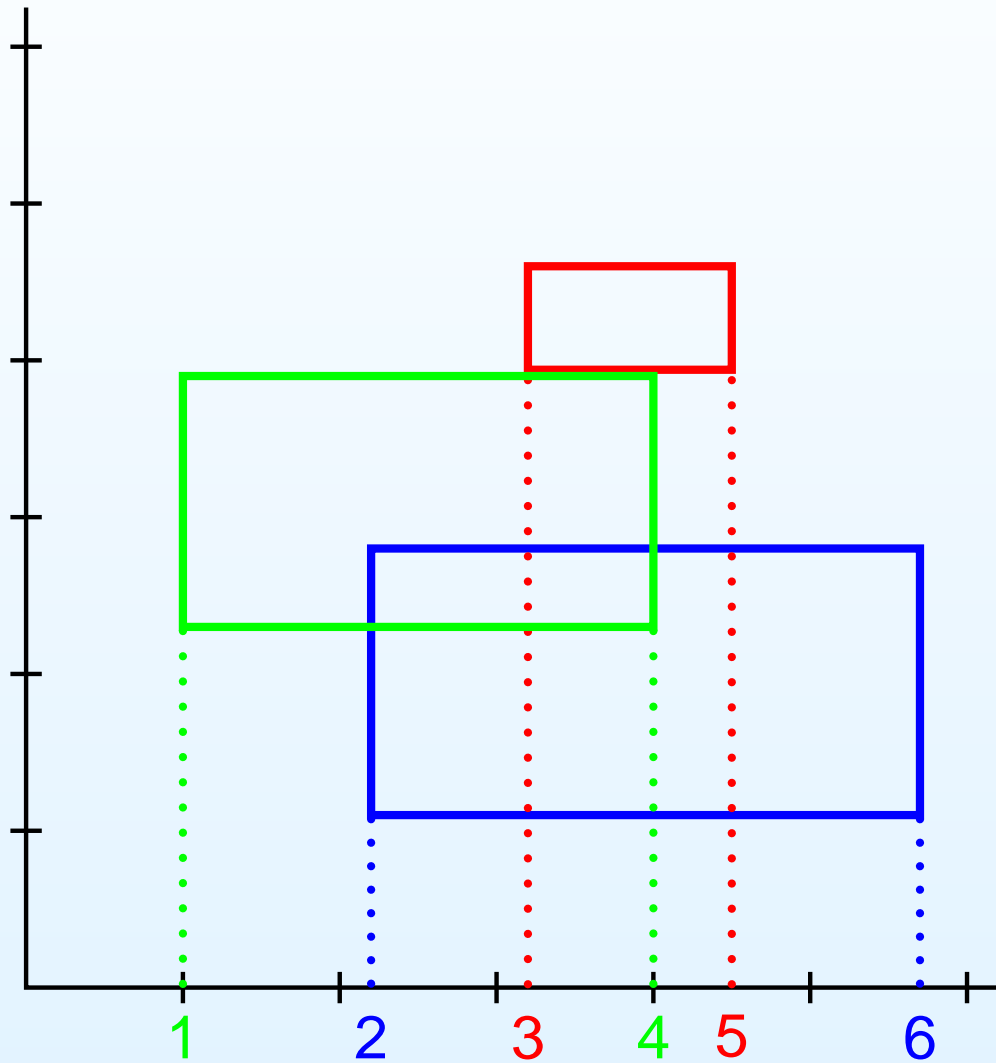


Transform rectangles into canonical rectangles



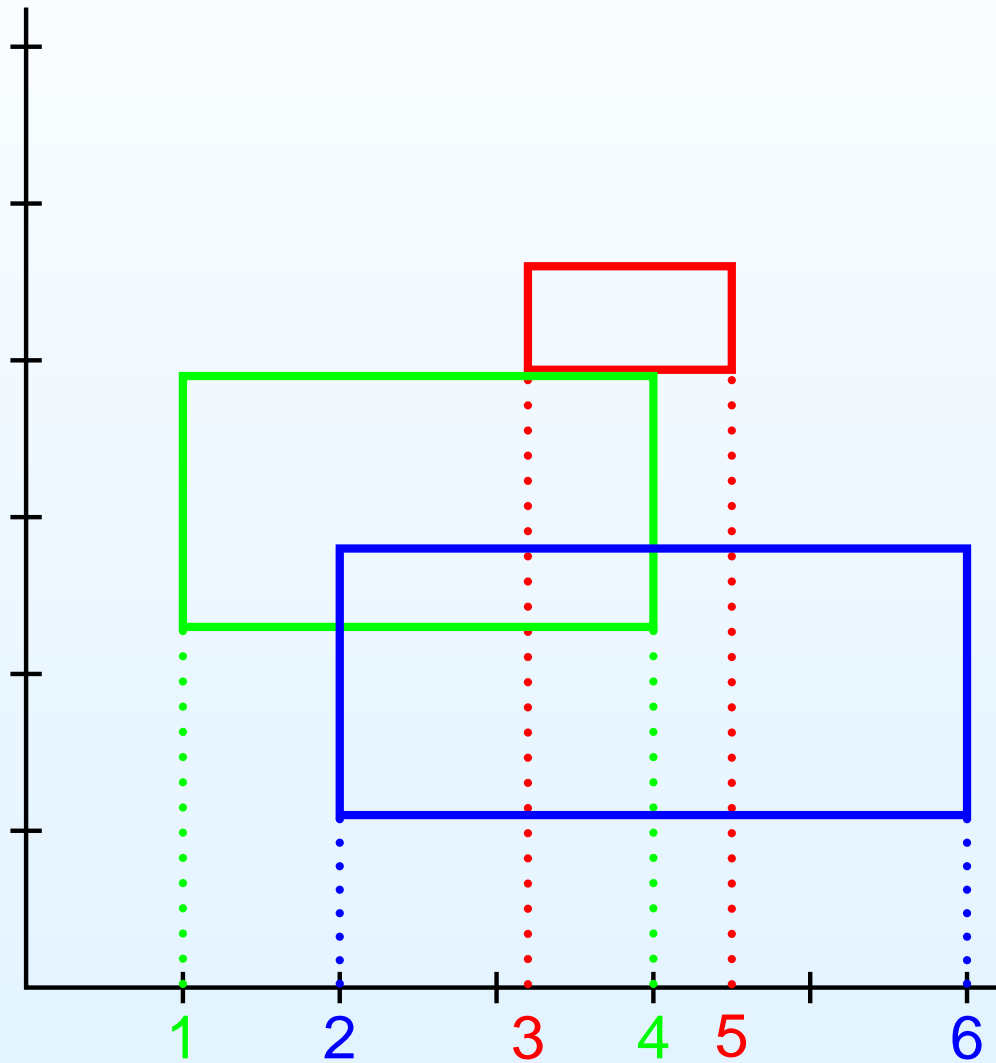
- Replace x -coordinates by their order statistics.

Transform rectangles into canonical rectangles



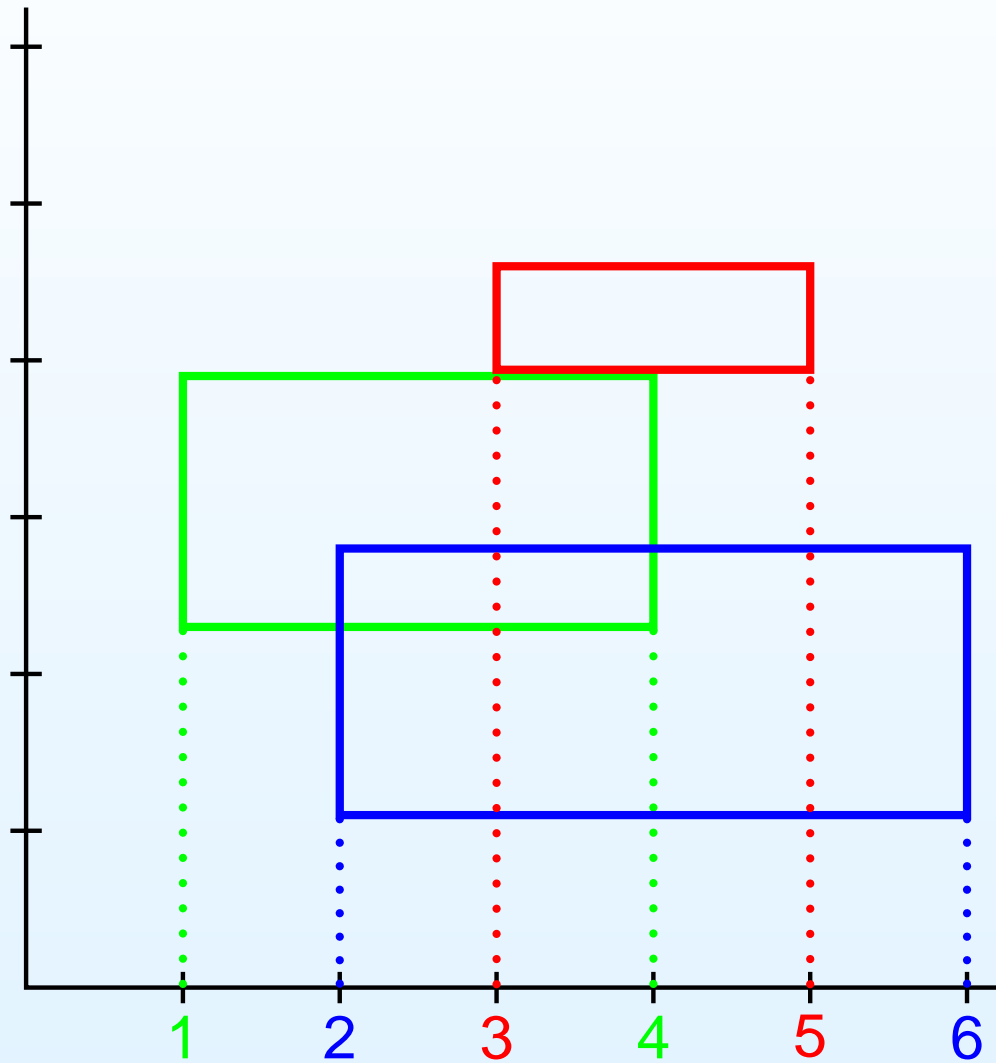
- Replace x -coordinates by their order statistics.

Transform rectangles into canonical rectangles



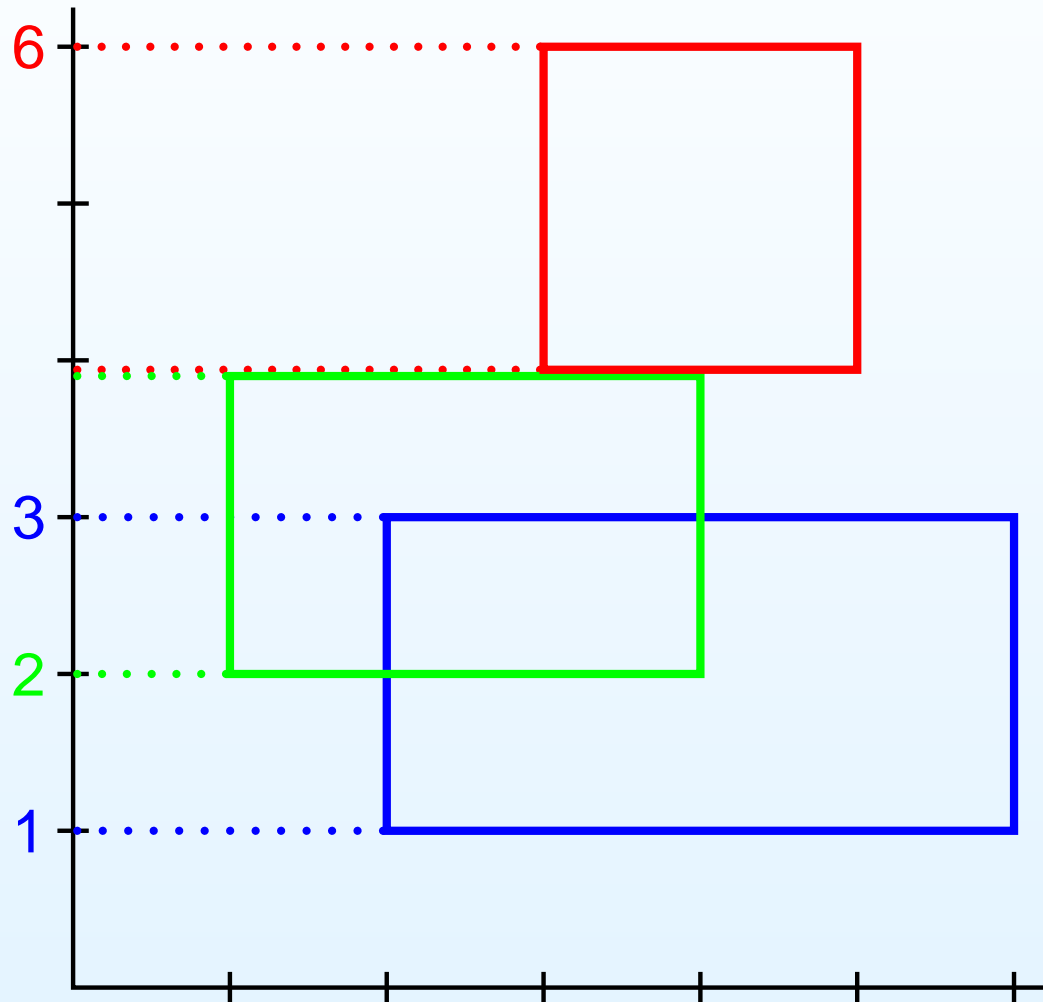
- Replace x -coordinates by their order statistics.

Transform rectangles into canonical rectangles



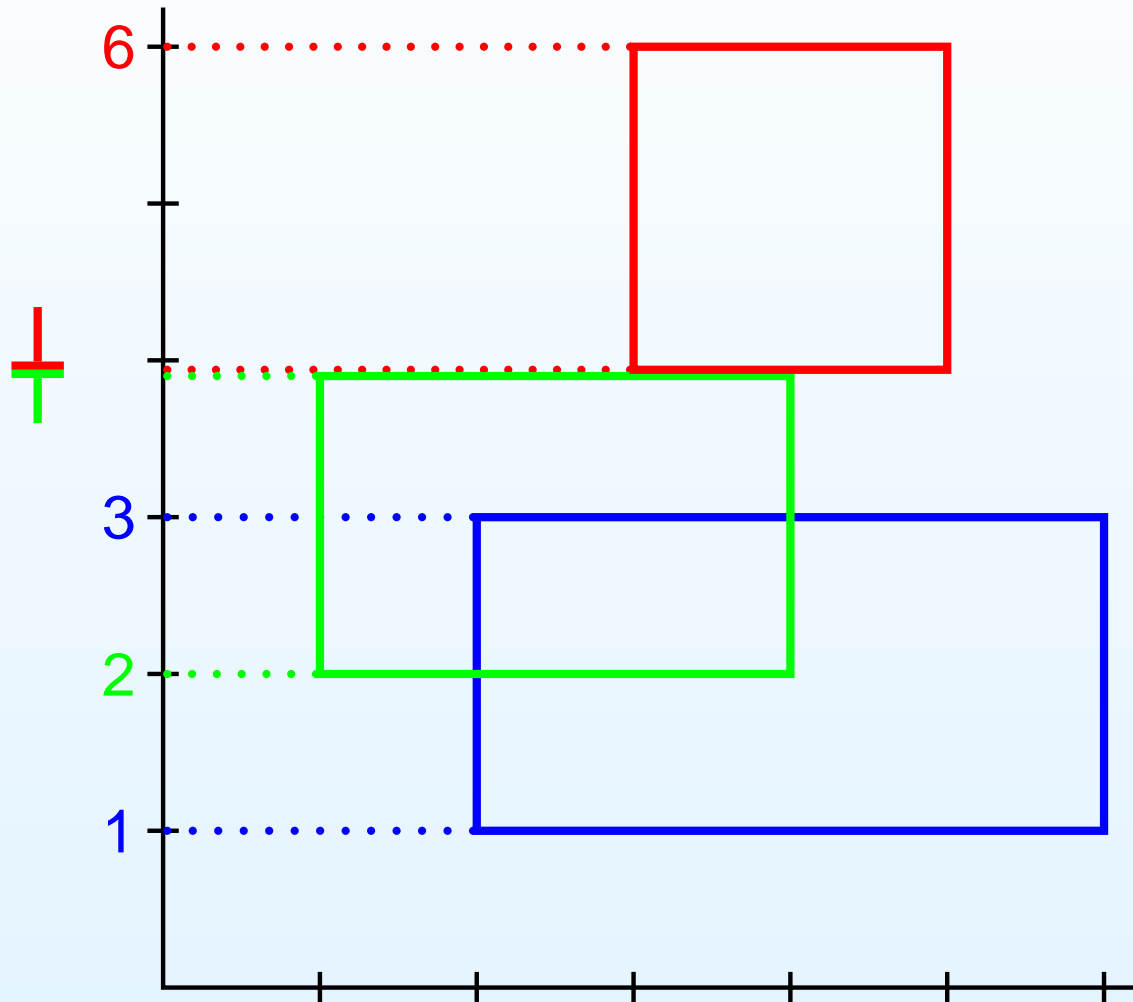
- Replace x -coordinates by their order statistics.

Transform rectangles into canonical rectangles



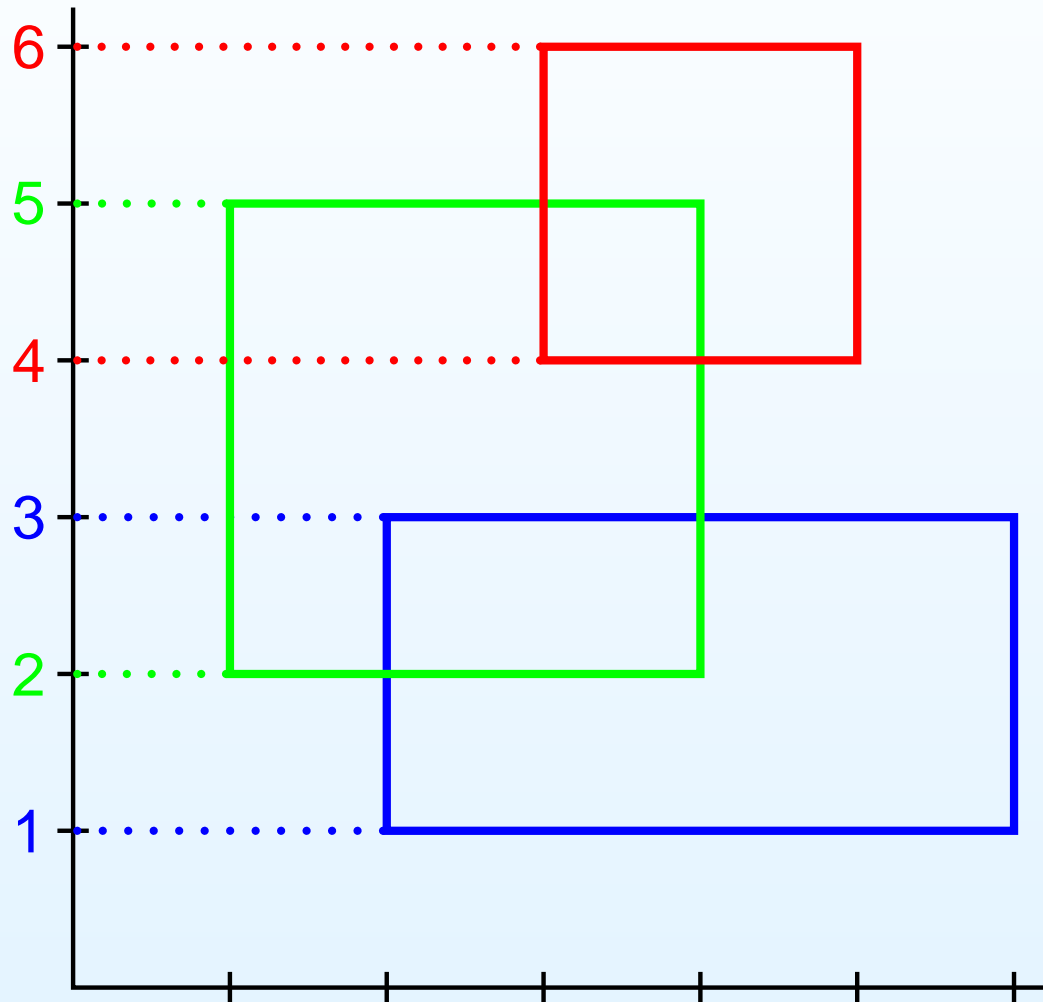
- Replace x -coordinates by their order statistics.
- Replace y -coordinates by their order statistics.

Transform rectangles into canonical rectangles



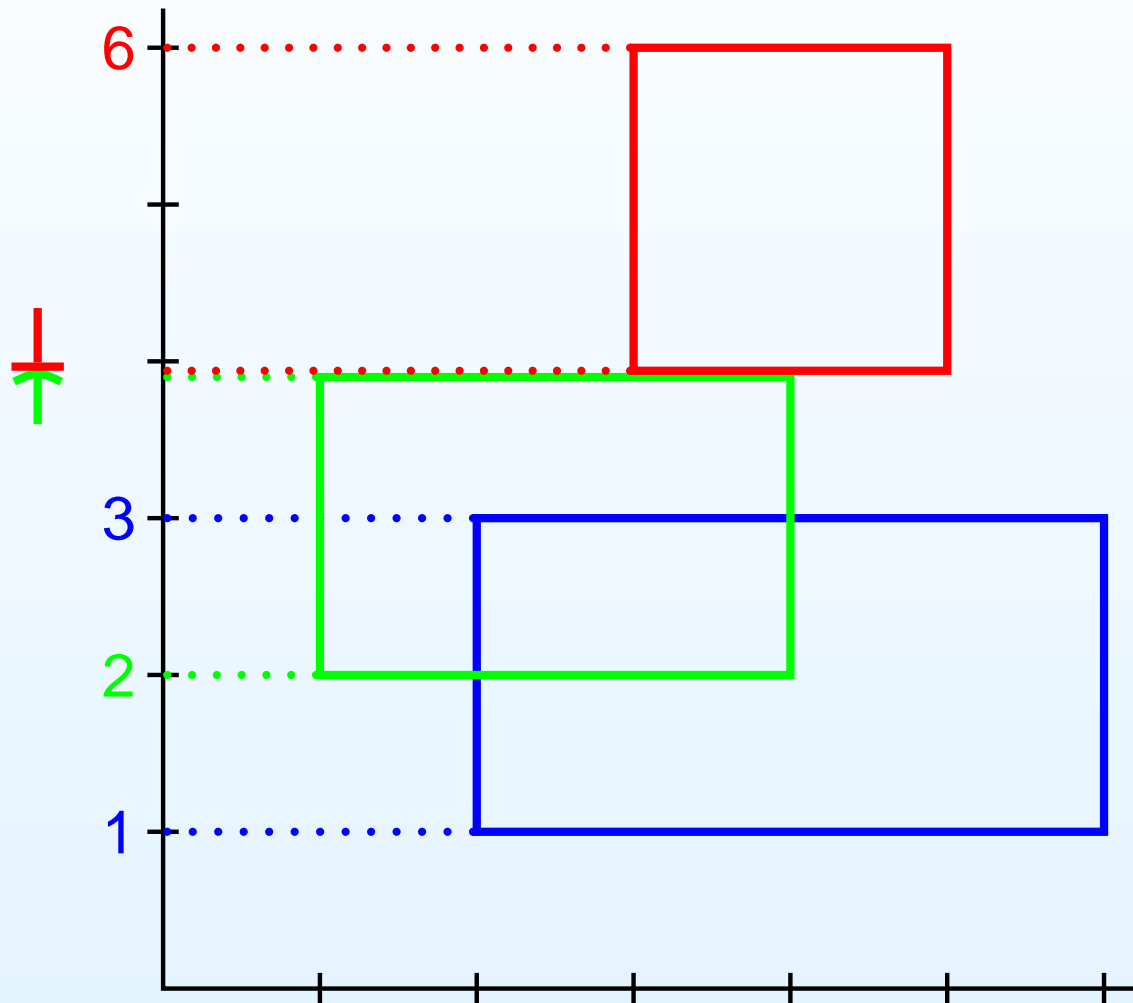
- Replace x -coordinates by their order statistics.
- Replace y -coordinates by their order statistics.

Transform rectangles into canonical rectangles



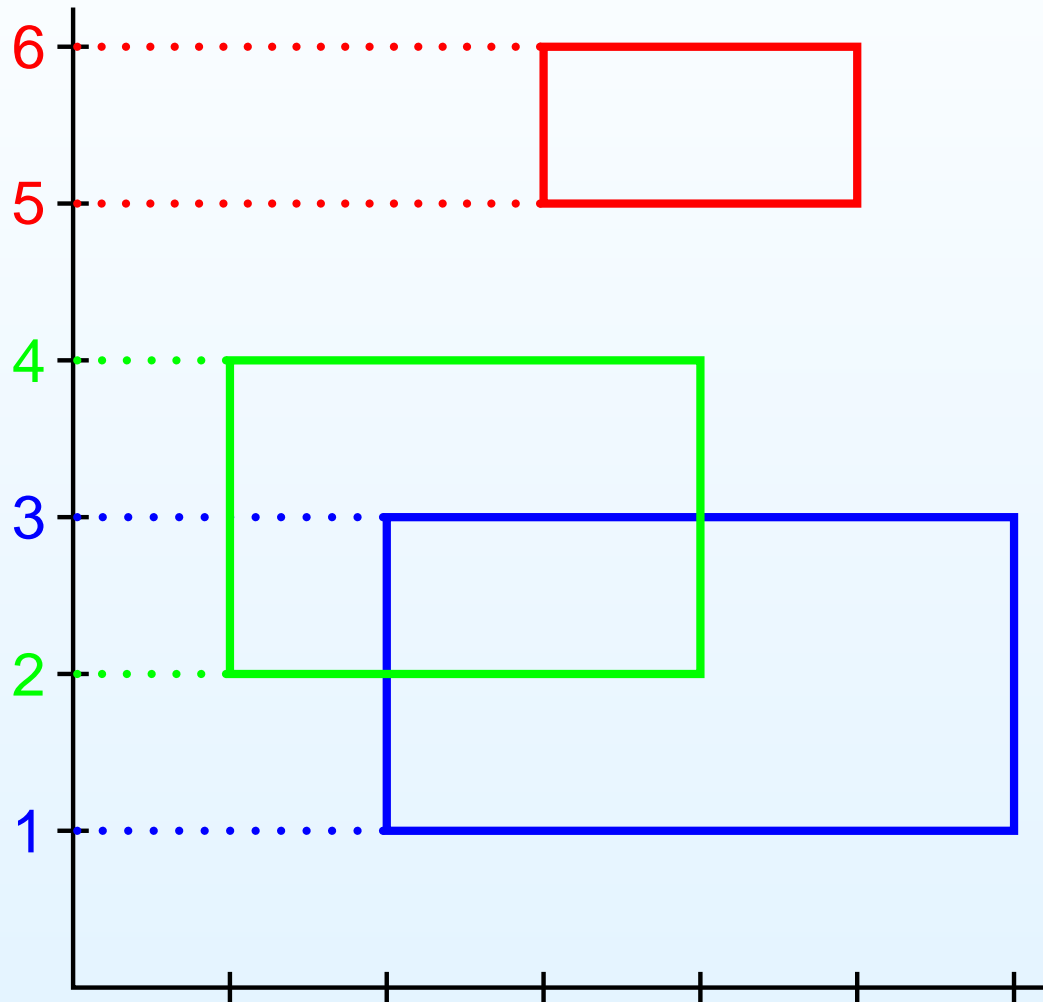
- Replace x -coordinates by their order statistics.
- Replace y -coordinates by their order statistics.

Transform rectangles into canonical rectangles



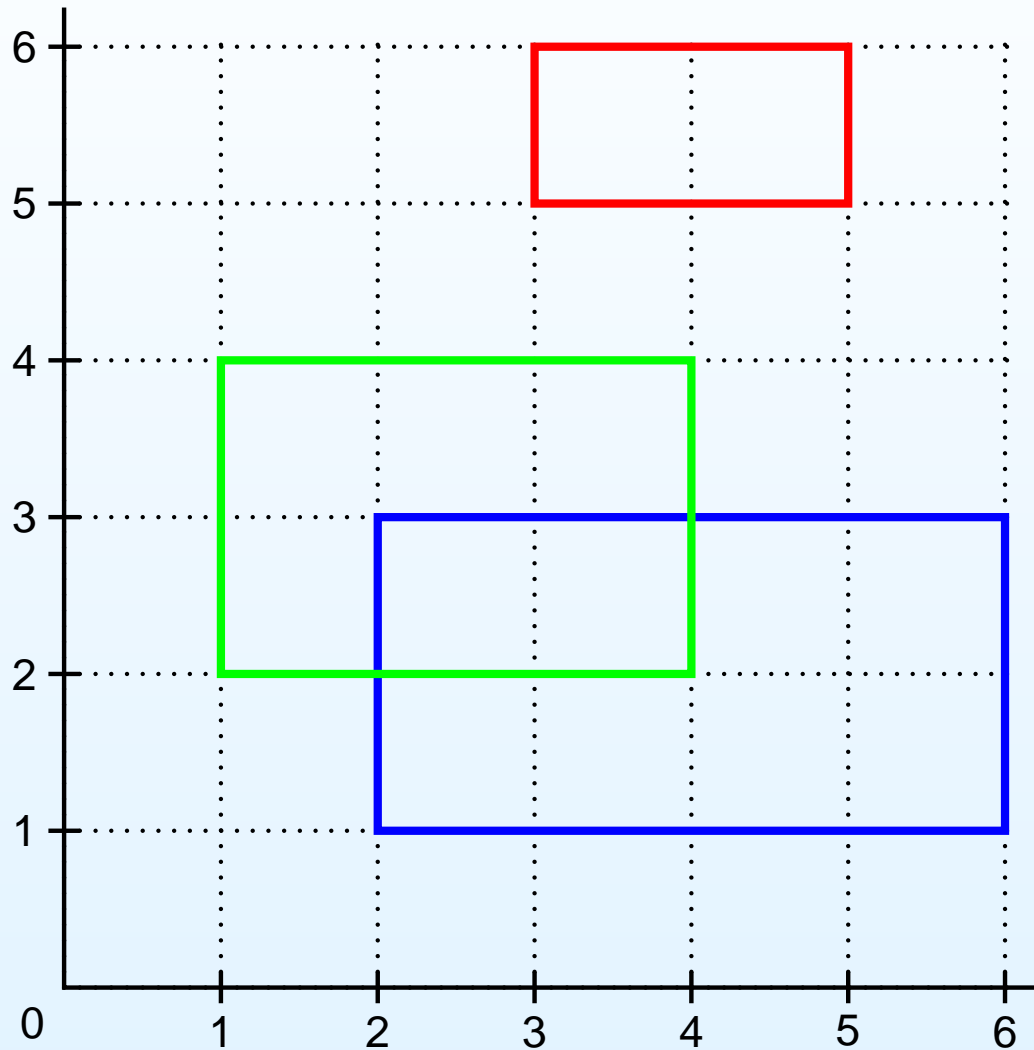
- Replace x -coordinates by their order statistics.
- Replace y -coordinates by their order statistics.

Transform rectangles into canonical rectangles

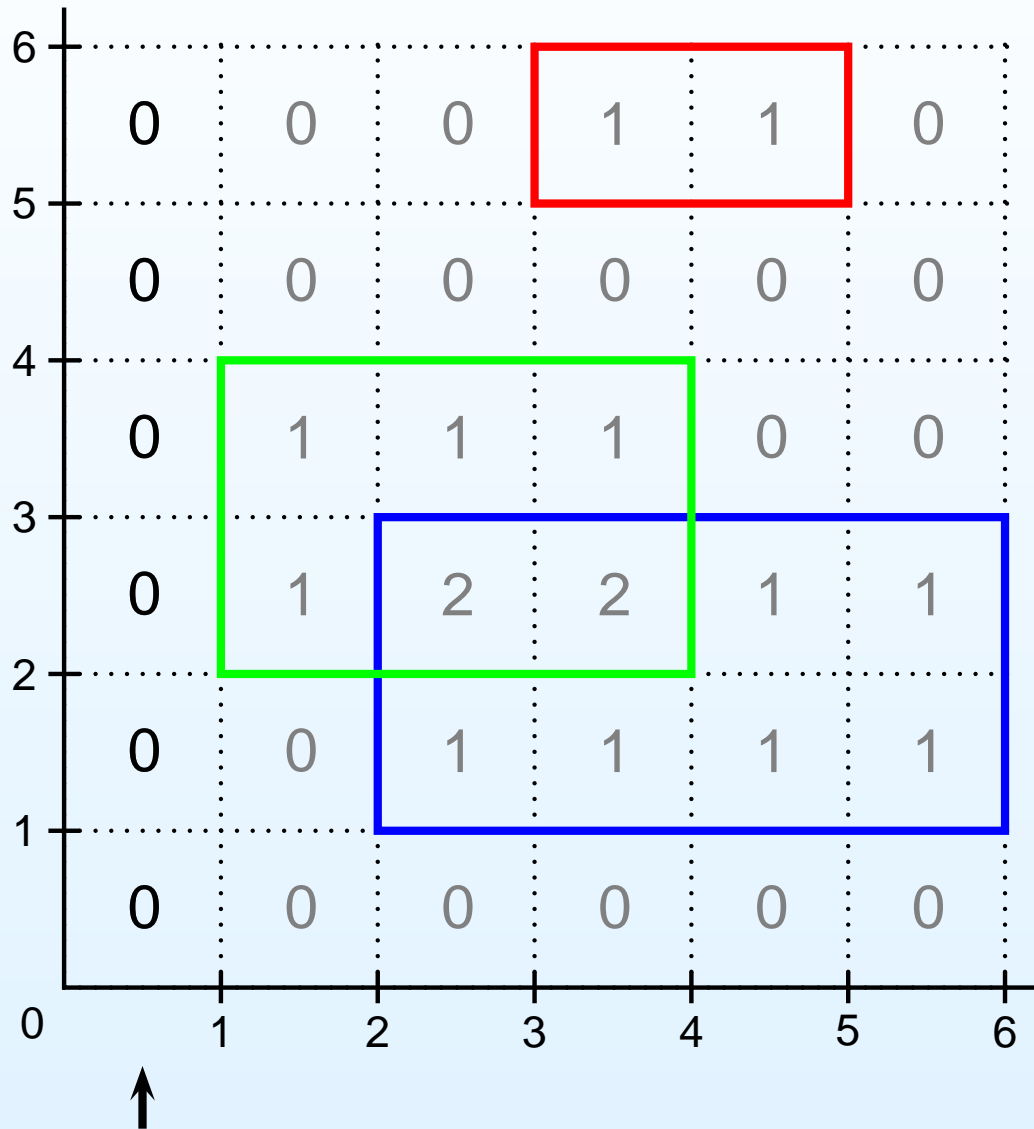


- Replace x -coordinates by their order statistics.
- Replace y -coordinates by their order statistics.

Find local maxima by sweeping through the height map

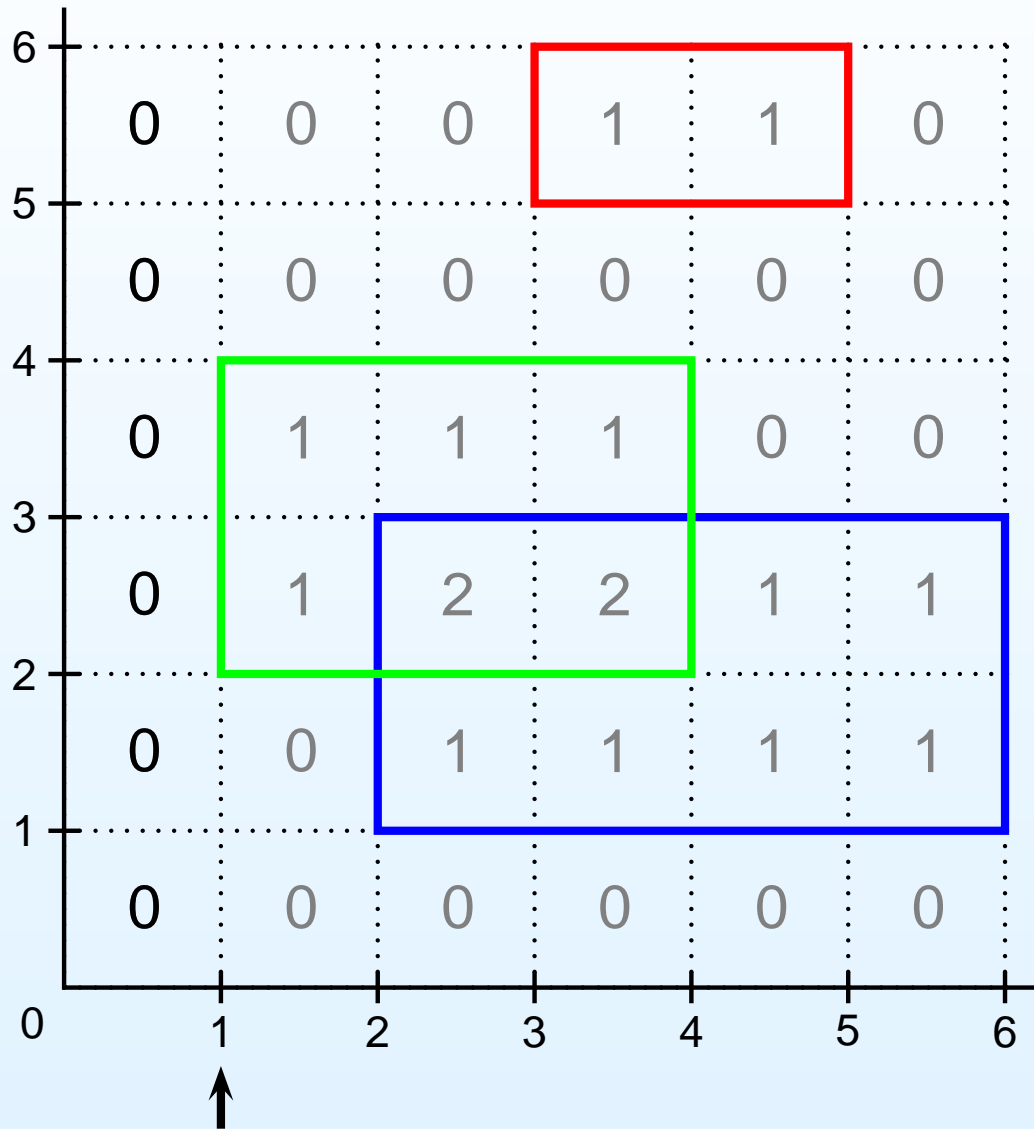


Find local maxima by sweeping through the height map



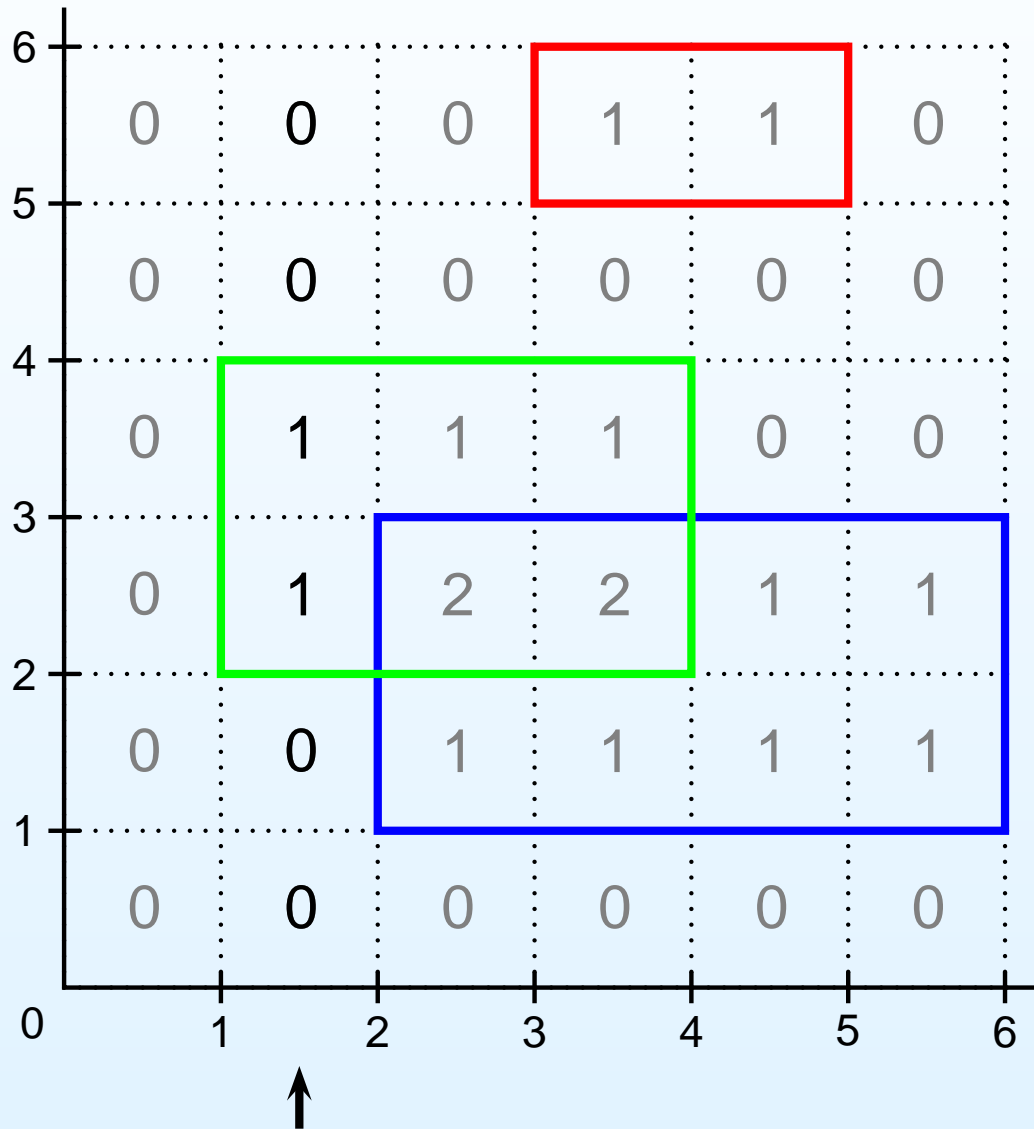
height map	last entered
0	
0	
0	
0	
0	
0	

Find local maxima by sweeping through the height map



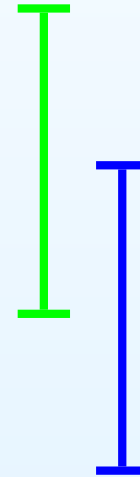
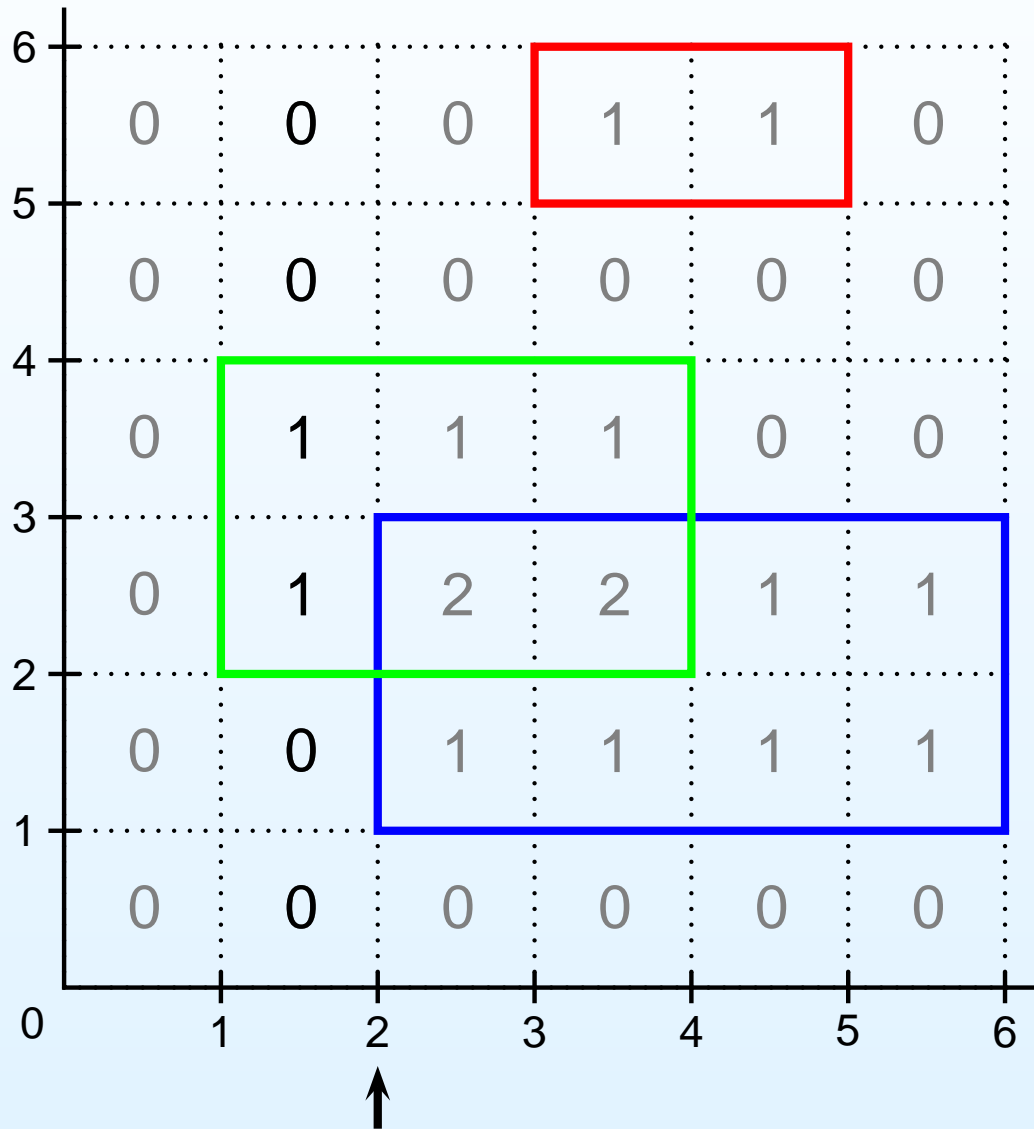
height map	last entered
0	
0	
0	
0	
0	
0	

Find local maxima by sweeping through the height map



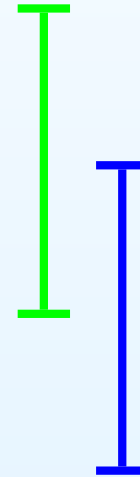
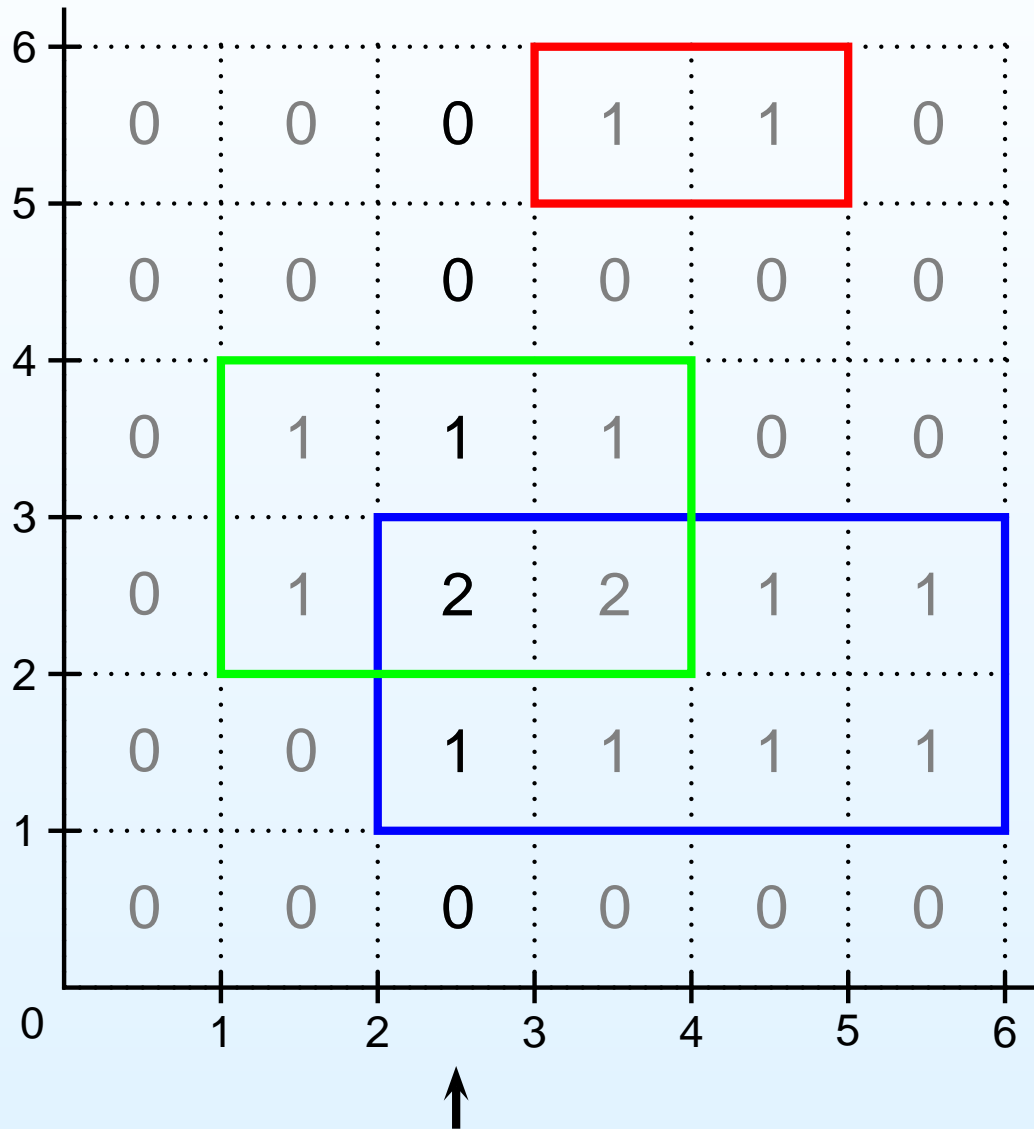
height map	last entered
0	
0	
1	
1	
0	
0	

Find local maxima by sweeping through the height map



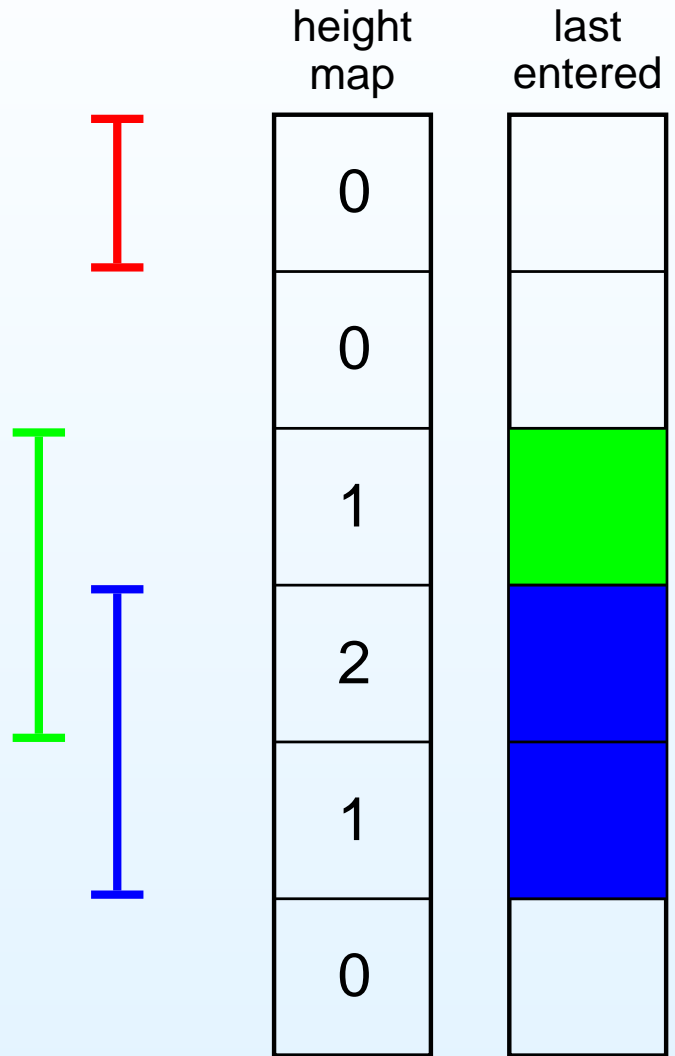
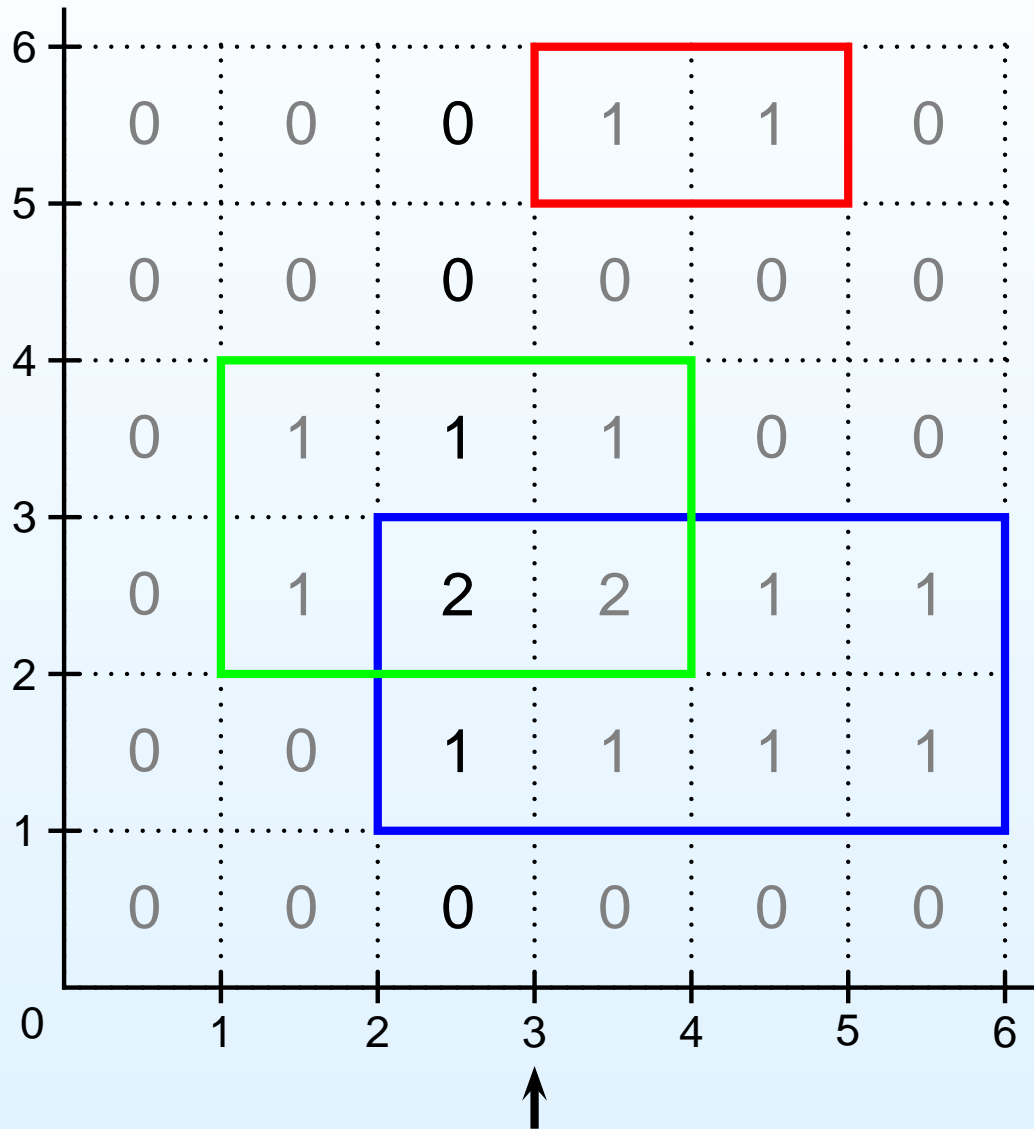
height map	last entered
0	
0	
1	
1	
0	
0	

Find local maxima by sweeping through the height map

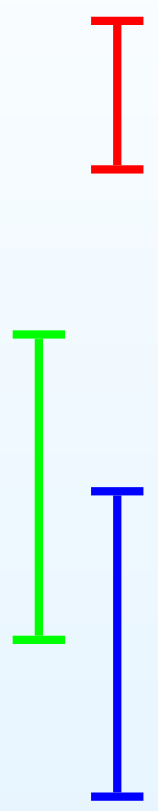
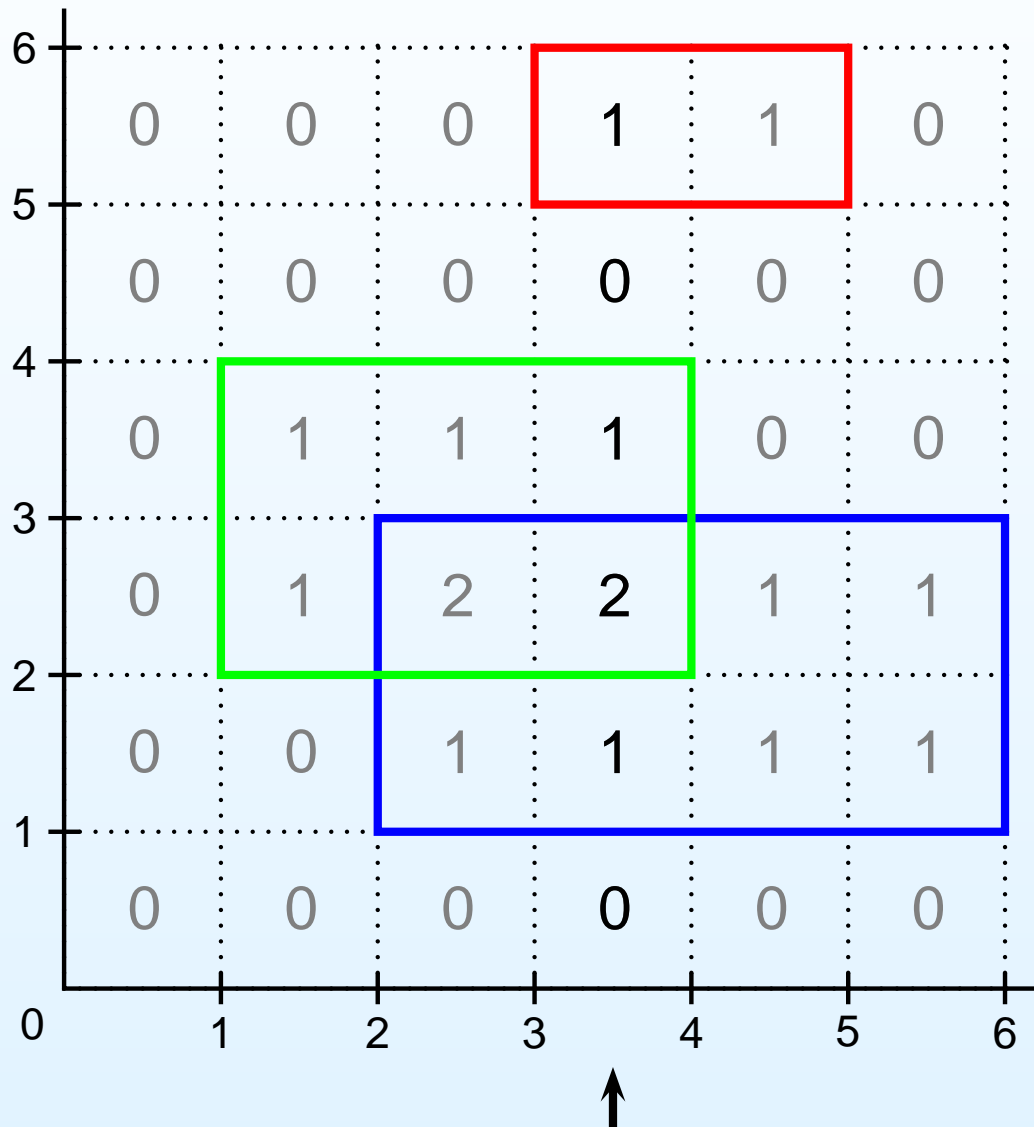


height map	last entered
0	
0	
1	
2	
1	
0	

Find local maxima by sweeping through the height map

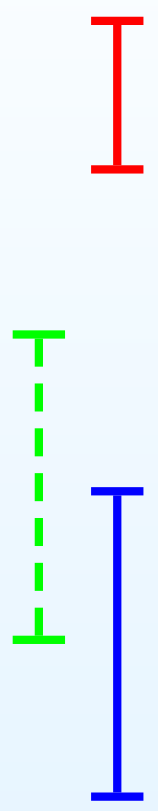
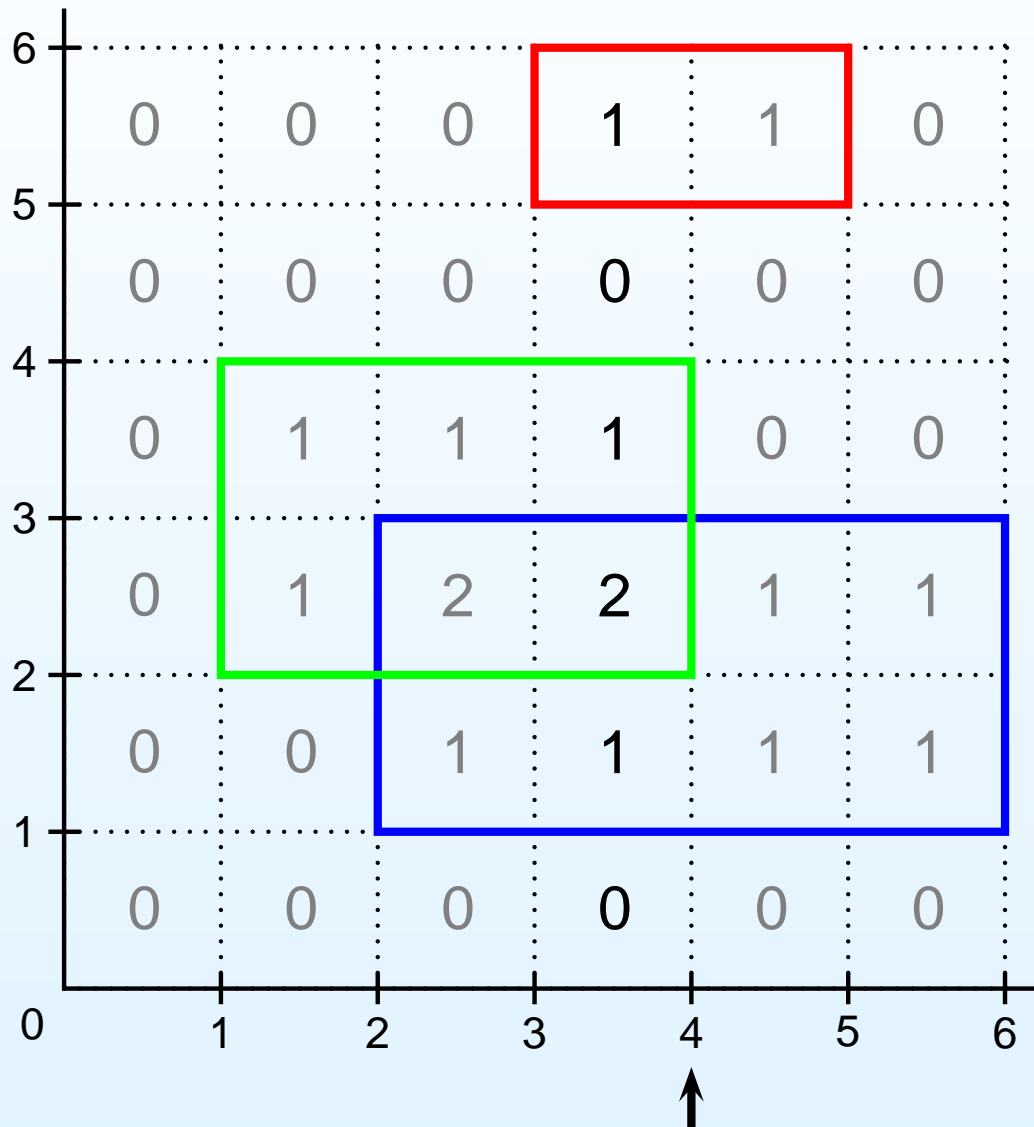


Find local maxima by sweeping through the height map



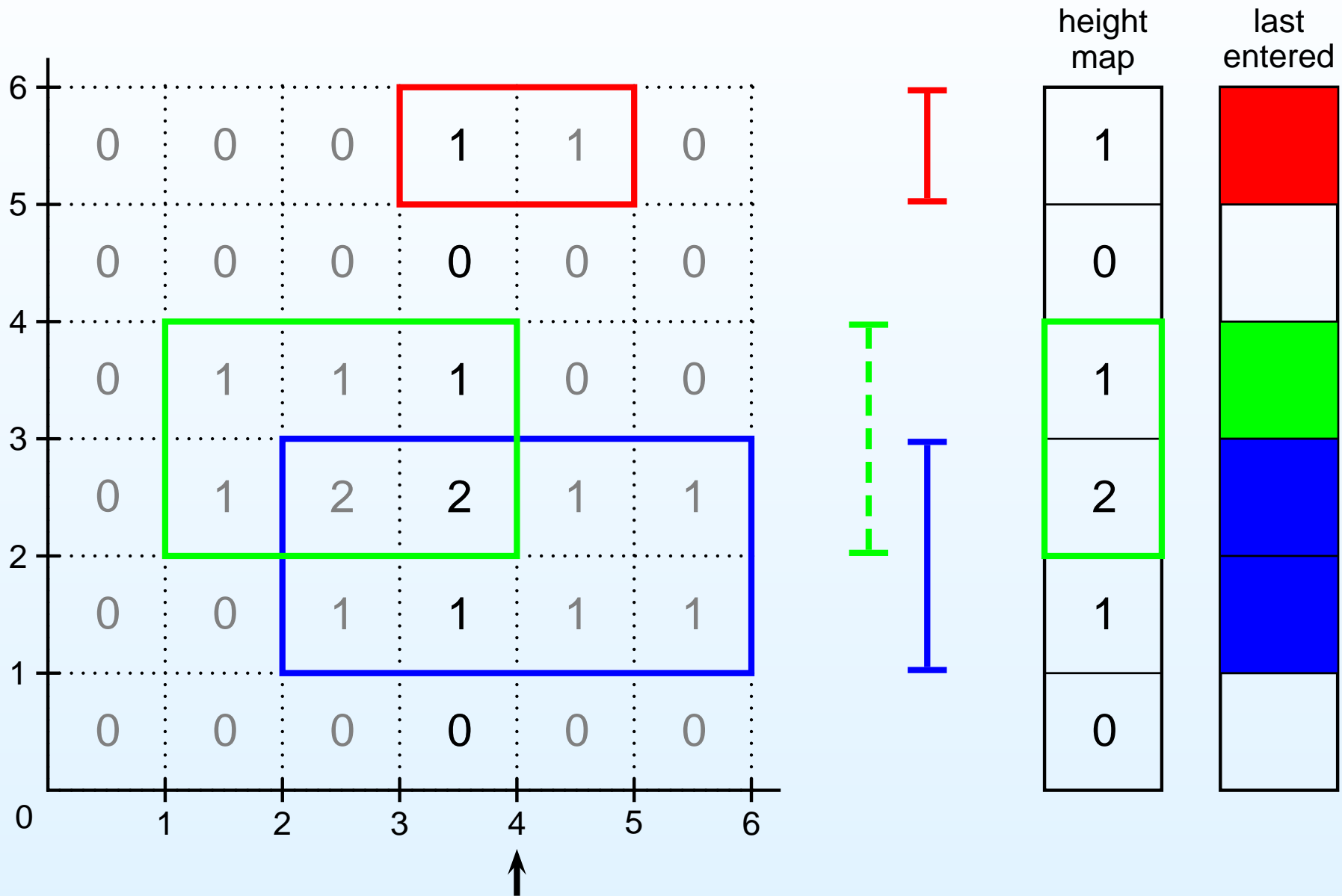
height map	last entered
1	
0	
1	
2	
1	
0	

Find local maxima by sweeping through the height map

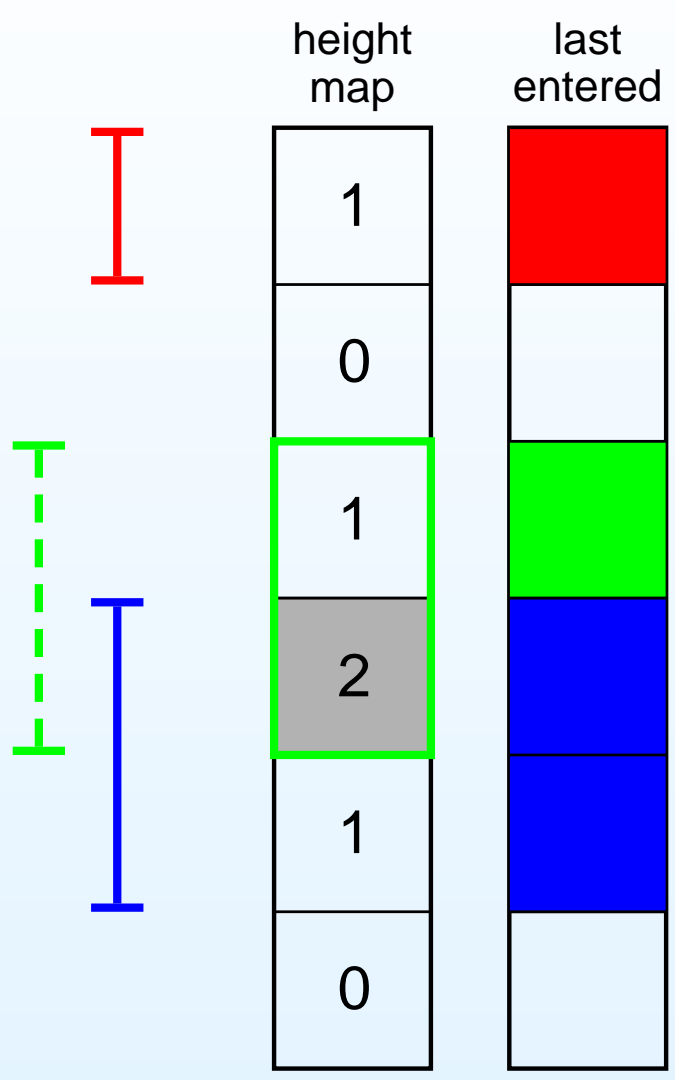
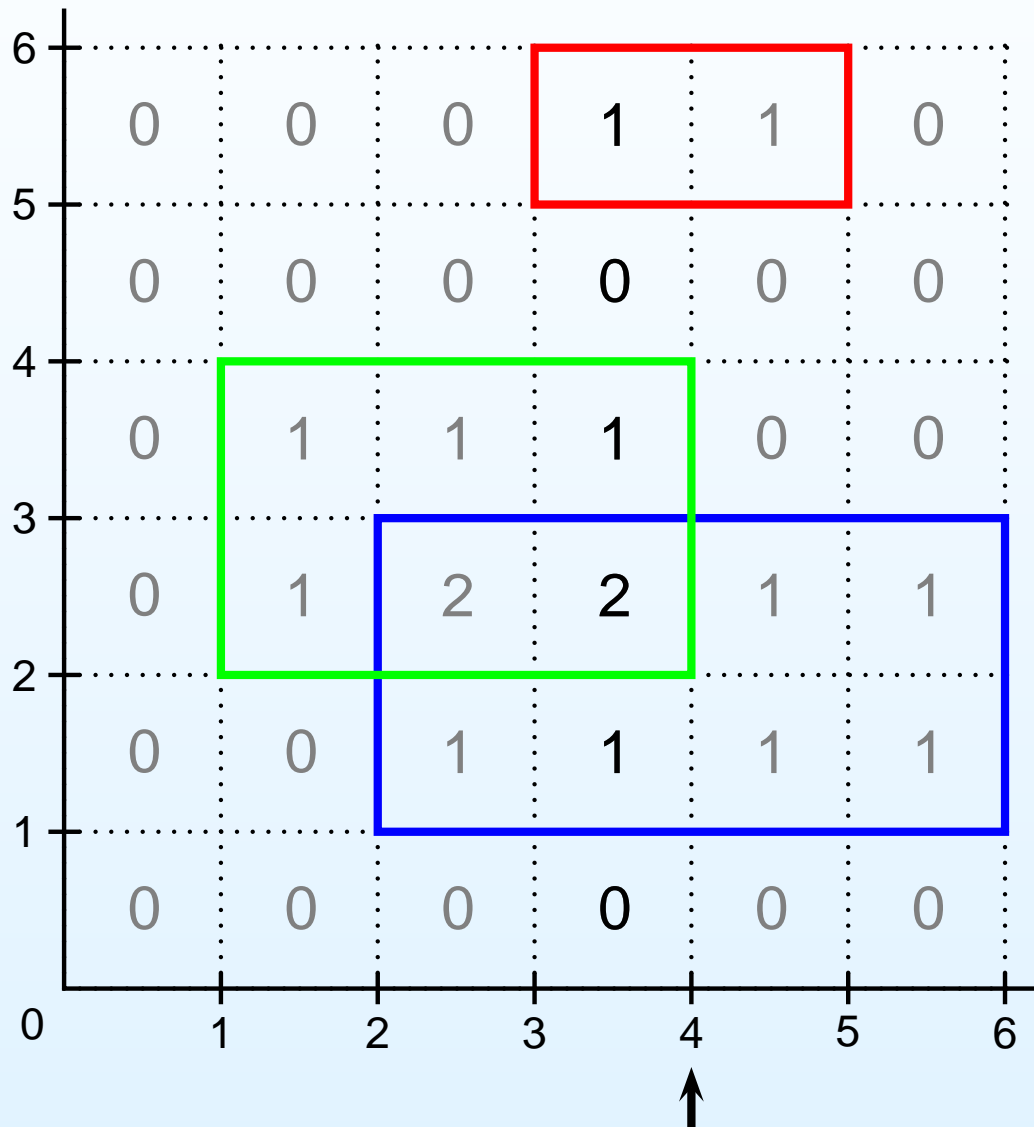


height map	last entered
1	
0	
1	
2	
1	
0	

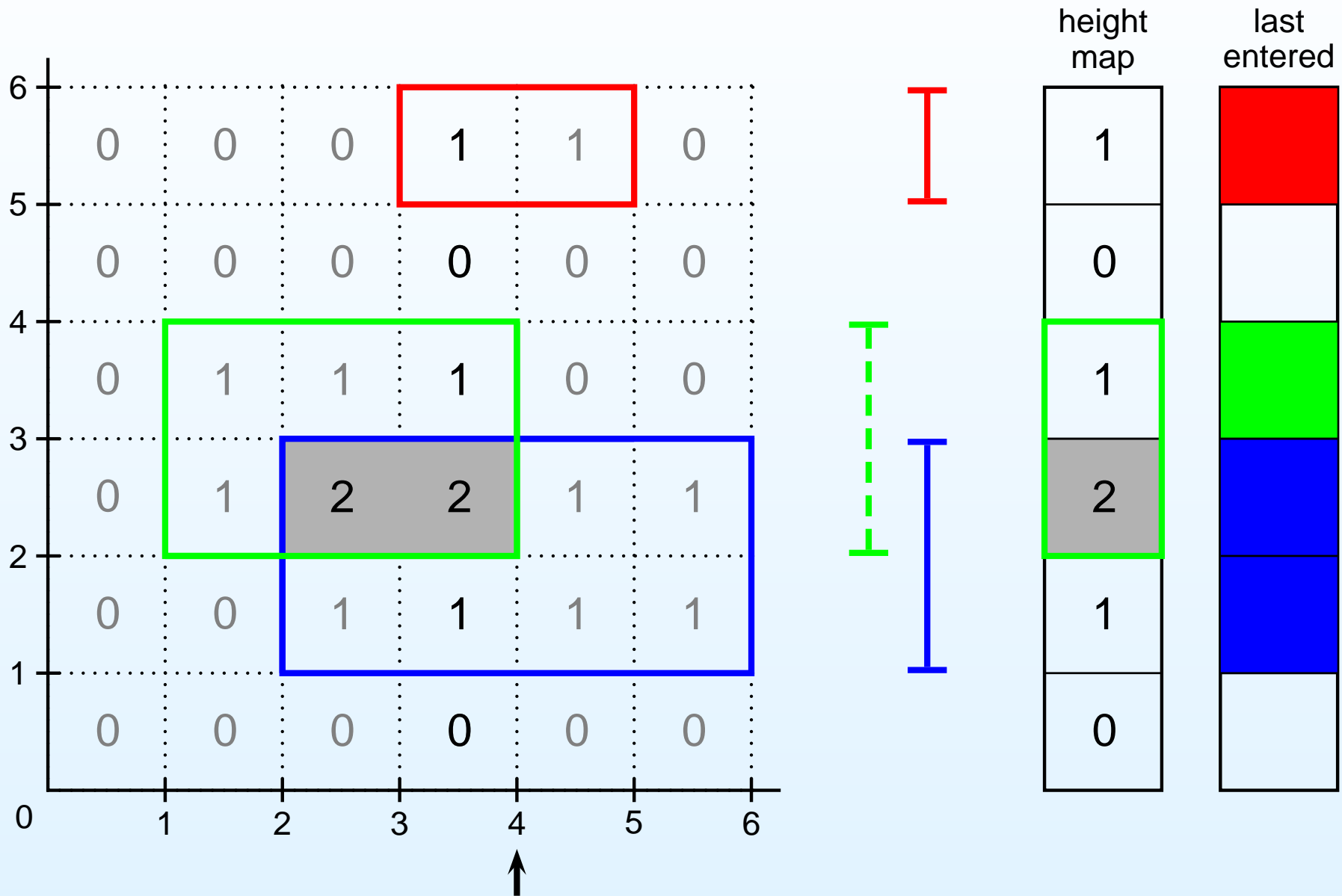
Find local maxima by sweeping through the height map



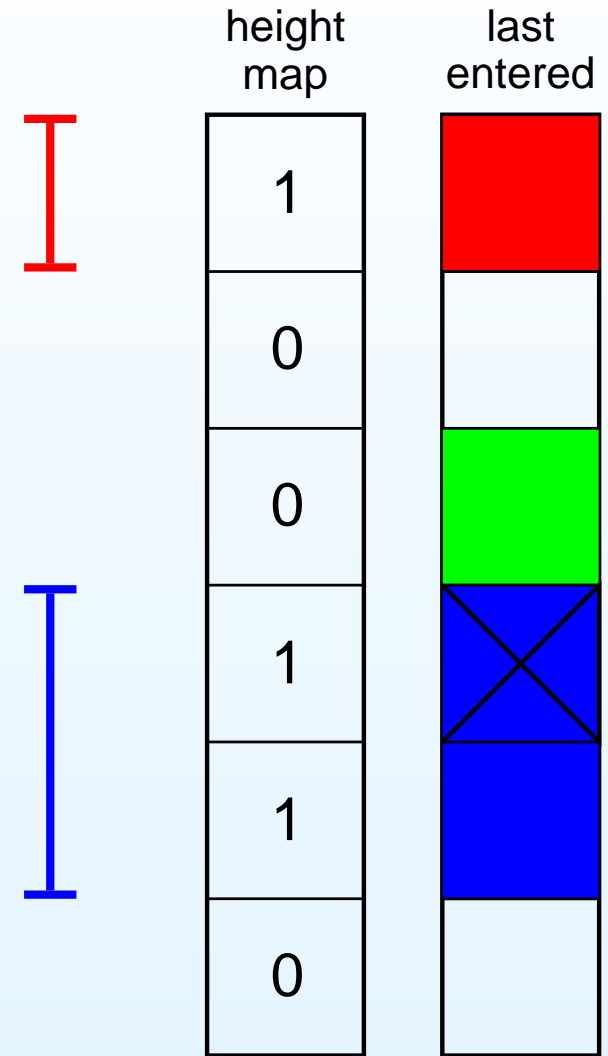
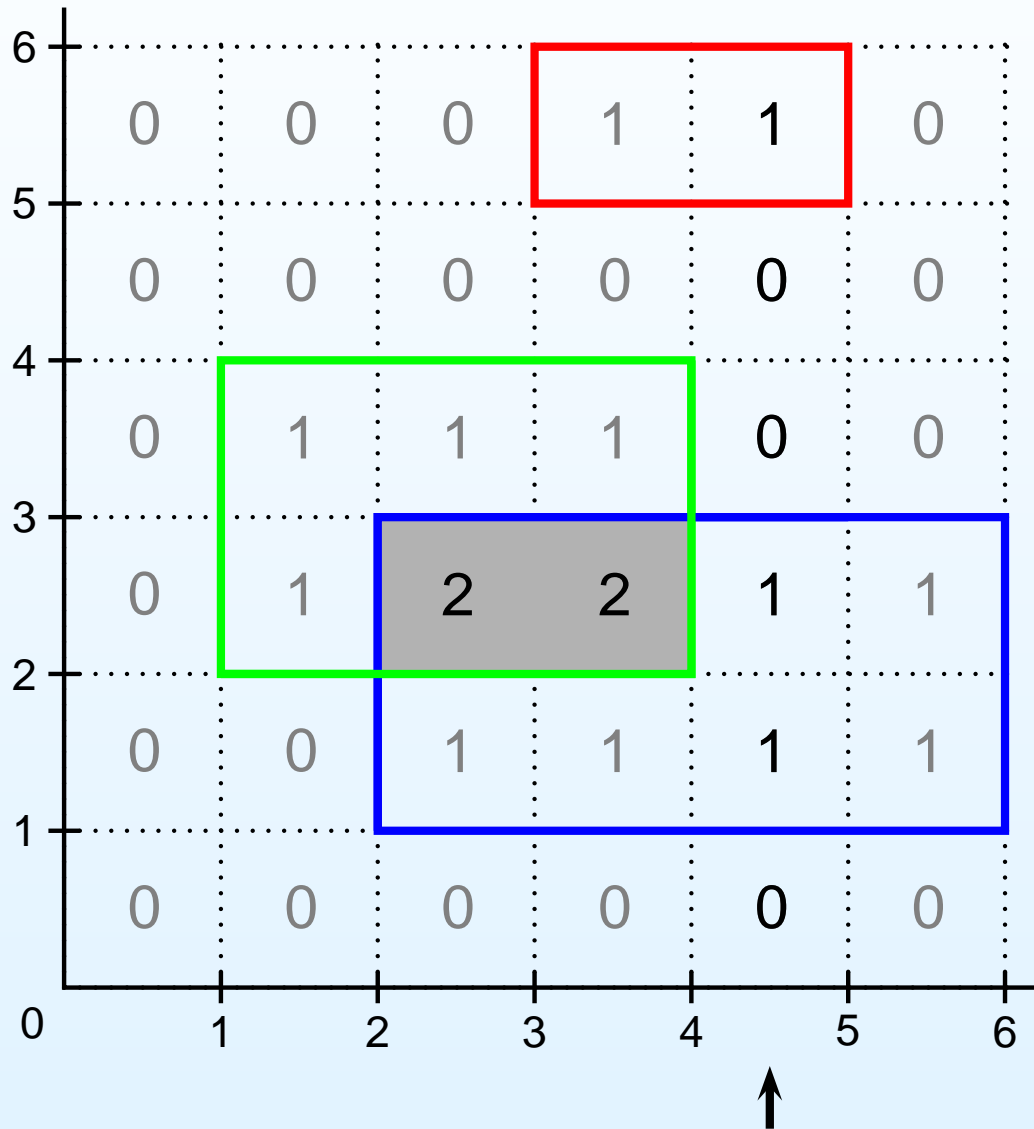
Find local maxima by sweeping through the height map



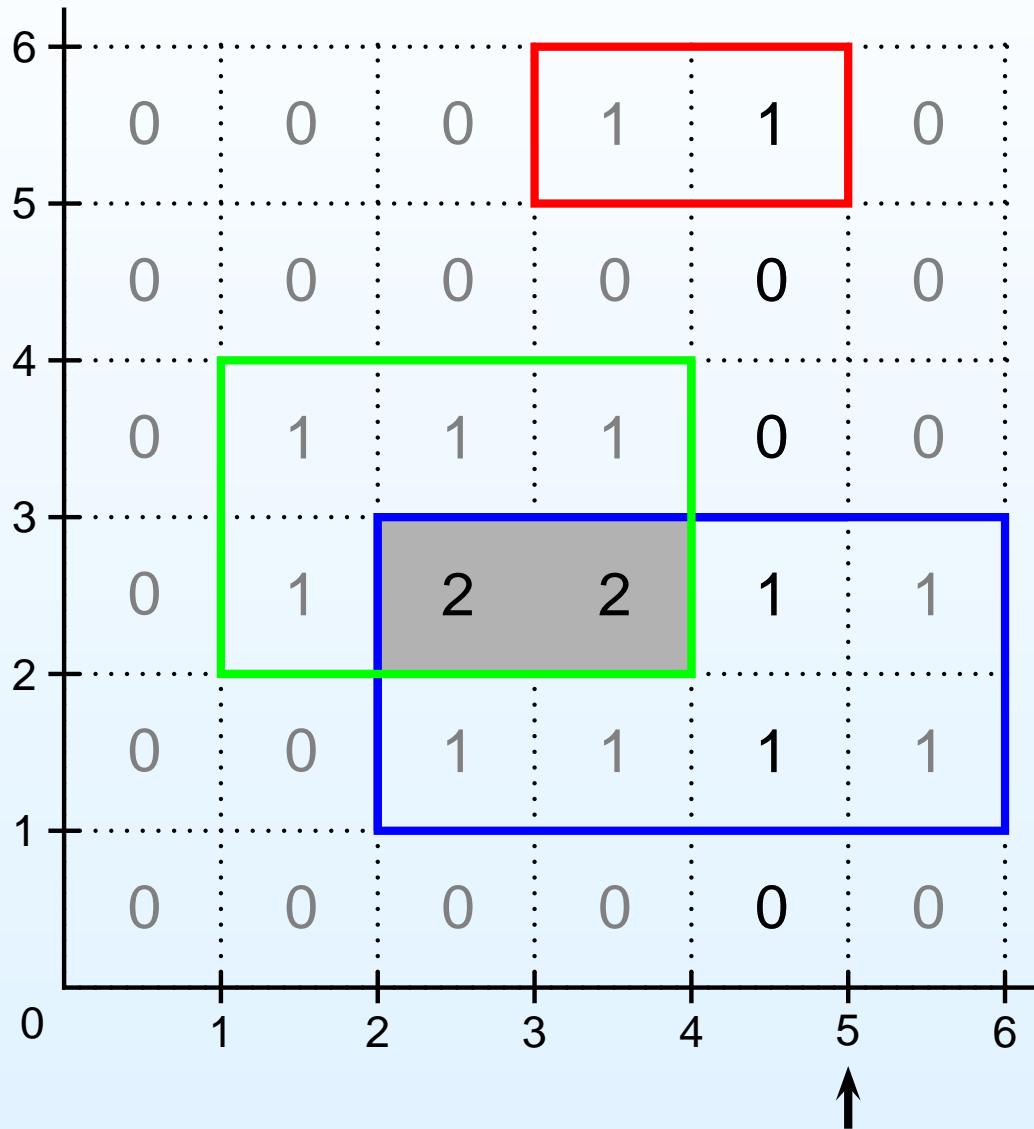
Find local maxima by sweeping through the height map



Find local maxima by sweeping through the height map



Find local maxima by sweeping through the height map



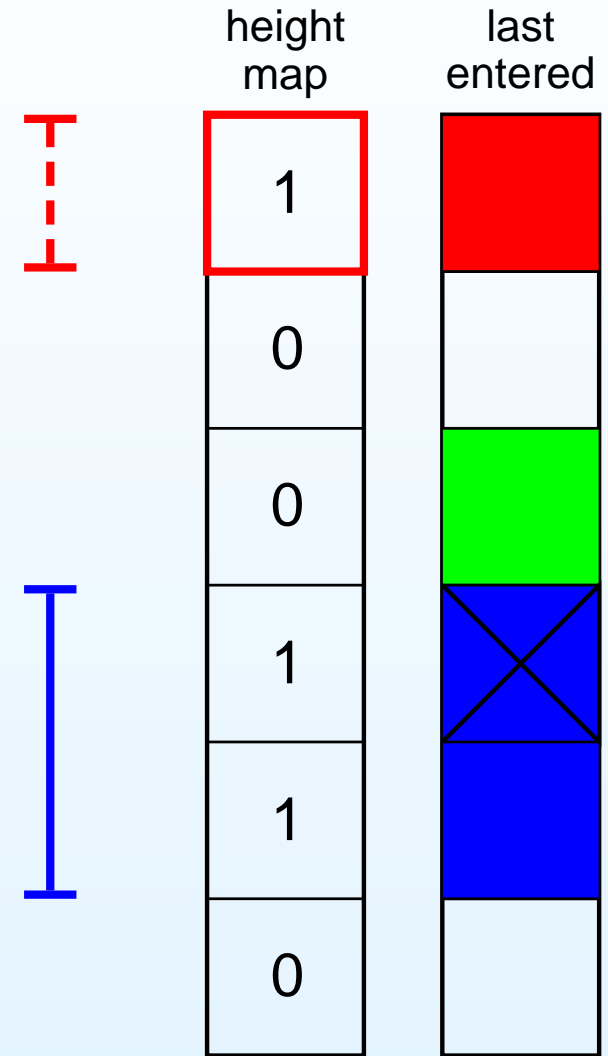
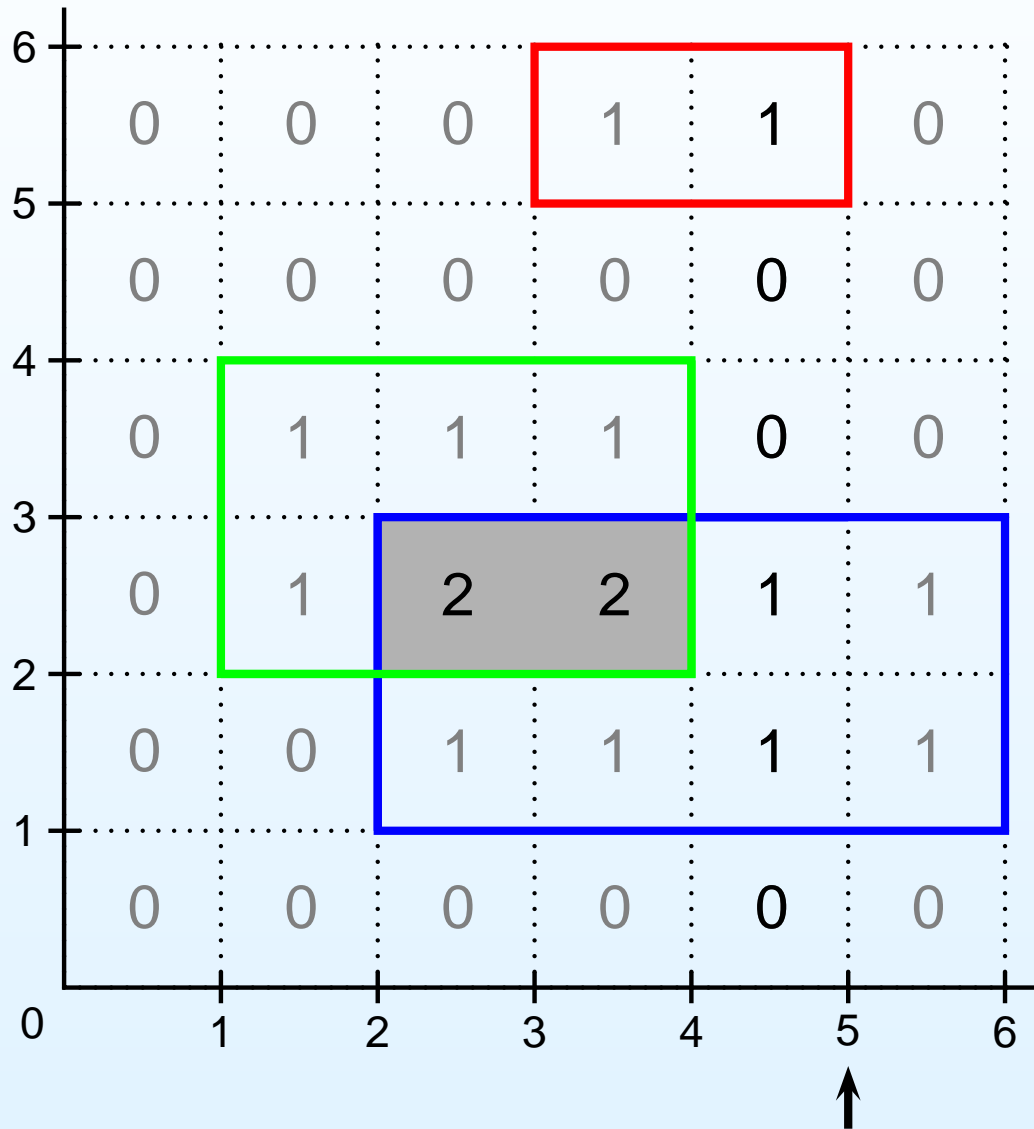
height map

1
0
0
1
1
0

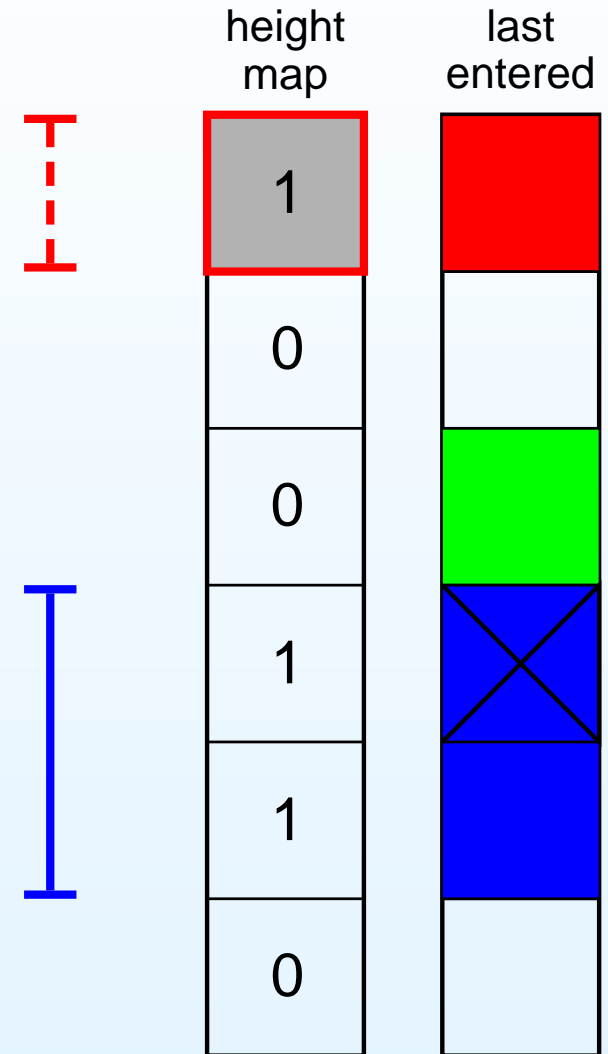
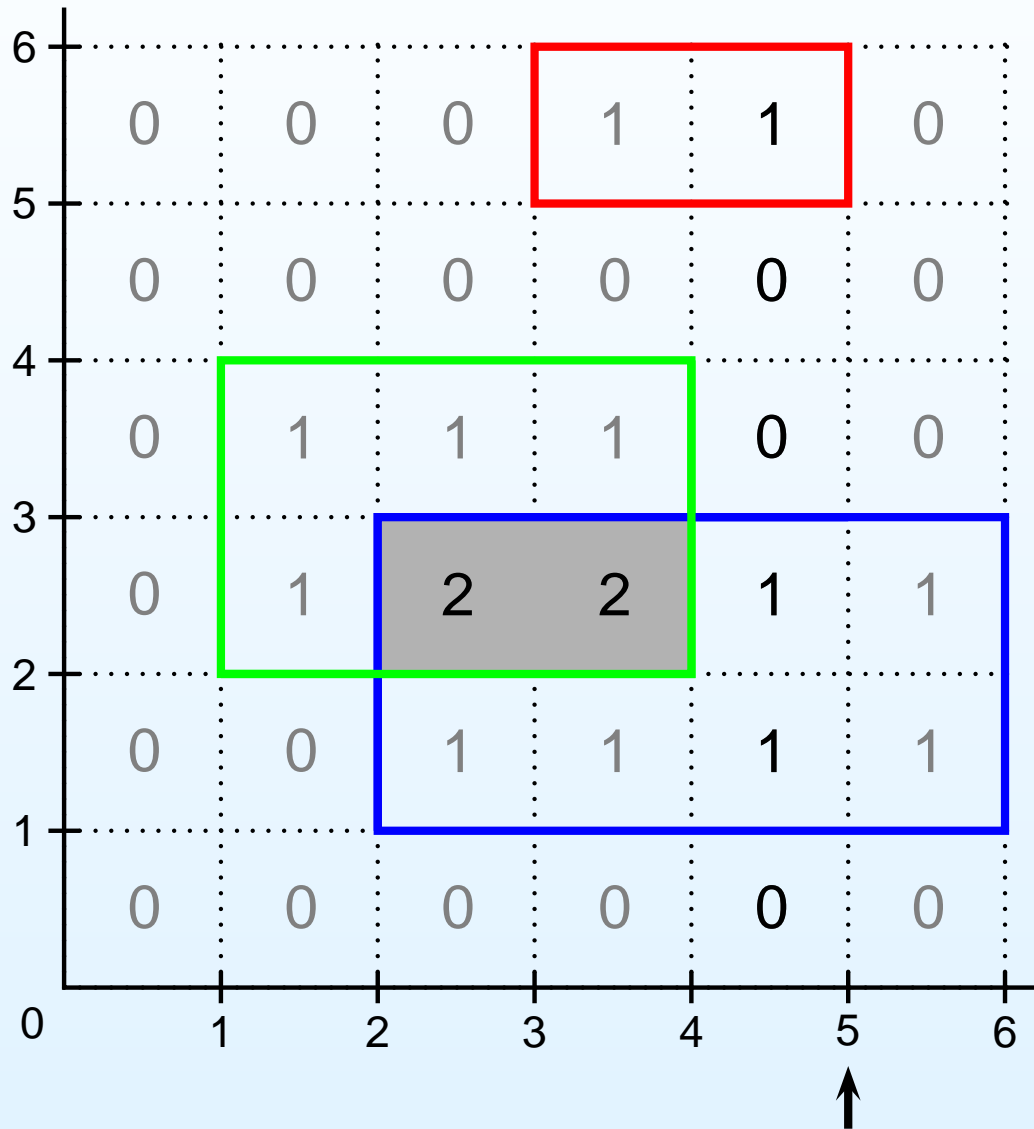
last entered

X

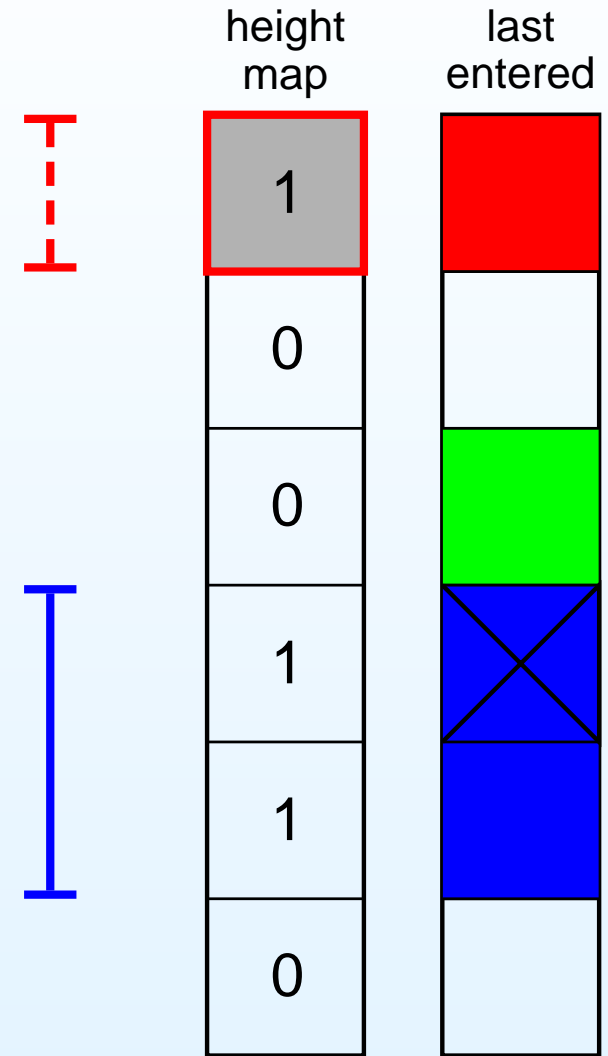
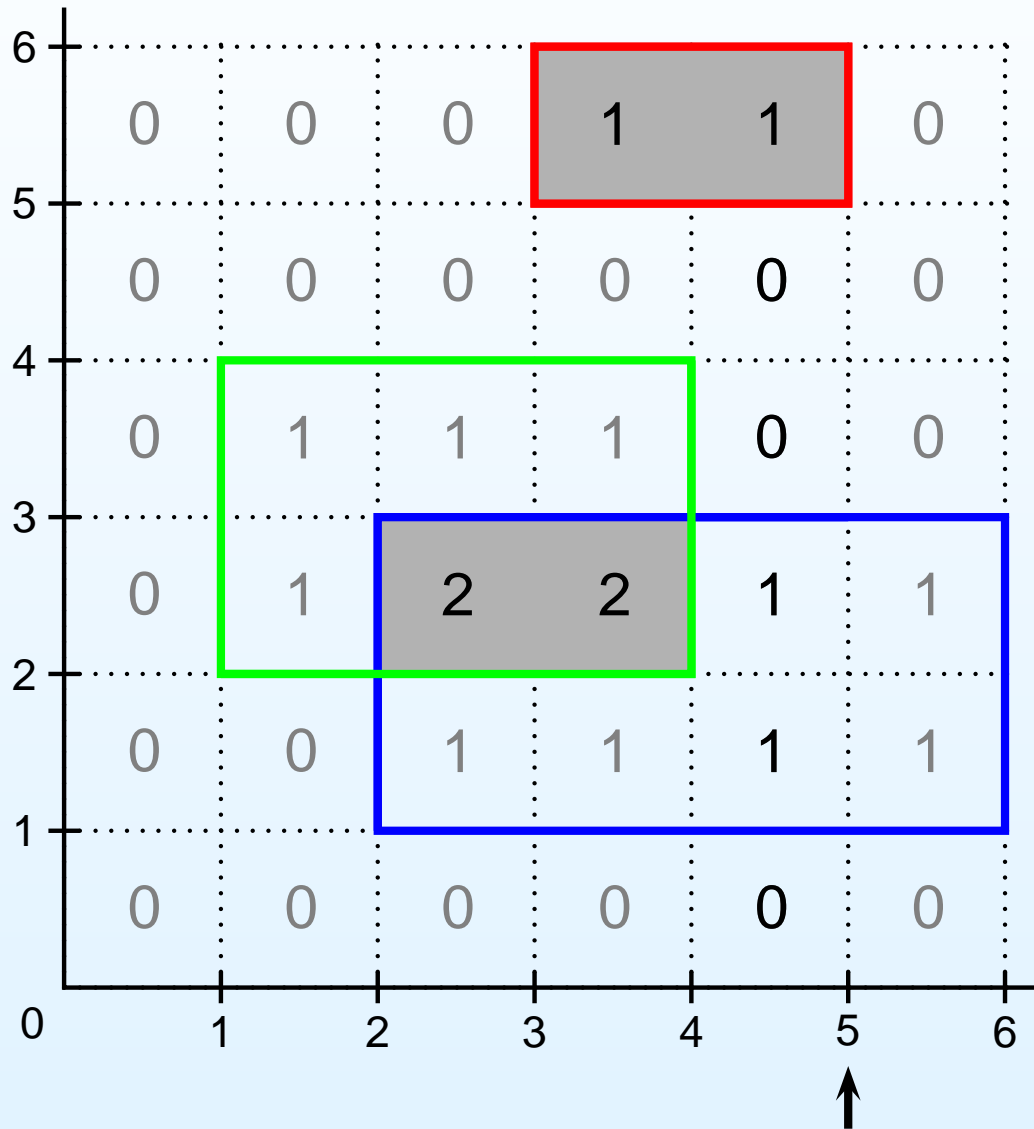
Find local maxima by sweeping through the height map



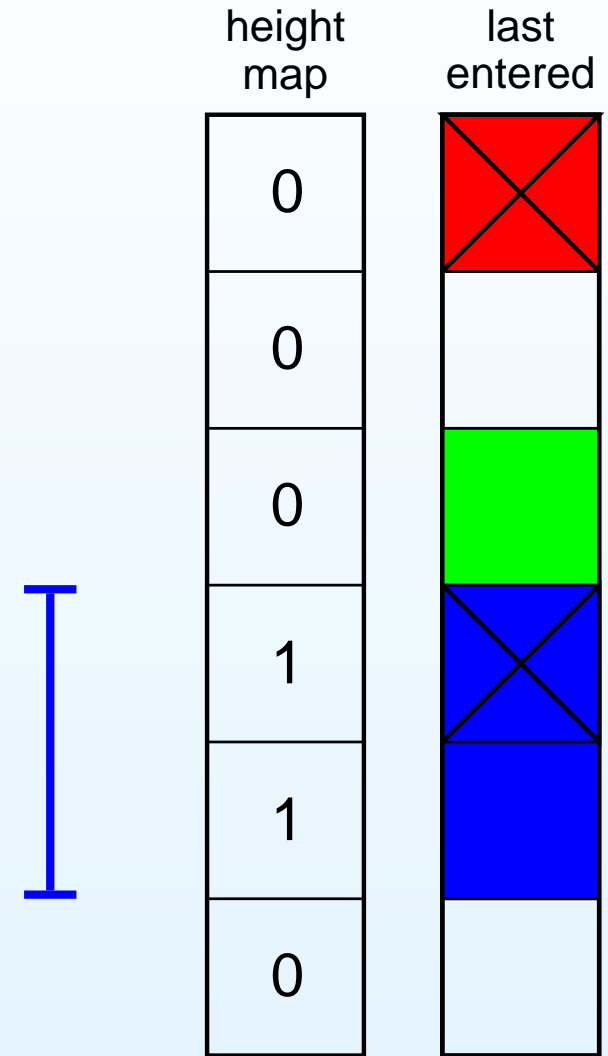
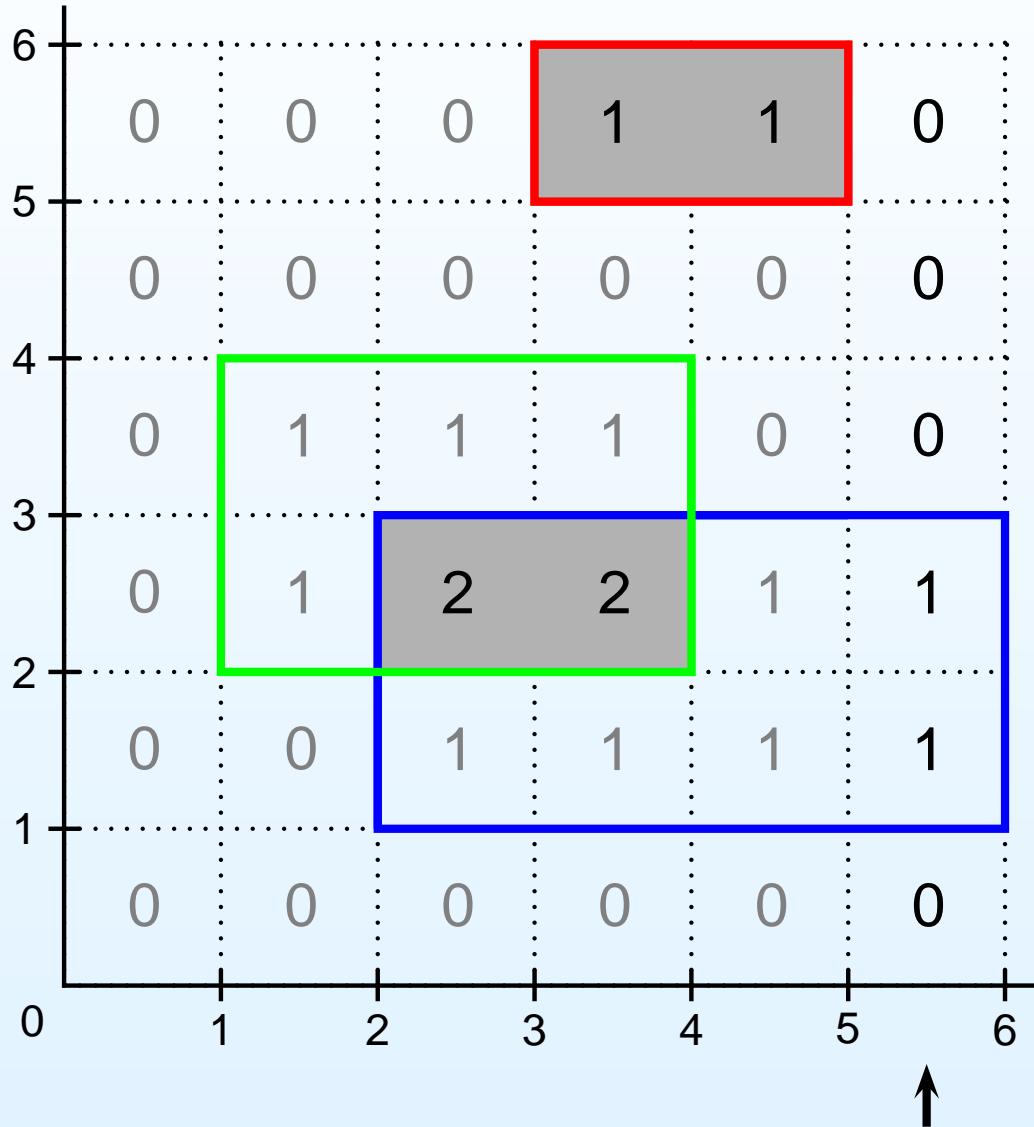
Find local maxima by sweeping through the height map



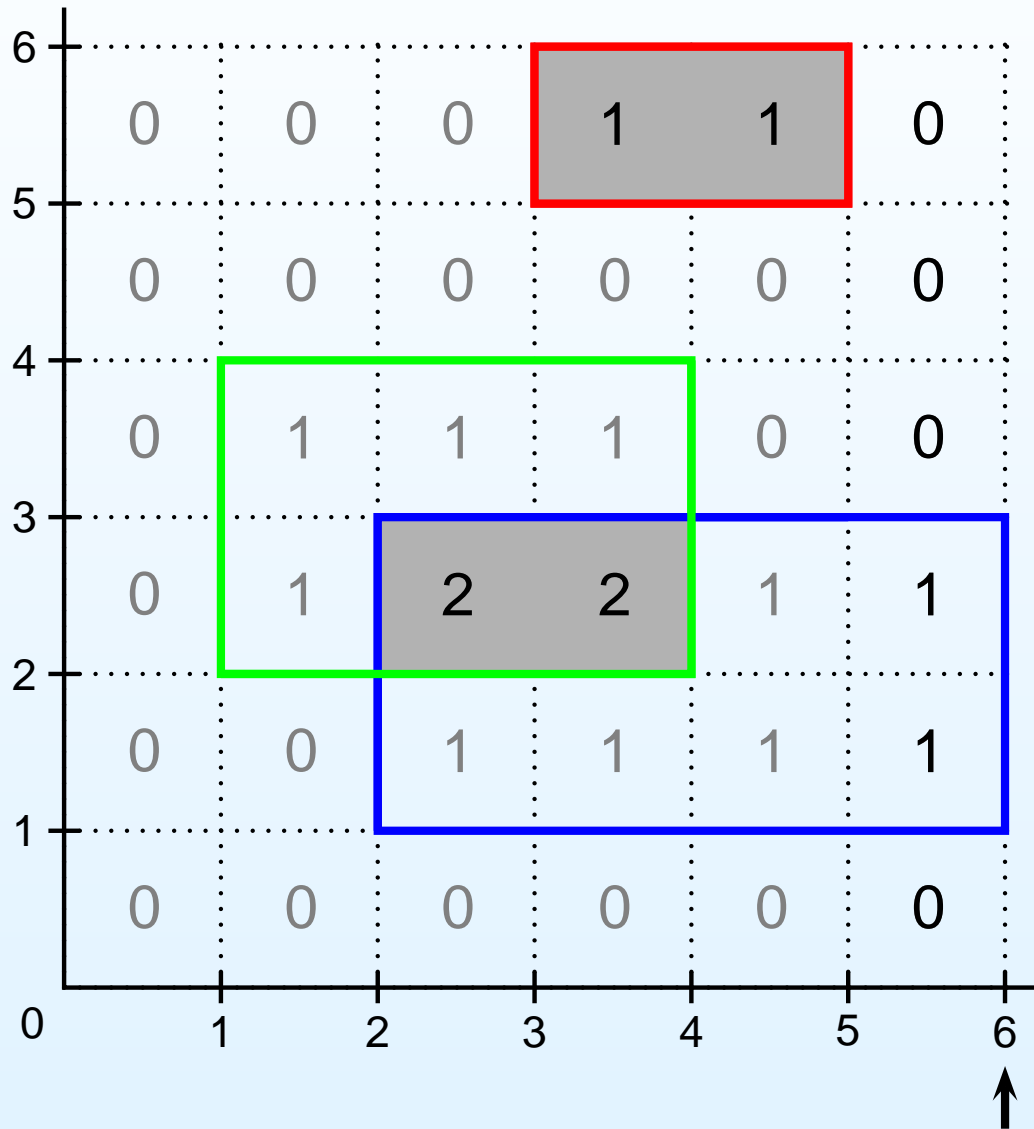
Find local maxima by sweeping through the height map



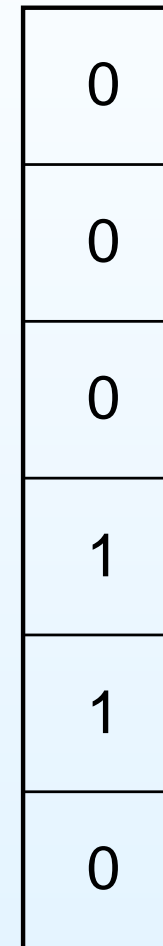
Find local maxima by sweeping through the height map



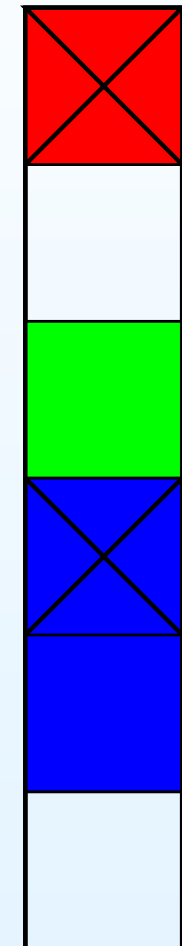
Find local maxima by sweeping through the height map



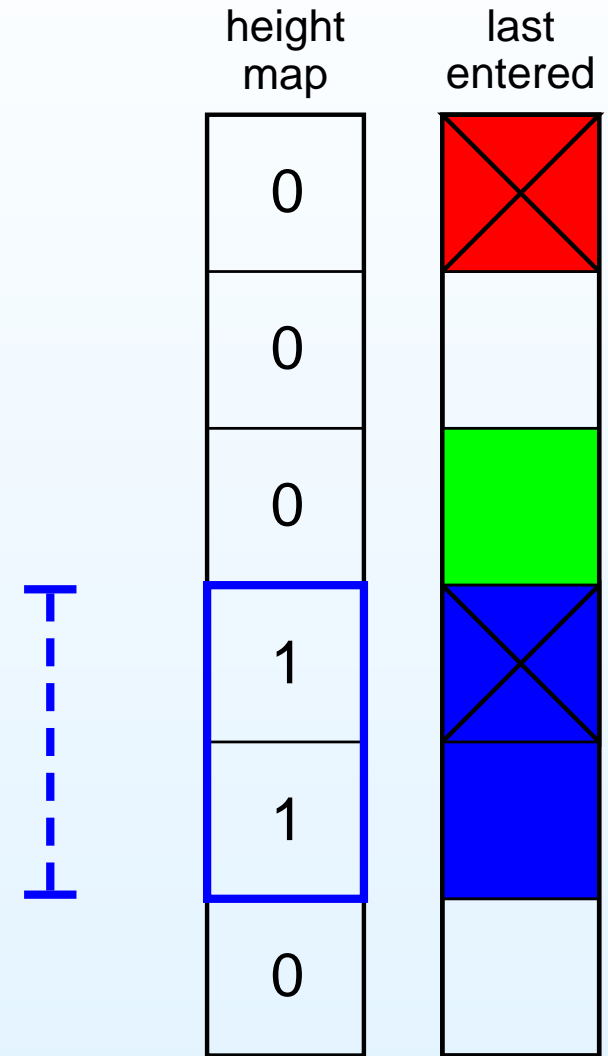
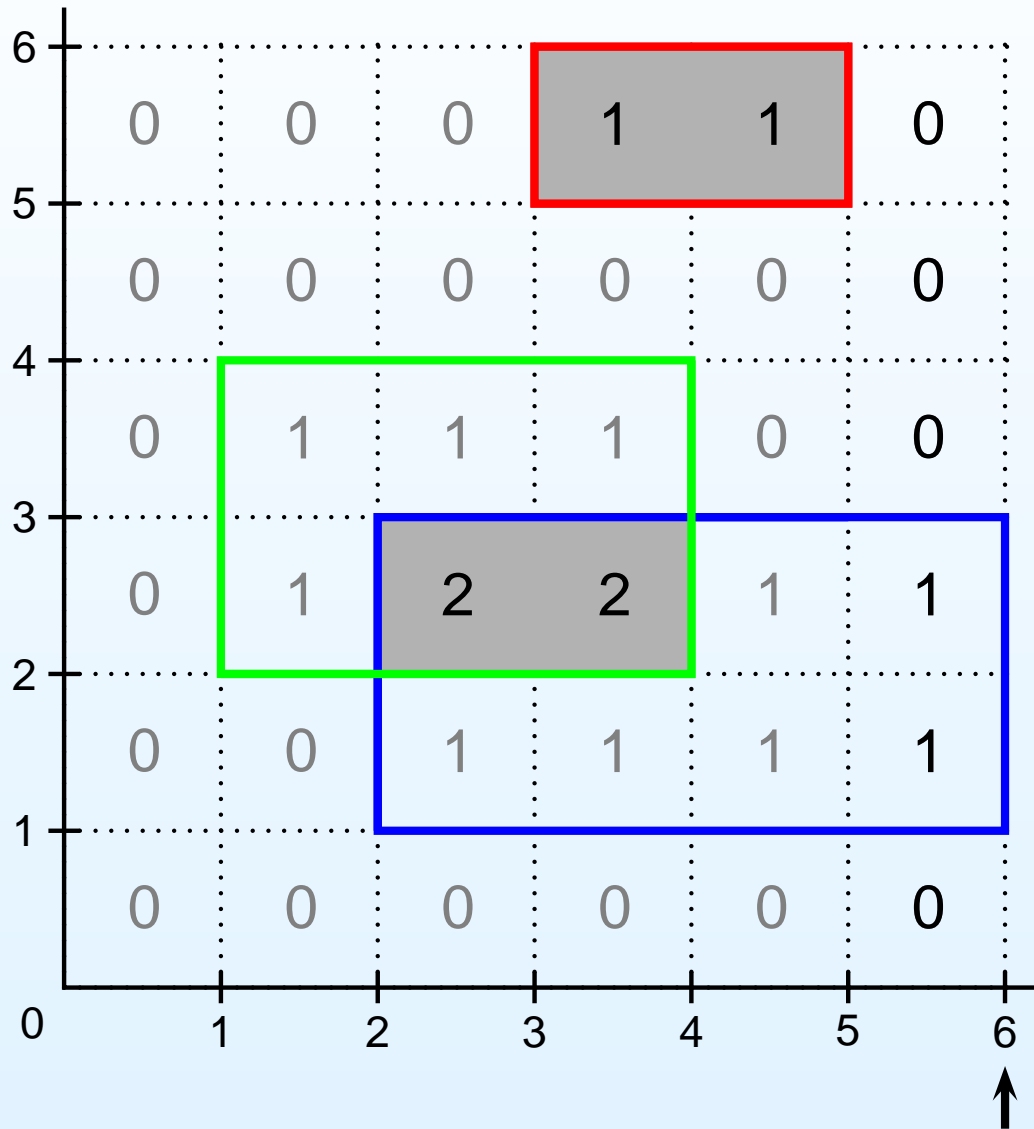
height map



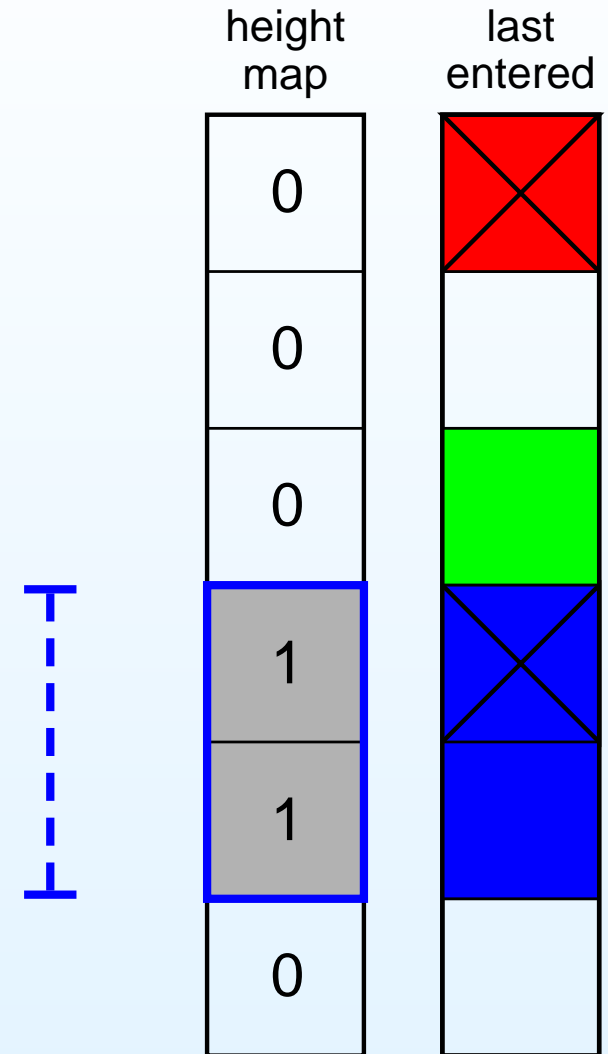
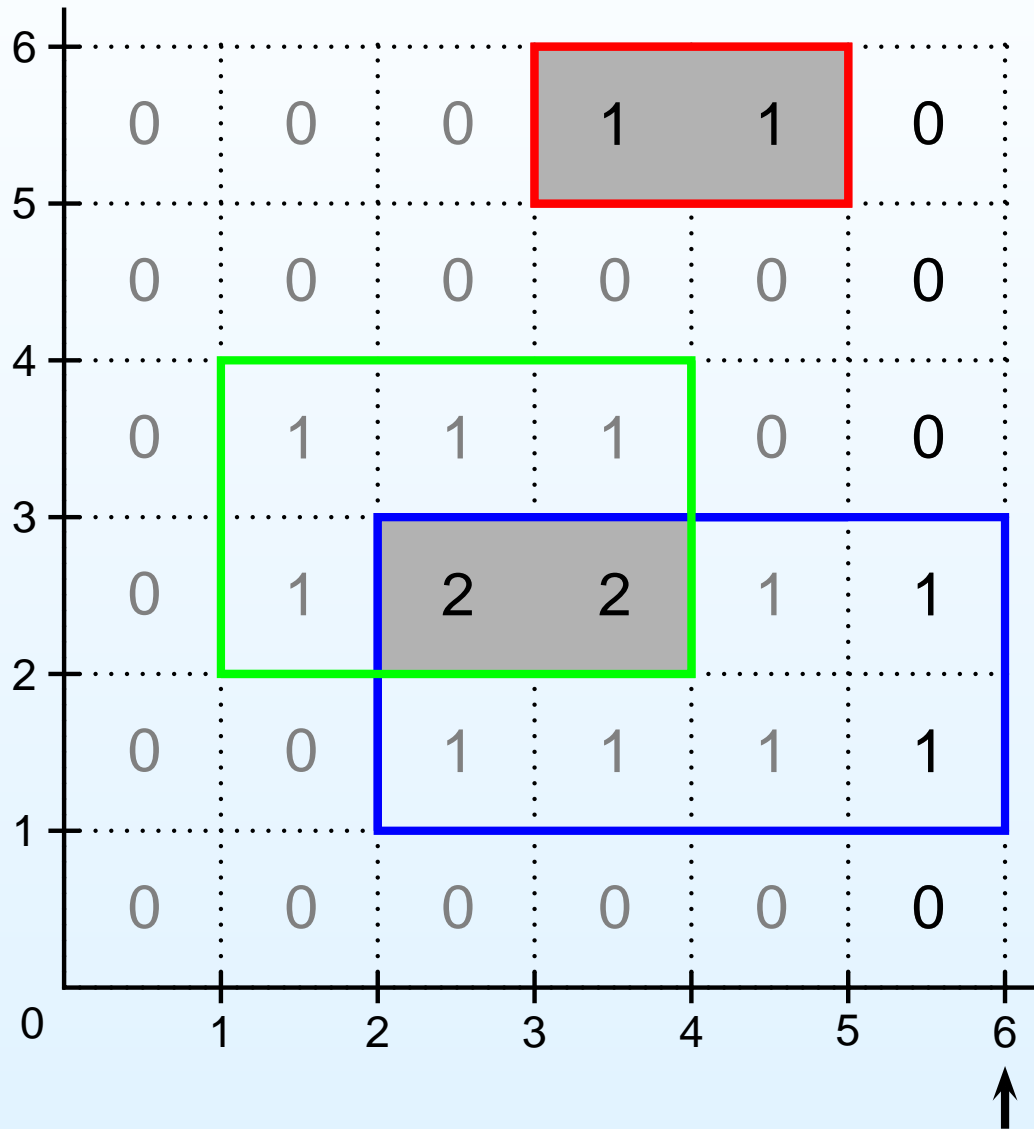
last entered



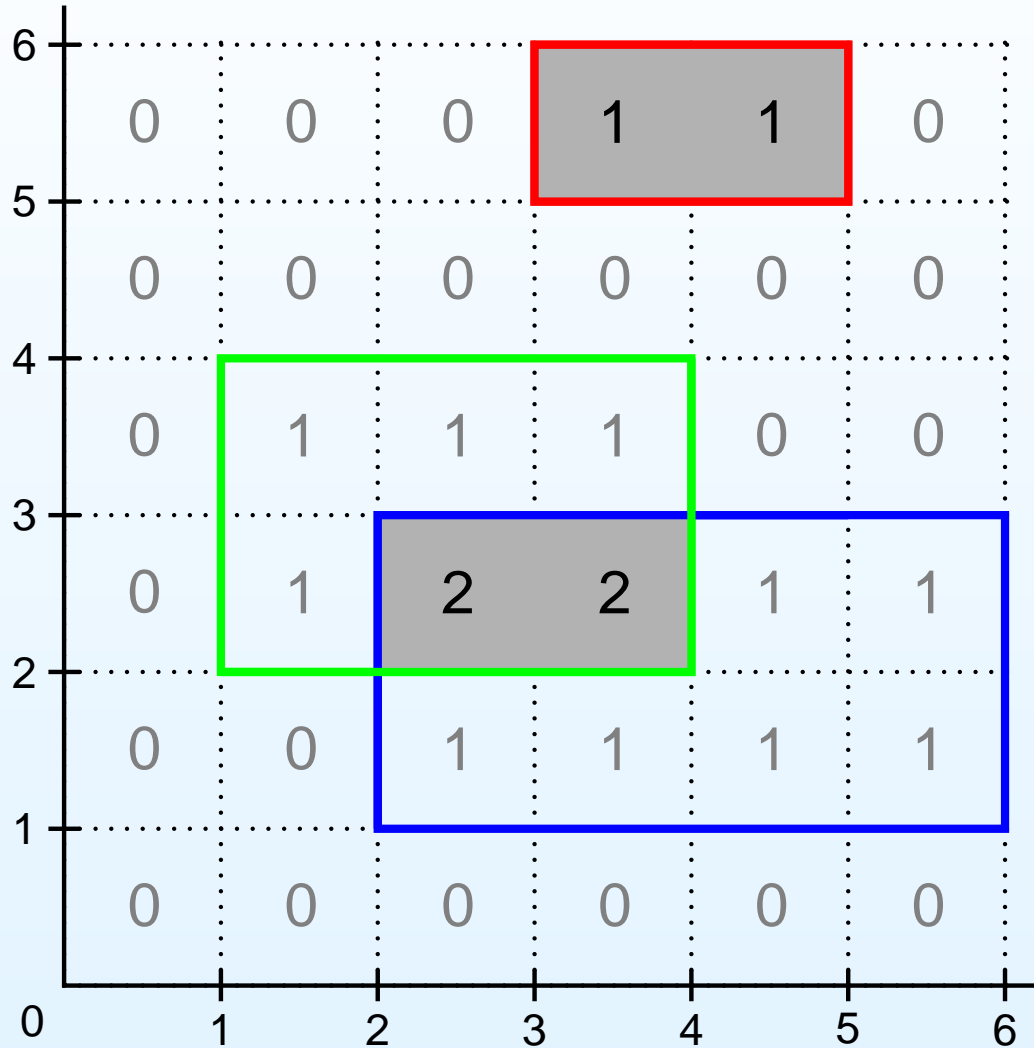
Find local maxima by sweeping through the height map



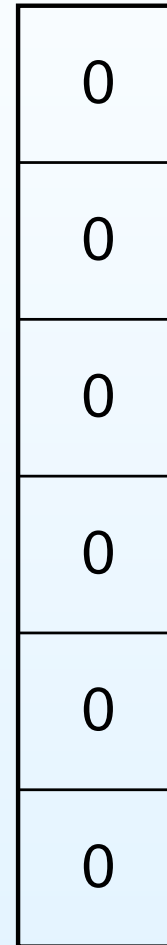
Find local maxima by sweeping through the height map



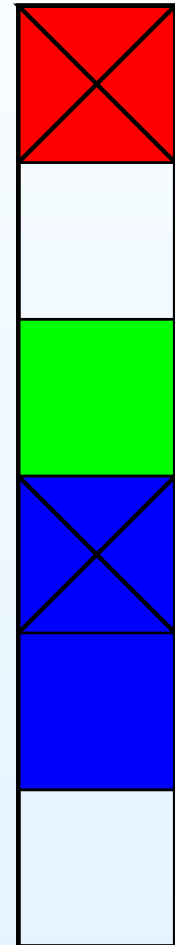
Find local maxima by sweeping through the height map



height map



last entered



Time and space complexity of the algorithm

For bivariate interval censored data:

- time complexity: $O(n^2)$
- space complexity:
 - computation: $O(n)$
 - output: $O(n^2)$

Time and space complexity of the algorithm

For bivariate interval censored data:

- time complexity: $O(n^2)$
- space complexity:
 - computation: $O(n)$
 - output: $O(n^2)$

For d -dimensional interval censored data:

- time complexity: $O(n^d)$
- space complexity:
 - computation: $O(n^{d-1})$
 - output: $O(n^d)$

Simulation study

Bivariate current status data from a simple exponential model:

- Variables of interest: $X, Y \sim \text{Exp}(1)$
- Observation times: $U, V \sim \text{Exp}(1)$
- X, Y, U, V mutually independent
- 50 simulations for sample sizes

50 100 250 500 1,000 2,500 5,000 10,000

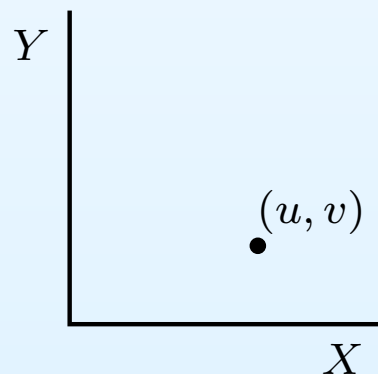
Simulation study

Bivariate current status data from a simple exponential model:

- Variables of interest: $X, Y \sim \text{Exp}(1)$
- Observation times: $U, V \sim \text{Exp}(1)$
- X, Y, U, V mutually independent
- 50 simulations for sample sizes

50 100 250 500 1,000 2,500 5,000 10,000

Observation rectangles:



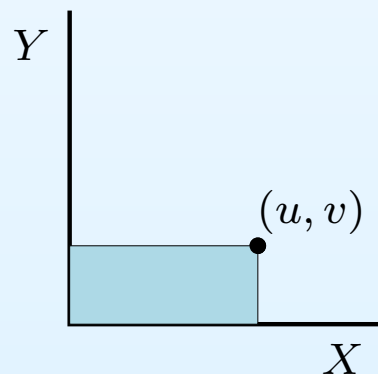
Simulation study

Bivariate current status data from a simple exponential model:

- Variables of interest: $X, Y \sim \text{Exp}(1)$
- Observation times: $U, V \sim \text{Exp}(1)$
- X, Y, U, V mutually independent
- 50 simulations for sample sizes

50 100 250 500 1,000 2,500 5,000 10,000

Observation rectangles:



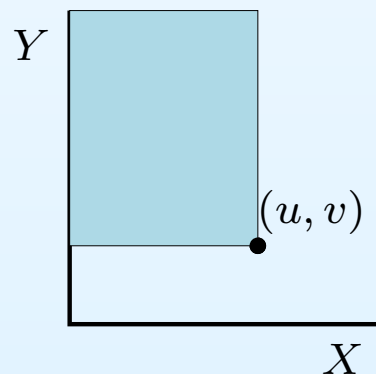
Simulation study

Bivariate current status data from a simple exponential model:

- Variables of interest: $X, Y \sim \text{Exp}(1)$
- Observation times: $U, V \sim \text{Exp}(1)$
- X, Y, U, V mutually independent
- 50 simulations for sample sizes

50 100 250 500 1,000 2,500 5,000 10,000

Observation rectangles:



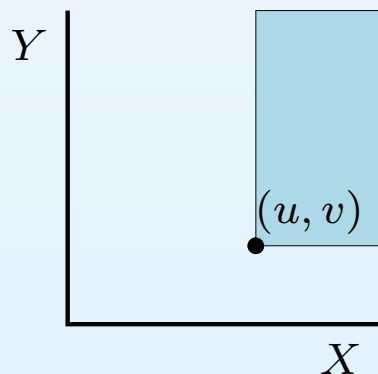
Simulation study

Bivariate current status data from a simple exponential model:

- Variables of interest: $X, Y \sim \text{Exp}(1)$
- Observation times: $U, V \sim \text{Exp}(1)$
- X, Y, U, V mutually independent
- 50 simulations for sample sizes

50 100 250 500 1,000 2,500 5,000 10,000

Observation rectangles:



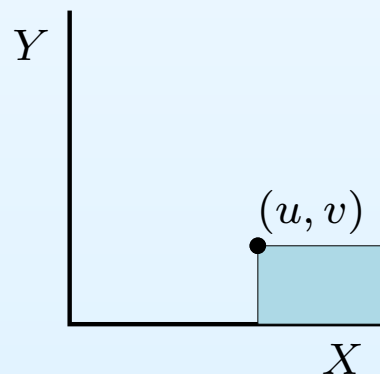
Simulation study

Bivariate current status data from a simple exponential model:

- Variables of interest: $X, Y \sim \text{Exp}(1)$
- Observation times: $U, V \sim \text{Exp}(1)$
- X, Y, U, V mutually independent
- 50 simulations for sample sizes

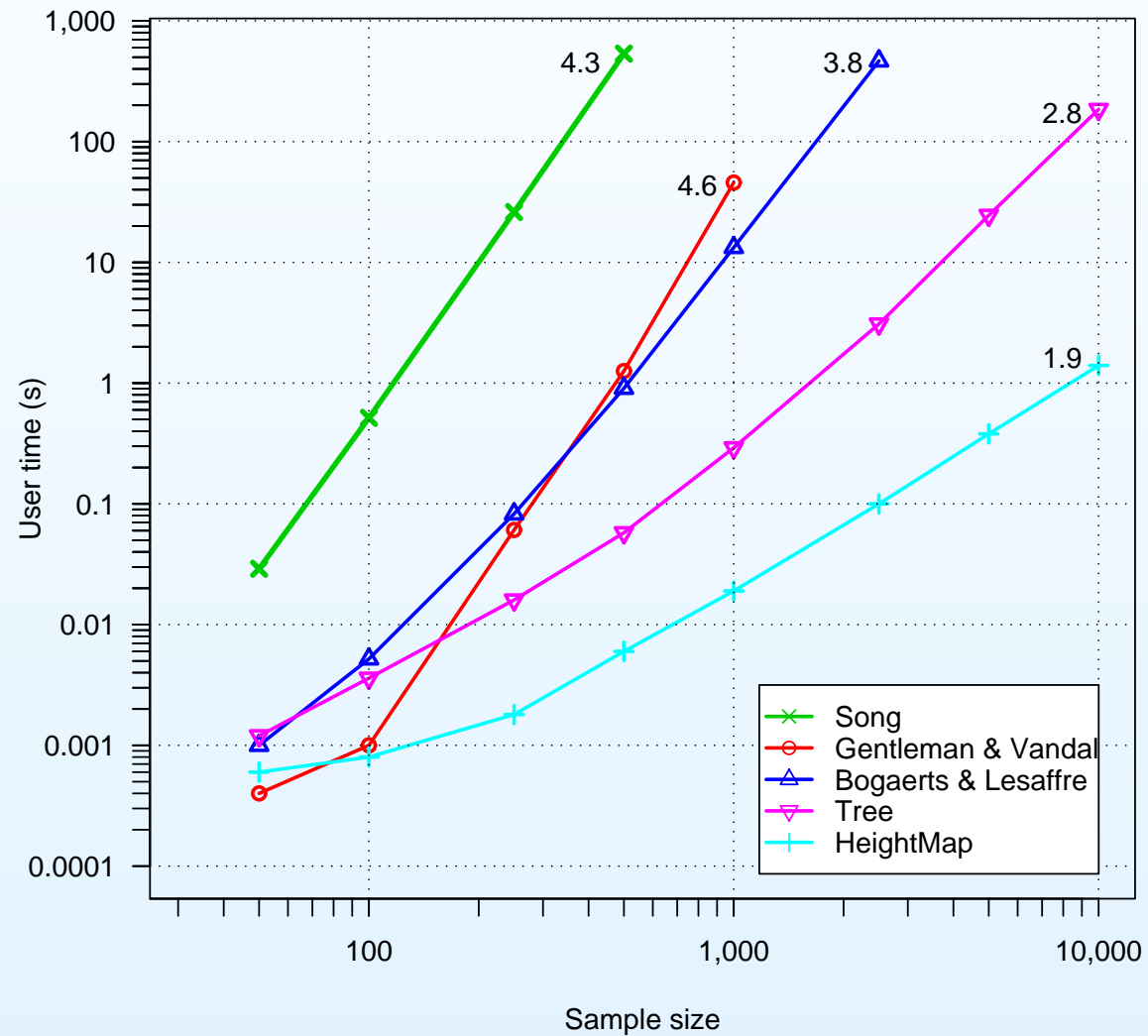
50 100 250 500 1,000 2,500 5,000 10,000

Observation rectangles:



Simulation study

Comparison of five reduction algorithms



Future work on the NPMLE

Computation:

- Optimization step

Future work on the NPMLE

Computation:

- Optimization step

Theory:

- Rate of convergence
- Limiting distribution

Thank you

Paper, presentation and R-package are available at
www.stat.washington.edu/marloes