Dynamic Planar Range Maxima Queries

(presented at ICALP 2011)

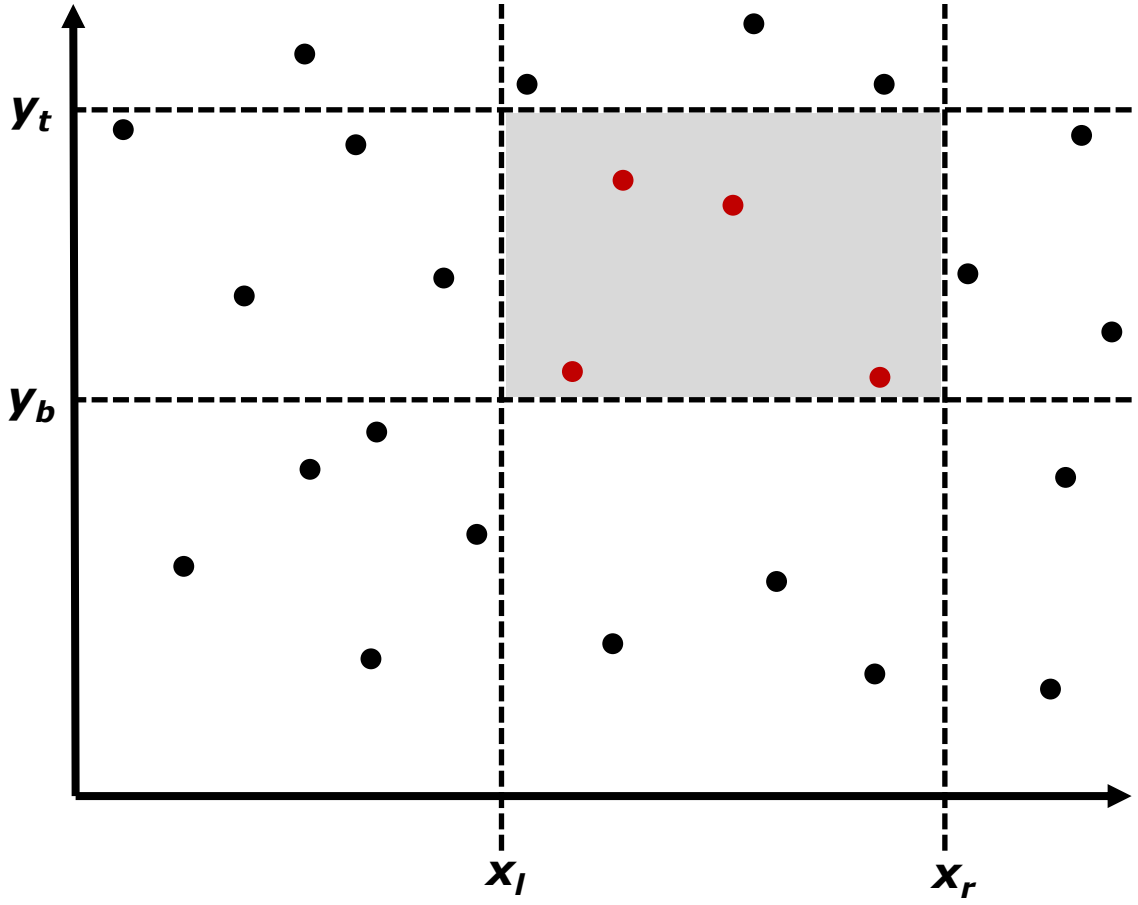
Gerth Stølting Brodal
Aarhus University

MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS

Kostas Tsakalidis

University of Primorska, October 10, 2011

Orthogonal Range Queries



Priority Search Tree [McCreight'75]

Space: O(*n*)

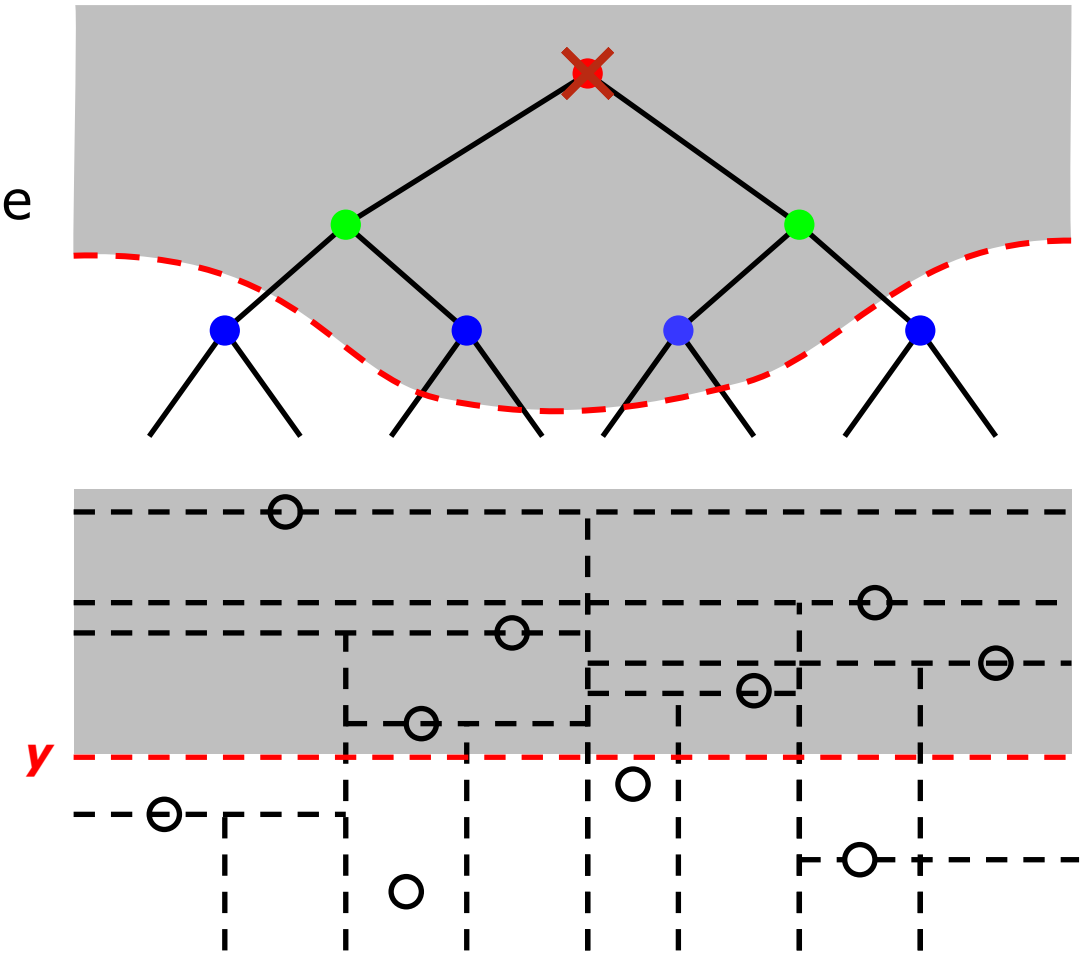
Update:

Recursively move up maximum *y*

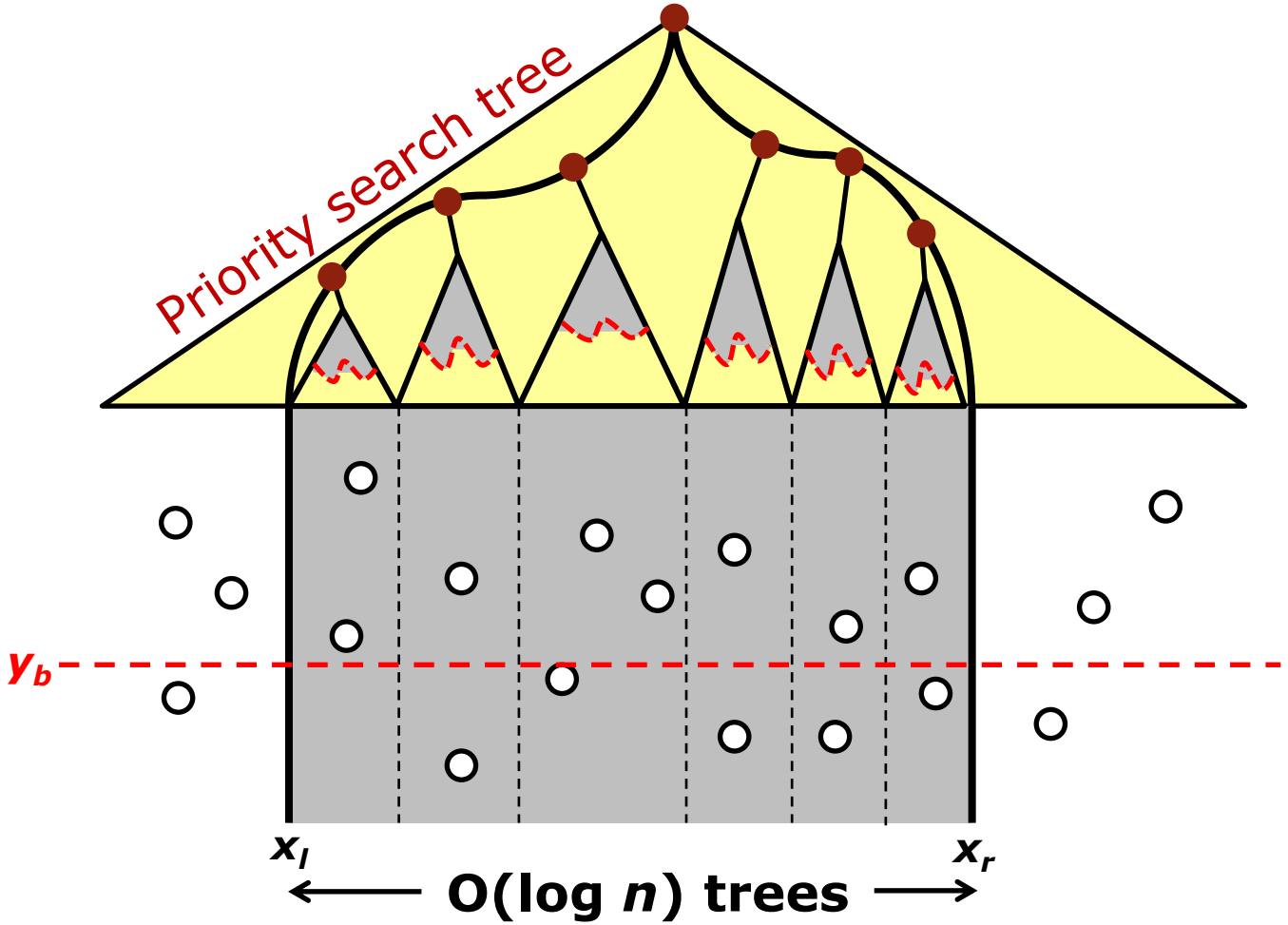
O(log *n*)

1-Sided reporting:

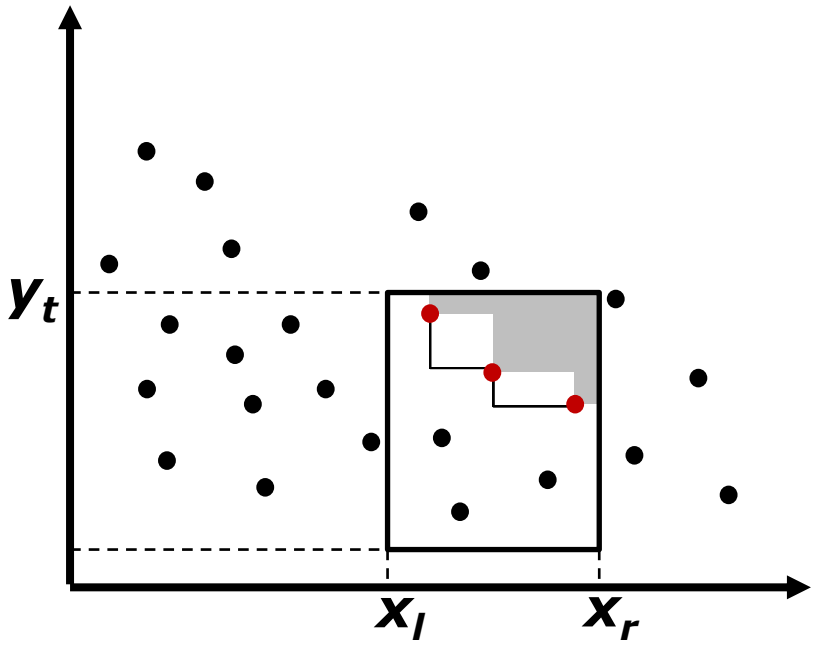
O(t)



3-Sided Reporting Queries O(log n + t)



Orthogonal Range MAXIMA Reporting alias "Generalized Planar SKYLINE Operator"



Dominance Maxima Queries

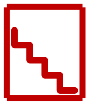
Report **all maximal points** among points with x in $[x_l,+\infty)$ and y in $[y_b,+\infty)$

Contour Maxima Queries

Report **all maximal points** among points with x in $(-\infty, x_r]$

3-Sided Maxima Queries

Report **all maximal points** among points with x in $[x_l, x_r]$ and y in $[y_b,+\infty)$



4-Sided Maxima Queries

Report **all maximal points** among points with x in $[x_l, x_r]$ and y in $[y_b,y_t]$

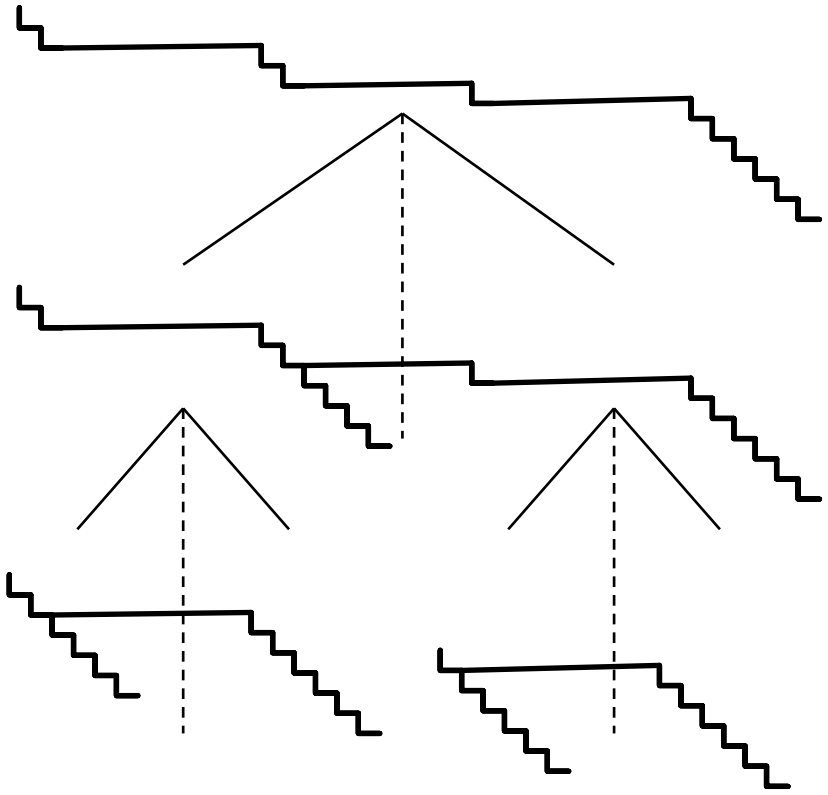
Static maximal points in O(n·log n) time [Kung, Luccio, Preparata, J.ACM'75]

Dynamic Range Maxima Reporting

	Space			Insert	Delete
Overmars, van Leeuwen '81	n	$\log n + t$	$\log^2 n + t$	$\log^2 n$	
Frederickson, Rodger '90	n	$\log n + t$	$\log^2 n + t$ $\log n(1+t)$	$\log n$	$\log^2 n$
Janardan '91	n	$\log n + t$		$\log n$	$\log^2 n$
Kapoor '00	n	$\log n + t$ amo.	-	$\log n$	
[ICALP '11] 	n	$\log n$ / loglog $n + t$		$\log n$ / loglog n	
	$n \cdot \log n$	$\log^2 n$ / loglog $n + t$		$\log^2 n$ / loglog n	

RAM

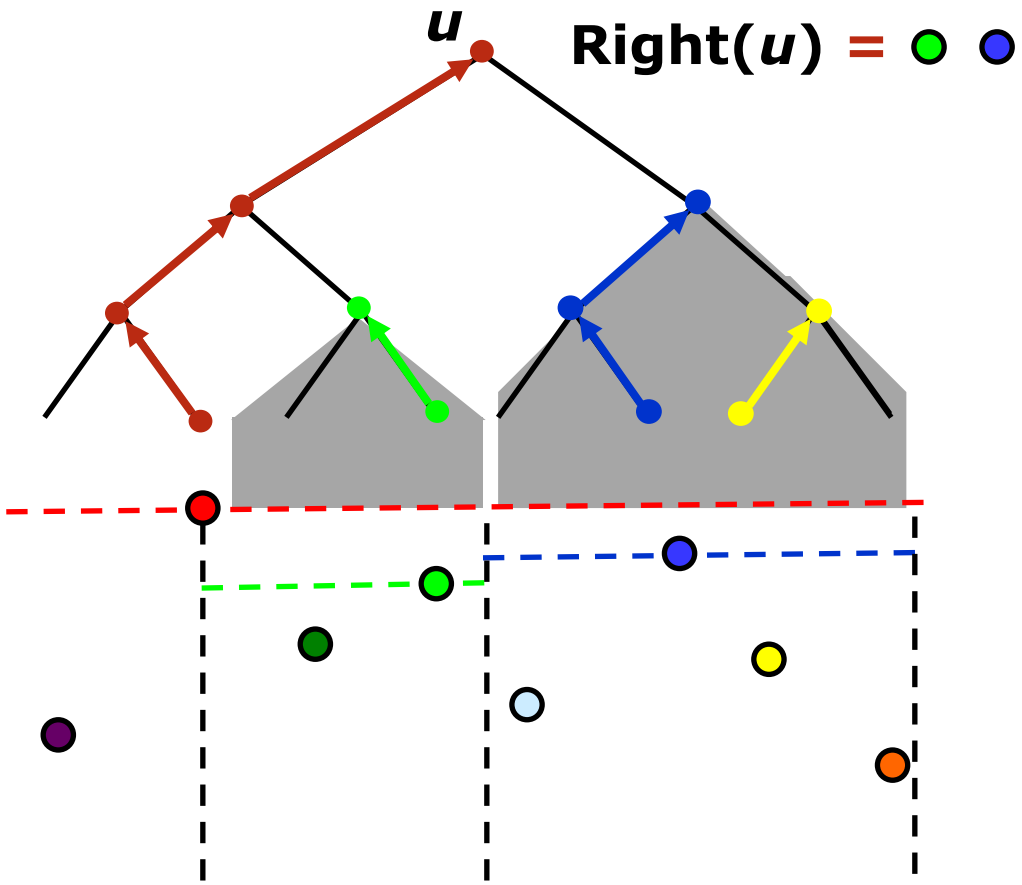
Overmars, van Leeuwen [JCSS '81]



O(log n + t)

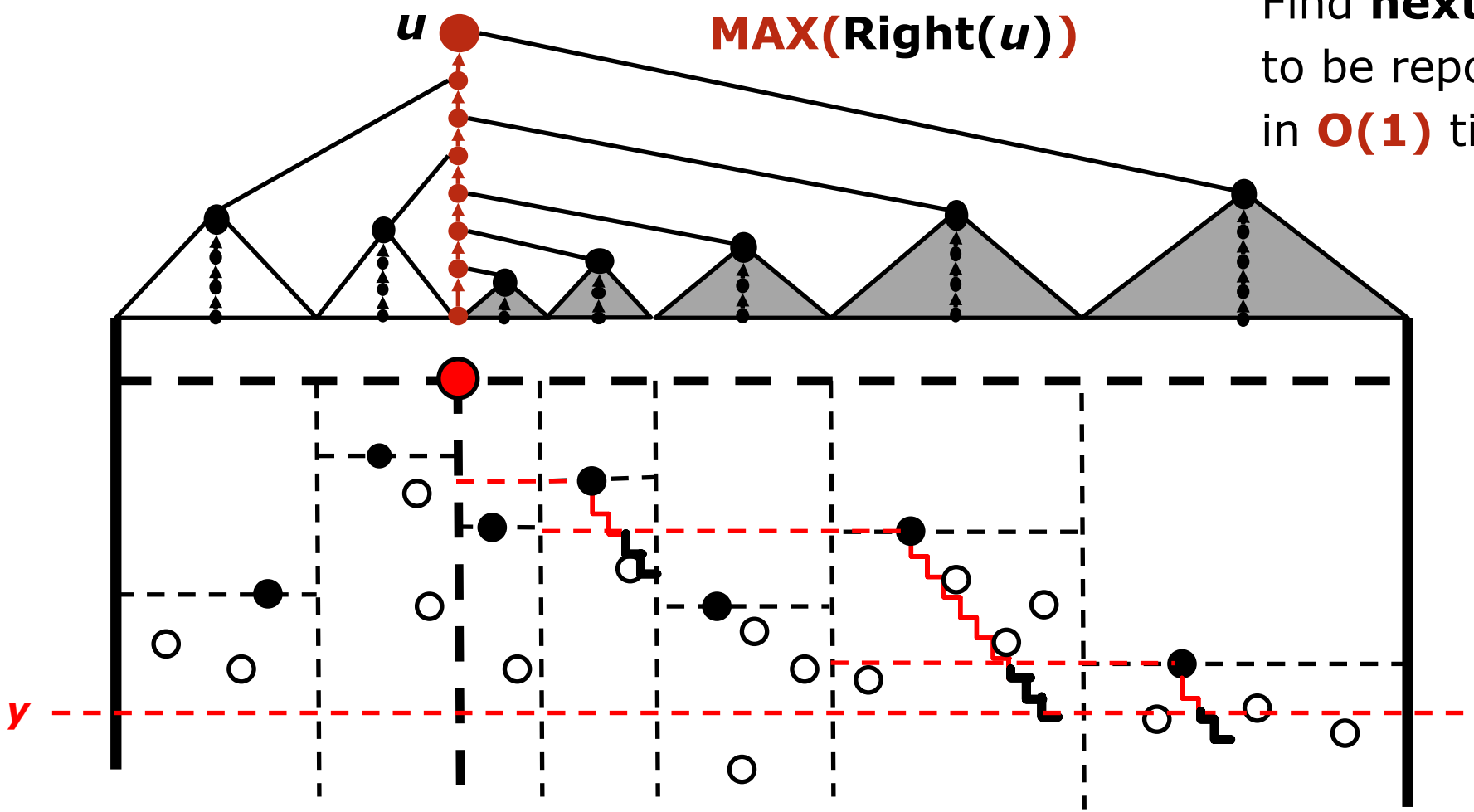
Updates: O(log^2 n)

Our Structure - Tournament Tree



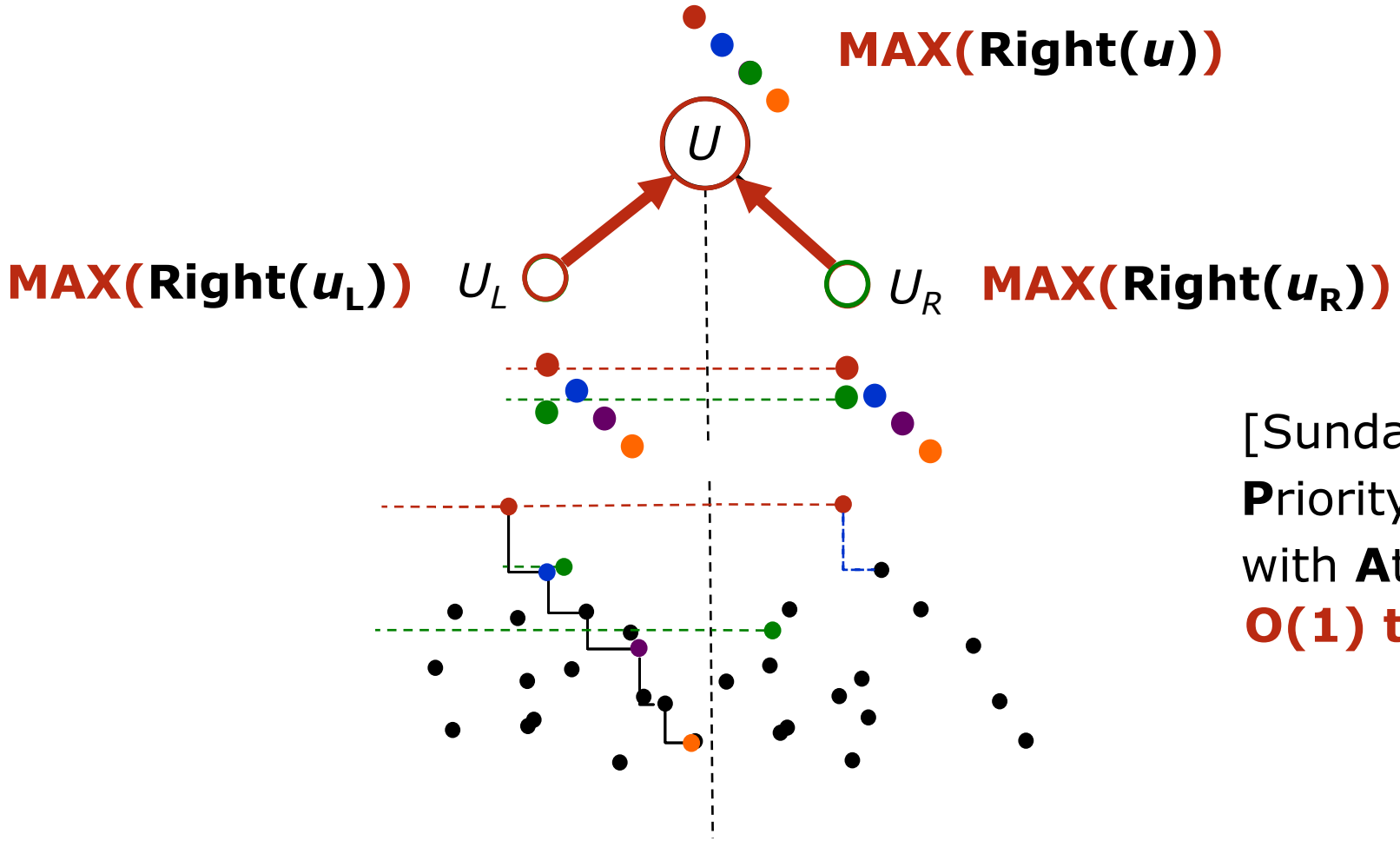
Copy Up
Maximum *y*

Tournament Tree



Find **next** point to be reported in **O(1)** time

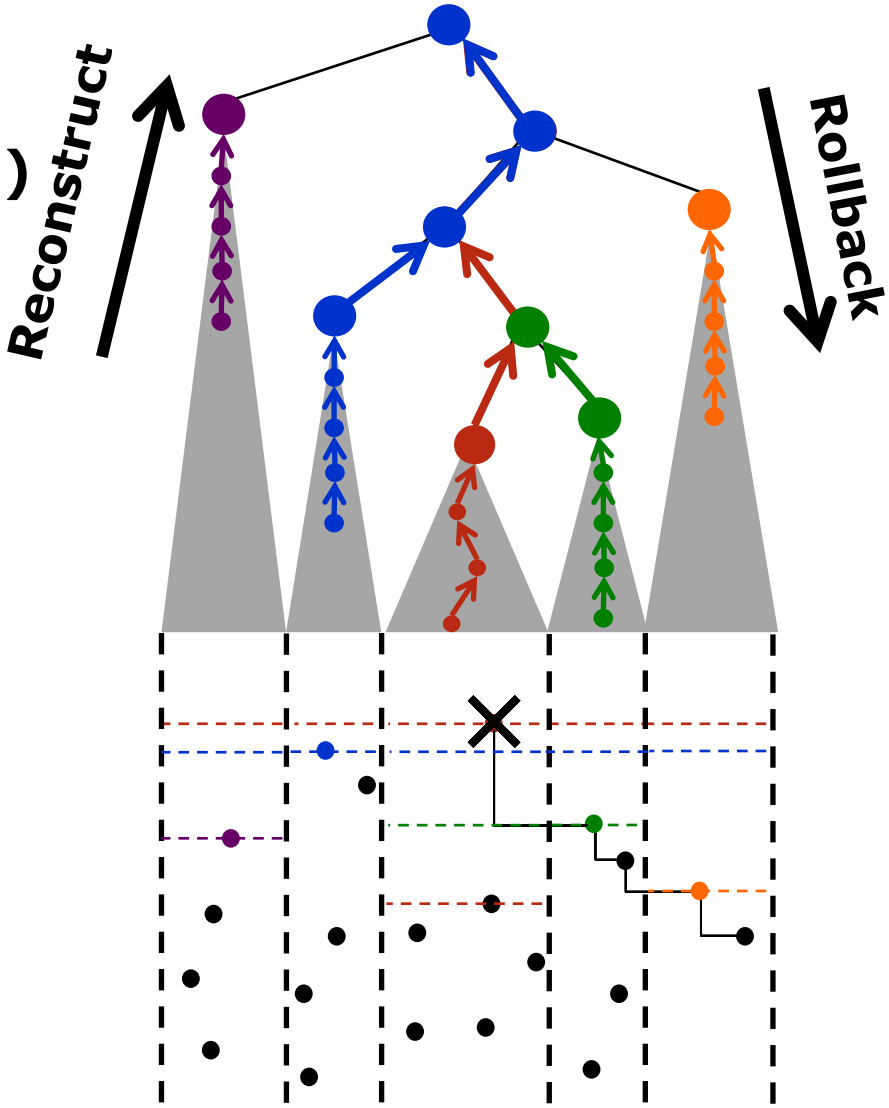
Computation of MAX(Right(*u*))



[Sundar '89]
Priority **Q**ueue with **A**ttrition
O(1) time

Update Operation

Space:**O(*n*)**
Update:**O(log *n*)**



Partially **P**ersistent **P**riority **Q**ueue with **A**ttrition

[Driscol et al. '89]

O(1) **amortized** time, space overhead per update step

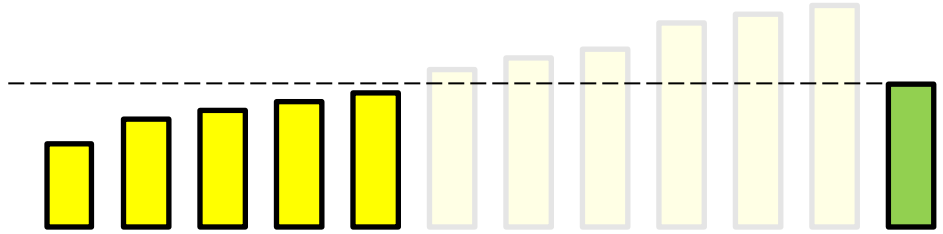
[Brodal '96]

worst case

Priority Queues with Attrition

[Sundar, IPL '89]

- Deletemin()
- InsertAndAttrite(element)

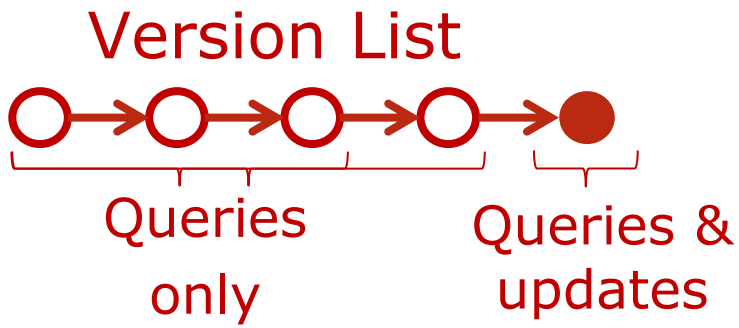


O(1) worst case time

Partial Persistent Data Structures

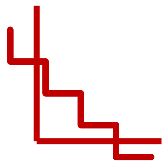
[Driscoll et al., JCSS ’89]

- “Persistent” = remember previous versions
- Any pointer-based structure with O(1) indegree

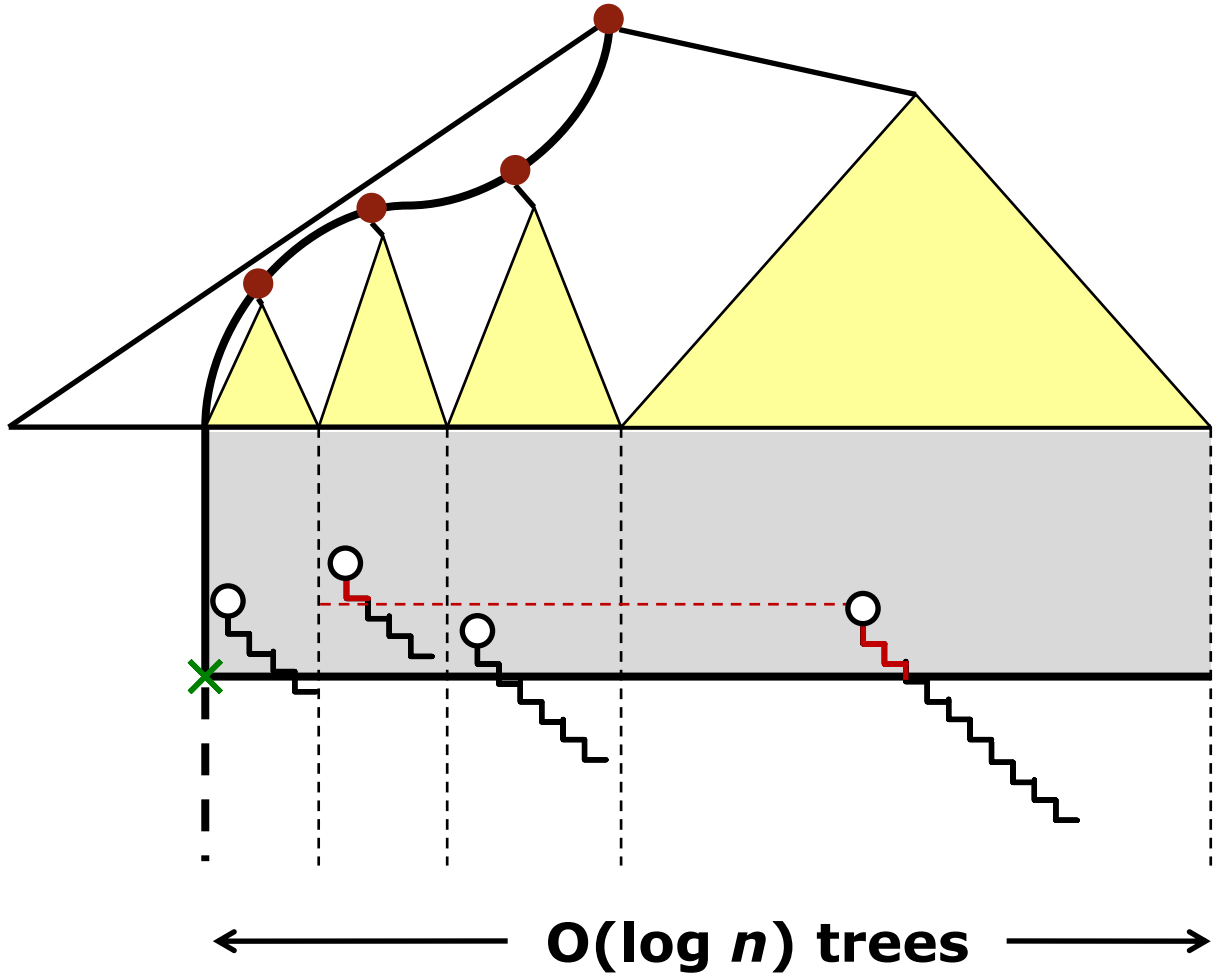


[Brodal, NJC ’96]

- O(1) worst case time overhead per access step
- O(1) **worst case** time, space overhead per update step
- “Rollback” = discard latest version in O(Update) time

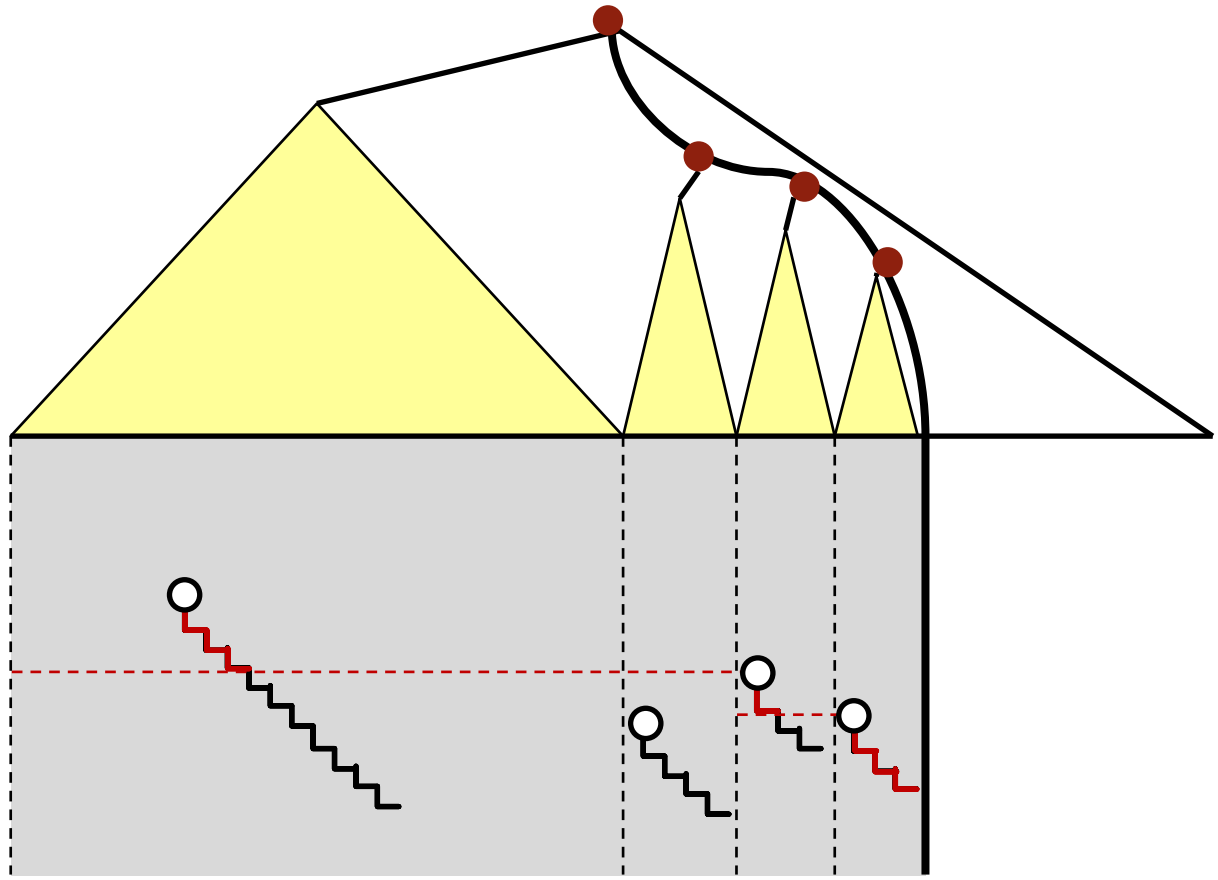


Dominance Range Maxima Queries



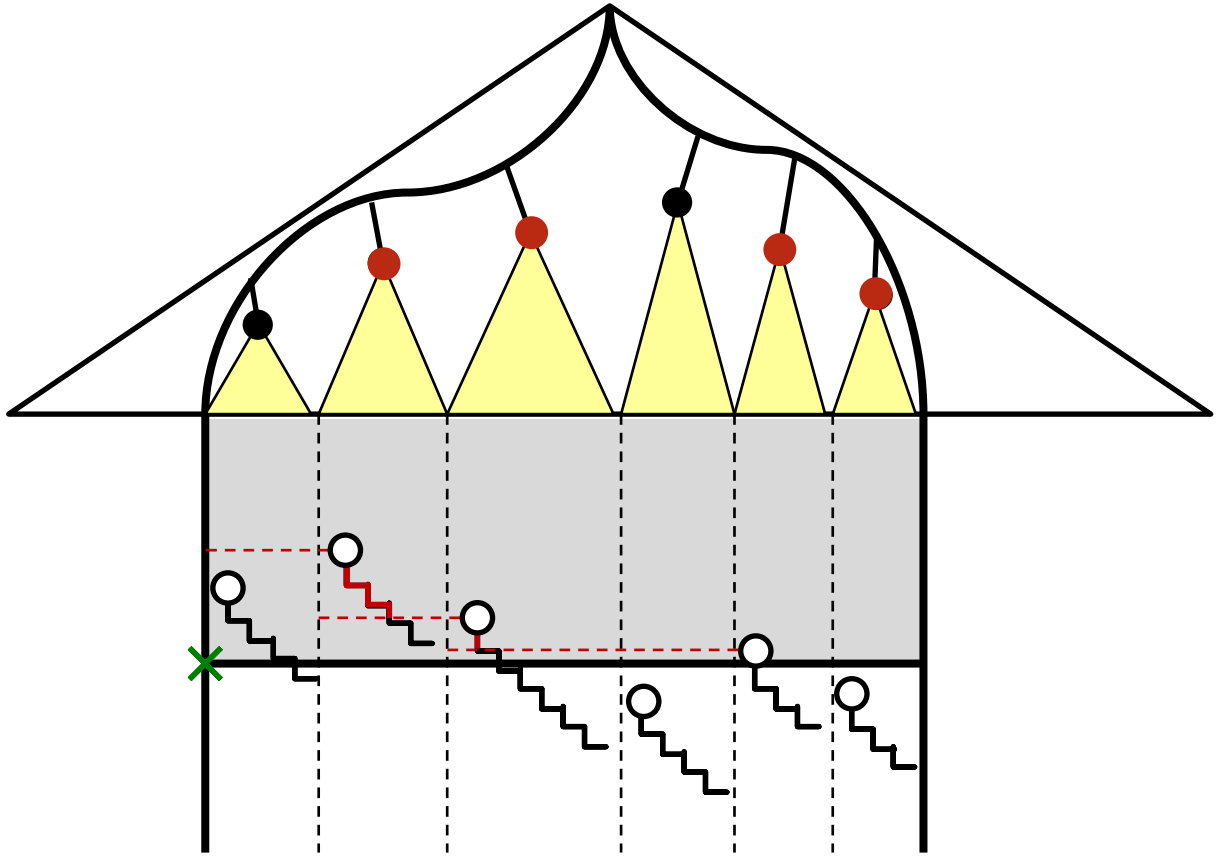
Query time O(log n + t)

Contour Range Maxima Queries



Query time O(log n + t)

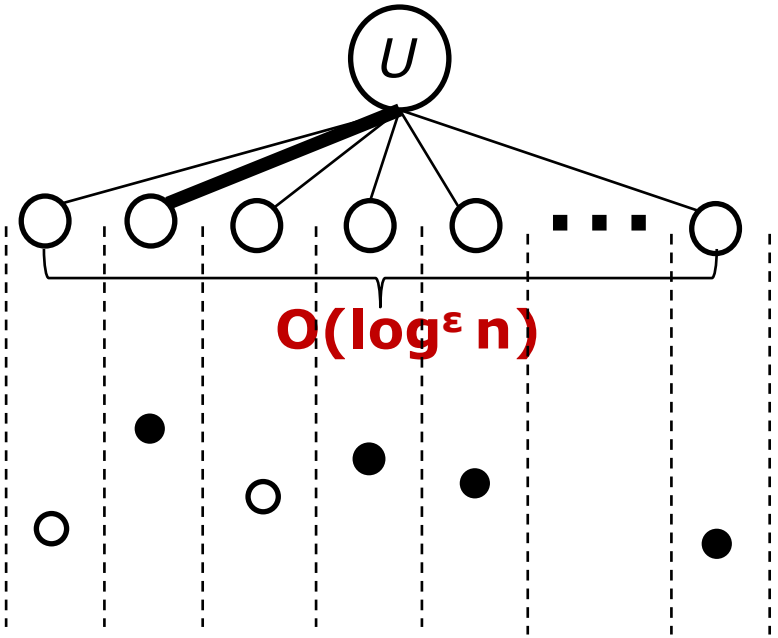
3-Sided Range Maxima Queries



← O(log *n*) trees →

Query time O(log *n* + *t*)

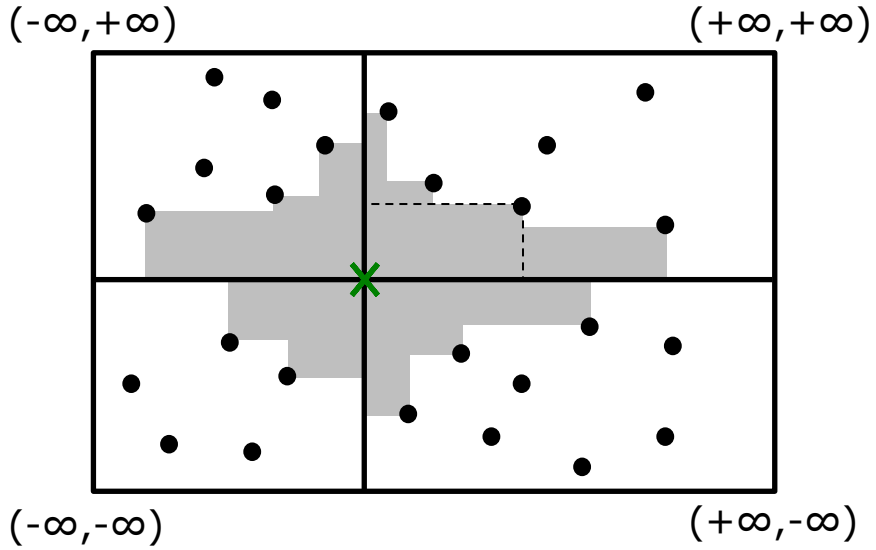
RAM – O(log *n*/loglog *n* + *t*)

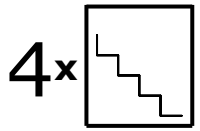


- Height O(log *n* / loglog *n*)
- MAX(Right(u)) maintained using **Q-heaps** [Fredman, Willard, JCSS ´94]

4-Sided Range MAXIMA Reporting and Rectangular Visibility Queries

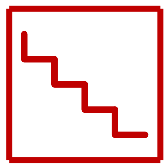
Proximity Queries/Similarity Search



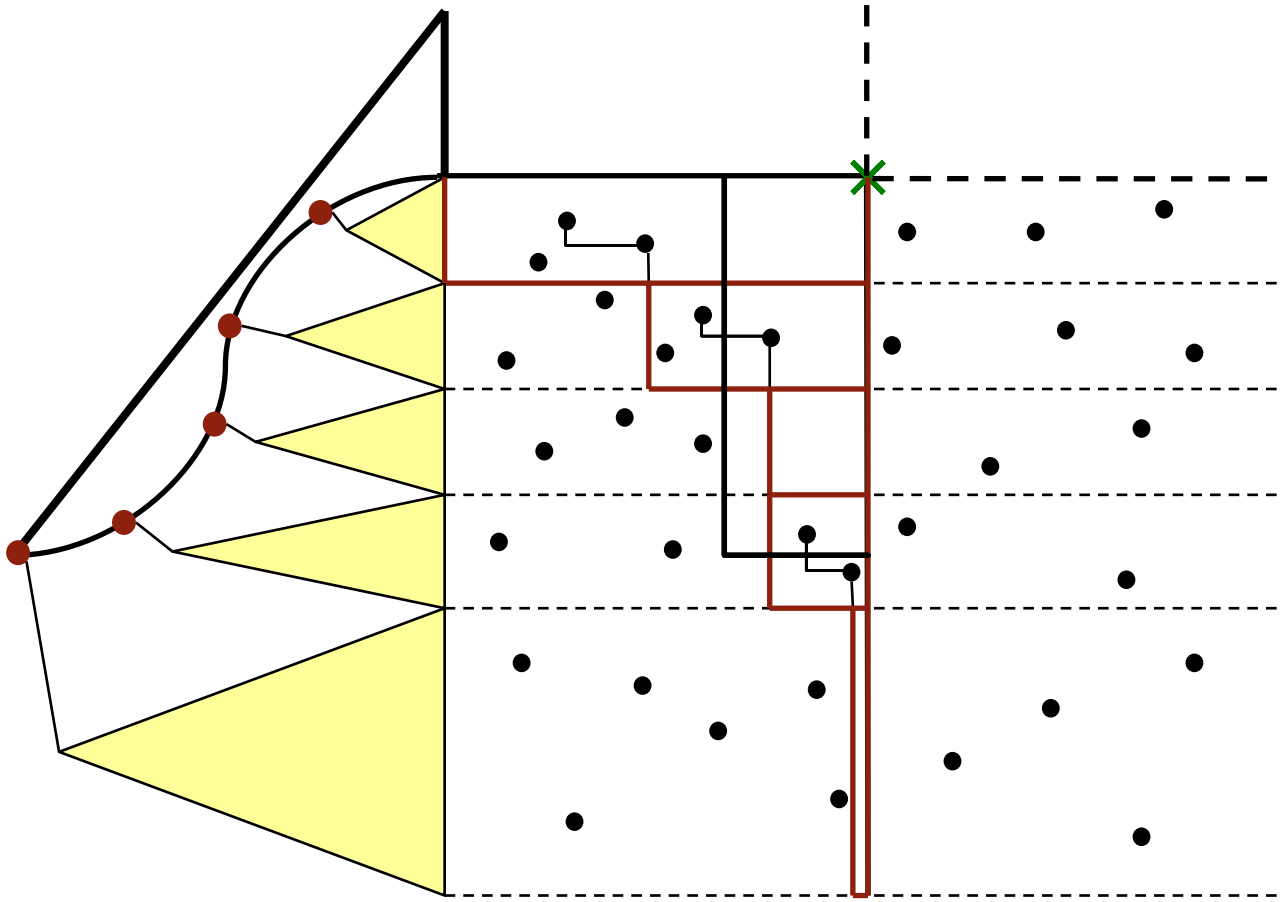


4-Sided Range Maxima Queries

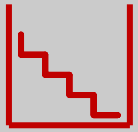
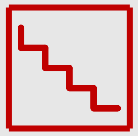
	Space	✿	Insert	Delete
Overmars, Wood '88	$n \cdot \log n$	$\log^2 n + t$ $\log^2 n + t \cdot \mathbf{\log n}$	$\log^2 n$	$\mathbf{\log^3 n}$ $\log^2 n$
[ICALP '11]	$\boldsymbol{n \cdot \log n}$	$\boldsymbol{\log^2 n + t}$	$\boldsymbol{\log^2 n}$	$\boldsymbol{\log^2 n}$



4-sided Range Maxima Queries



Query time O(log^2 n + t), space O(n·log n)

	Space	Query	Insert/Delete	
	$O(n)$	$O(\log n/\log\log n + t)$	$O(\log n/\log\log n)$	RAM
	$O(n \cdot \log n)$	$O(\log^2 n + t)$	$O(\log^2 n/\log\log n)$	

Thank You

Gerth Stølting Brodal
Aarhus University

MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS