Fun with Recursion and Tree Drawings

Timothy Chan
U. of Illinois

I like recurrences!

From “Klee’s measure problem made easy” [C.’12]:

$$T(n) \;=\; 2\,T\left(\frac{n}{2^{2/3}}\right) + O(n)$$

$$\Rightarrow\; T(n) = \boxed{O(n^{3/2})}$$

From "Transdichotomous results in computational geometry, II. . ." [C.–Pătraşcu'10]:

$$Q(n, U_L, U_R) \;\le\; Q(b, H, H) + \max\left\{Q\left(\frac{n}{b}, U_L, U_R\right), Q\left(n, \frac{U_L}{H}, U_R\right), Q\left(n, U_L, \frac{U_R}{H}\right)\right\} + \widetilde{O}((\log U_L + \log U_R)/w)$$

with $Q(b, H, H) = O(1)$ if $b \log H \le w$

$$\Rightarrow\; Q(n, U, U) = 2^{O(\sqrt{\log\log n})}$$

From "Clustered integer 3SUM via additive combinatorics" [C.–Lewenstein'15]:

$$T(n) \;\le\; O\left(\alpha\left(\frac{n}{\ell}\right)^2\right)T(\ell) + \widetilde{O}\left(\frac{n\ell}{\alpha^6} + \left(\frac{n}{\ell}\right)^2\right)$$

for any $\alpha < 1$ and ℓ

$$\Rightarrow\; T(n) = \widetilde{O}(n^{(9+\sqrt{177})/2}) = O(n^{1.859})$$

From “Conflict-free coloring of points w.r.t. rectangles…” [C.’12]:

$$G(n,v,h) \;\geq\; \min_{r \geq r_0} \left\{ \frac{n}{r_0},\; \widetilde{\Omega}(r)\, G\left(G\left(\frac{n}{r}, \frac{v}{r}, r \right), r, \frac{h}{r} \right) \right\}$$

for any r_0, with $G(n,v,h) \geq n/v$ and $G(n,v,h) = G(n,h,v)$

$$\Rightarrow\; G(n,n,n) = \boxed{\Omega(n^{0.632})}$$

From “Improved bounds for drawing trees on fixed points with L-shaped edges” [Biedl–C.–Derka–Jain–Lubiw (GD’17)]:

$$\begin{aligned} f(n) &\le 2f(n_1) + g(n_2) \\ f(n) &\le 2g(n_1) + 2f(n_{21}) + g(n_{22}) \\ f(n) &\le \max\{2g(n_1) + f(n_{22}) + n,\ g(n_1) + g(n_{21}) + f(n_{22})\} \\ g(n) &\le f(n_1) + g(n_2) \end{aligned}$$

for some $n_1 \le n_2, n_{21} \le n_{22}, n_1 + n_2 = n, n_{21} + n_{22} = n_2$

$$\Rightarrow\ f(n), g(n) = \boxed{O(n^{1.22})}$$

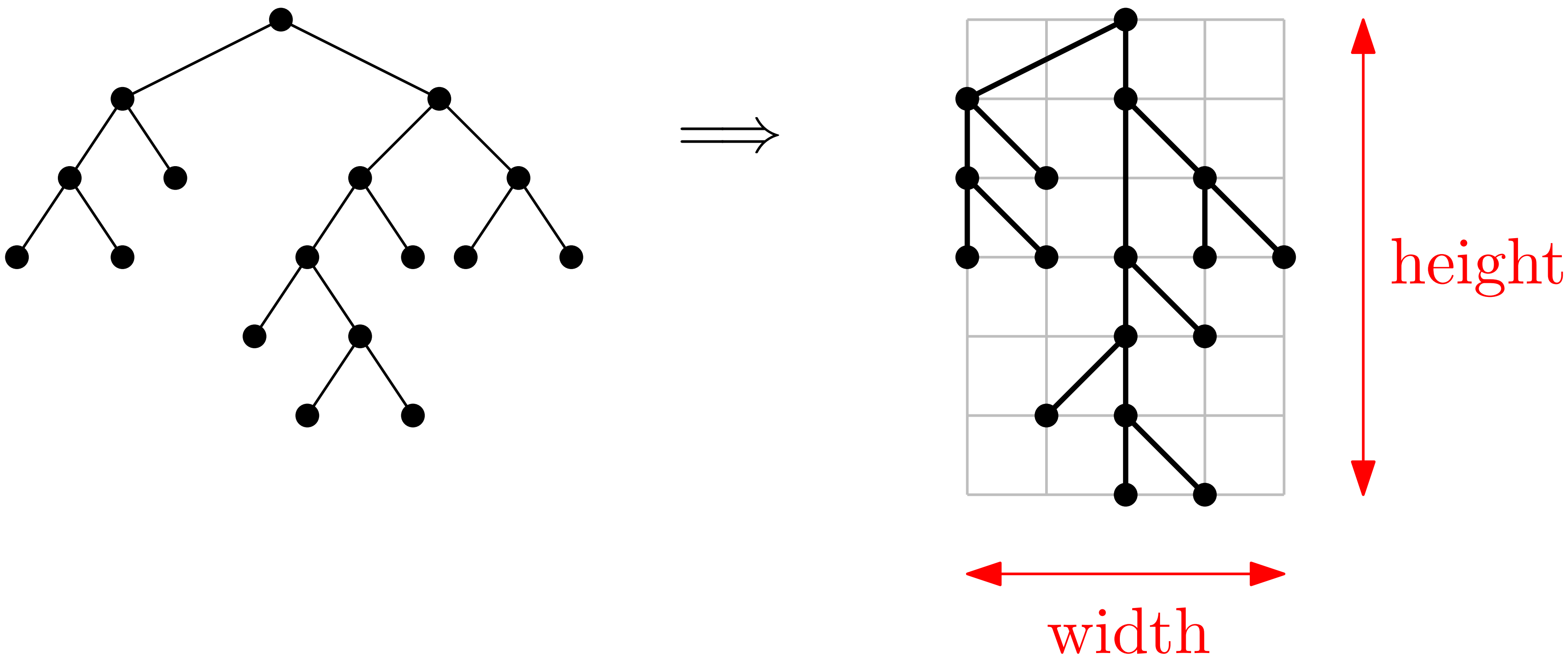
Tree Drawings

The Problem(s)

- obtain worst-case area bounds for different types of planar, straight-line drawing of trees on a grid

The Problem(s)

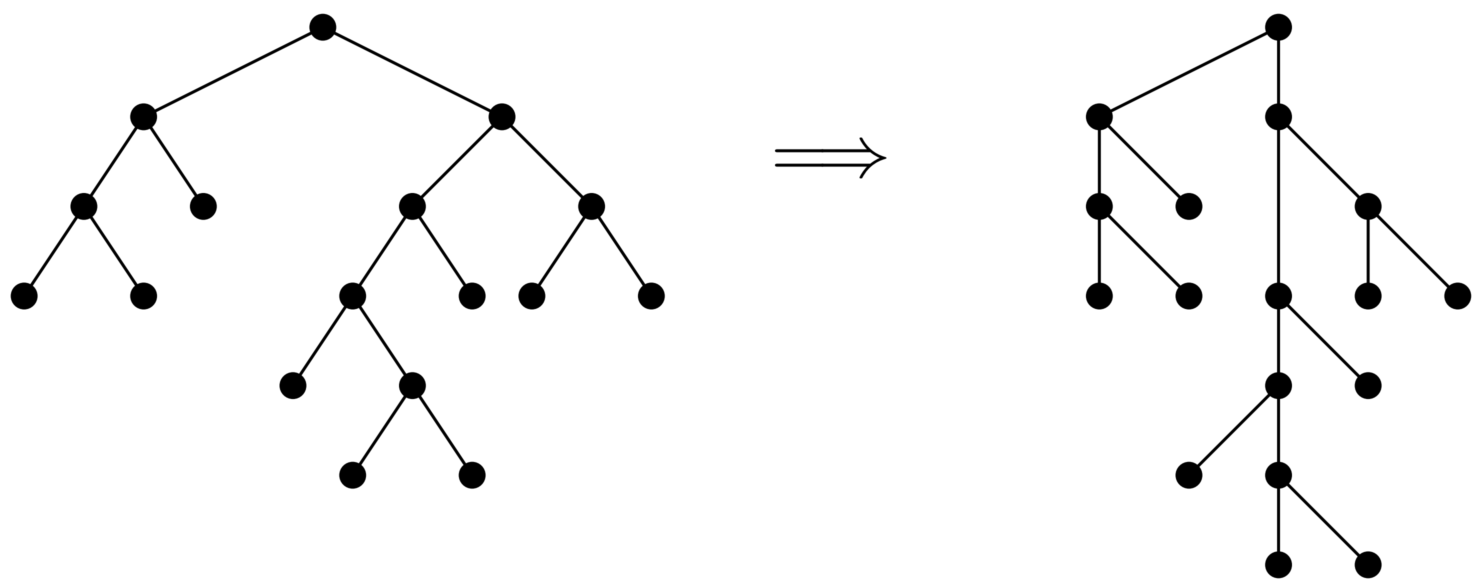
- obtain worst-case area bounds for different types of planar, straight-line drawing of trees on a grid



(area = width × height)

The Problem(s)

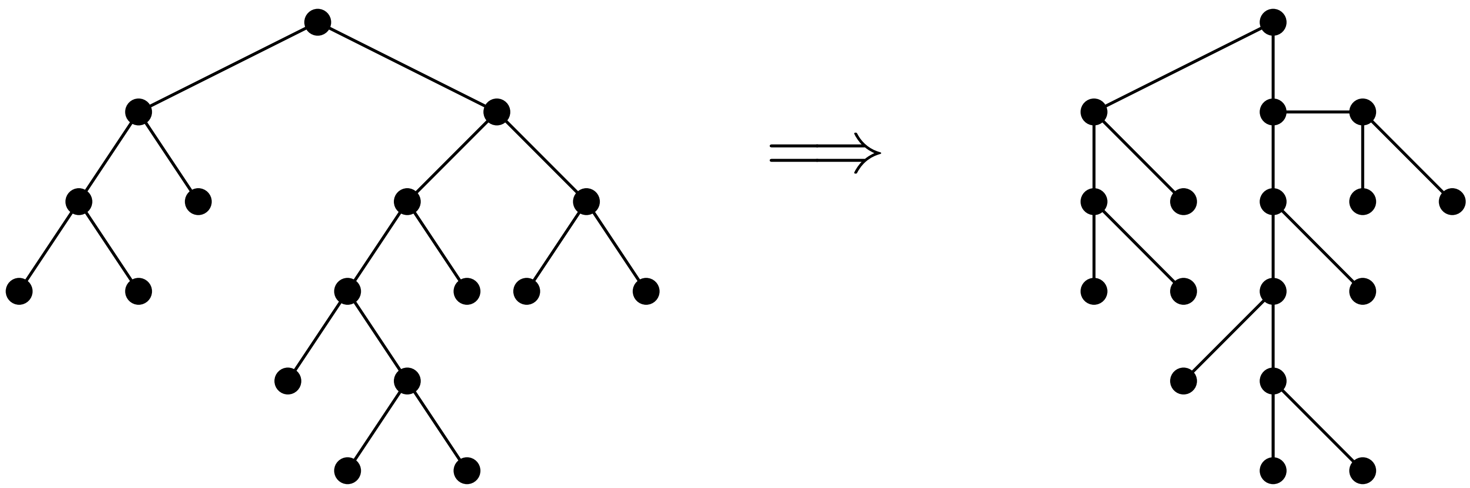
- obtain worst-case area bounds for different types of planar, straight-line drawing of trees on a grid



strictly upward

The Problem(s)

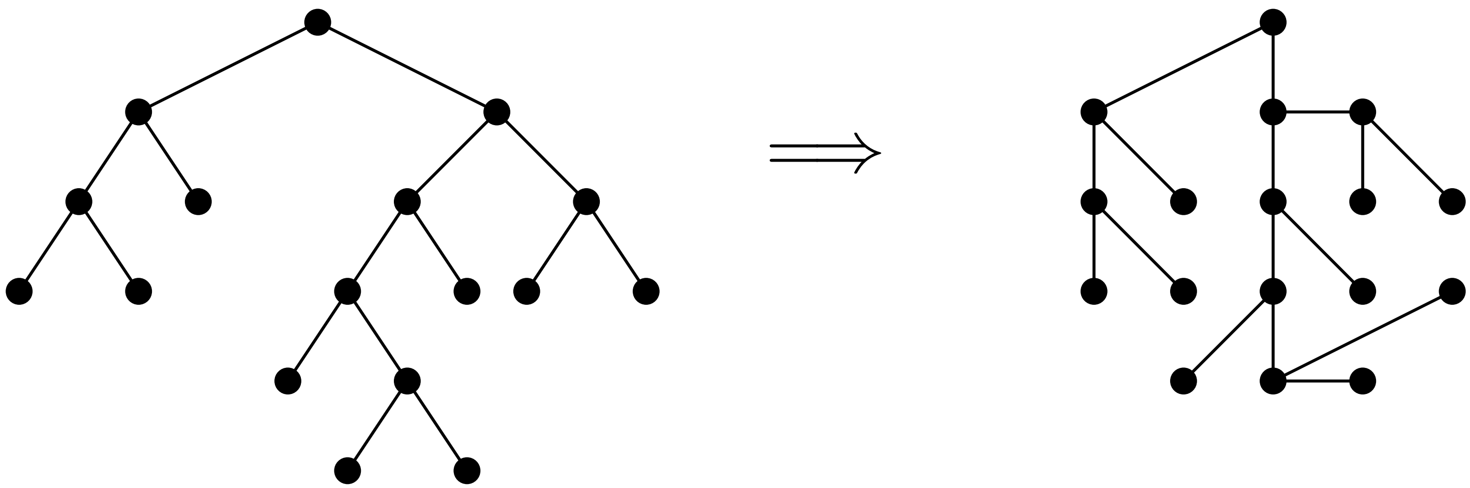
- obtain worst-case area bounds for different types of planar, straight-line drawing of trees on a grid



upward (“upw.”)

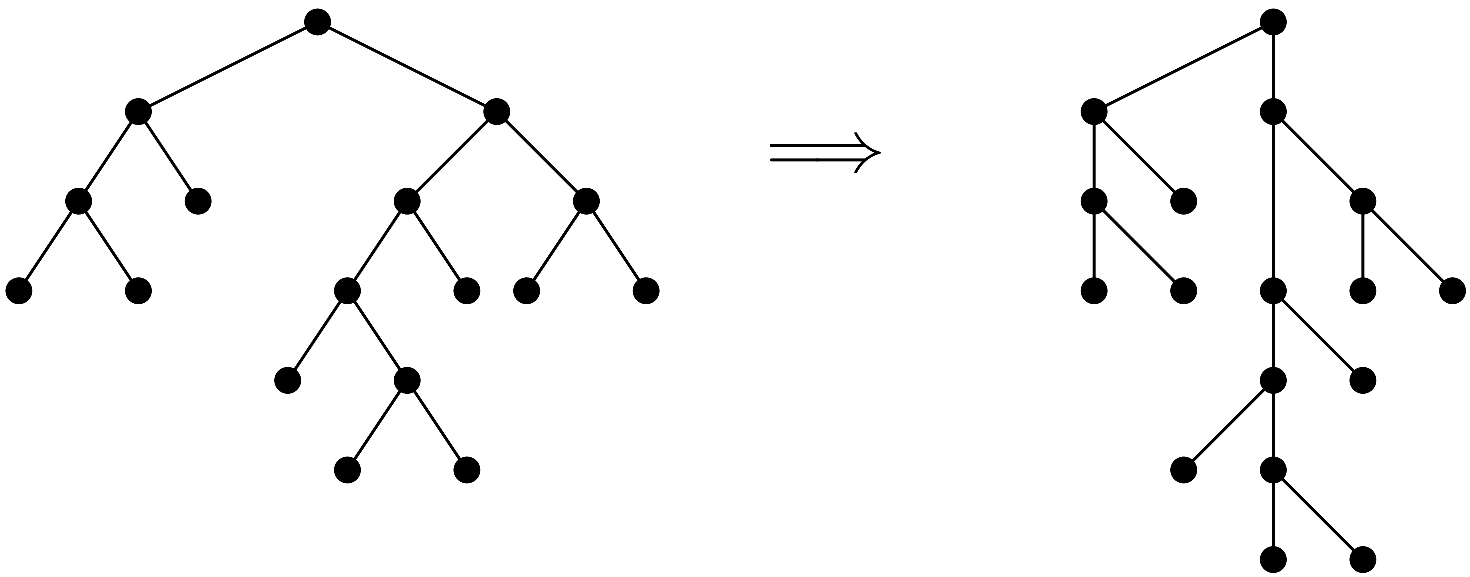
The Problem(s)

- obtain worst-case area bounds for different types of planar, straight-line drawing of trees on a grid



The Problem(s)

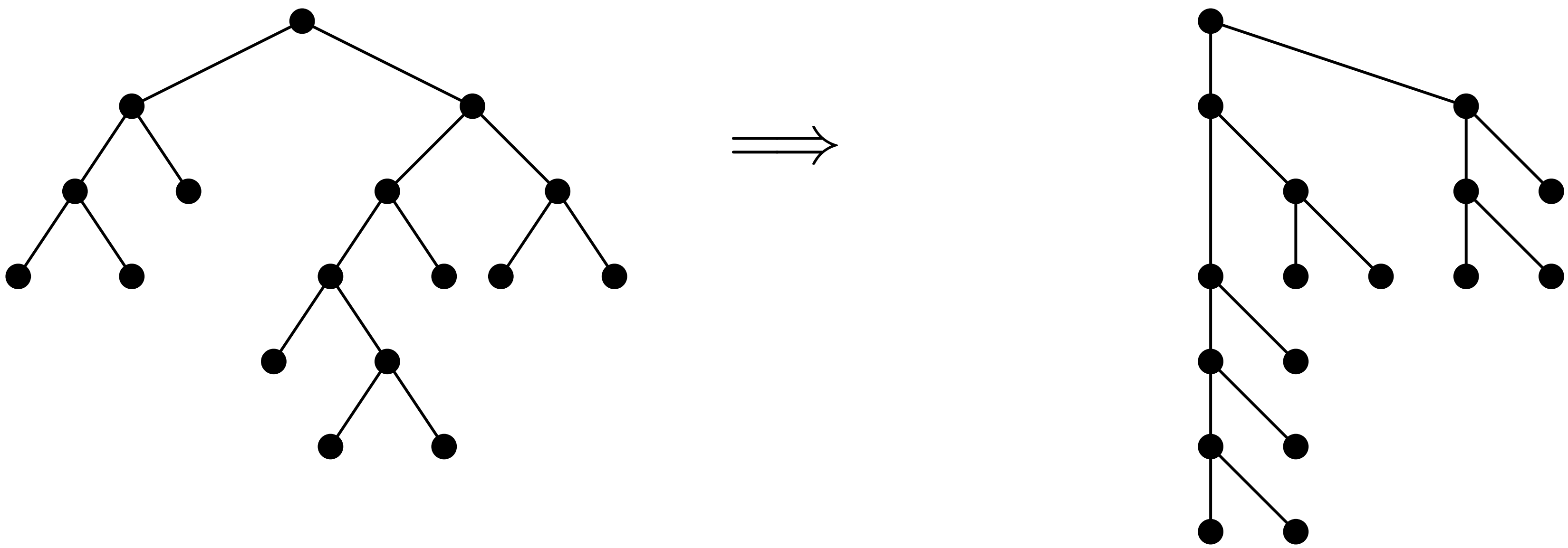
- obtain worst-case area bounds for different types of planar, straight-line drawing of trees on a grid



order-preserving ("*ordered*")

The Problem(s)

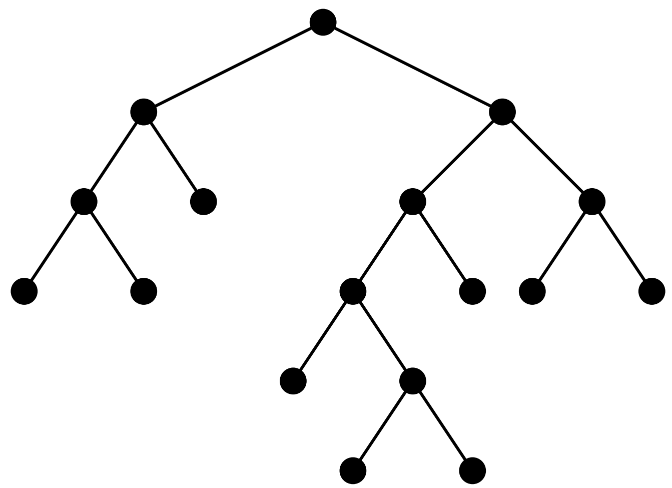
- obtain worst-case area bounds for different types of planar, straight-line drawing of trees on a grid



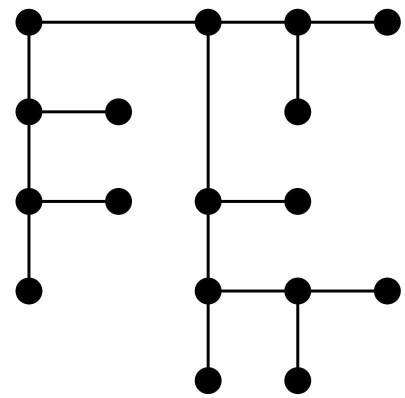
non-order-preserving ("*unordered*")

The Problem(s)

- obtain worst-case area bounds for different types of planar, straight-line drawing of trees on a grid



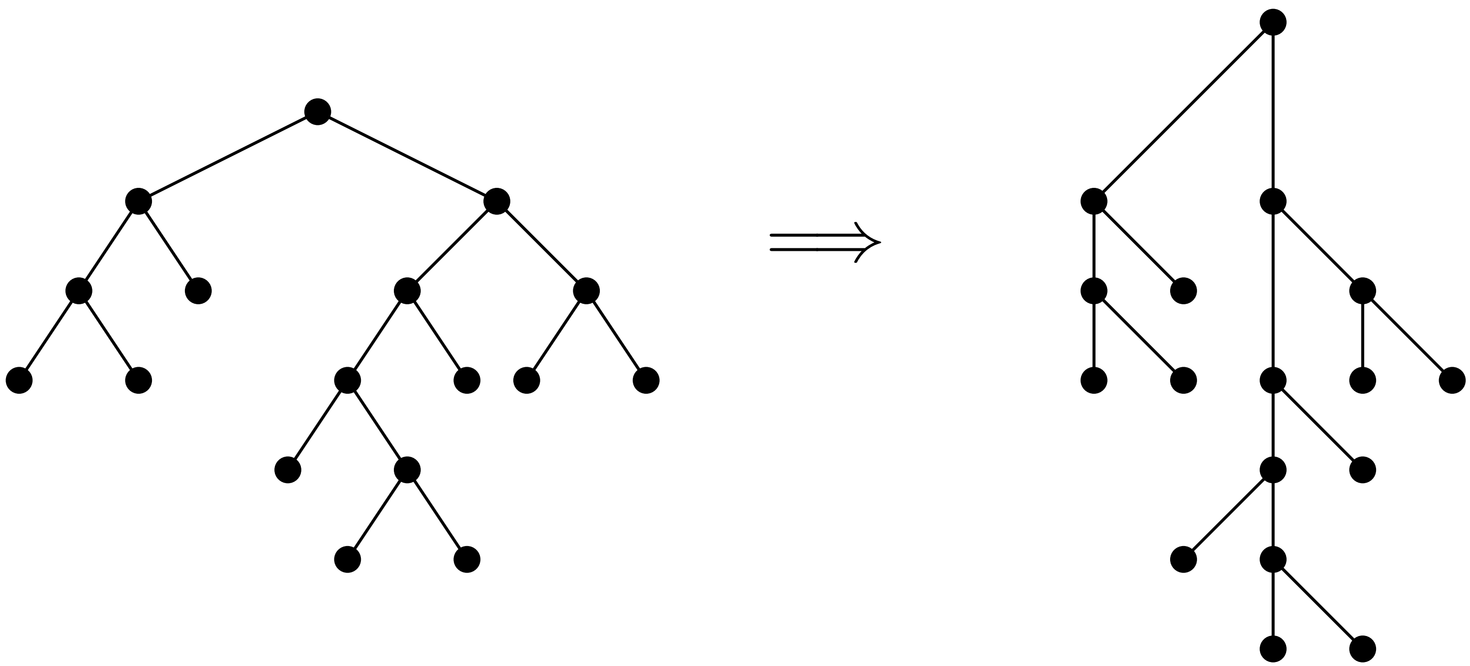
$\Longrightarrow$



orthogonal

The Problem(s)

- obtain worst-case area bounds for different types of planar, straight-line drawing of trees on a grid



octilinear (45° angles)

Survey of Known Area Bounds

[see Di Battista–Frati'14]

Binary trees

	unordered	*ordered*
non-upw.	$\Theta(n)$ [Garg–Goodrich–Tamassia'93]	$O(n \log\log n)$ [Garg–Rusu'03]
upw.	$O(n \log\log n)$ [Shin–Kim–Chwa'96]	$O(n^{3/2})$ [C.'99]
strict upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$O(n^{3/2})$ [C.'99]

Binary trees

	unordered	*ordered*
non-upw.	$\Theta(n)$ [Garg–Goodrich–Tamassia'93]	$O(n \log \log n)$ [Garg–Rusu'03]
upw.	$O(n \log \log n)$ [Shin–Kim–Chwa'96]	$O(n^{1.48})$ [C.'99]
strict upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$O(n^{1.48})$ [C.'99]

Binary trees

	unordered	*ordered*
non-upw.	$\Theta(n)$ [Garg–Goodrich–Tamassia'93]	$O(n \log \log n)$ [Garg–Rusu'03]
upw.	$O(n \log \log n)$ [Shin–Kim–Chwa'96]	$O(nc^{\sqrt{\log n}})$ [C.'99]
strict upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$O(nc^{\sqrt{\log n}})$ [C.'99]

Binary trees

	unordered	*ordered*
non-upw.	$\Theta(n)$ [Garg–Goodrich–Tamassia'93]	$O(n \log \log n)$ [Garg–Rusu'03]
upw.	$O(n \log \log n)$ [Shin–Kim–Chwa'96]	$O(n \log n)$ [Garg–Rusu'03]
strict upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$\Theta(n \log n)$ [Garg–Rusu'03]

Binary trees

	unordered	*ordered*
non-upw.	$\Theta(n)$ [Garg–Goodrich–Tamassia'93]	$O(n \log\log n)$ [Garg–Rusu'03] open
upw.	$O(n \log\log n)$ [Shin–Kim–Chwa'96] open	$O(n \log n)$ [Garg–Rusu'03] open
strict upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$\Theta(n \log n)$ [Garg–Rusu'03]

Binary trees

	unordered	*ordered*
non-upw.	$\Theta(n)$ [Garg–Goodrich–Tamassia'93]	$O(n \log \log n)$ [Garg–Rusu'03] new: $O(nc^{\log^* n})$
upw.	$O(n \log \log n)$ [Shin–Kim–Chwa'96] open	$O(n \log n)$ [Garg–Rusu'03] open
strict upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$\Theta(n \log n)$ [Garg–Rusu'03]

Binary trees, orthogonal

	unordered	*ordered*
non-upw.	$O(n \log \log n)$ [C.–Goodrich–Kosaraju–Tamassia (GD'96), Shin–Kim–Chwa'96]	$O(n^{3/2})$ [Frati'07]
upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$\Theta(n^2)$

Binary trees, orthogonal

	unordered	*ordered*
non-upw.	$O(n \log \log n)$ [C.–Goodrich–Kosaraju–Tamassia (GD'96), Shin–Kim–Chwa'96] open	$O(n^{3/2})$ [Frati'07] open
upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$\Theta(n^2)$

Binary trees, orthogonal

	unordered	*ordered*
non-upw.	$O(n \log \log n)$ [C.–Goodrich–Kosaraju–Tamassia (GD'96), Shin–Kim–Chwa'96] new: $O(nc^{\log^* n})$	$O(n^{3/2})$ [Frati'07] open
upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$\Theta(n^2)$

Binary trees, orthogonal

	unordered	*ordered*
non-upw.	$O(n \log \log n)$ [C.–Goodrich–Kosaraju–Tamassia (GD'96), Shin–Kim–Chwa'96] new: $O(nc^{\log^* n})$	$O(n^{3/2})$ [Frati'07] new: $O(nc^{\sqrt{\log n}})$
upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$\Theta(n^2)$

General trees

	unordered	*ordered*
non-upw.	$O(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$O(n \log n)$ [Garg–Rusu'03]
upw.	$O(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$O(nc^{\sqrt{\log n}})$ [C.'99]
strict upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$O(nc^{\sqrt{\log n}})$ [C.'99]

General trees

	unordered	*ordered*
non-upw.	$O(n \log n)$ [Crescenzi–Di Battista–Piperno'93] open	$O(n \log n)$ [Garg–Rusu'03] open
upw.	$O(n \log n)$ [Crescenzi–Di Battista–Piperno'93] open	$O(nc^{\sqrt{\log n}})$ [C.'99] open
strict upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$O(nc^{\sqrt{\log n}})$ [C.'99] open

General trees

	unordered	*ordered*
non-upw.	$O(n \log n)$ [Crescenzi–Di Battista–Piperno'93] new: $O(nc^{\sqrt{\log\log n \log\log\log n}})$	$O(n \log n)$ [Garg–Rusu'03] open
upw.	$O(n \log n)$ [Crescenzi–Di Battista–Piperno'93] open	$O(nc^{\sqrt{\log n}})$ [C.'99] open
strict upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$O(nc^{\sqrt{\log n}})$ [C.'99] open

General trees

	unordered	*ordered*
non-upw.	$O(n \log n)$ [Crescenzi–Di Battista–Piperno'93] new: $O(nc^{\sqrt{\log\log n \log\log\log n}})$	$O(n \log n)$ [Garg–Rusu'03] open
upw.	$O(n \log n)$ [Crescenzi–Di Battista–Piperno'93] new: $O(n\sqrt{\log n}\,\mathrm{polyloglog}\, n)$	$O(nc^{\sqrt{\log n}})$ [C.'99] open
strict upw.	$\Theta(n \log n)$ [Crescenzi–Di Battista–Piperno'93]	$O(nc^{\sqrt{\log n}})$ [C.'99] open

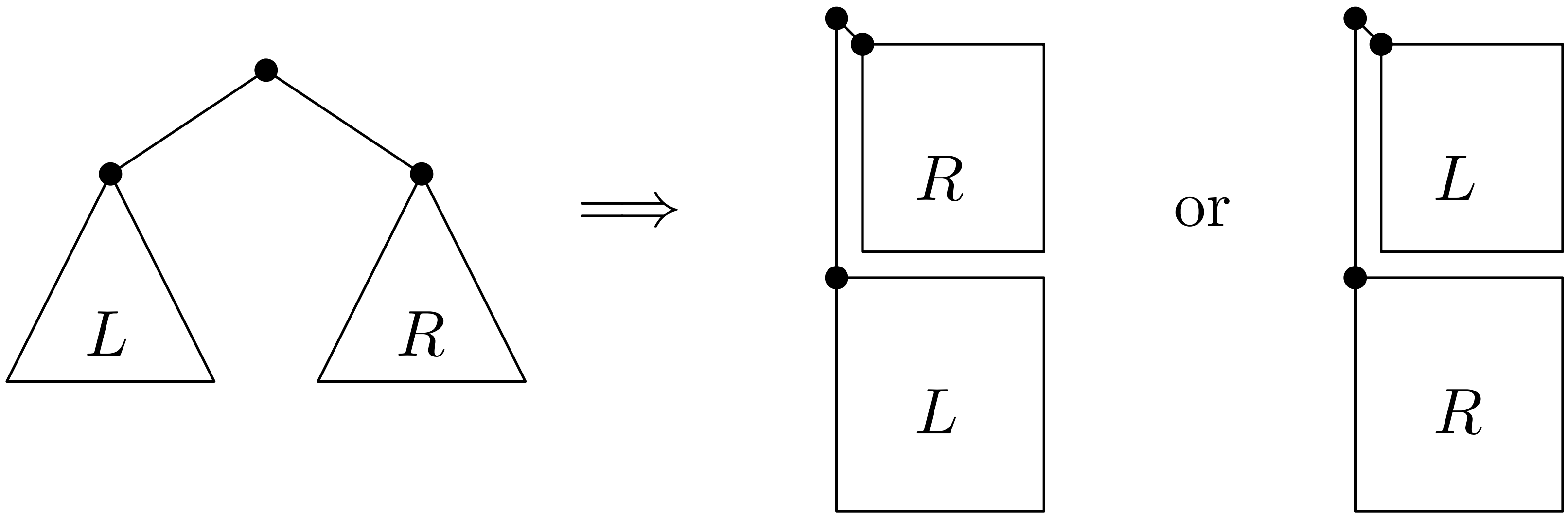
Technique 1: The “Heavy Path”

Ex: binary, strict upw.

[Crescenzi–Di Battista–Piperno'93]

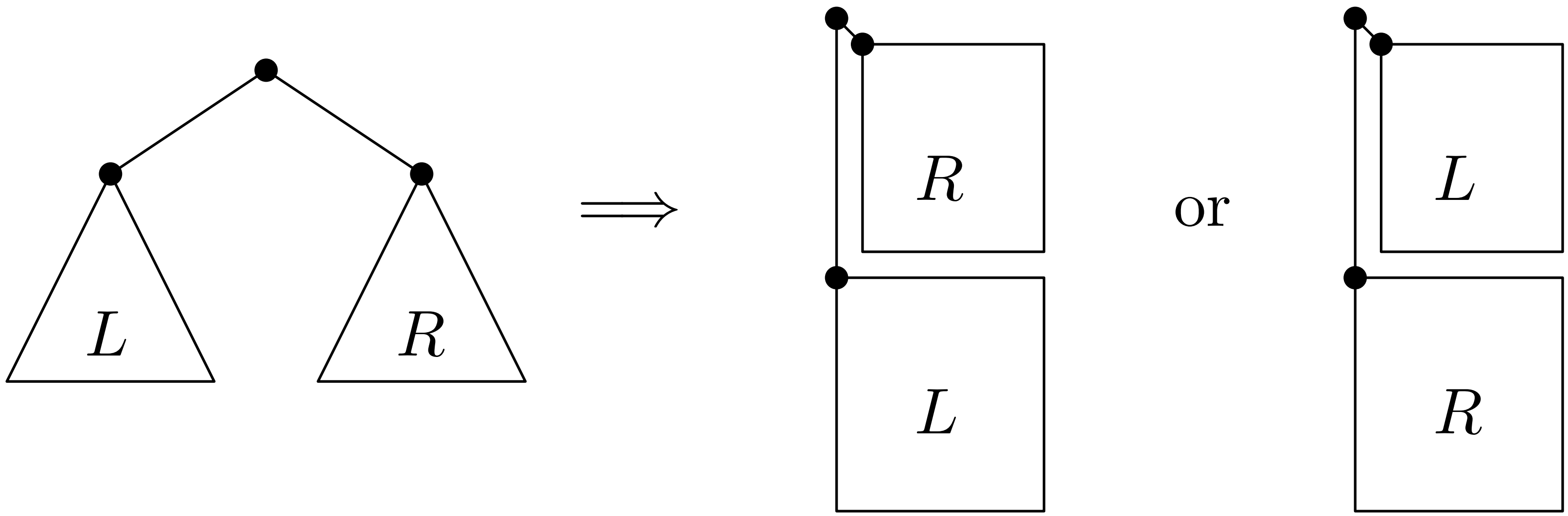
Ex: binary, strict upw.

[Crescenzi–Di Battista–Piperno'93]



Ex: binary, strict upw.

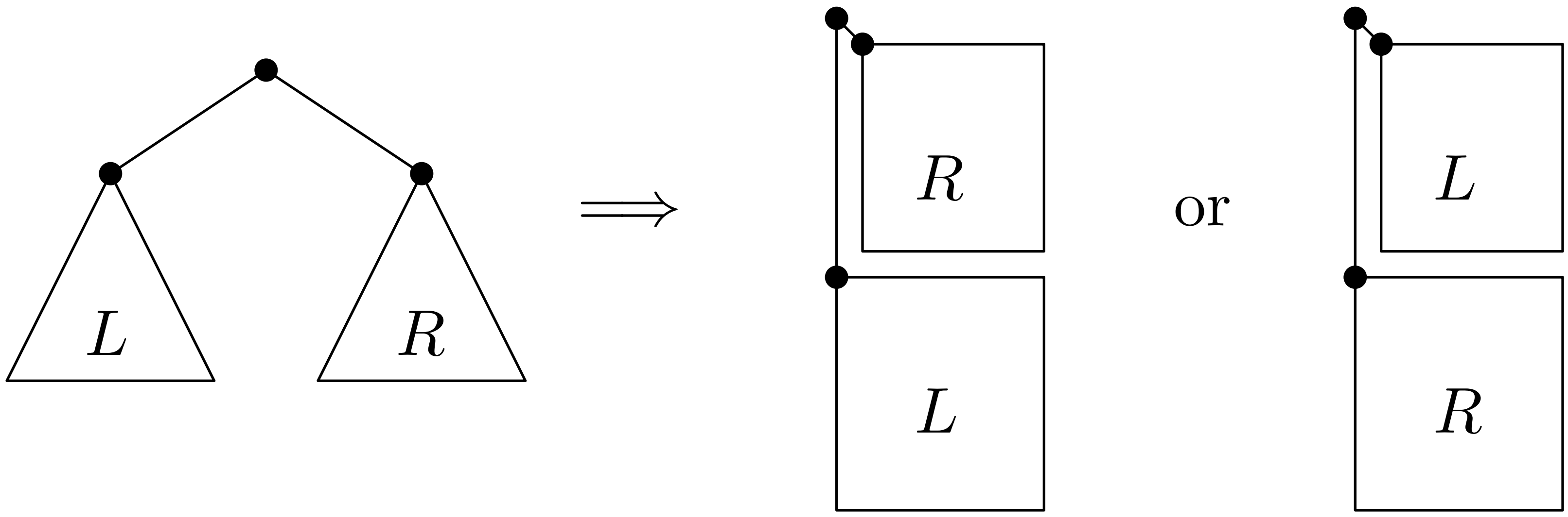
[Crescenzi–Di Battista–Piperno’93]



- if $R \leq L$, left option, else right option

Ex: binary, strict upw.

[Crescenzi–Di Battista–Piperno'93]

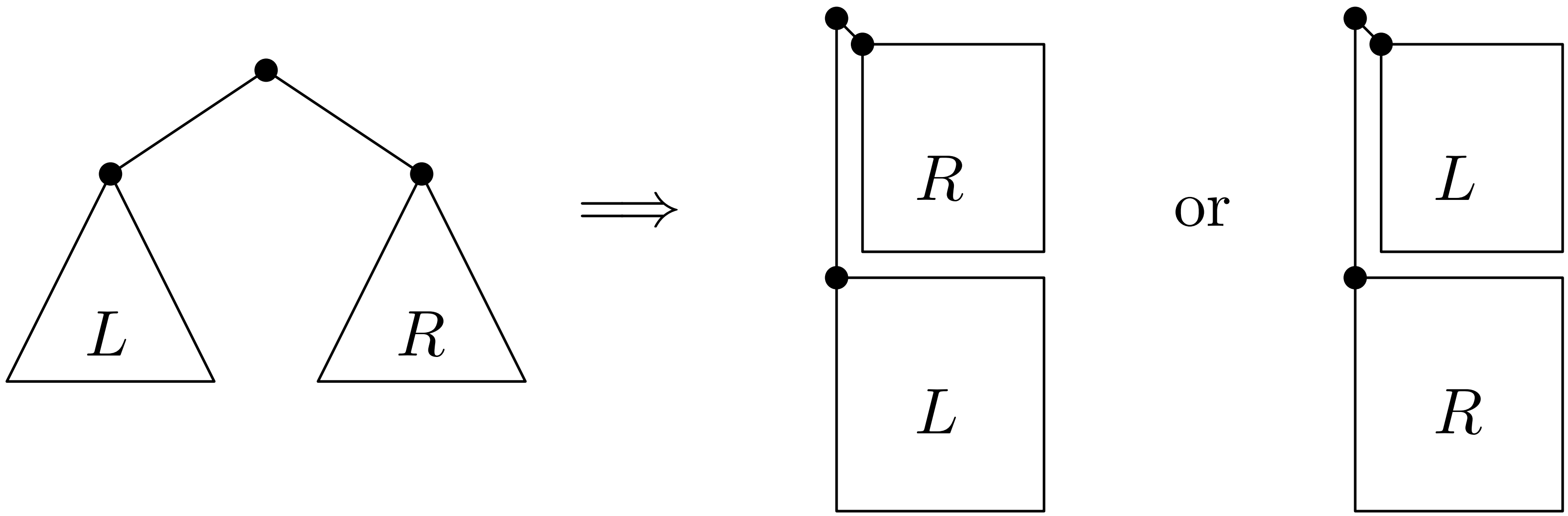


- if $R \leq L$, left option, else right option

$\Rightarrow$ $\boxed{W(n) \leq W(n/2) + O(1)}$

Ex: binary, strict upw.

[Crescenzi–Di Battista–Piperno'93]



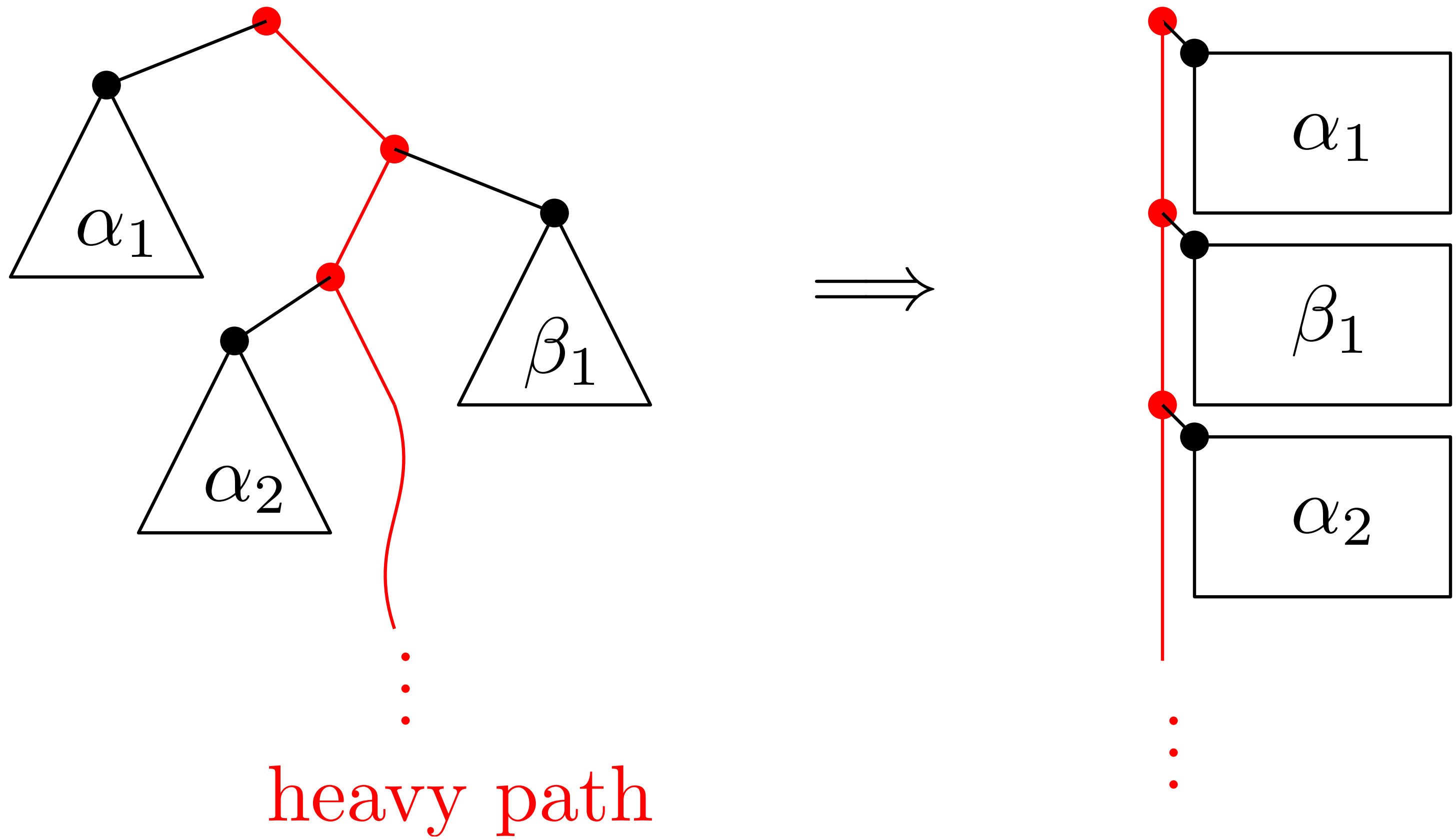
- if $R \leq L$, left option, else right option

$\Rightarrow$ $\boxed{W(n) \leq W(n/2) + O(1)}$

$\Rightarrow$ $\boxed{O(\log n)}$ width

Ex: binary, strict upw.

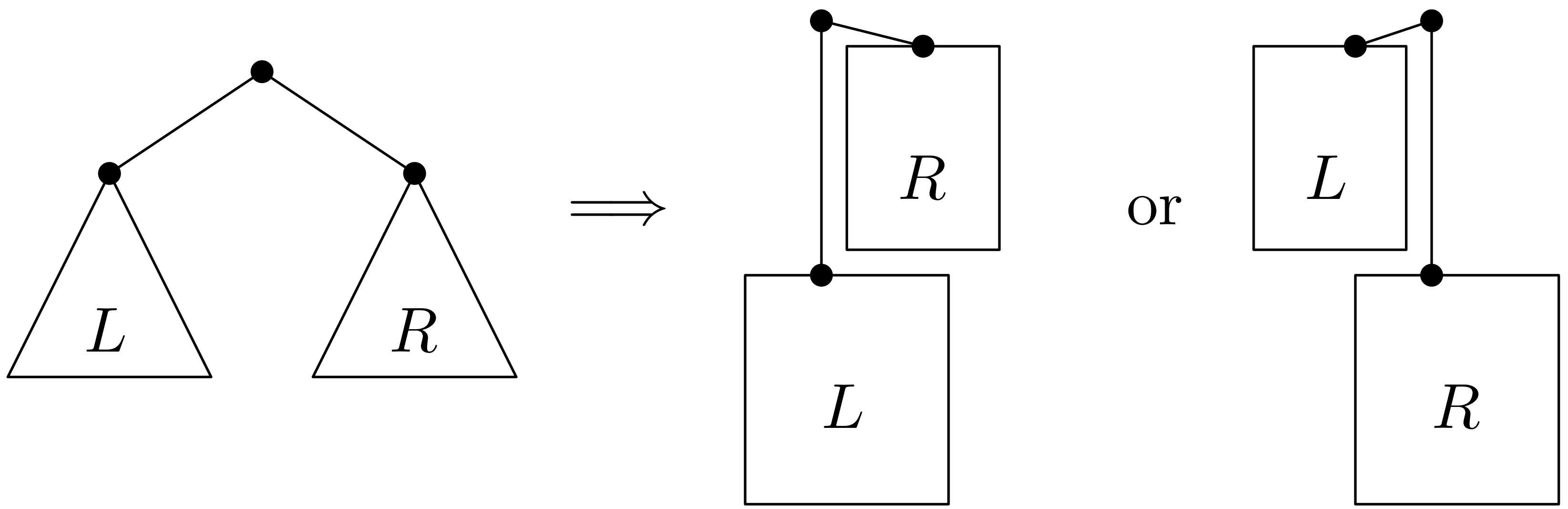
equivalent to



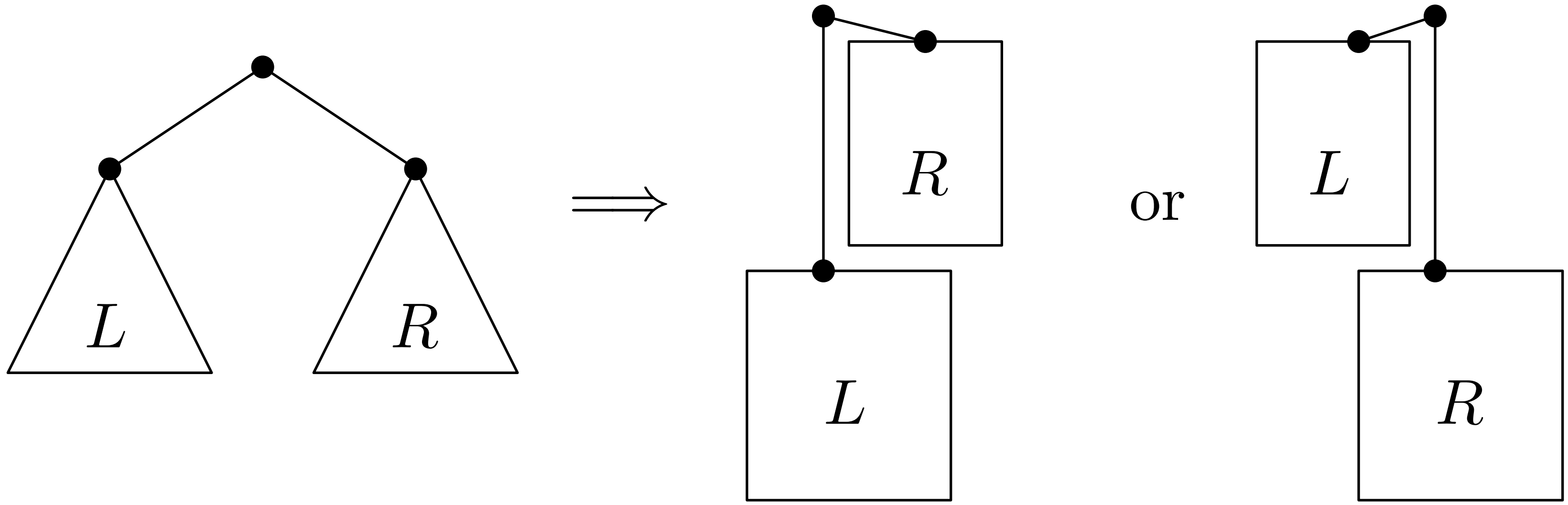
Technique 2: “LR Path”

Ex: binary, strict upw., *ordered* [C.'99]

Ex: binary, strict upw., *ordered* [C.'99]

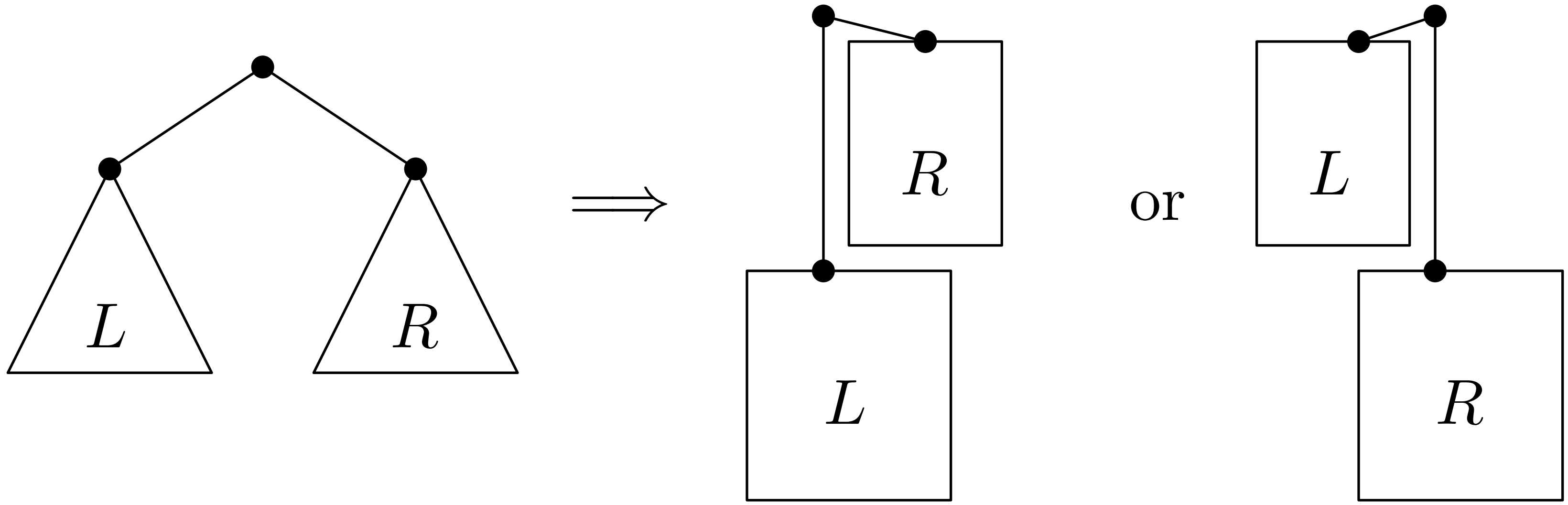


Ex: binary, strict upw., *ordered* [C.'99]



(called "LR drawings")

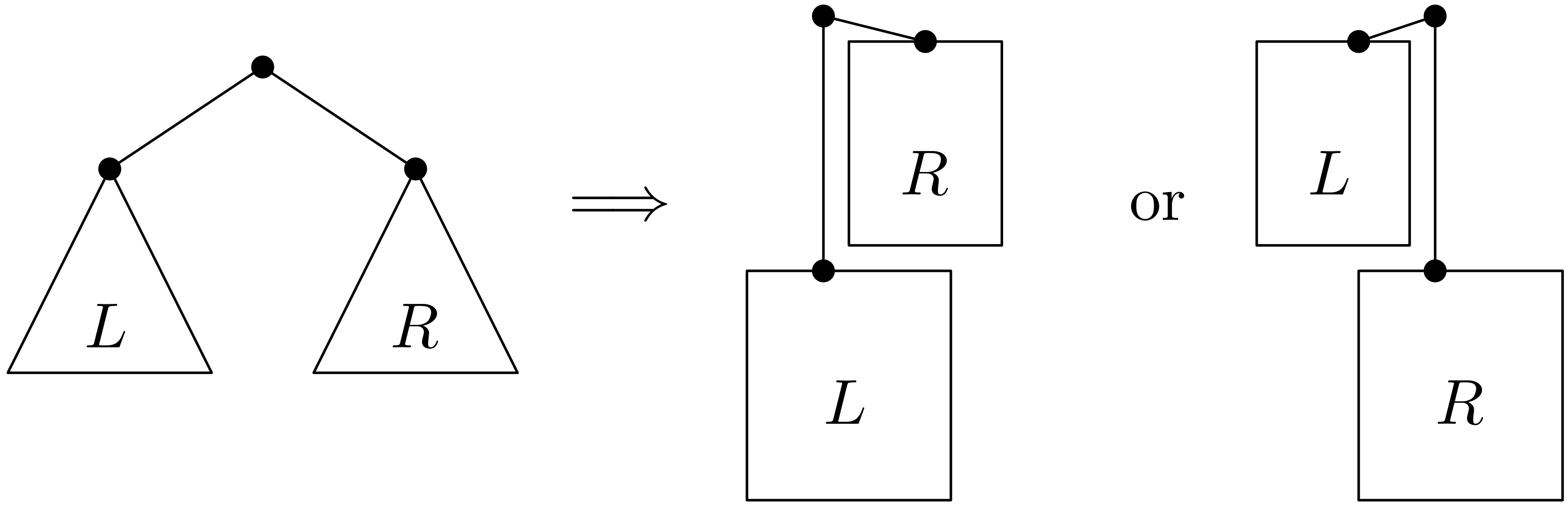
Ex: binary, strict upw., *ordered* [C.'99]



(called "LR drawings")

- how to decide which option?

Ex: binary, strict upw., *ordered* [C.'99]

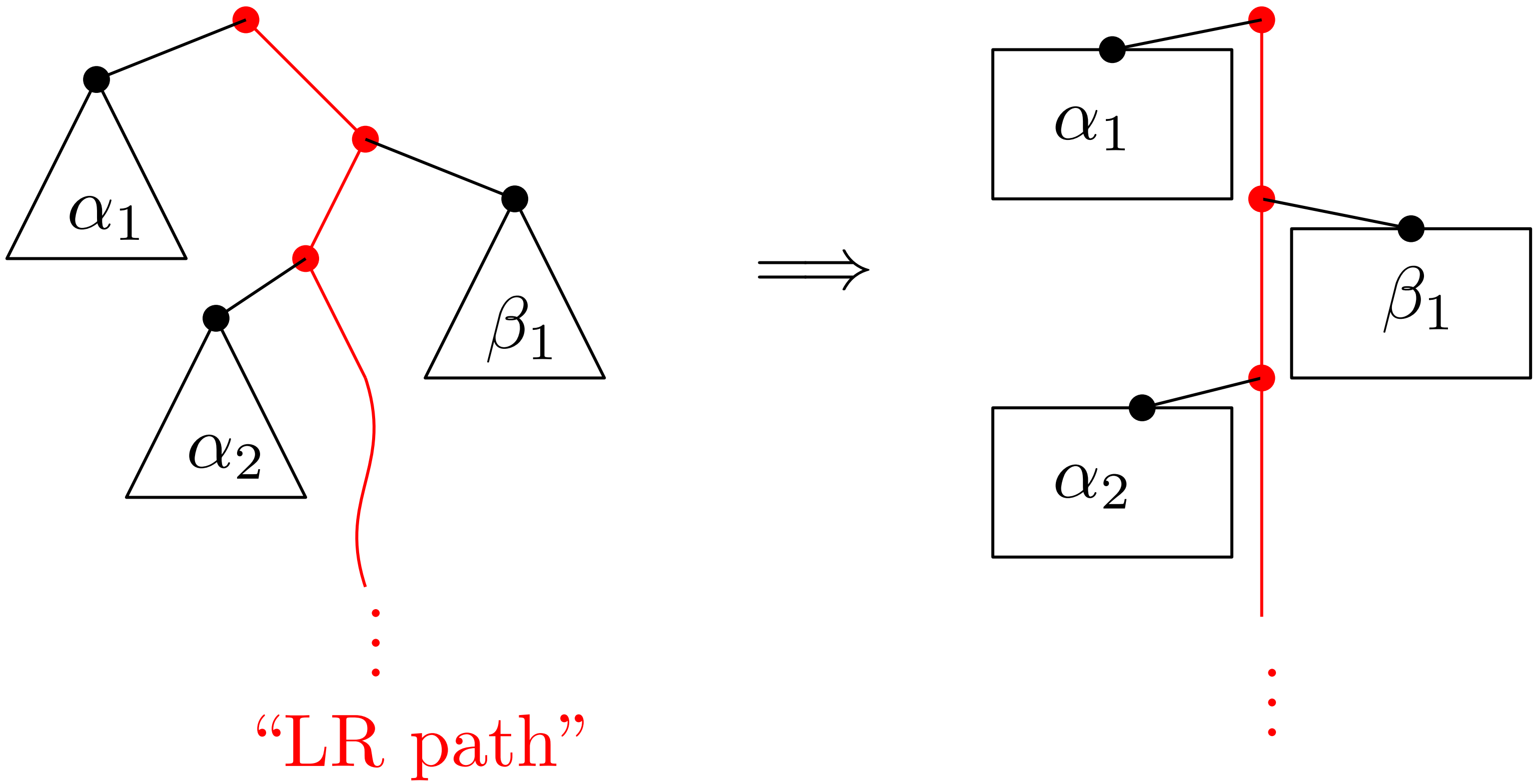


(called "LR drawings")

- how to decide which option? tricky...

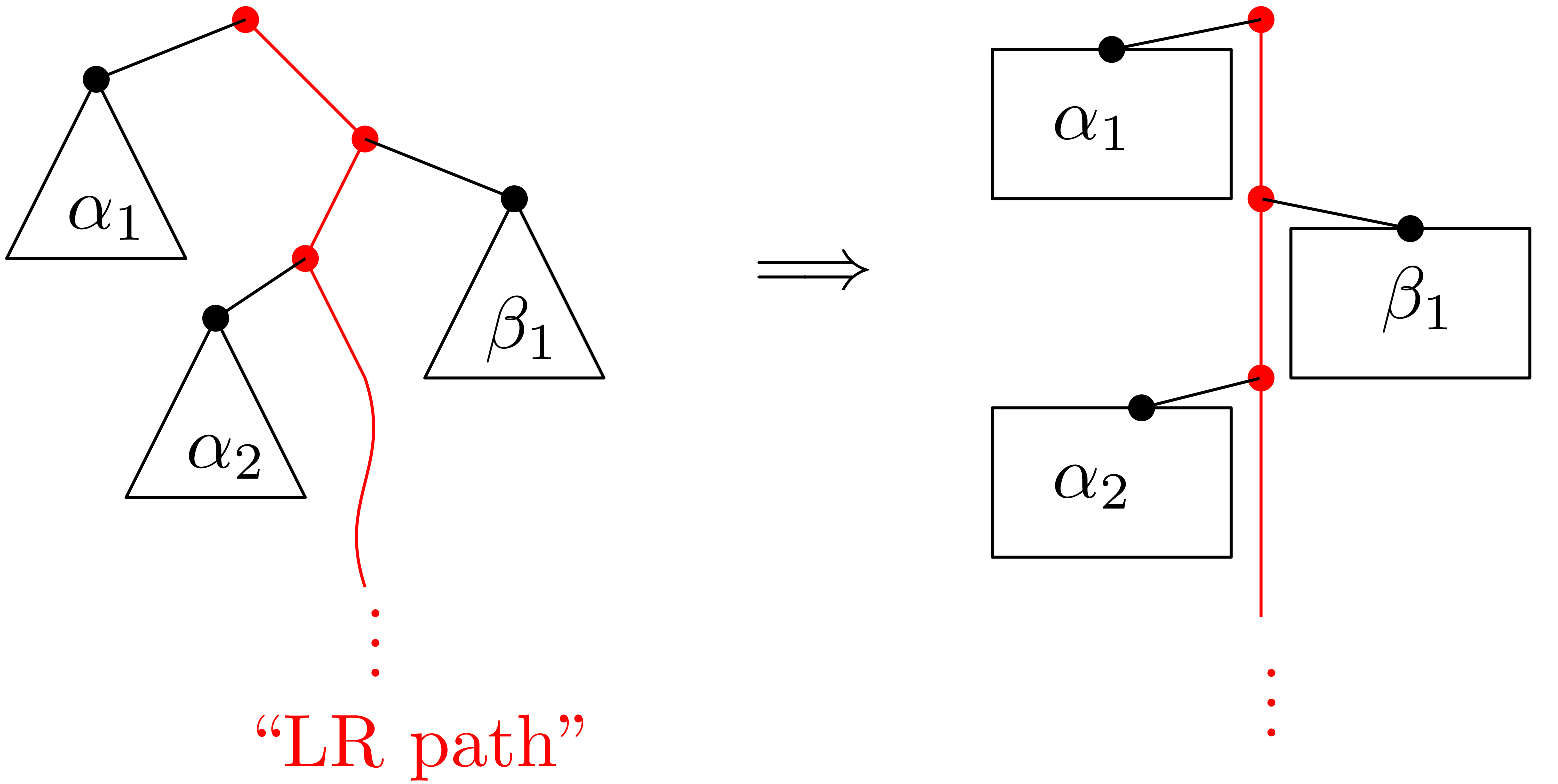
Ex: binary, strict upw., *ordered* [C.'99]

equivalent to



Ex: binary, strict upw., *ordered* [C.'99]

equivalent to



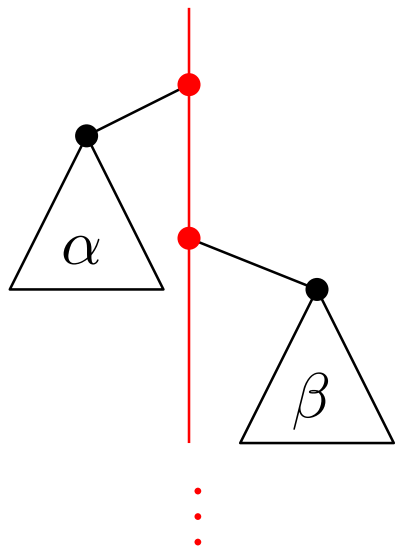
$$W(n) = \min_{\text{path } \pi} \max_{\substack{\text{left subtree } \alpha, \\ \text{right subtree } \beta \text{ of } \pi}} (W(\alpha) + W(\beta)) + O(1)$$

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 1: just use heavy path

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 1: just use heavy path



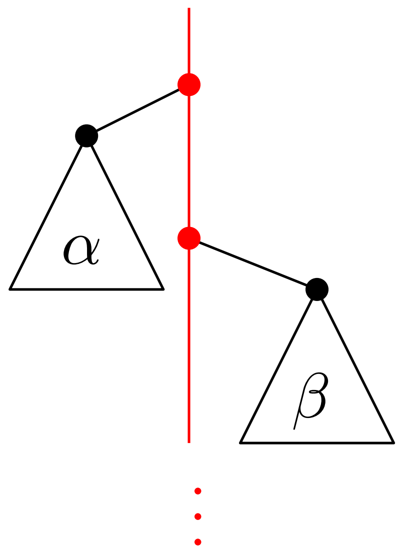
α = max left subtree

β = max right subtree

$$W(n) = \max_{\substack{\alpha \le n/2, \\ \beta \le (n-\alpha)/2}} (W(\alpha) + W(\beta)) + O(1)$$

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 1: just use heavy path



α = max left subtree

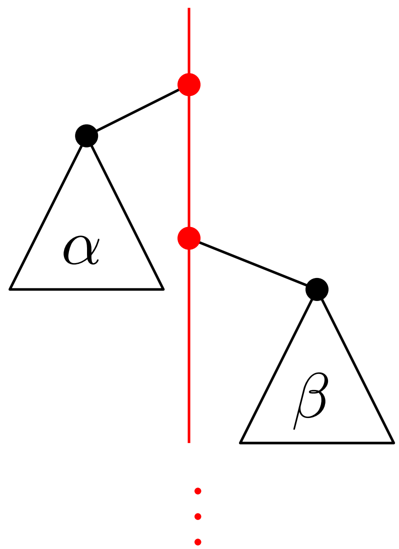
β = max right subtree

$$W(n) = \max_{\substack{\alpha \le n/2,\\ \beta \le (n-\alpha)/2}} (W(\alpha) + W(\beta)) + O(1)$$

$$= W(n/2) + W(n/4) + O(1)$$

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 1: just use heavy path



α = max left subtree

β = max right subtree

$$W(n) = \max_{\substack{\alpha \le n/2, \\ \beta \le (n-\alpha)/2}} (W(\alpha) + W(\beta)) + O(1)$$

$$= W(n/2) + W(n/4) + O(1)$$

$\Rightarrow O(n^{\log_2 \phi}) = \boxed{O(n^{0.695})}$ width

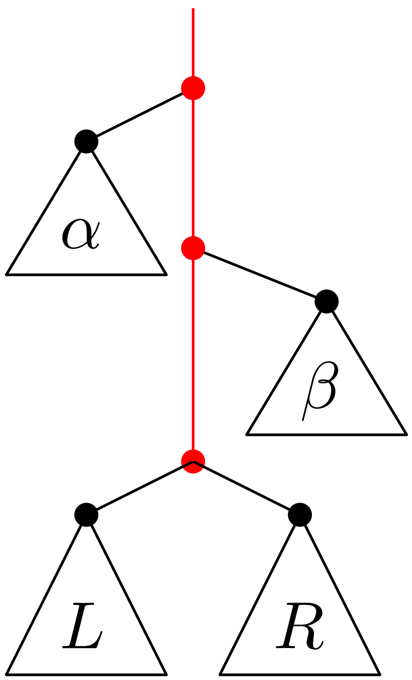
Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 2: $\exists$ path with $\alpha + \beta \leq n/2$

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 2: $\exists$ path with $\alpha + \beta \leq n/2$

Proof:



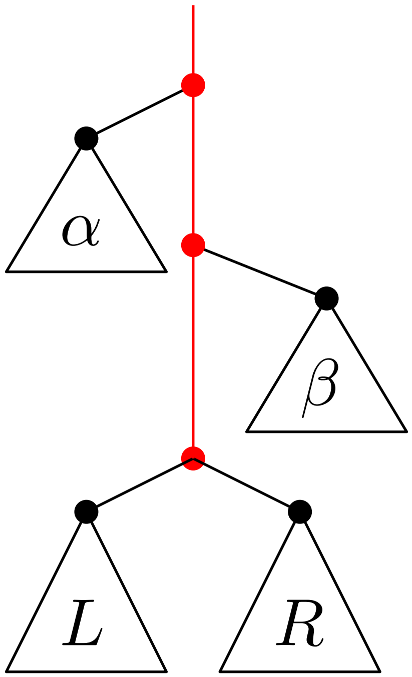
α = current max left subtree

β = current max right subtree

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 2: $\exists$ path with $\alpha + \beta \leq n/2$

Proof:



$\alpha =$ current max left subtree

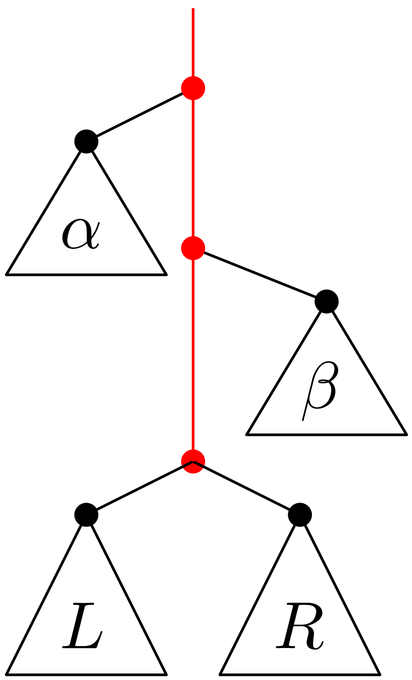
$\beta =$ current max right subtree

if $R + \alpha \leq n/2$, go left

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 2: $\exists$ path with $\alpha + \beta \leq n/2$

Proof:



$\alpha =$ current max left subtree

$\beta =$ current max right subtree

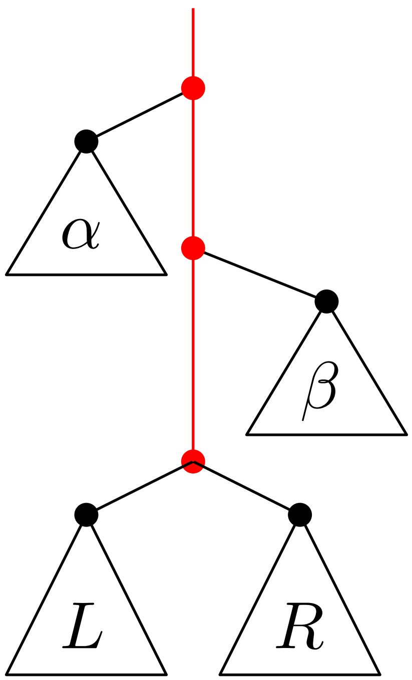
if $R + \alpha \leq n/2$, go left

if $L + \beta \leq n/2$, go right.

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 2: $\exists$ path with $\alpha + \beta \le n/2$

Proof:



$\alpha =$ current max left subtree

$\beta =$ current max right subtree

if $R + \alpha \le n/2$, go left

if $L+\beta \le n/2$, go right. Q.E.D.

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 2: $\exists$ path with $\alpha + \beta \leq n/2$

$$W(n) = \max_{\alpha+\beta\leq n/2} (W(\alpha) + W(\beta)) + O(1)$$

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 2: $\exists$ path with $\alpha + \beta \leq n/2$

$$\begin{aligned} W(n) &= \max_{\alpha+\beta\leq n/2} (W(\alpha) + W(\beta)) + O(1) \\ &= 2\,W(n/4) + O(1) \end{aligned}$$

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 2: $\exists$ path with $\alpha + \beta \leq n/2$

$$W(n) = \max_{\alpha+\beta\leq n/2} (W(\alpha) + W(\beta)) + O(1)$$
$$= 2\,W(n/4) + O(1)$$

$\Rightarrow$ $\boxed{O(\sqrt{n})}$ width

Ex: binary, strict upw., *ordered* [C.'99]

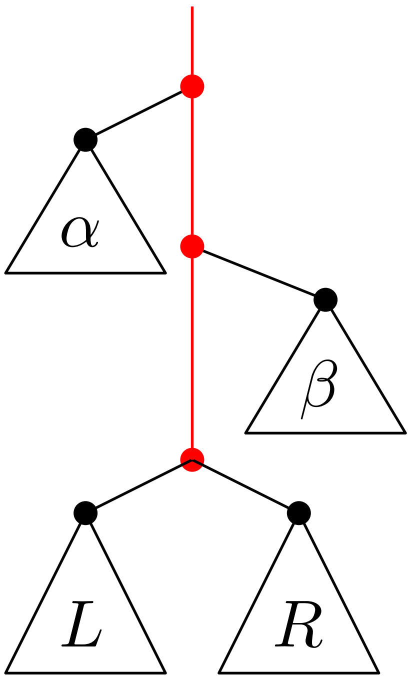
Upper bound 3: $\boxed{O(n^{0.48})}$ width for LR drawings

Ex: binary, strict upw., *ordered* [C.'99]

Upper bound 3: $\boxed{O(n^{0.48})}$ width for LR drawings

New upper bound: $\boxed{O(n^{0.44})}$ width for LR drawings

Ex: binary, strict upw., *ordered* [new]

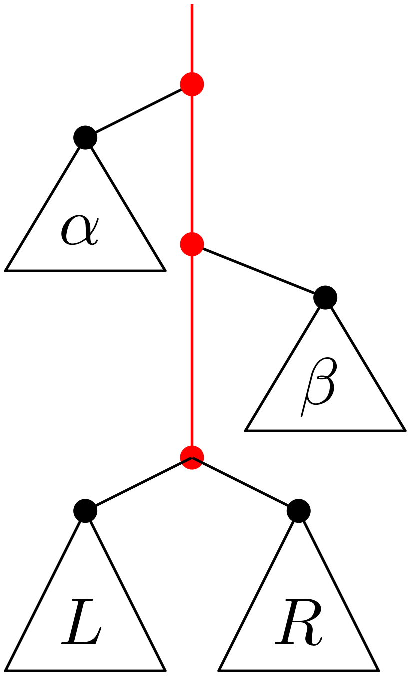


$\alpha =$ current max left subtree

$\beta =$ current max right subtree

assume $W(\alpha) + W(\beta) \leq \widehat{W}$

Ex: binary, strict upw., *ordered* [new]



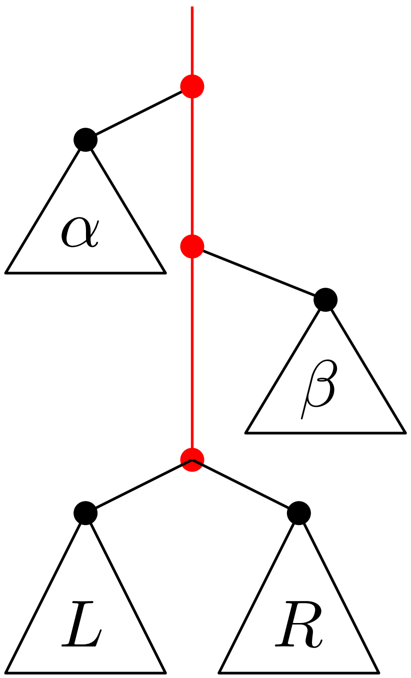
$\alpha =$ current max left subtree

$\beta =$ current max right subtree

assume $W(\alpha) + W(\beta) \leq \widehat{W}$

if $W(R) + W(\alpha) \leq \widehat{W}$, go left

Ex: binary, strict upw., *ordered* [new]

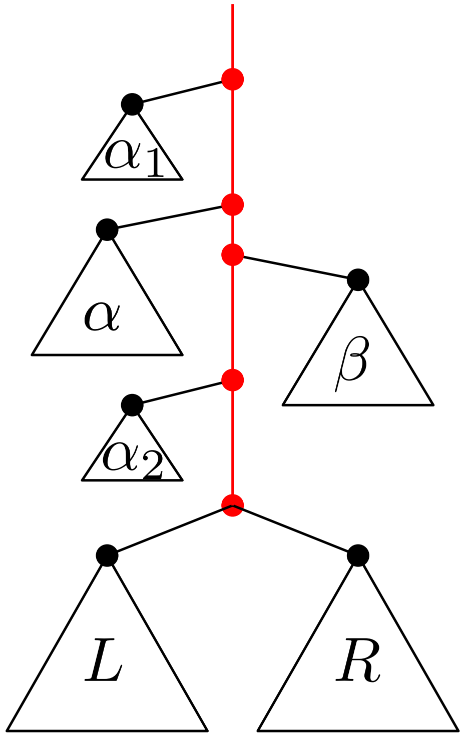


α = current max left subtree
β = current max right subtree

assume $W(\alpha) + W(\beta) \leq \widehat{W}$

if $W(R) + W(\alpha) \leq \widehat{W}$, go left
if $W(L) + W(\beta) \leq \widehat{W}$, go right

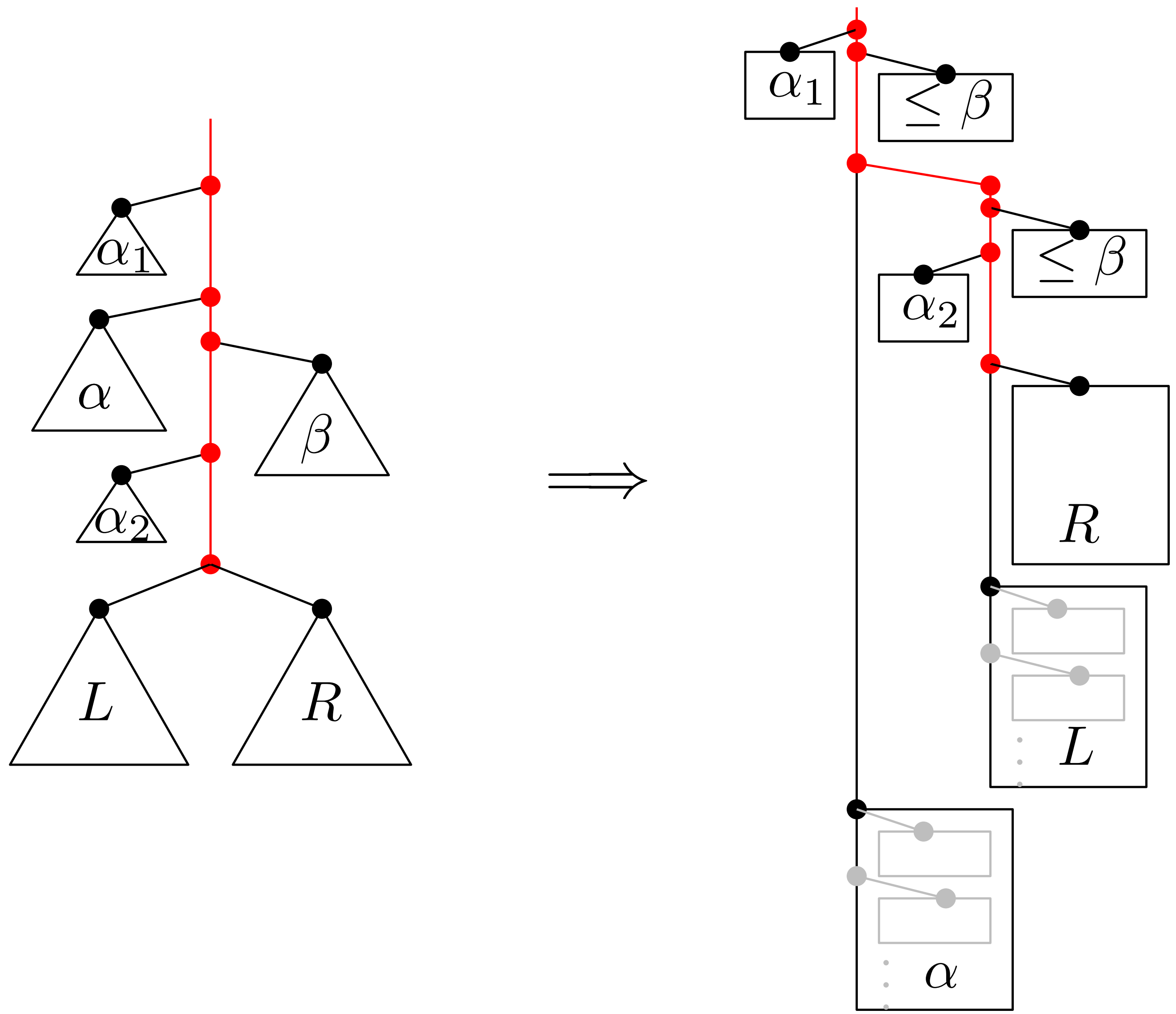
Ex: binary, strict upw., *ordered* [new]



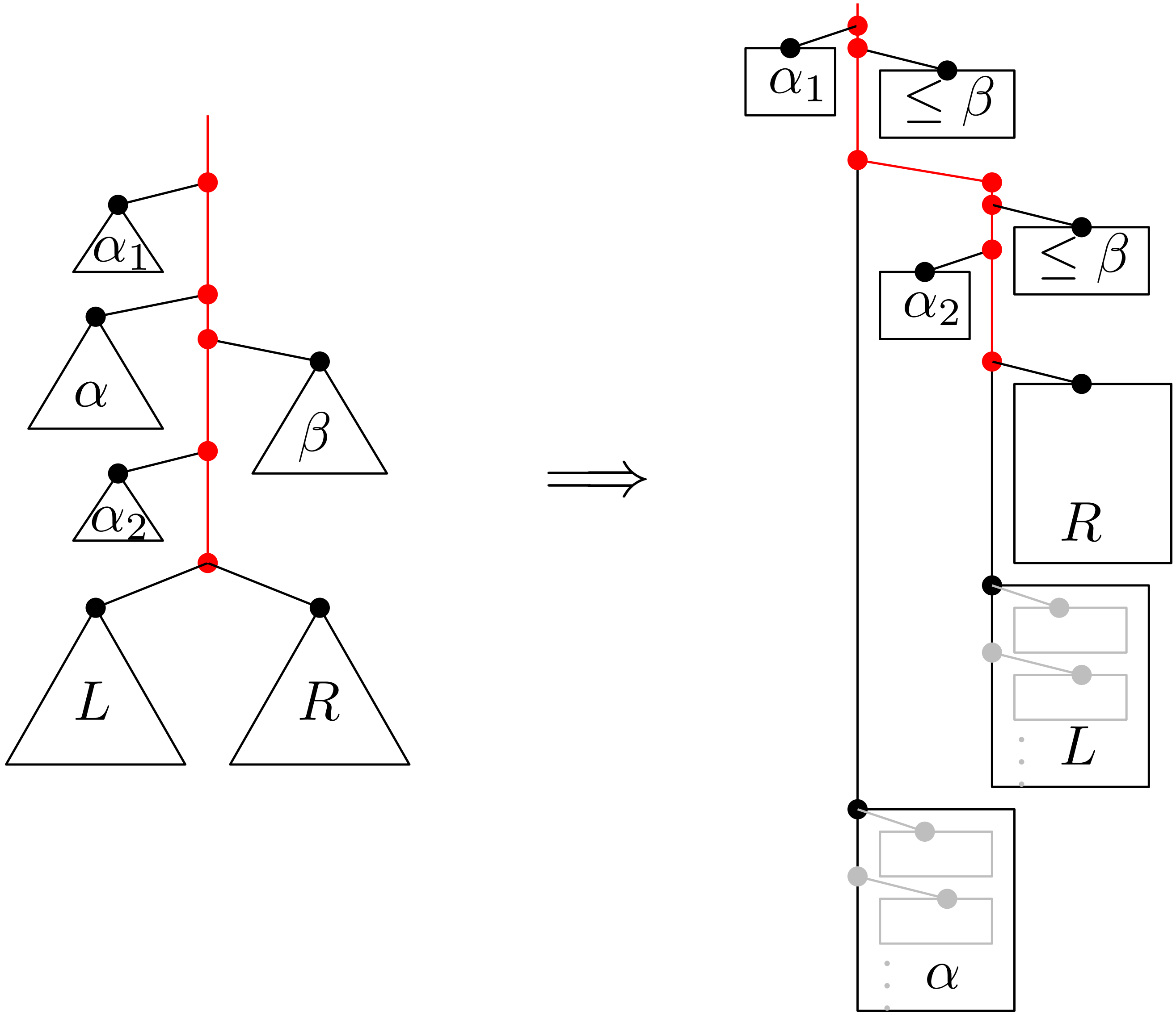
$\alpha_1 =$ max left subtree above α

$\alpha_2 =$ max left subtree below α

Ex: binary, strict upw., *ordered* [new]



Ex: binary, strict upw., *ordered* [new]



if $W(\max(L, R)) + W(\alpha_1) + W(\alpha_2) \leq \widehat{W}$, ok

Ex: binary, strict upw., *ordered* [new]

$$W(n) \leq \max_{\substack{L+R+ \\ \alpha+\beta+ \\ \alpha_1+\alpha_2=n}} \min \begin{cases} W(R) + W(\alpha) \\ W(L) + W(\beta) \\ W(\max(L, R)) + W(\alpha_1) + W(\alpha_2) \end{cases}$$

$$+ O(1)$$

Ex: binary, strict upw., *ordered* [new]

$$W(n) \leq \max_{\substack{L+R+ \\ \alpha+\beta+ \\ \alpha_1+\alpha_2+ \\ \beta_1+\beta_2=n}} \min \begin{cases} W(R) + W(\alpha) \\ W(L) + W(\beta) \\ W(\max(L,R)) + W(\alpha_1) + W(\alpha_2) \\ W(\max(L,R)) + W(\beta_1) + W(\beta_2) \end{cases}$$

$$+\,O(1)$$

Ex: binary, strict upw., *ordered* [new]

$$W(n) \le \max_{\substack{L+R+\\ \alpha+\beta+\\ \alpha_1+\alpha_2+\cdots+\\ \beta_1+\beta_2=n}} \min \begin{cases} W(R)+W(\alpha) \\ W(L)+W(\beta) \\ W(\max(L,R))+W(\alpha_1)+W(\alpha_2) \\ W(\max(L,R))+W(\beta_1)+W(\beta_2) \\ W(\max(L,R))+W(\alpha_3)+\cdots+W(\alpha_6) \end{cases}$$

$$+\,O(1)$$

Ex: binary, strict upw., *ordered* [new]

$$W(n) \le \max_{\substack{L+R+\\ \alpha+\beta+\\ \alpha_1+\alpha_2+\cdots+\\ \beta_1+\beta_2+\cdots=n}} \min \begin{cases} W(R)+W(\alpha) \\ W(L)+W(\beta) \\ W(\max(L,R))+W(\alpha_1)+W(\alpha_2) \\ W(\max(L,R))+W(\beta_1)+W(\beta_2) \\ W(\max(L,R))+W(\alpha_3)+\cdots+W(\alpha_6) \\ W(\max(L,R))+W(\beta_3)+\cdots+W(\beta_6) \\ \vdots \end{cases}$$

$$+\,O(1)$$

Ex: binary, strict upw., *ordered* [new]

$$W(n) \;\le\; \max_{\substack{L+R+\\ \alpha+\beta+\\ \alpha_1+\alpha_2+\cdots+\\ \beta_1+\beta_2+\cdots=n}} \quad \min \begin{cases} W(R)+W(\alpha) \\ W(L)+W(\beta) \\ W(\max(L,R))+W(\alpha_1)+W(\alpha_2) \\ W(\max(L,R))+W(\beta_1)+W(\beta_2) \\ W(\max(L,R))+W(\alpha_3)+\cdots+W(\alpha_6) \\ W(\max(L,R))+W(\beta_3)+\cdots+W(\beta_6) \\ \vdots \end{cases}$$

$$+\,O(1)$$

$\Rightarrow$ $\boxed{O(n^{0.44})}$ width

(by induction, taking convex comb. & using Hölder's inequality)

Ex: binary, strict upw., *ordered*

[Frati–Patrignani–Roselli'17]

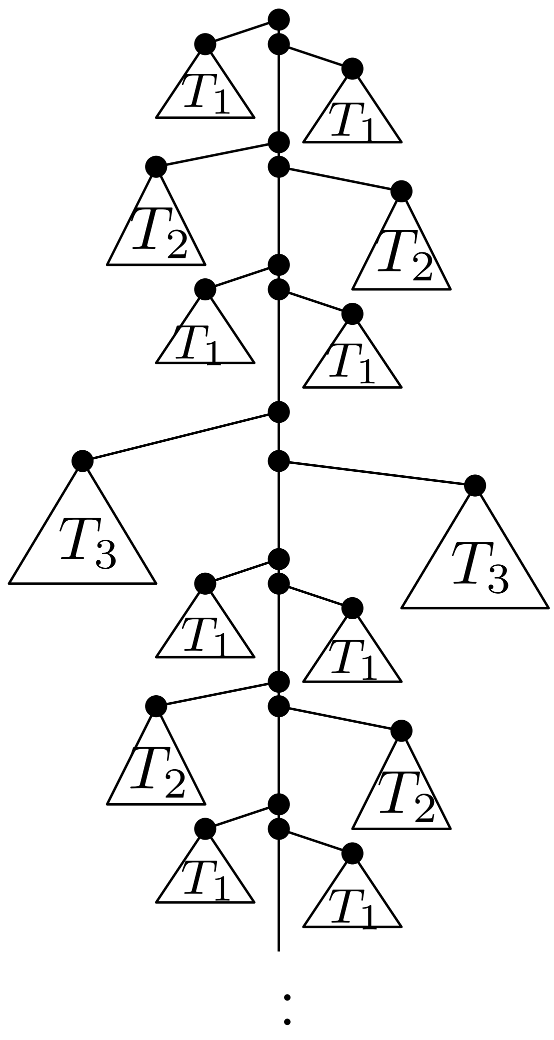
Lower bound: $\boxed{\Omega(n^{0.418})}$ width for LR drawings

Ex: binary, strict upw., *ordered*

[Frati–Patrignani–Roselli'17]

Lower bound: $\Omega(n^{0.418})$ width for LR drawings

$T_k \ =$

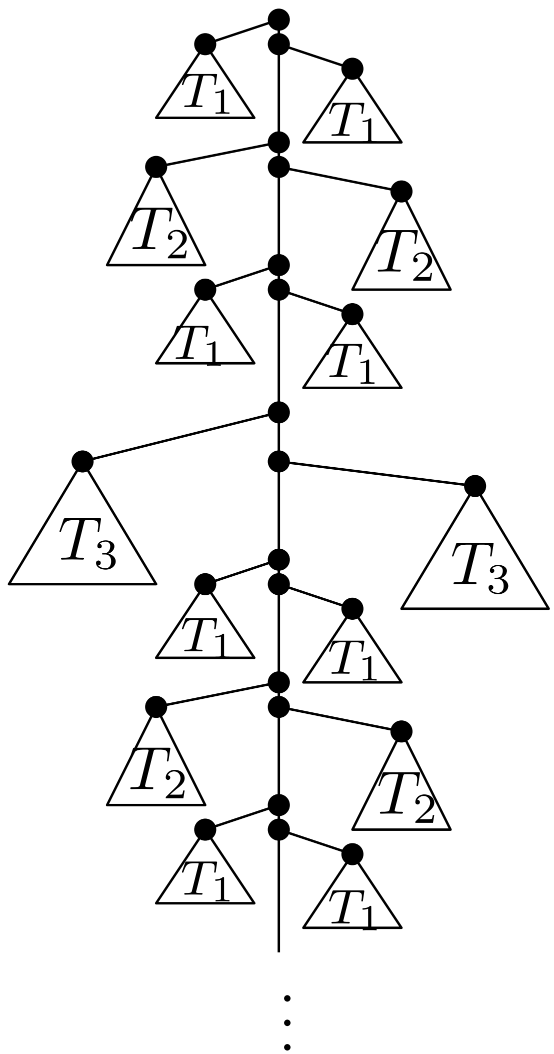


Ex: binary, strict upw., *ordered*

[Frati–Patrignani–Roselli'17]

Lower bound: $\Omega(n^{0.418})$ width for LR drawings

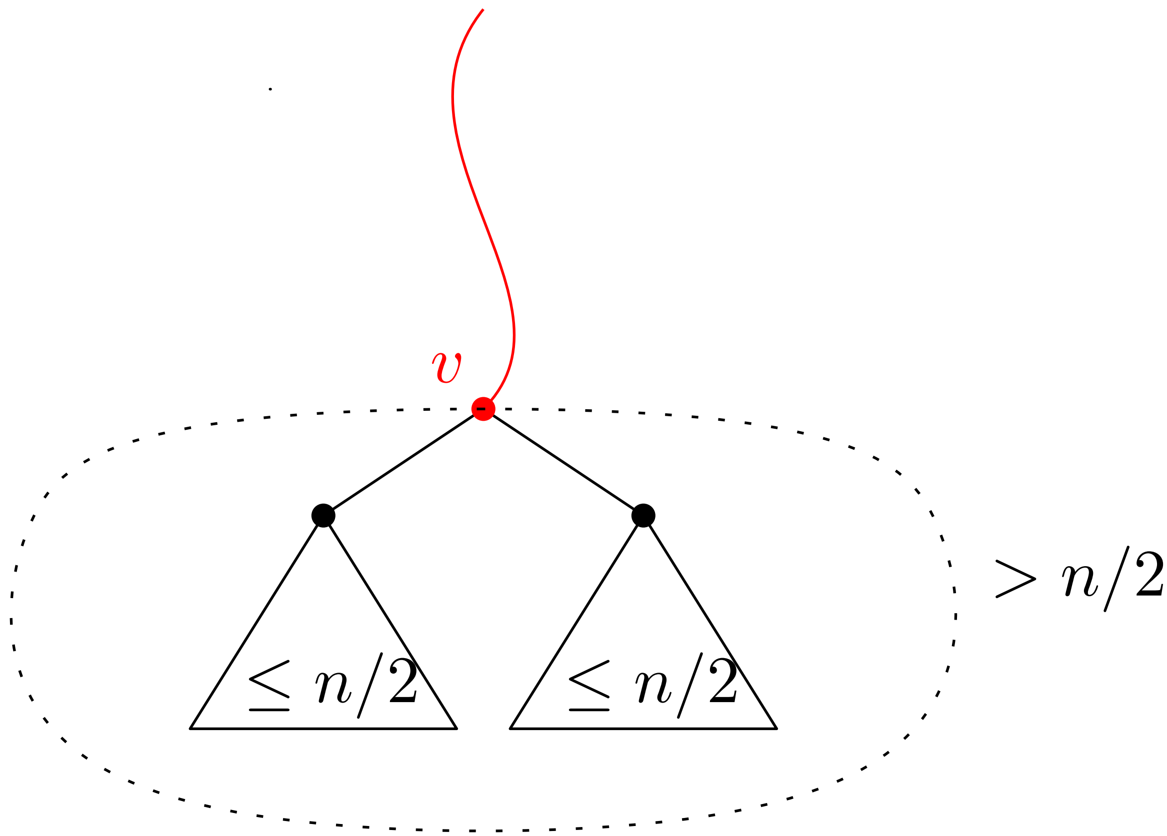
$T_k =$



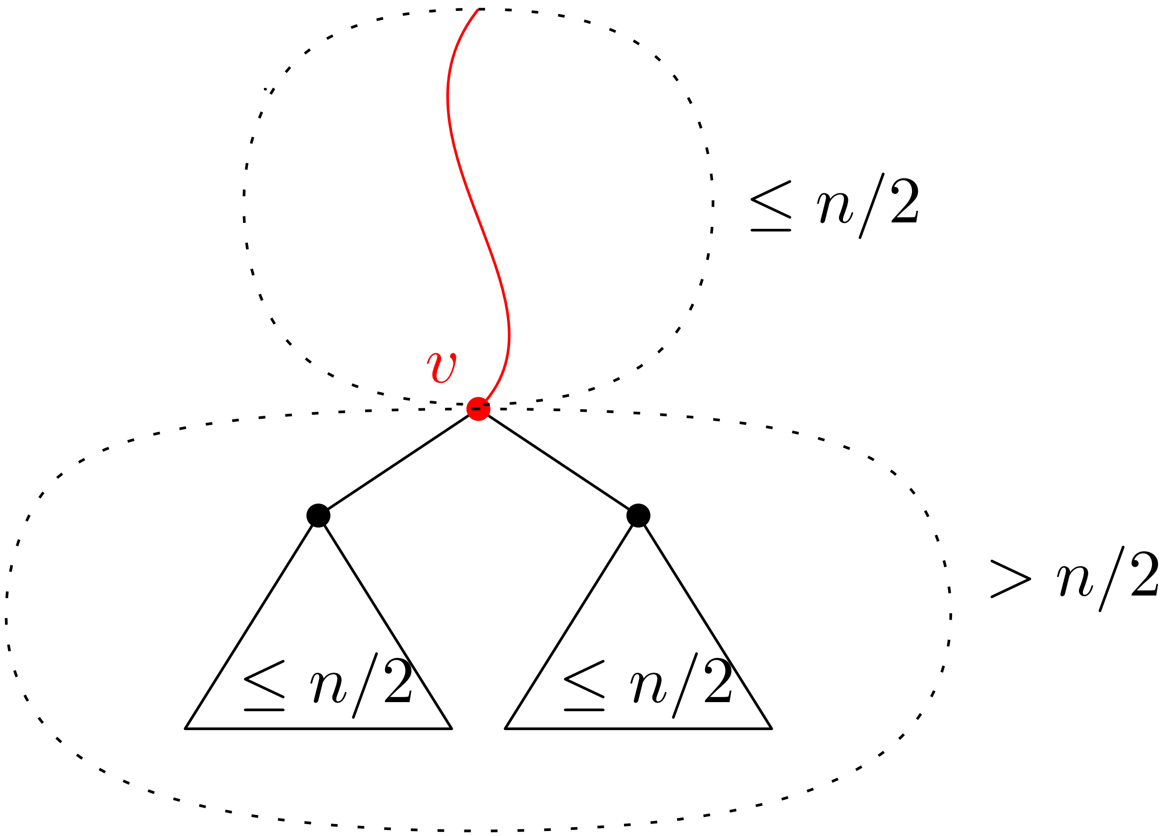
(open: best exponent?)

Technique 3: “Skewed Centroid”

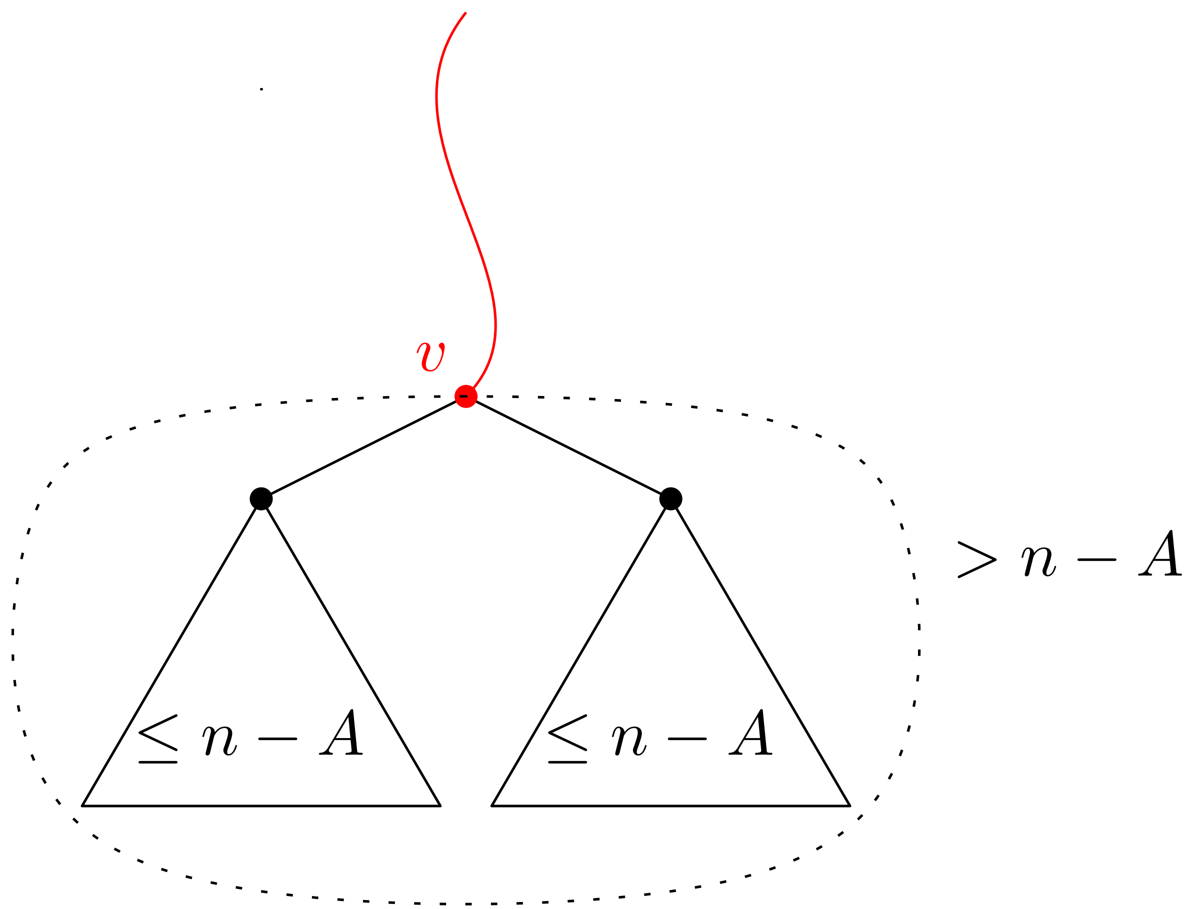
centroid = lowest node v with subtree size $> n/2$



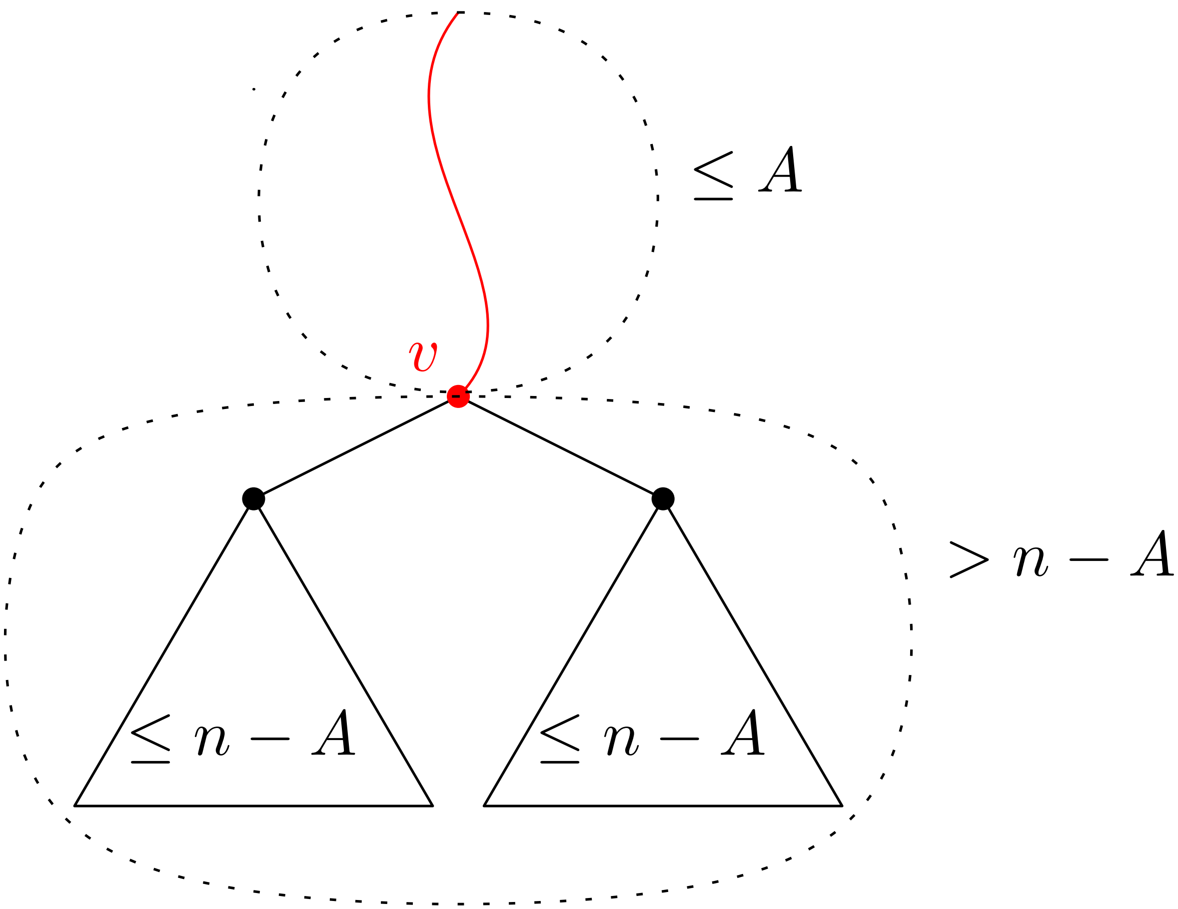
centroid $=$ lowest node v with subtree size $> n/2$



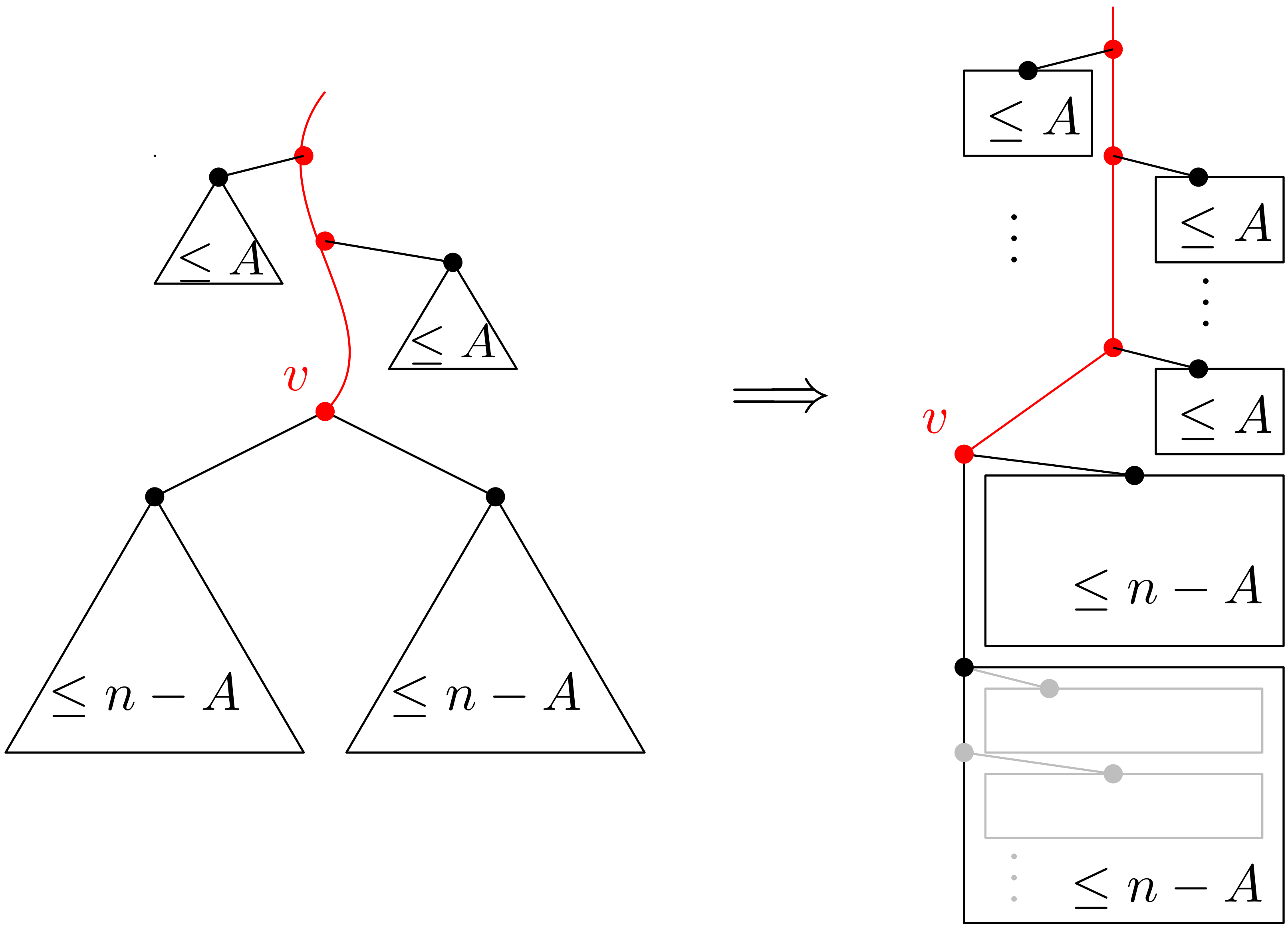
"skewed" centroid $=$ lowest node v with subtree size $> n - A$



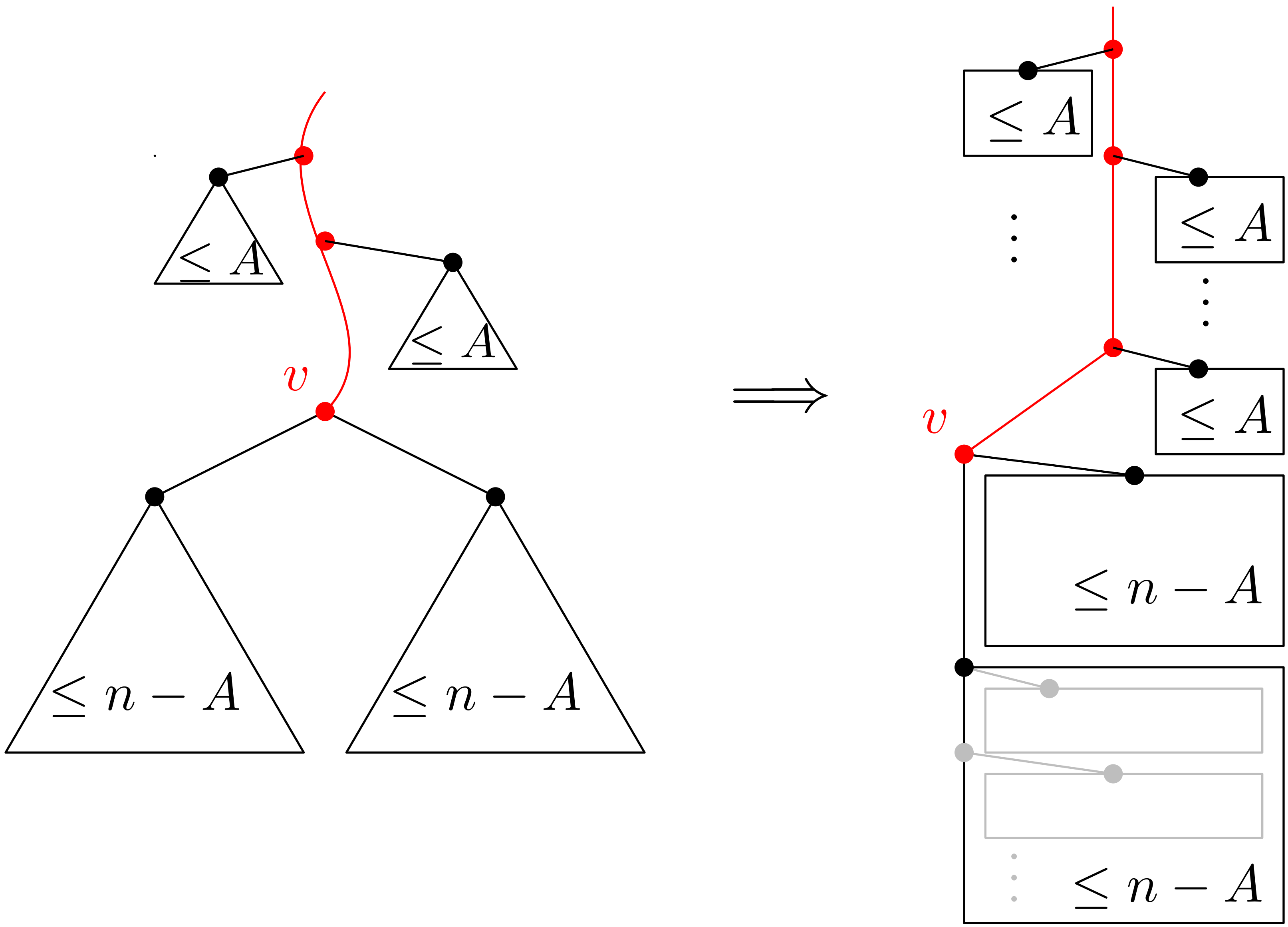
“skewed” centroid $=$ lowest node v with subtree size $> n - A$



Ex: binary, strict upw., *ordered* [C.'99]



Ex: binary, strict upw., *ordered* [C.'99]



$$W(n) \leq \max\{2\,W(A),\, W(n-A)\} + O(1)$$

Ex: binary, strict upw., *ordered* [C.'99]

$$\boxed{W(n) \;\le\; \max\{2\,W(A),\, W(n-A)\} + O(1)}$$

Ex: binary, strict upw., *ordered* [C.'99]

$$\boxed{W(n) \;\le\; \max\{2\,W(A),\, W(n-A)\} + O(1)}$$

set $A = n/b$ for large constant b

$\Rightarrow W(n) \;\le\; \max\{2\,W(n/b),\, W((1-1/b)n)\} + O(1)$

$\Rightarrow W(n) = O(n^{1/\log b})$

Ex: binary, strict upw., *ordered* [C.'99]

$$\boxed{W(n) \;\le\; \max\{2\,W(A),\, W(n-A)\} + O(1)}$$

set $A = n/b$ for large constant b

$\Rightarrow W(n) \;\le\; \max\{2\,W(n/b),\, W((1-1/b)n)\} + O(1)$

$\Rightarrow W(n) = O(n^{1/\log b}) \;\;\Rightarrow\;\; \boxed{O(n^{\varepsilon})}$ width

Ex: binary, strict upw., *ordered* [C.'99]

$$\boxed{W(n) \;\le\; \max\{2\,W(A),\, W(n-A)\} + O(1)}$$

set $A = n/b$ for large constant b

$\Rightarrow W(n) \;\le\; \max\{2\,W(n/b),\, W((1-1/b)n)\} + O(1)$

$\Rightarrow W(n) = O(n^{1/\log b}) \;\;\Rightarrow\;\; \boxed{O(n^{\varepsilon})}$ width

nonconstant $b \;\Rightarrow\; \boxed{O(c^{\sqrt{\log n}})}$ width

Ex: binary, strict upw., *ordered* [C.'99]

$$\boxed{W(n) \;\le\; \max\{2\,W(A),\, W(n-A)\} + O(1)}$$

set $A = n/b$ for large constant b

$\Rightarrow W(n) \;\le\; \max\{2\,W(n/b),\, W((1-1/b)n)\} + O(1)$

$\Rightarrow W(n) = O(n^{1/\log b}) \;\;\Rightarrow\;\; \boxed{O(n^{\varepsilon})}$ width

nonconstant $b \;\Rightarrow\; \boxed{O(c^{\sqrt{\log n}})}$ width

("unfortunately", Garg–Rusu'03 showed that heavy path technique can be modified to work for strict upw., *ordered* $\Rightarrow\; O(\log n)$ width)

Next Ex: binary, orthogonal, *ordered*

Next Ex: binary, orthogonal, *ordered*

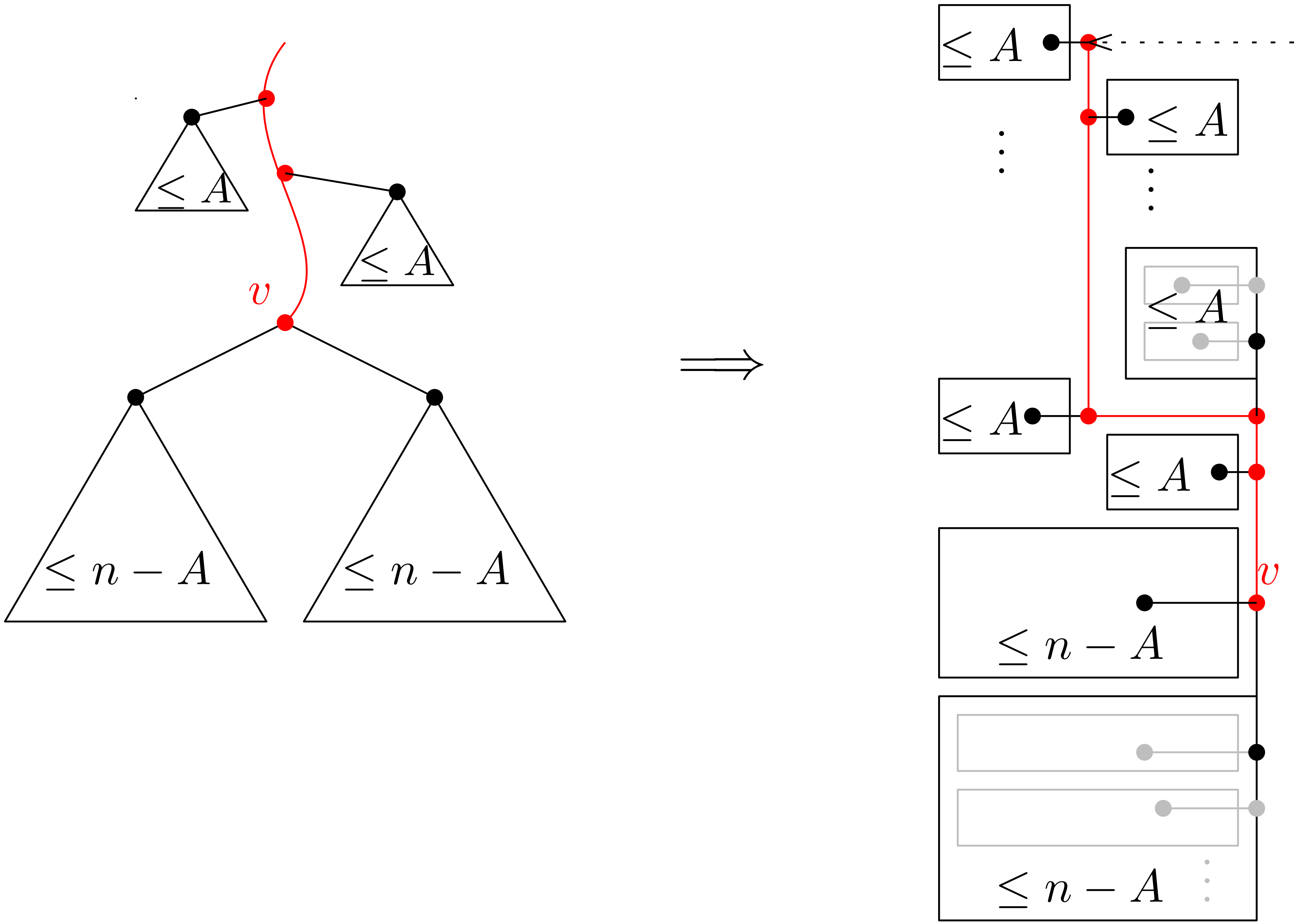
Frati'07: $\boxed{O(\sqrt{n})}$ width (via LR path technique)

Next Ex: binary, orthogonal, *ordered*

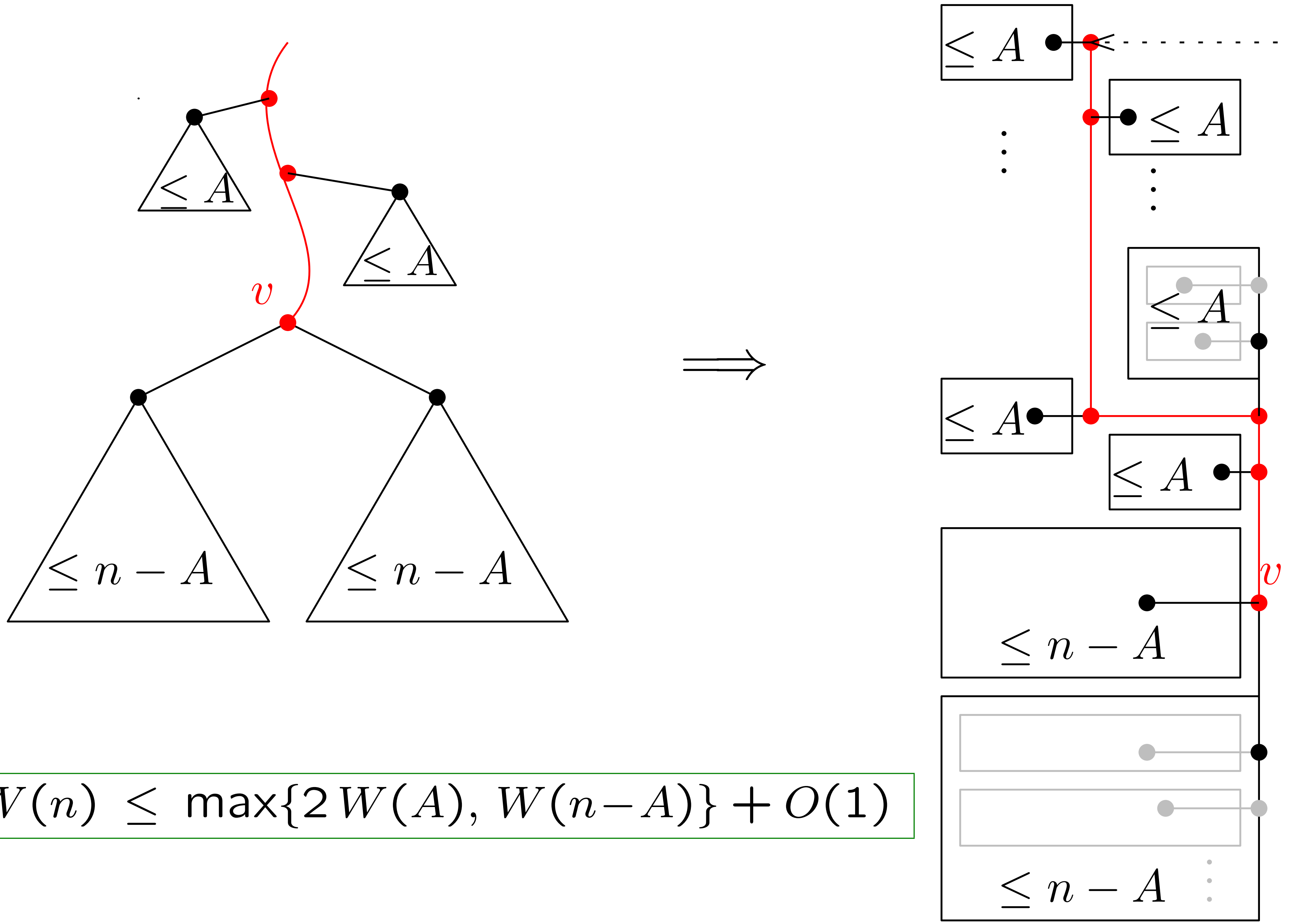
Frati'07: $\boxed{O(\sqrt{n})}$ width (via LR path technique)

new: $\boxed{O(c^{\sqrt{\log n}})}$ width

Next Ex: binary, orthogonal, *ordered* [new]

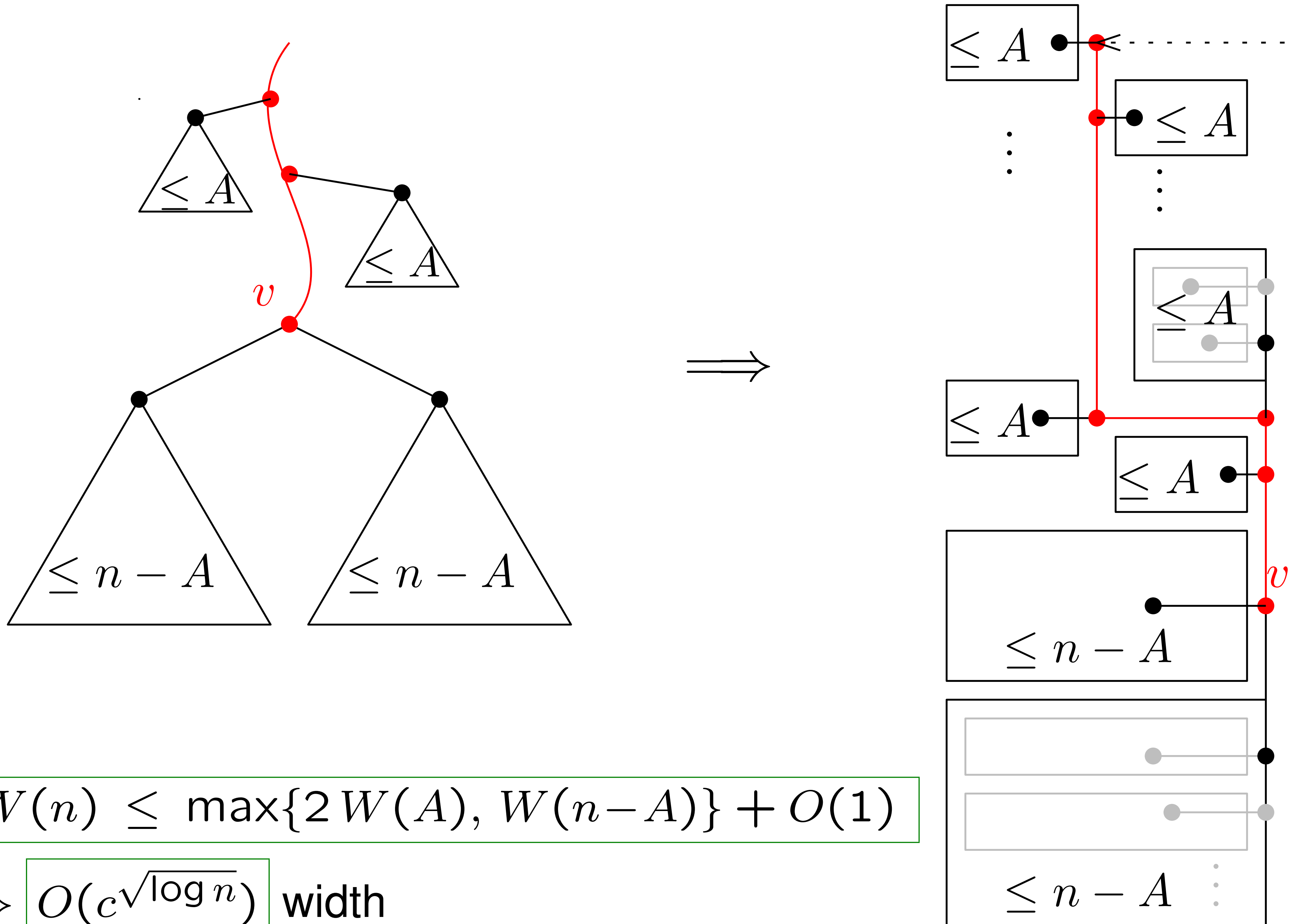


Next Ex: binary, orthogonal, *ordered* [new]



$$W(n) \leq \max\{2\,W(A),\, W(n-A)\} + O(1)$$

Next Ex: binary, orthogonal, *ordered* [new]



$$W(n) \;\leq\; \max\{2\,W(A),\, W(n-A)\} + O(1)$$

$\Rightarrow$ $O(c^{\sqrt{\log n}})$ width

Technique 4: “Double Recurrence”

Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

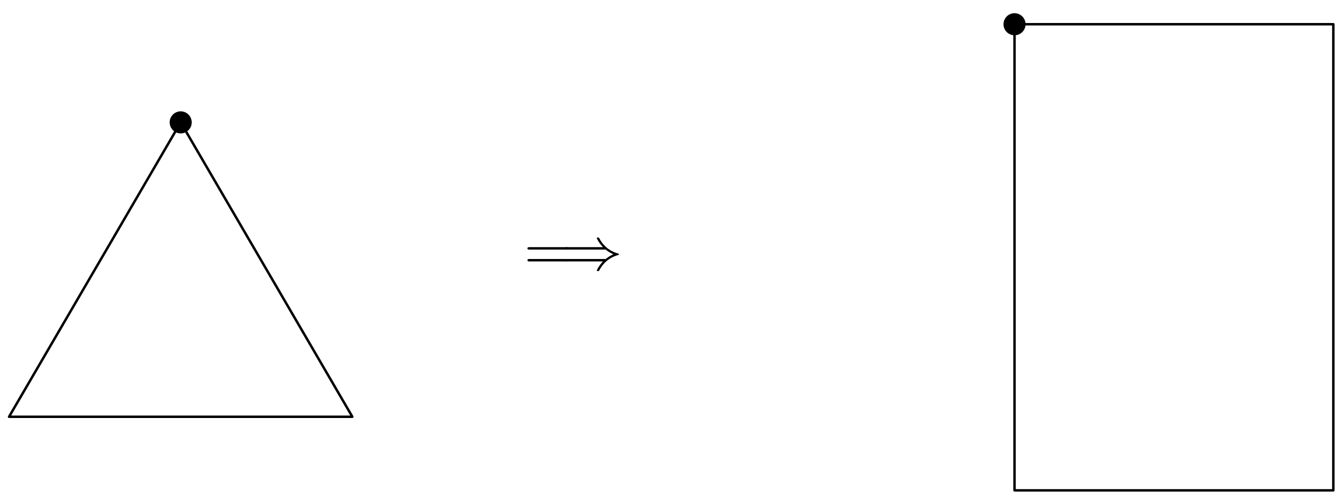
$O(\log^2 n)$ width

Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

$O(\log^2 n)$ width

2 recursive alg'ms:

- Main alg'm

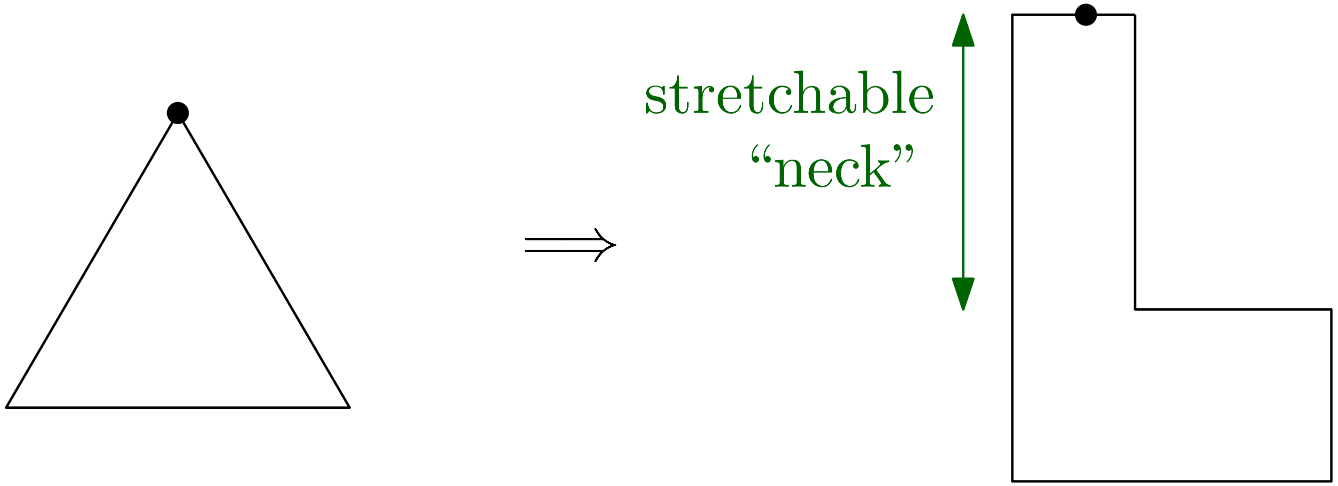


Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

$O(\log^2 n)$ width

2 recursive alg'ms:

- Main alg'm
- "Narrow-neck" alg'm $(*)$

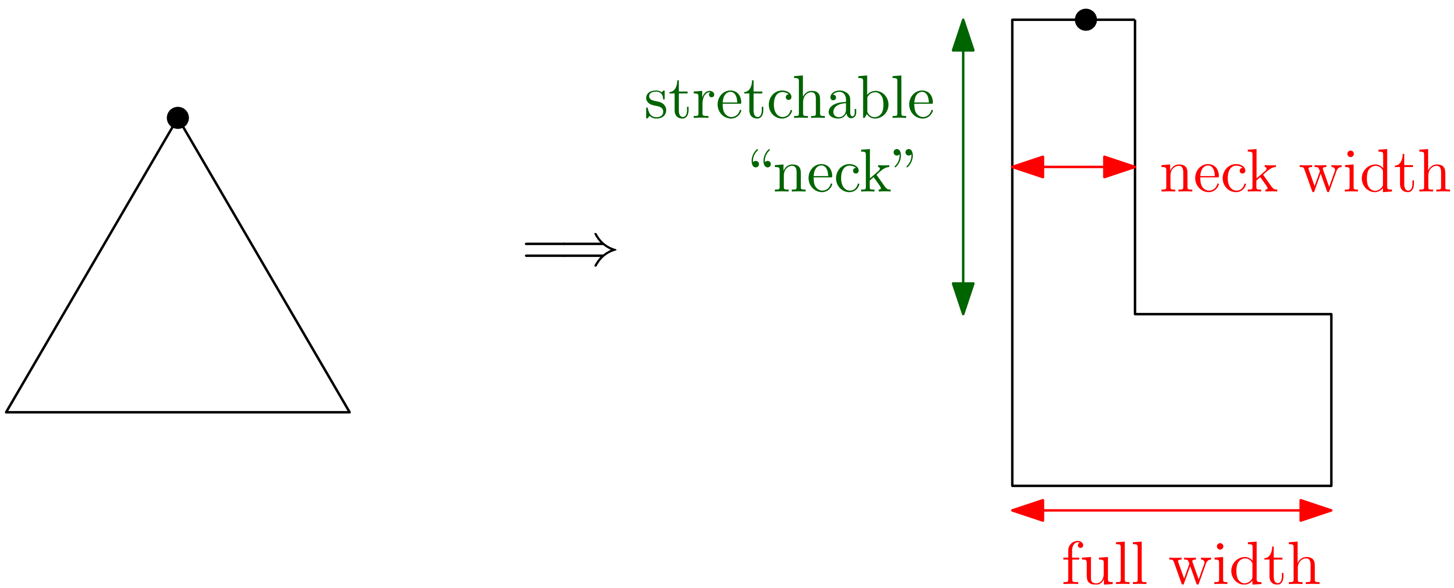


Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

$O(\log^2 n)$ width

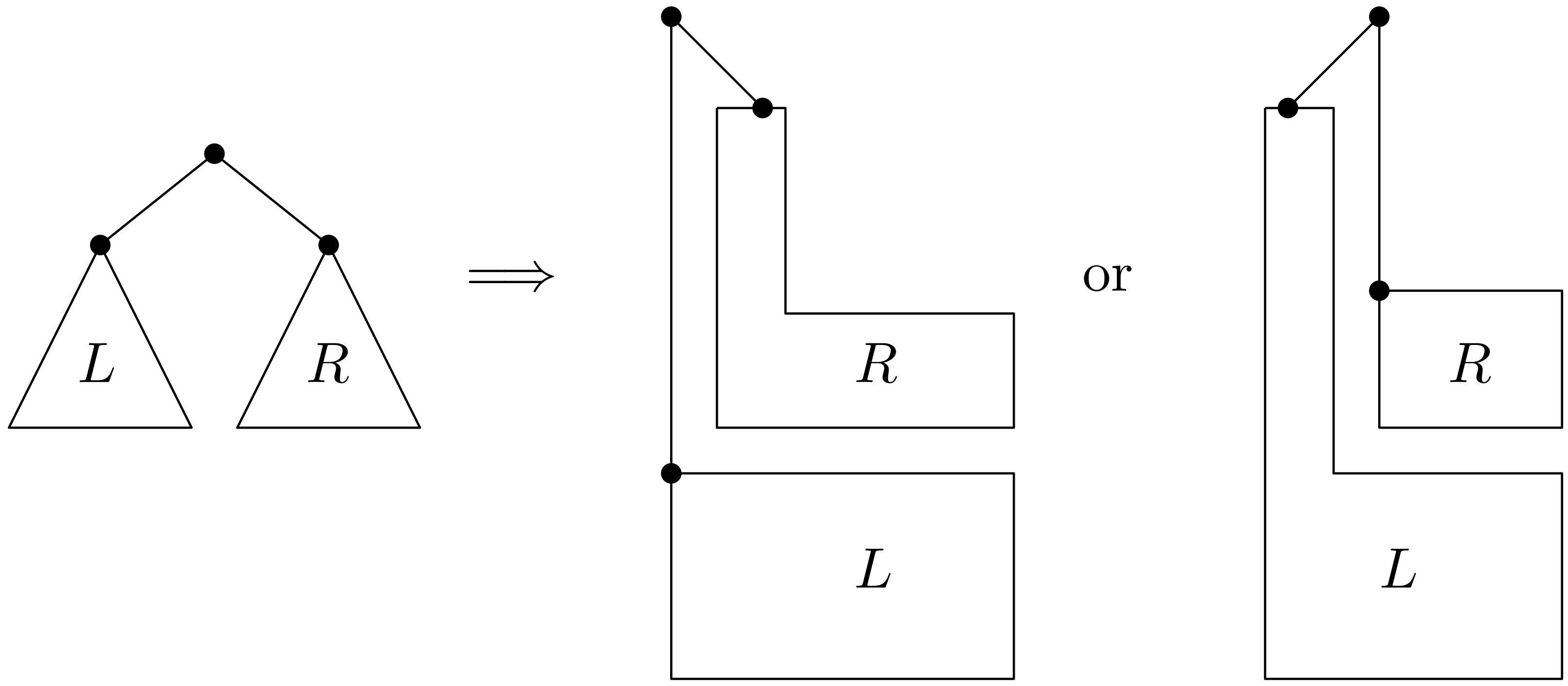
2 recursive alg'ms:

- Main alg'm
- "Narrow-neck" alg'm $(*)$



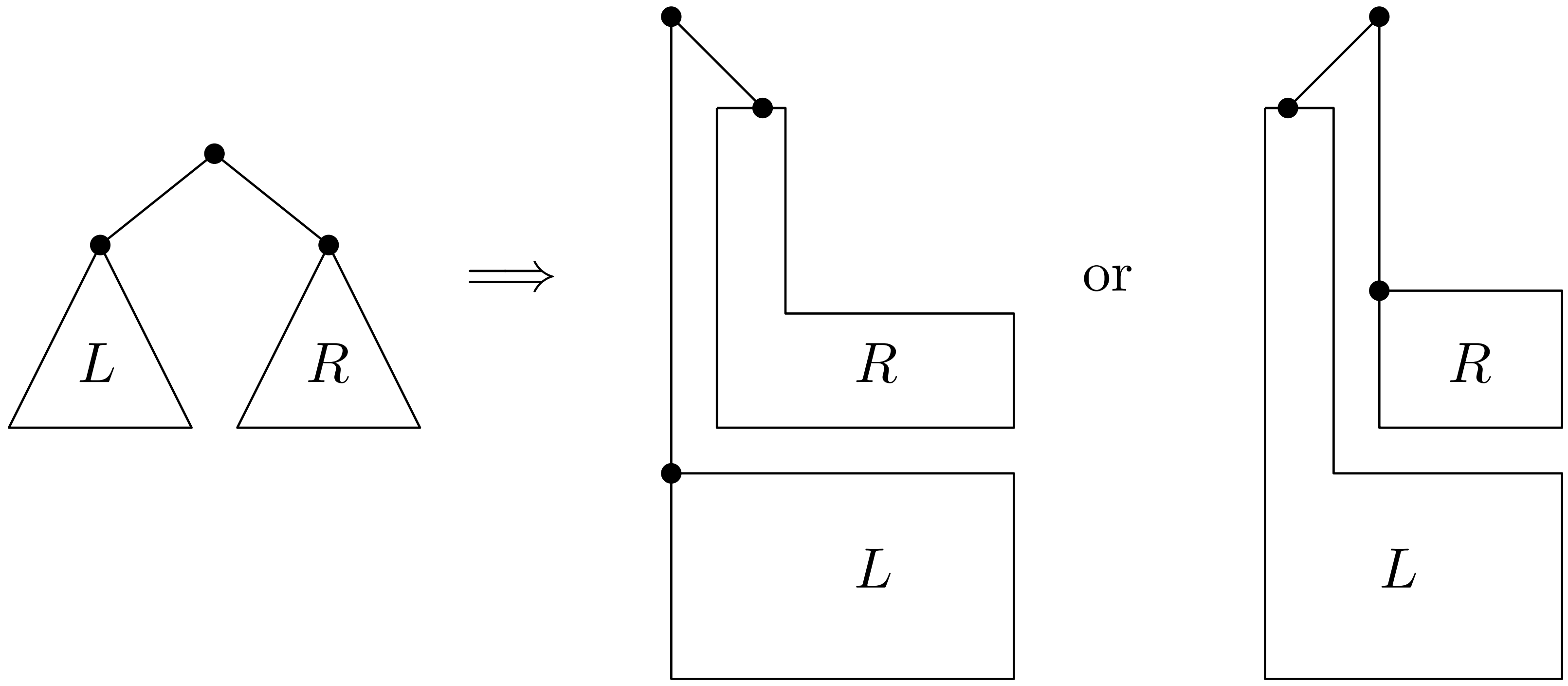
Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

"Narrow-neck" alg'm $(*)$:



Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

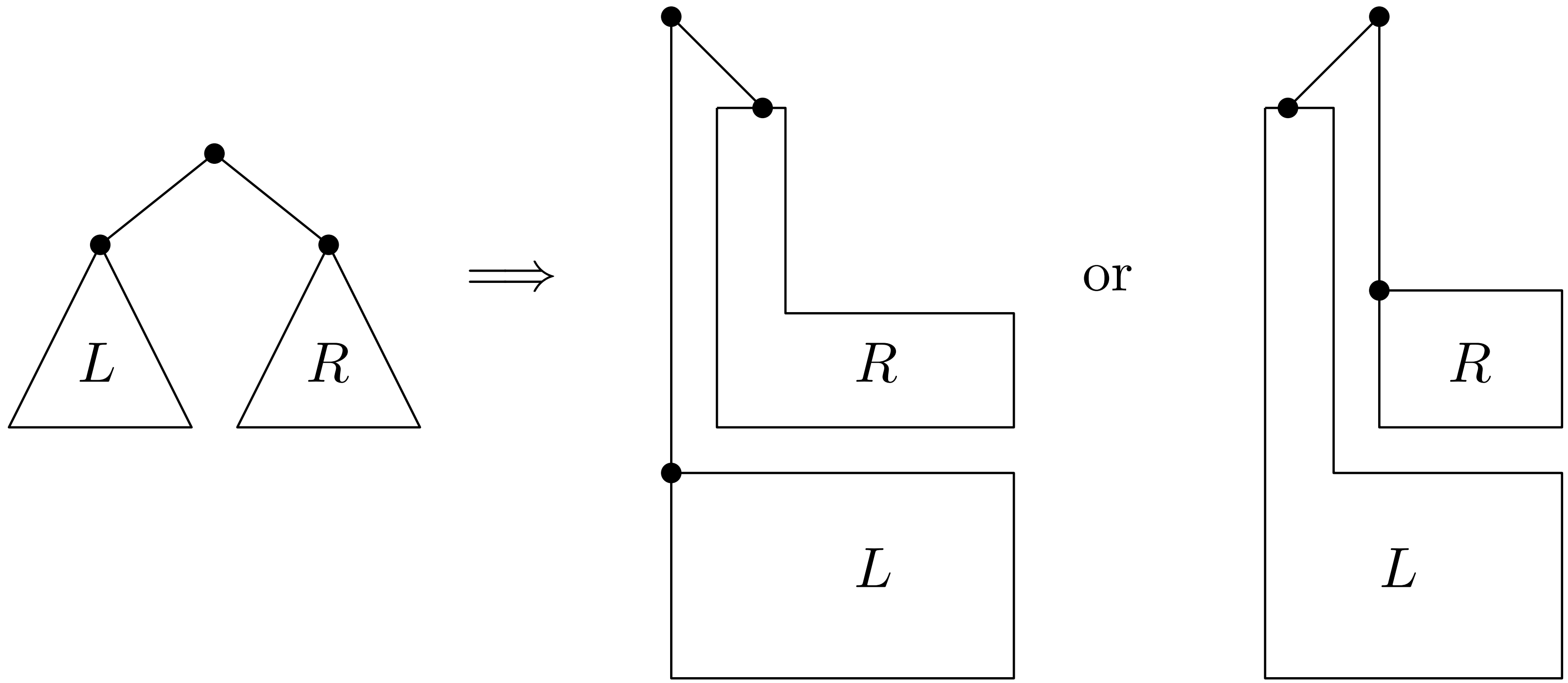
"Narrow-neck" alg'm $(*)$:



- if $R \leq L$, left option, else right option

Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

"Narrow-neck" alg'm $(*)$:

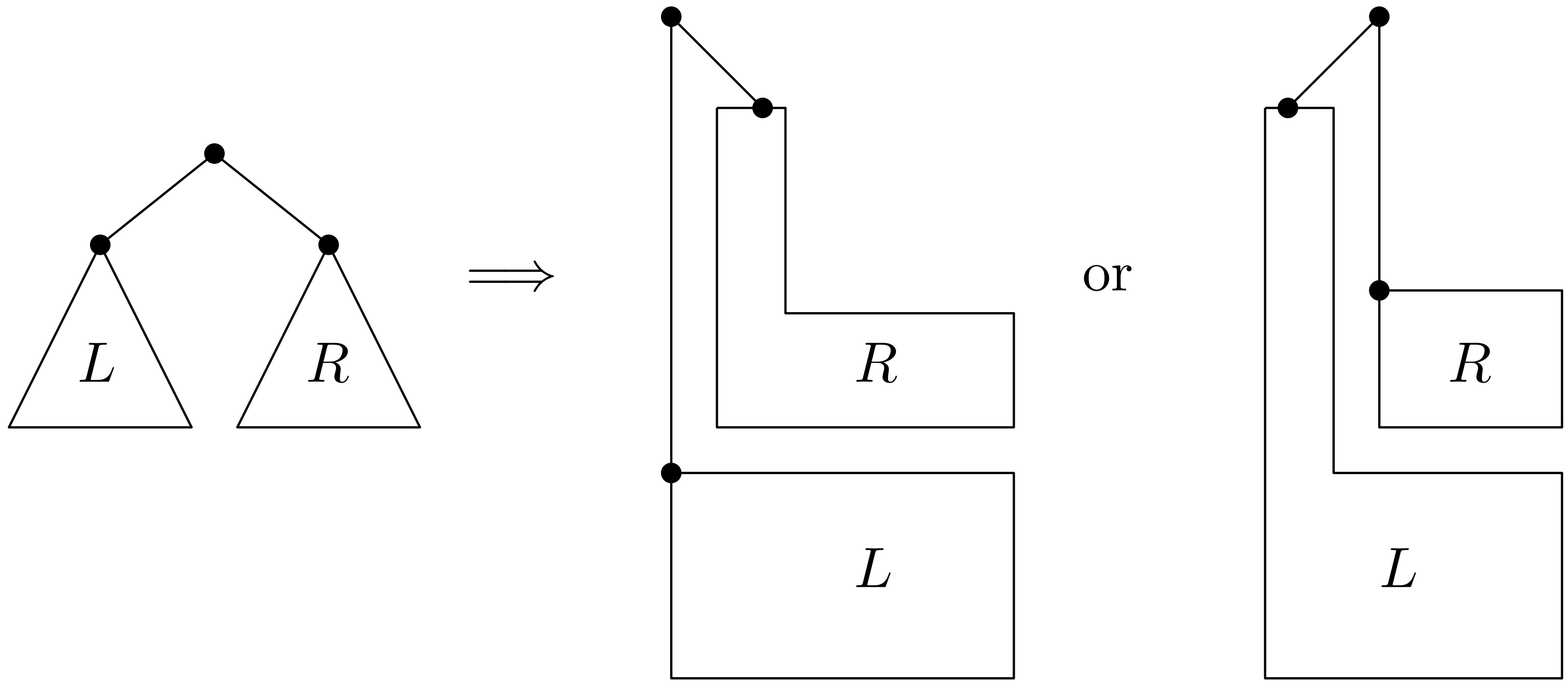


- if $R \leq L$, left option, else right option

$$W^*_{\text{neck}}(n) \;\leq\; W^*_{\text{neck}}(n/2) + O(1)$$

Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

"Narrow-neck" alg'm $(*)$:

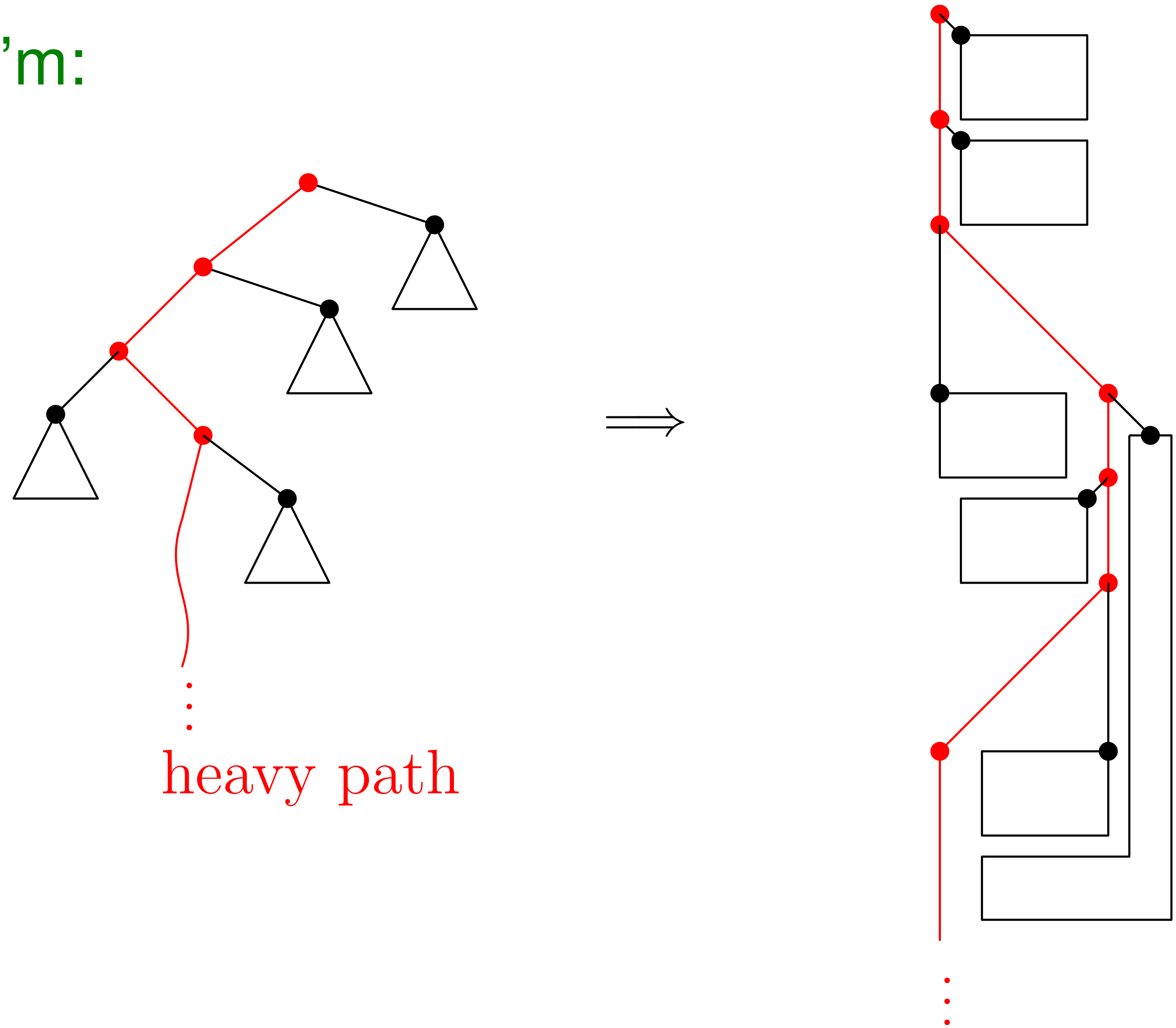


- if $R \leq L$, left option, else right option

$$W^*_{\text{neck}}(n) \leq W^*_{\text{neck}}(n/2) + O(1)$$
$$W^*_{\text{full}}(n) \leq W(n) + W^*_{\text{neck}}(n/2) + O(1)$$

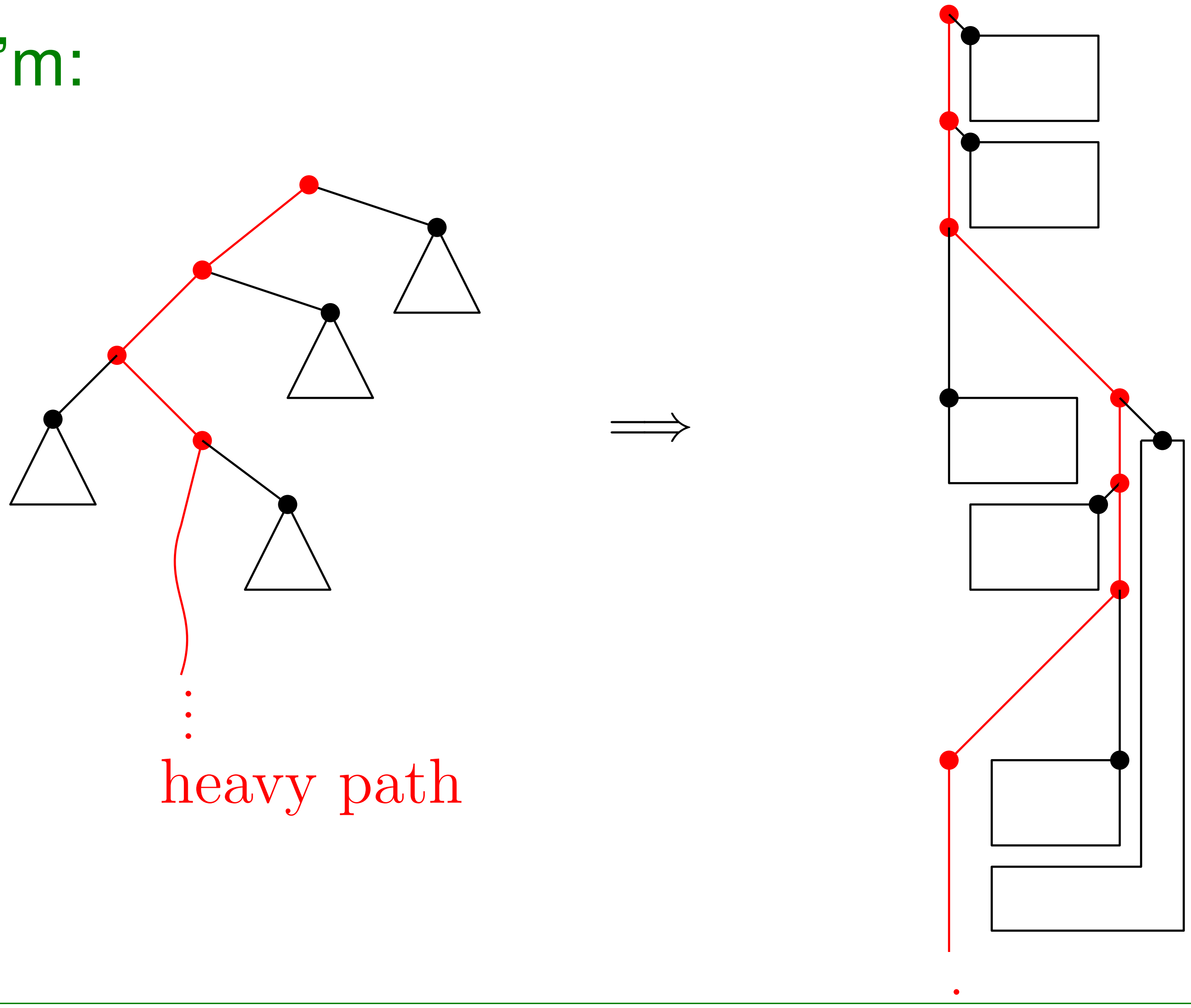
Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

Main alg'm:



Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

Main alg'm:



$$W(n) \ \leq \ \max\{W(n/2), W^*_{\text{full}}(n/2)\} + O(W^*_{\text{neck}}(n/2)) + O(1)$$

Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

$$
\begin{aligned}
W^*_{\text{neck}}(n) &\leq W^*_{\text{neck}}(n/2) + O(1) \\
W^*_{\text{full}}(n) &\leq W(n) + W^*_{\text{neck}}(n/2) + O(1) \\
W(n) &\leq \max\{W(n/2), W^*_{\text{full}}(n/2)\} + O(W^*_{\text{neck}}(n/2)) + O(1)
\end{aligned}
$$

Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

$$
\begin{aligned}
W^*_{\text{neck}}(n) &\leq W^*_{\text{neck}}(n/2) + O(1) \\
W^*_{\text{full}}(n) &\leq W(n) + W^*_{\text{neck}}(n/2) + O(1) \\
W(n) &\leq \max\{W(n/2), W^*_{\text{full}}(n/2)\} + O(W^*_{\text{neck}}(n/2)) + O(1)
\end{aligned}
$$

$$\Rightarrow \; W^*_{\text{neck}}(n) = O(\log n)$$

Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

$$\begin{aligned}
W^*_{\text{neck}}(n) &\leq W^*_{\text{neck}}(n/2) + O(1) \\
W^*_{\text{full}}(n) &\leq W(n) + W^*_{\text{neck}}(n/2) + O(1) \\
W(n) &\leq \max\{W(n/2), W^*_{\text{full}}(n/2)\} + O(W^*_{\text{neck}}(n/2)) + O(1)
\end{aligned}$$

$$\Rightarrow \; W^*_{\text{neck}}(n) = O(\log n)$$

$$\Rightarrow \; W(n) \leq W(n/2) + O(\log n)$$

Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

$$
\begin{aligned}
W^*_{\text{neck}}(n) &\leq W^*_{\text{neck}}(n/2) + O(1) \\
W^*_{\text{full}}(n) &\leq W(n) + W^*_{\text{neck}}(n/2) + O(1) \\
W(n) &\leq \max\{W(n/2), W^*_{\text{full}}(n/2)\} + O(W^*_{\text{neck}}(n/2)) + O(1)
\end{aligned}
$$

$\Rightarrow\ W^*_{\text{neck}}(n) = O(\log n)$

$\Rightarrow\ W(n) \leq W(n/2) + O(\log n)$

$\Rightarrow\ W(n) = \boxed{O(\log^2 n)}$ width

Ex: binary, octilinear, strict upw., *ordered* [Biedl'17]

$$\begin{aligned} W^*_{\text{neck}}(n) &\leq W^*_{\text{neck}}(n/2) + O(1) \\ W^*_{\text{full}}(n) &\leq W(n) + W^*_{\text{neck}}(n/2) + O(1) \\ W(n) &\leq \max\{W(n/2), W^*_{\text{full}}(n/2)\} + O(W^*_{\text{neck}}(n/2)) + O(1) \end{aligned}$$

$\Rightarrow \; W^*_{\text{neck}}(n) = O(\log n)$

$\Rightarrow \; W(n) \leq W(n/2) + O(\log n)$

$\Rightarrow \; W(n) = \boxed{O(\log^2 n)}$ width (open: single log?)

Technique 5: Height–Width Tradeoff

Ex: binary, orthogonal

[C.–Goodrich–Kosaraju–Tamassia'96/Shin–Kim–Chwa'96]

Ex: binary, orthogonal

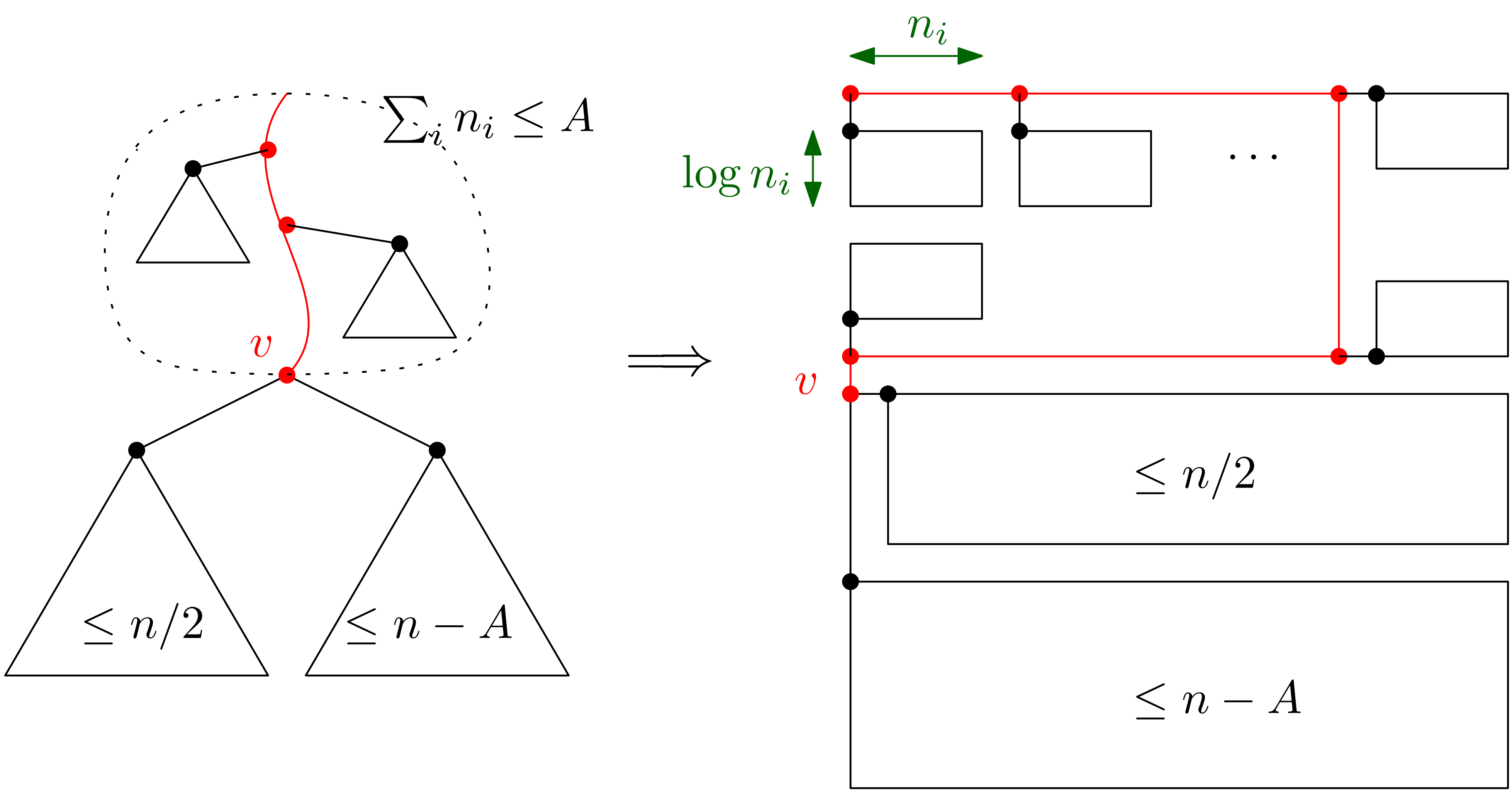
[C.–Goodrich–Kosaraju–Tamassia’96/Shin–Kim–Chwa’96]

skewed centroid again!

Ex: binary, orthogonal

[C.–Goodrich–Kosaraju–Tamassia'96/Shin–Kim–Chwa'96]

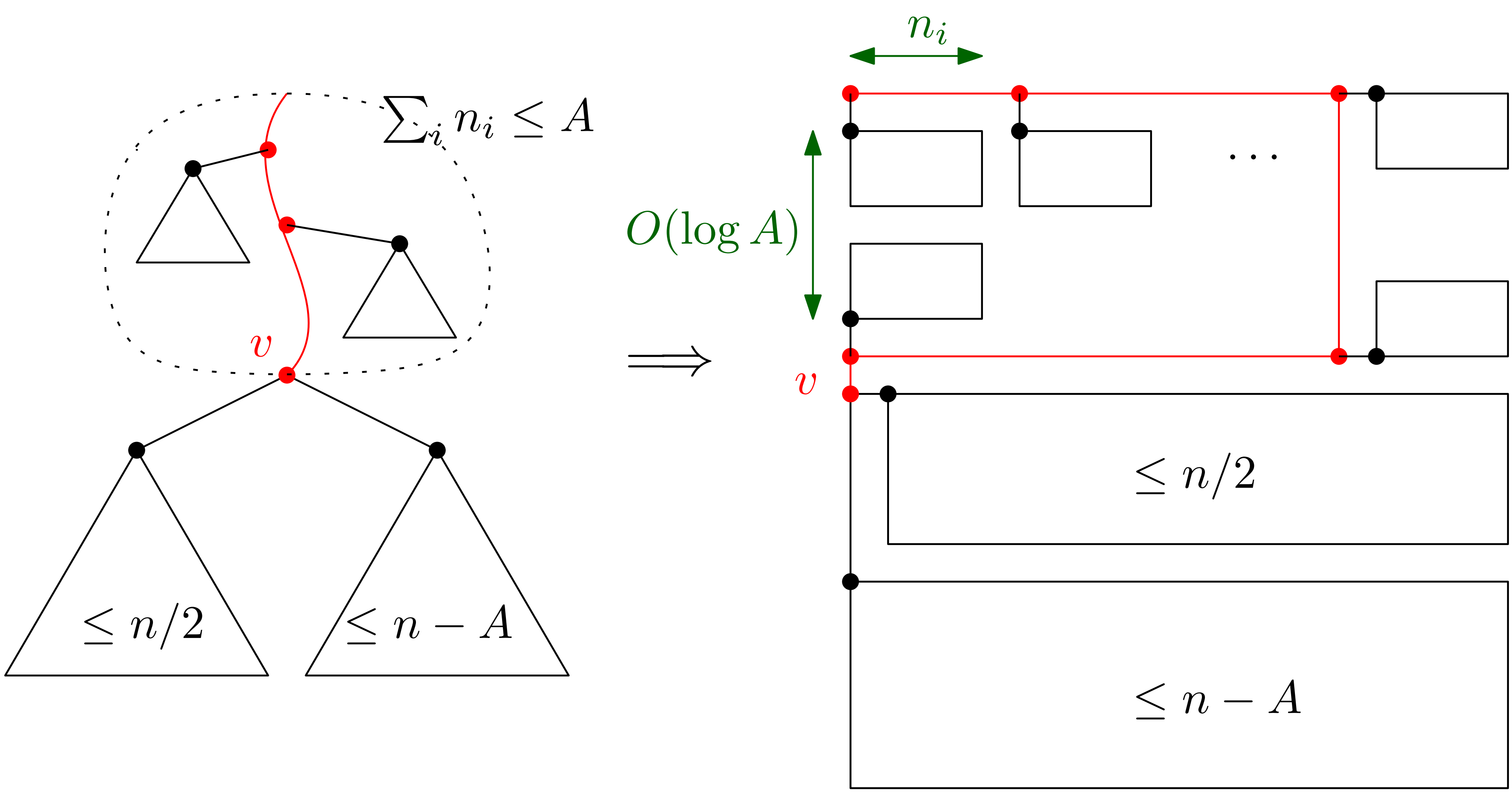
skewed centroid again!



Ex: binary, orthogonal

[C.–Goodrich–Kosaraju–Tamassia'96/Shin–Kim–Chwa'96]

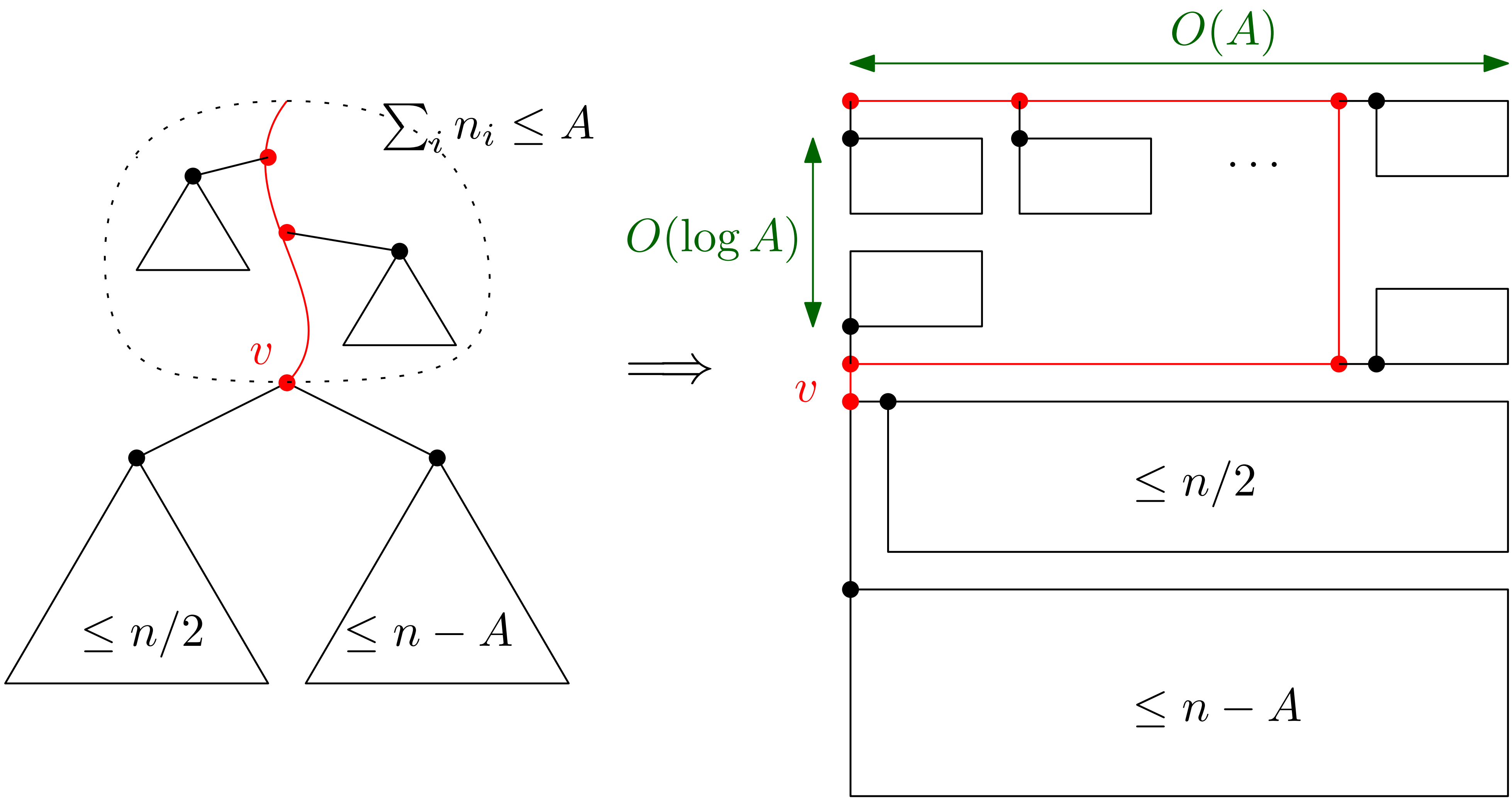
skewed centroid again!



Ex: binary, orthogonal

[C.–Goodrich–Kosaraju–Tamassia'96/Shin–Kim–Chwa'96]

skewed centroid again!



Ex: binary, orthogonal

[C.–Goodrich–Kosaraju–Tamassia'96/Shin–Kim–Chwa'96]

$$W(n) \le \max\{W(n/2) + O(1),\ O(A)\}$$

$$H(n) \le \max_{\substack{L+R \le n \\ L,R \le n-A}} (H(L) + H(R)) + O(\log A)$$

Ex: binary, orthogonal

[C.–Goodrich–Kosaraju–Tamassia'96/Shin–Kim–Chwa'96]

$$W(n) \leq \max\{W(n/2) + O(1),\ O(A)\}$$

$$H(n) \leq \max_{\substack{L+R\leq n \\ L,R\leq n-A}} (H(L) + H(R)) + O(\log A)$$

$$\Rightarrow \quad W(n) = O(A + \log n)$$

$$H(n) = O(\lceil n/A \rceil \log A)$$

Ex: binary, orthogonal

[C.–Goodrich–Kosaraju–Tamassia'96/Shin–Kim–Chwa'96]

$$W(n) \le \max\{W(n/2) + O(1),\ O(A)\}$$

$$H(n) \le \max_{\substack{L+R\le n \\ L,R\le n-A}} (H(L) + H(R)) + O(\log A)$$

$$\Rightarrow \begin{array}{l} W(n) = O(A + \log n) \\ H(n) = O(\lceil n/A \rceil \log A) \end{array}$$

set $A = \log n \Rightarrow$ $O(n \log \log n)$ area

Ex: binary, orthogonal

[C.–Goodrich–Kosaraju–Tamassia'96/Shin–Kim–Chwa'96]

$$W(n) \le \max\{W(n/2) + O(1),\ O(A)\}$$

$$H(n) \le \max_{\substack{L+R \le n \\ L,R \le n-A}} (H(L) + H(R)) + O(\log A)$$

$$\Rightarrow \begin{aligned} W(n) &= O(A + \log n) \\ H(n) &= O(\lceil n/A \rceil \log A) \end{aligned}$$

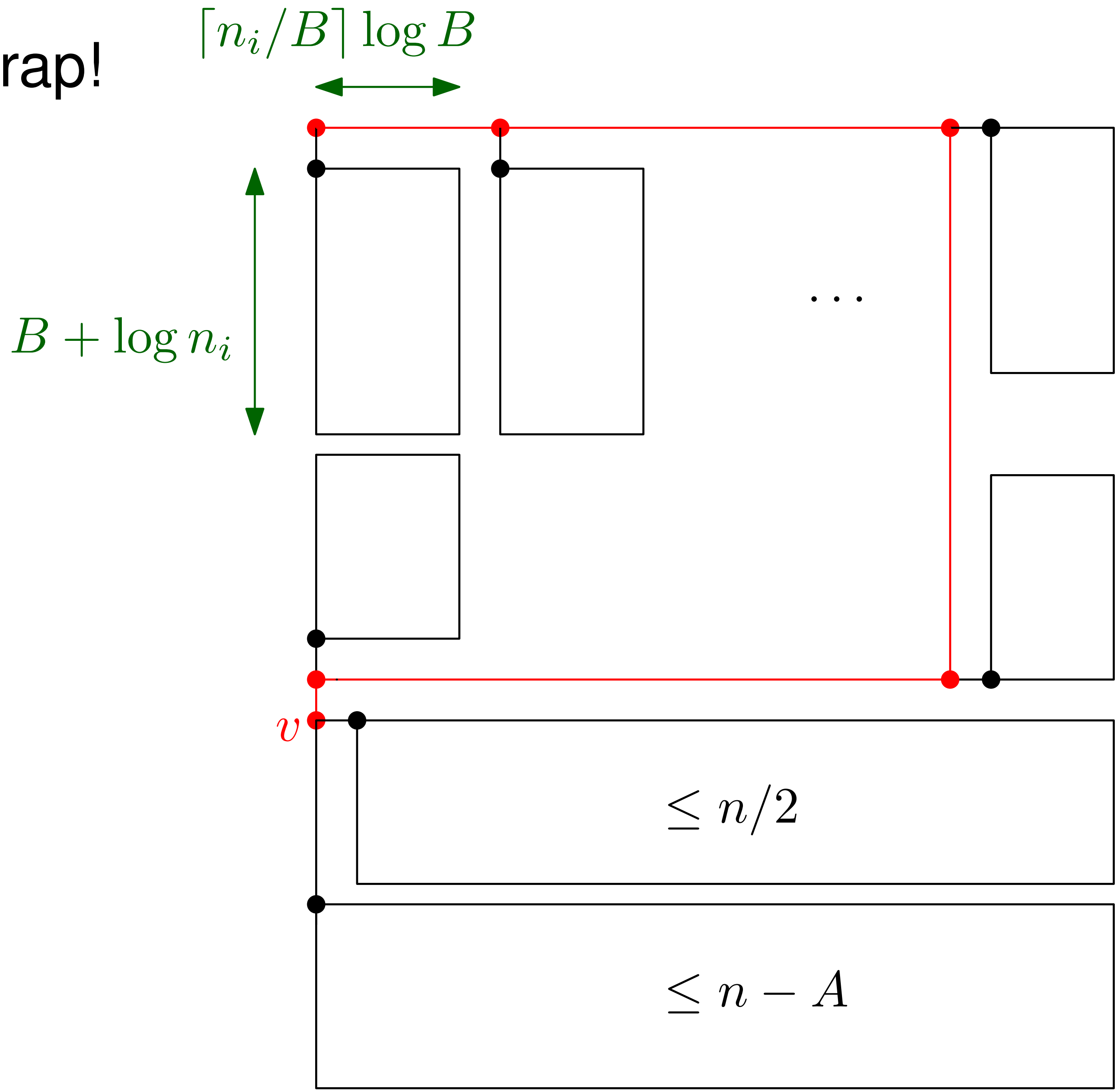
set $A = \log n \Rightarrow$ $O(n \log \log n)$ area better?

Ex: binary, orthogonal [new]

Idea: bootstrap!

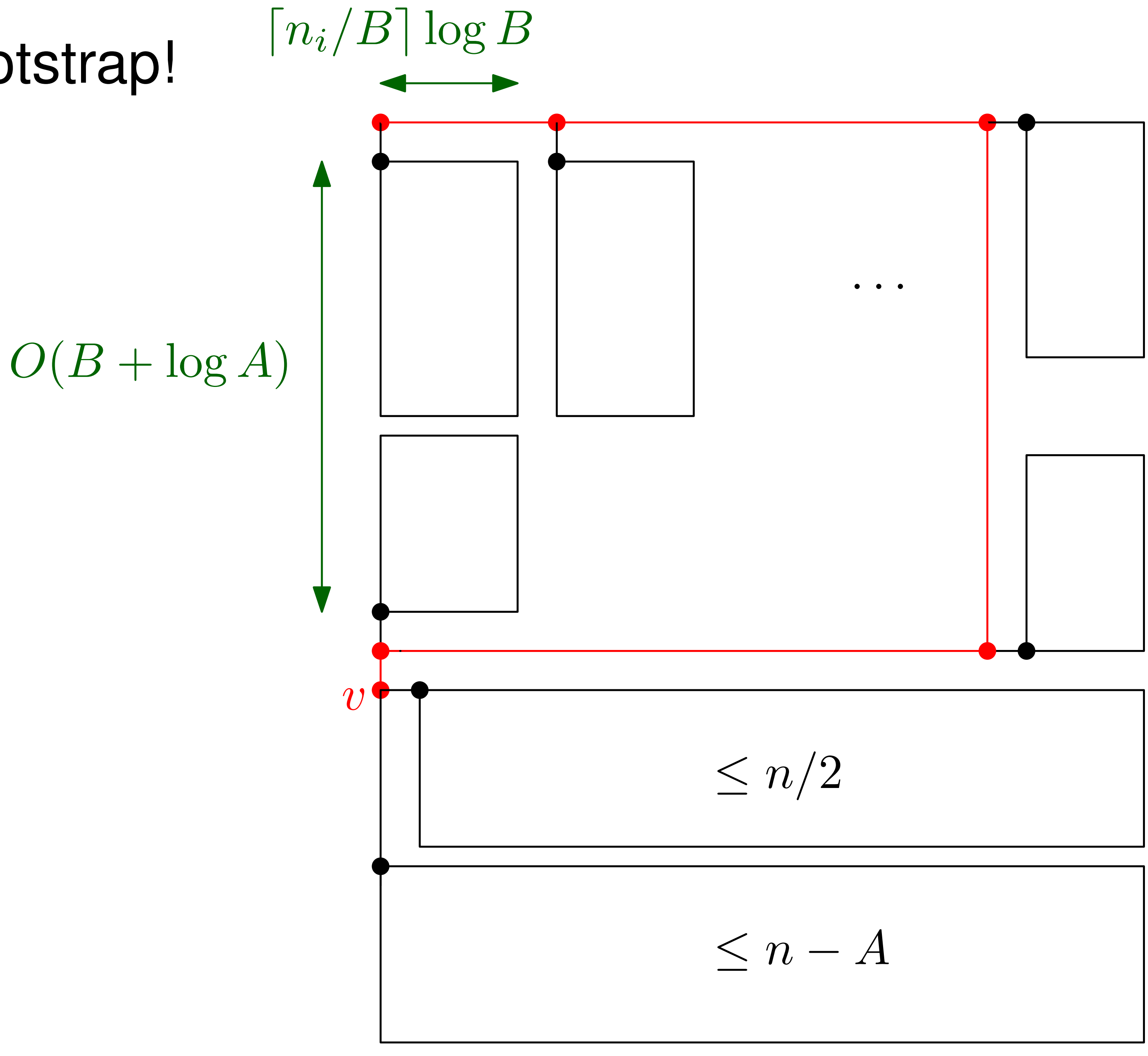
Ex: binary, orthogonal [new]

Idea: bootstrap!



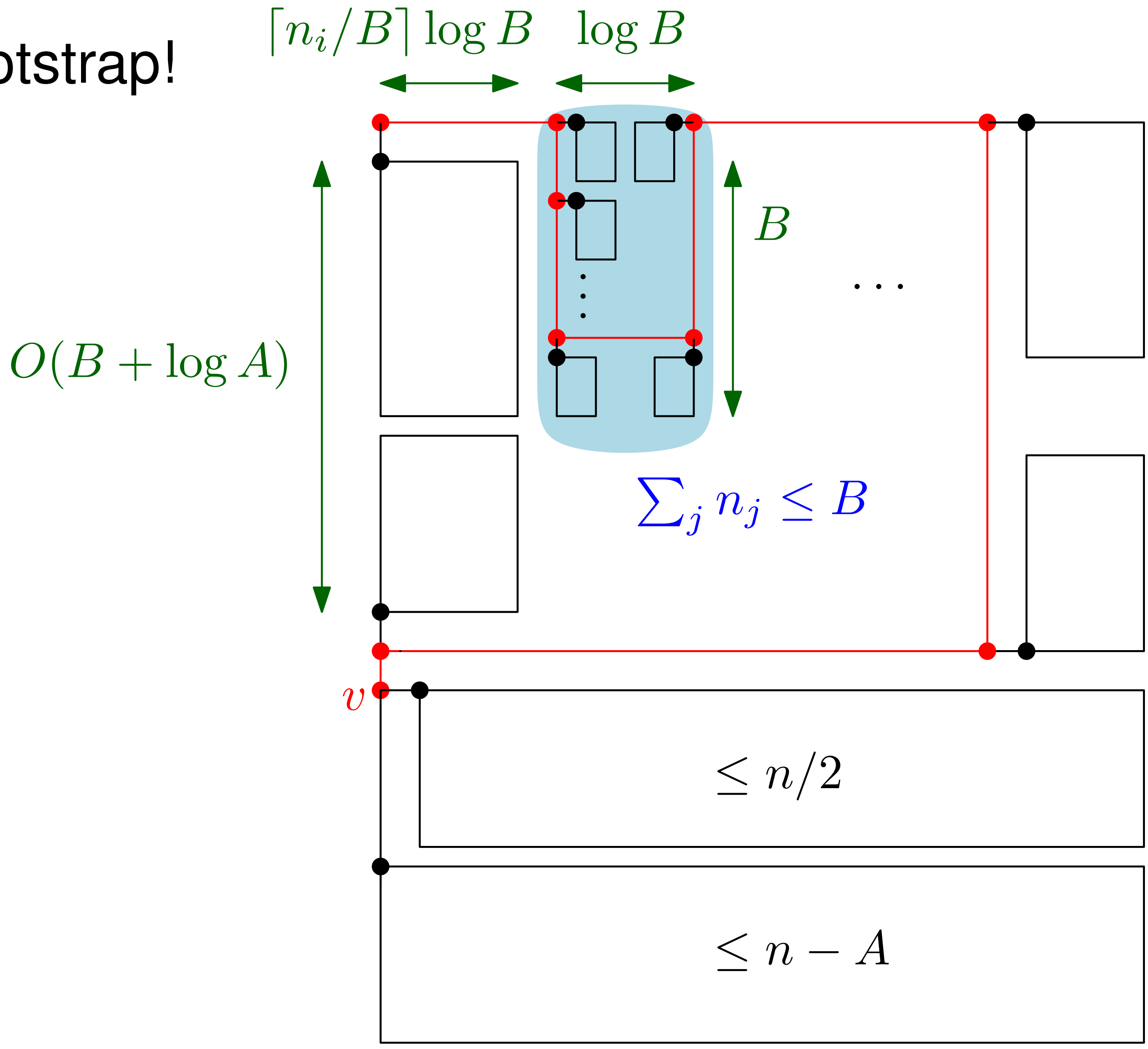
Ex: binary, orthogonal [new]

Idea: bootstrap!



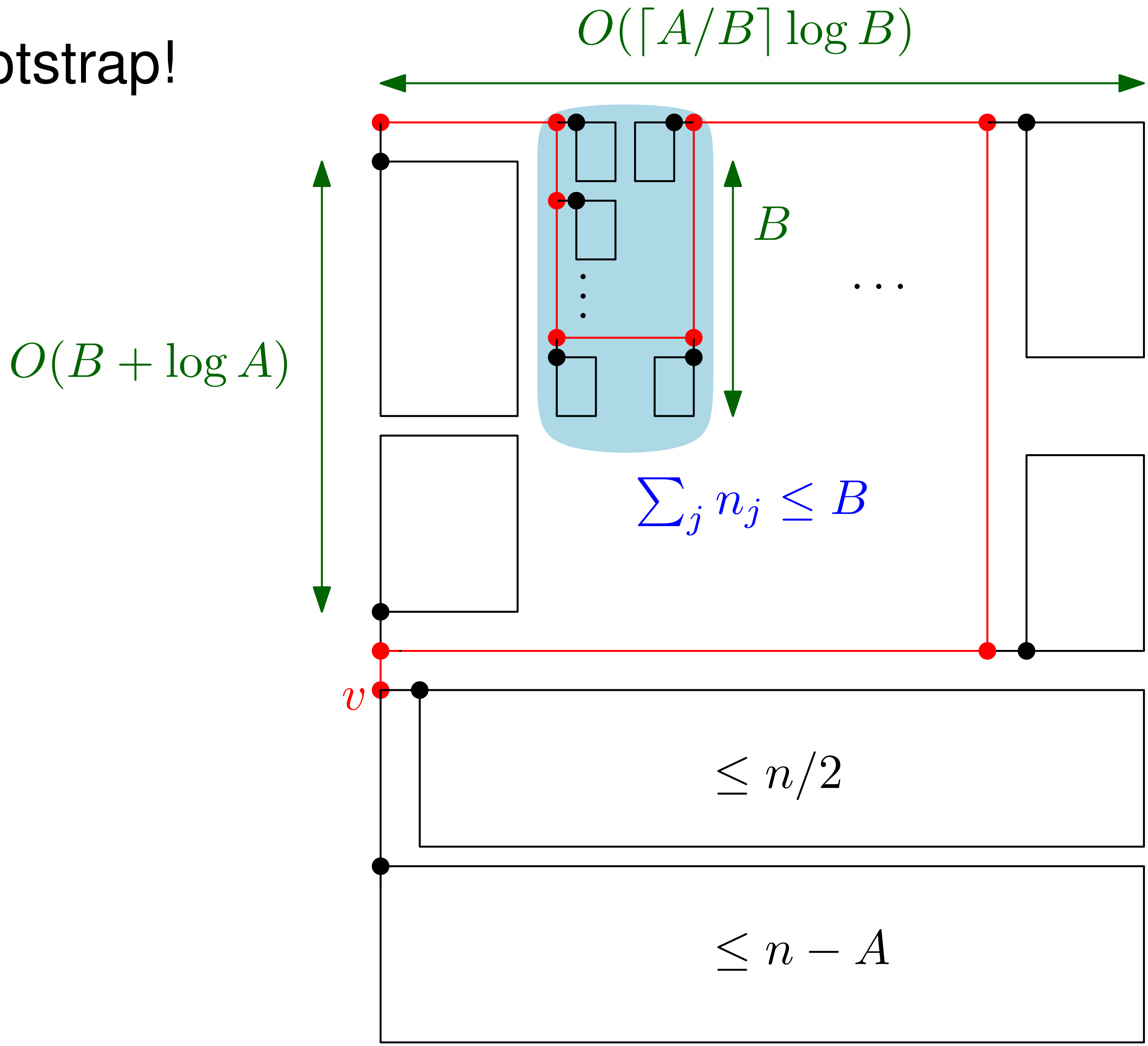
Ex: binary, orthogonal [new]

Idea: bootstrap!



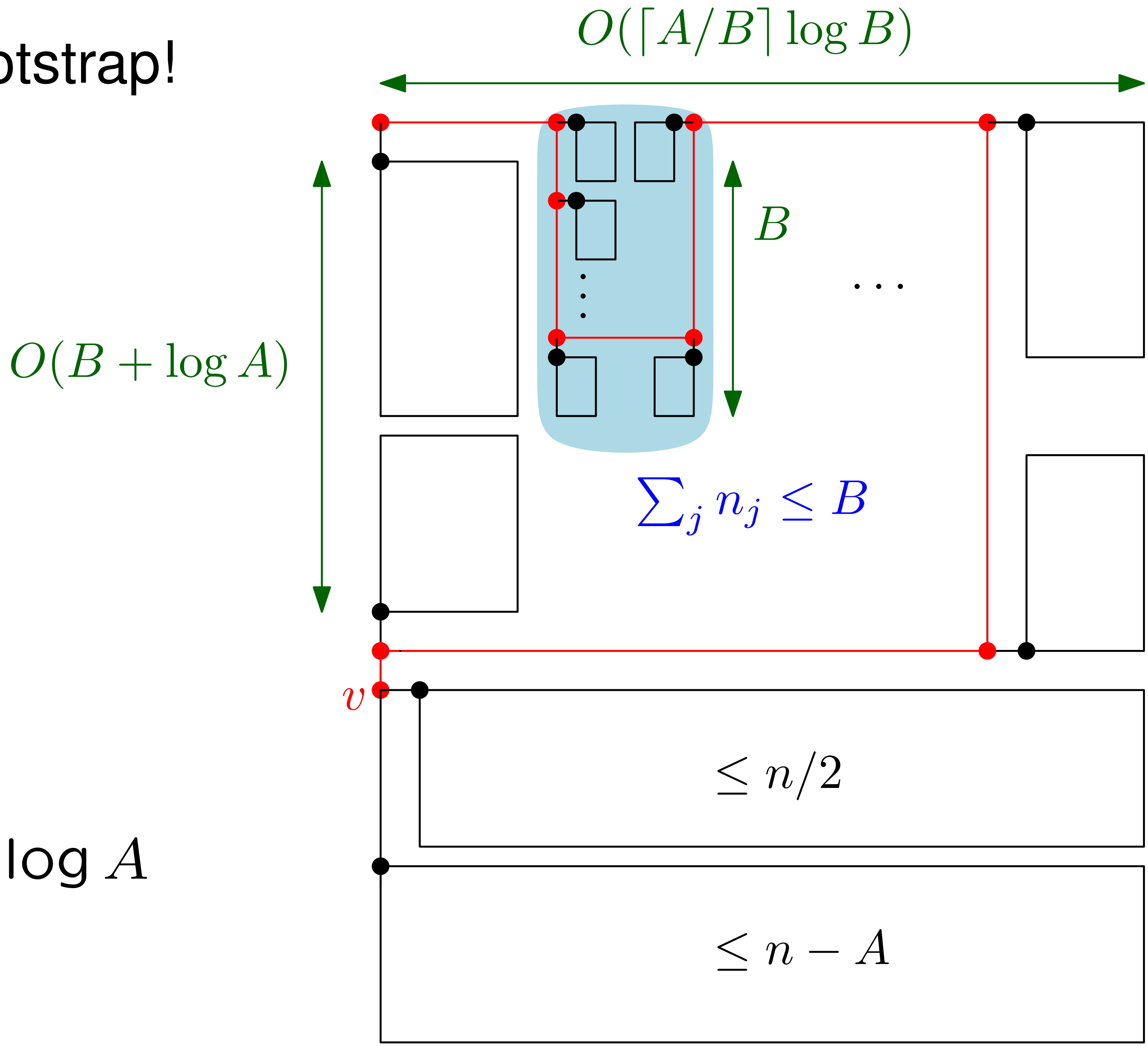
Ex: binary, orthogonal [new]

Idea: bootstrap!



Ex: binary, orthogonal [new]

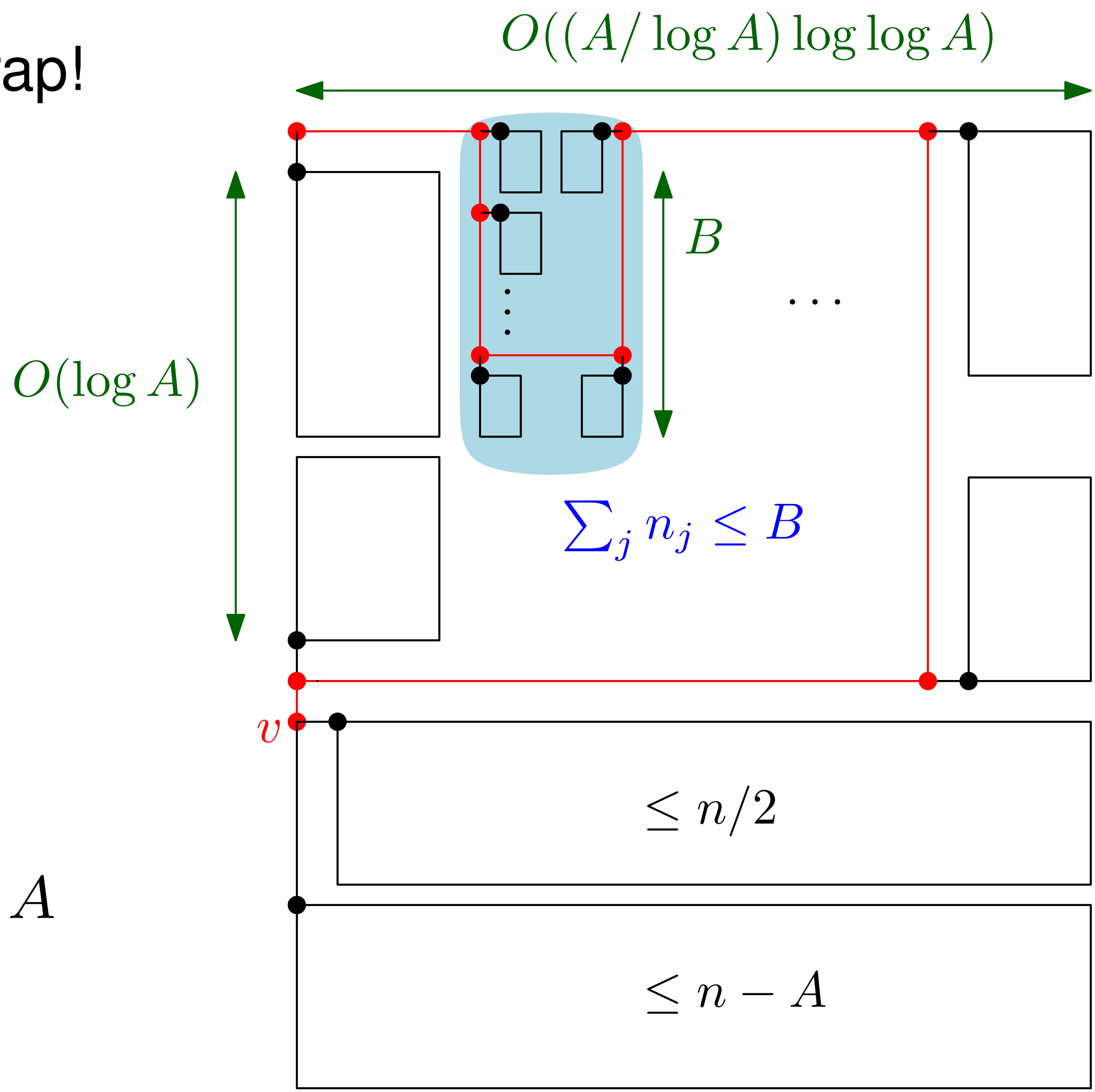
Idea: bootstrap!



set $B = \log A$

Ex: binary, orthogonal [new]

Idea: bootstrap!



set $B = \log A$

Ex: binary, orthogonal [new]

$$W(n) \leq \max\{W(n/2) + O(1),\ O((A/\log A)\log\log A)\}$$

$$H(n) \leq \max_{\substack{L+R\leq n \\ L,R\leq n-A}} (H(L) + H(R)) + O(\log A)$$

Ex: binary, orthogonal [new]

$$\begin{aligned} W(n) &\le \max\{W(n/2) + O(1),\ O((A/\log A)\log\log A)\} \\ H(n) &\le \max_{\substack{L+R\le n \\ L,R\le n-A}} (H(L) + H(R)) + O(\log A) \end{aligned}$$

$$\Rightarrow \begin{aligned} W(n) &= O((A/\log A)\log\log A + \log n) \\ H(n) &= O(\lceil n/A \rceil \log A) \end{aligned}$$

Ex: binary, orthogonal [new]

$$\begin{aligned} W(n) &\le \max\{W(n/2) + O(1),\ O((A/\log A)\log\log A)\} \\ H(n) &\le \max_{\substack{L+R\le n \\ L,R\le n-A}} (H(L)+H(R)) + O(\log A) \end{aligned}$$

$$\Rightarrow \begin{aligned} W(n) &= O((A/\log A)\log\log A + \log n) \\ H(n) &= O(\lceil n/A \rceil \log A) \end{aligned}$$

set $A = \text{polylog}\, n \Rightarrow$ $O(n \log\log\log n)$ area

Ex: binary, orthogonal [new]

$$\begin{aligned} W(n) &\le \max\{W(n/2) + O(1),\ O((A/\log A)\log\log A)\} \\ H(n) &\le \max_{\substack{L+R\le n \\ L,R\le n-A}} (H(L) + H(R)) + O(\log A) \end{aligned}$$

$$\Rightarrow \begin{aligned} W(n) &= O((A/\log A)\log\log A + \log n) \\ H(n) &= O(\lceil n/A \rceil \log A) \end{aligned}$$

set $A = \text{polylog}\, n \ \Rightarrow$ $O(n \log\log\log n)$ area

bootstrap again $\Rightarrow$ $O(n \log\log\log\log n)$ area

Ex: binary, orthogonal [new]

$$
\begin{aligned}
W(n) &\le \max\{W(n/2) + O(1),\ O((A/\log A)\log\log A)\} \\
H(n) &\le \max_{\substack{L+R \le n \\ L,R \le n-A}} (H(L) + H(R)) + O(\log A)
\end{aligned}
$$

$$
\Rightarrow \boxed{\begin{aligned}
W(n) &= O((A/\log A)\log\log A + \log n) \\
H(n) &= O(\lceil n/A \rceil \log A)
\end{aligned}}
$$

set $A = \text{polylog}\, n \ \Rightarrow \boxed{O(n \log\log\log n)}$ area

bootstrap again $\Rightarrow \boxed{O(n \log\log\log\log\log n)}$ area

Ex: binary, orthogonal [new]

$$W(n) \le \max\{W(n/2) + O(1),\ O((A/\log A)\log\log A)\}$$

$$H(n) \le \max_{\substack{L+R\le n \\ L,R\le n-A}} (H(L) + H(R)) + O(\log A)$$

$$\Rightarrow \begin{array}{l} W(n) = O((A/\log A)\log\log A + \log n) \\ H(n) = O(\lceil n/A\rceil \log A) \end{array}$$

set $A = \text{polylog}\, n \ \Rightarrow\ O(n \log\log\log n)$ area

bootstrap again $\Rightarrow\ O(n \log\log\log\log\log\log n)$ area

Ex: binary, orthogonal [new]

$$\begin{aligned} W(n) &\le \max\{W(n/2) + O(1),\ O((A/\log A)\log\log A)\} \\ H(n) &\le \max_{\substack{L+R\le n \\ L,R\le n-A}} (H(L) + H(R)) + O(\log A) \end{aligned}$$

$$\Rightarrow \boxed{\begin{aligned} W(n) &= O((A/\log A)\log\log A + \log n) \\ H(n) &= O(\lceil n/A \rceil \log A) \end{aligned}}$$

set $A = \text{polylog}\, n \ \Rightarrow$ $\boxed{O(n \log\log\log n)}$ area

bootstrap again $\Rightarrow$ $\boxed{O(nc^{\log^* n})}$ area

Ex: binary, orthogonal [new]

$$W(n) \leq \max\{W(n/2) + O(1),\ O((A/\log A)\log\log A)\}$$
$$H(n) \leq \max_{\substack{L+R\leq n \\ L,R\leq n-A}} (H(L) + H(R)) + O(\log A)$$

$$\Rightarrow \begin{aligned} W(n) &= O((A/\log A)\log\log A + \log n) \\ H(n) &= O(\lceil n/A \rceil \log A) \end{aligned}$$

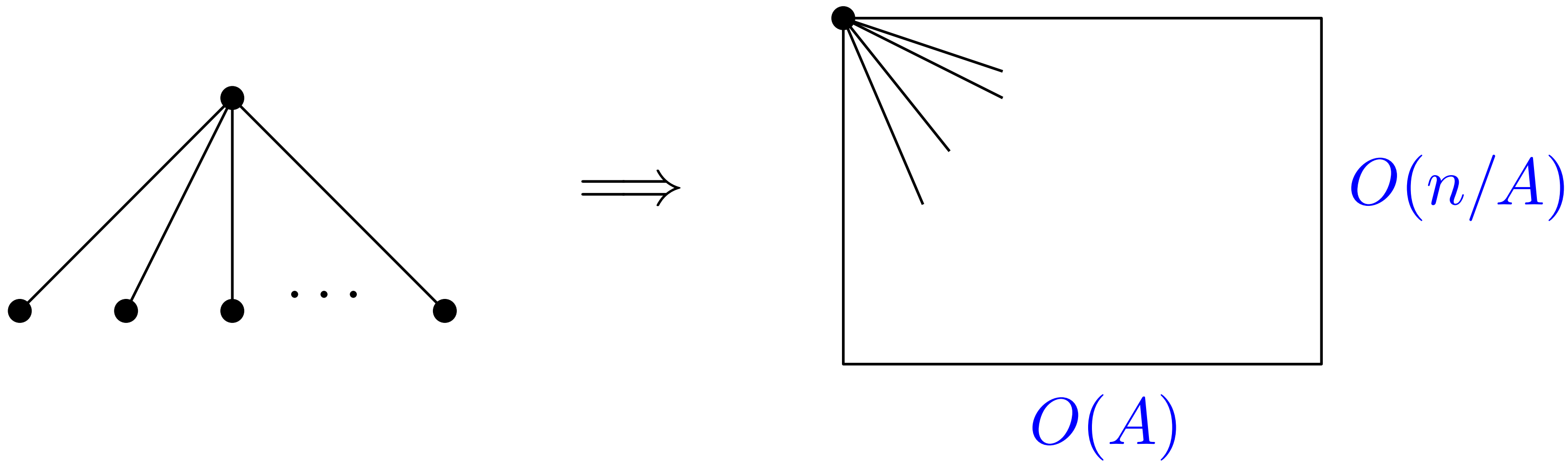
set $A = \text{polylog}\, n \Rightarrow$ $O(n \log\log\log n)$ area

bootstrap again $\Rightarrow$ $O(nc^{\log^* n})$ area (open: linear?)

Last Ex: general trees [new]

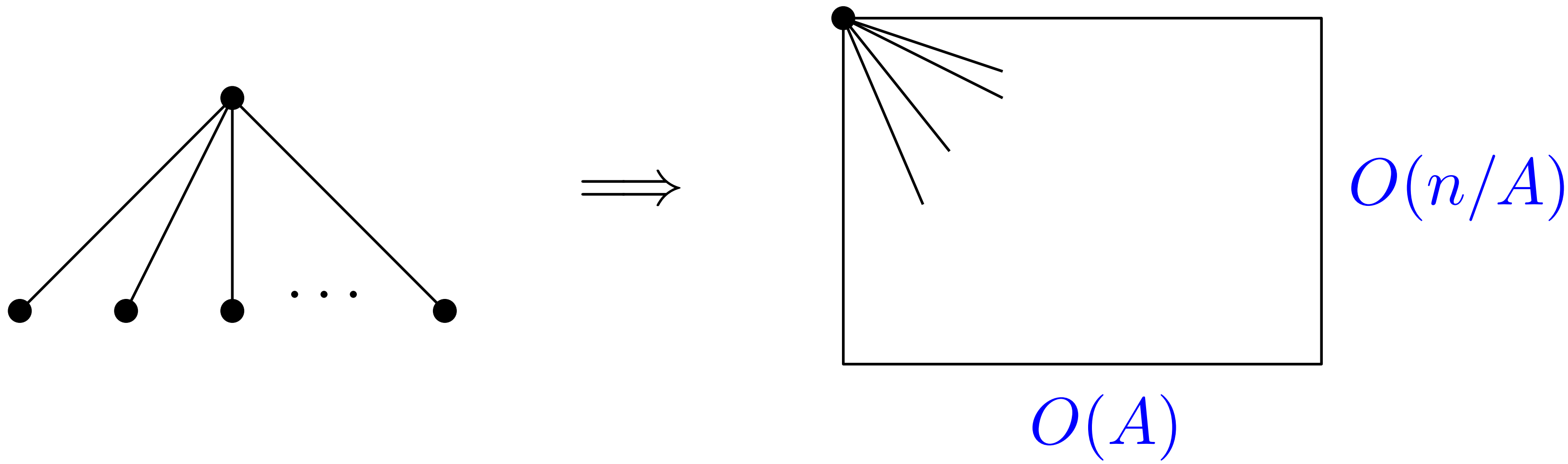
Last Ex: general trees [new]

Lemma: for (almost) any A,



Last Ex: general trees [new]

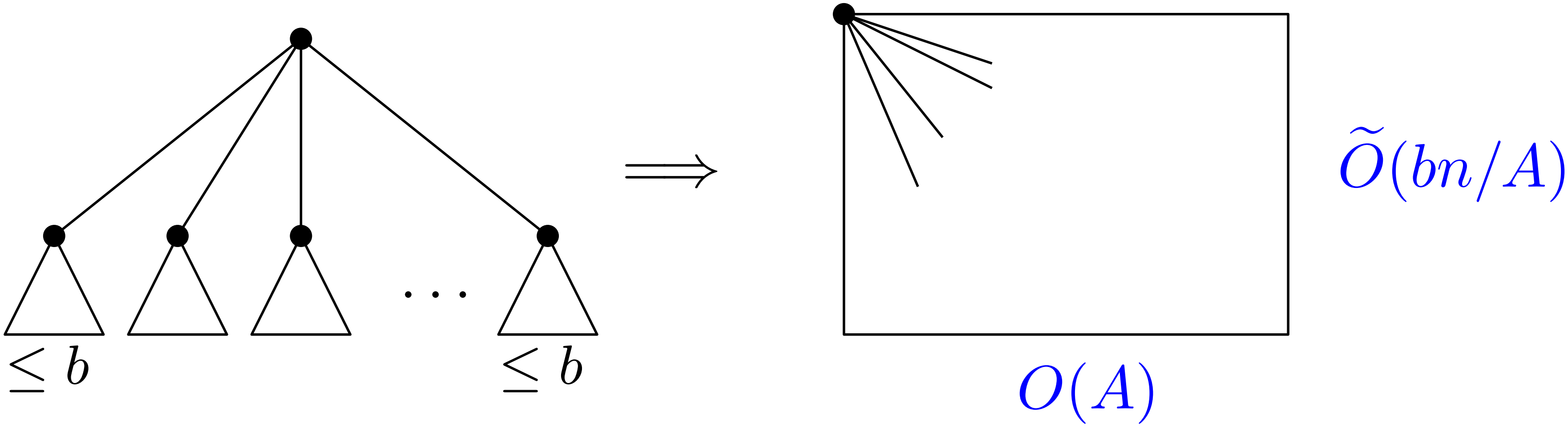
Lemma: for (almost) any A,



Proof: take all points with co-prime (x, y). Q.E.D.

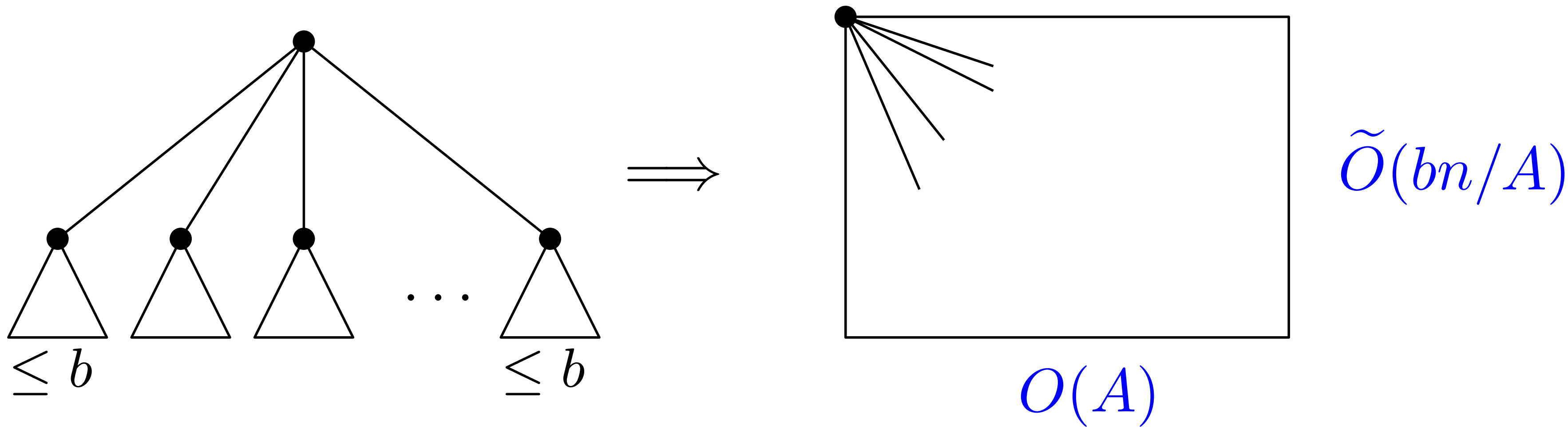
Last Ex: general trees [new]

Lemma: for (almost) any A,



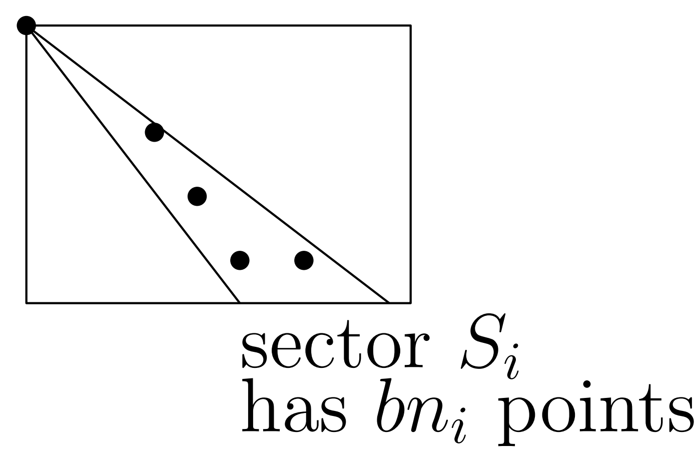
Last Ex: general trees [new]

Lemma: for (almost) any A,

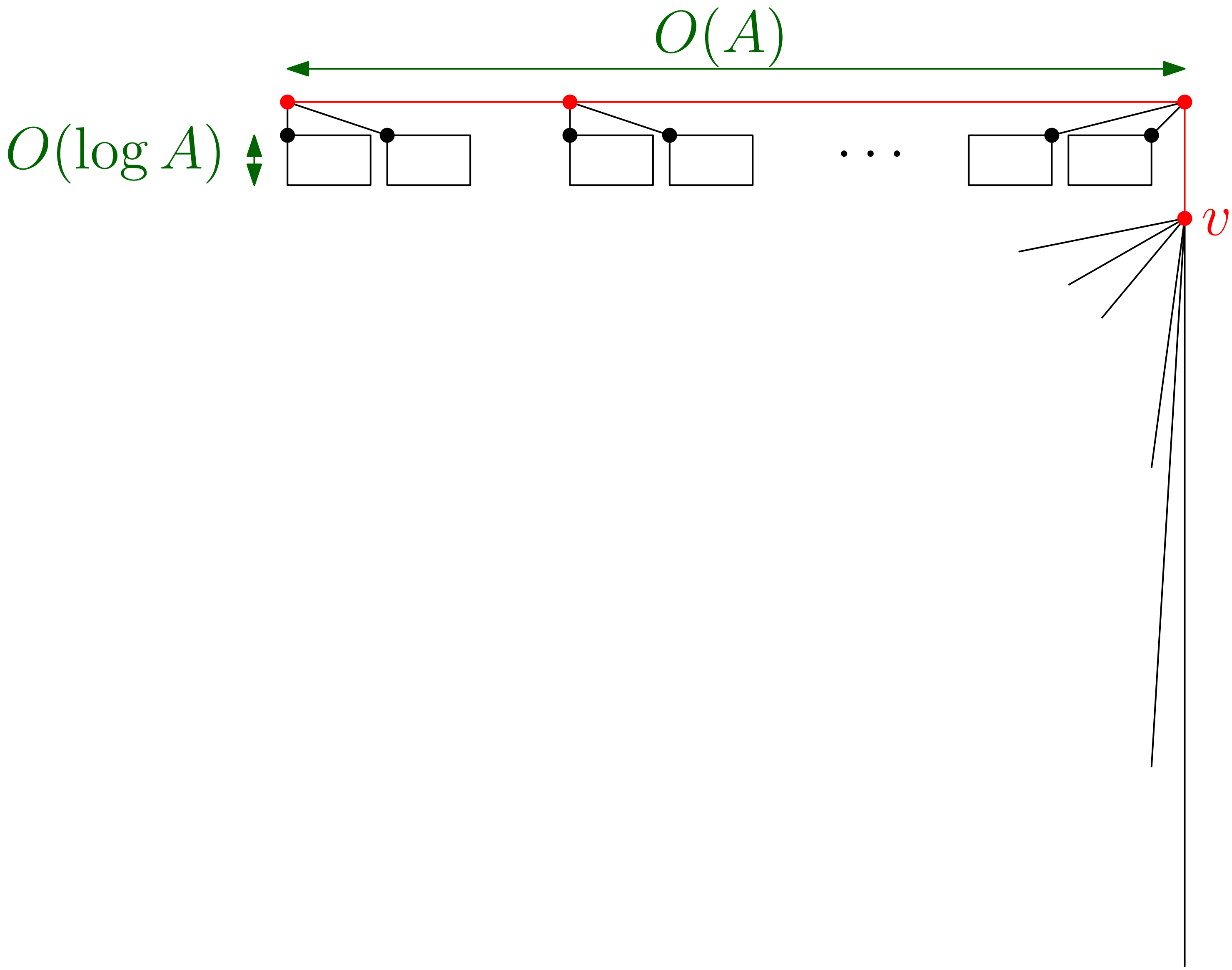


Proof Sketch:

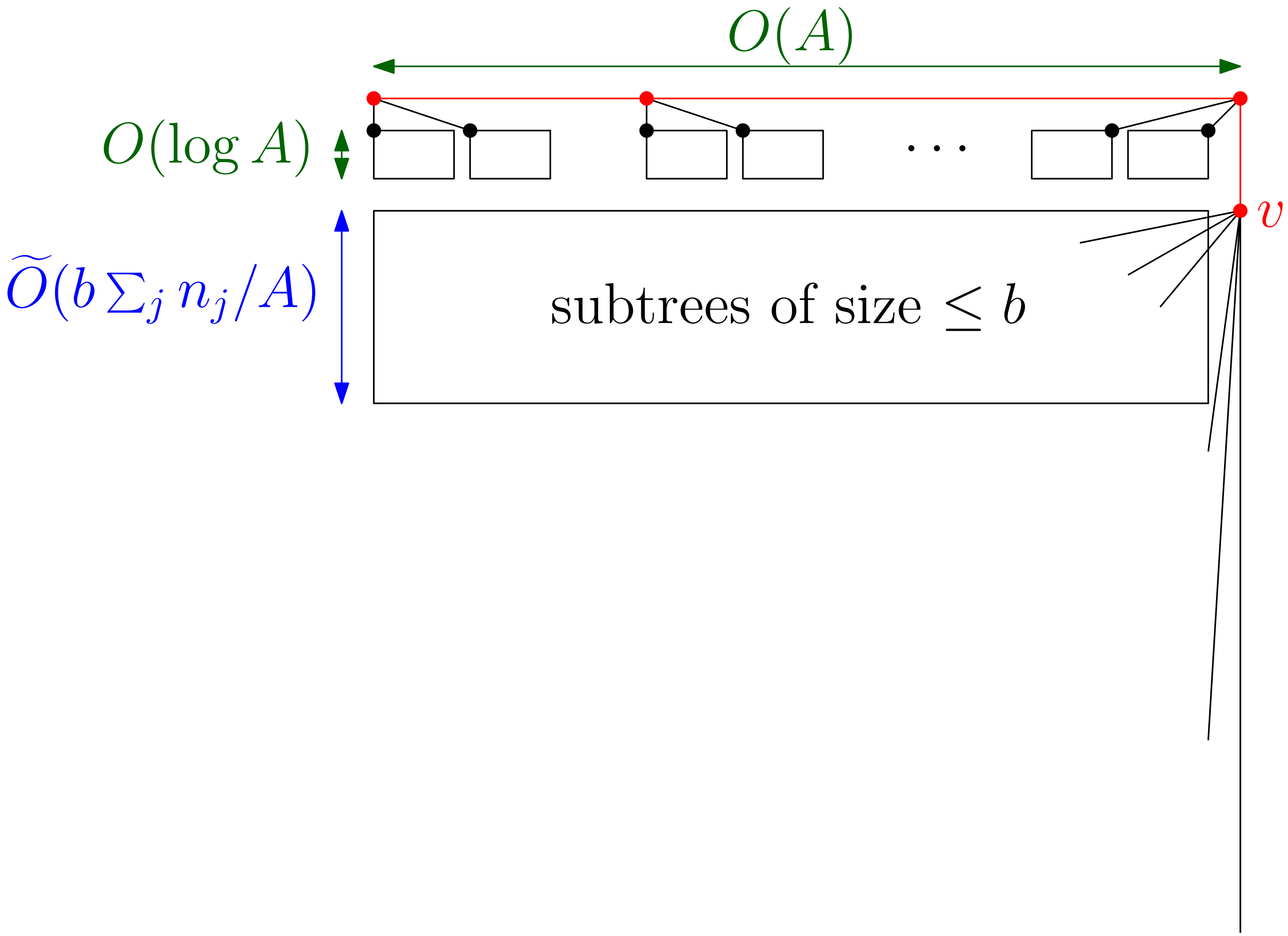
- divide into sectors S_i
- if no b points of S_i lie on a line, ok
- if b points of S_i lie on a line, magnify by factor $\log b$, and simulate drawing on $b \times \log b$ grid



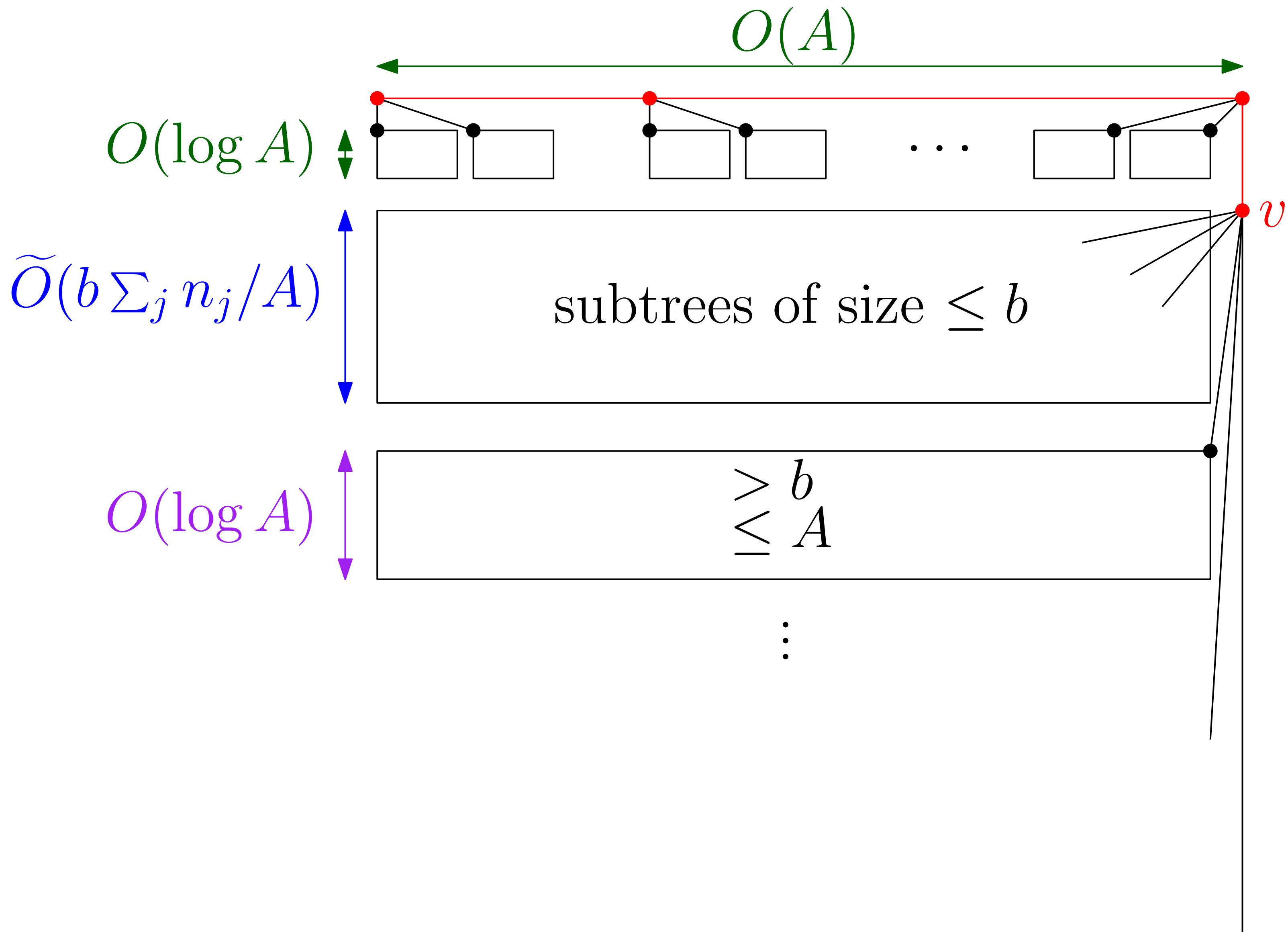
Last Ex: general trees [new]



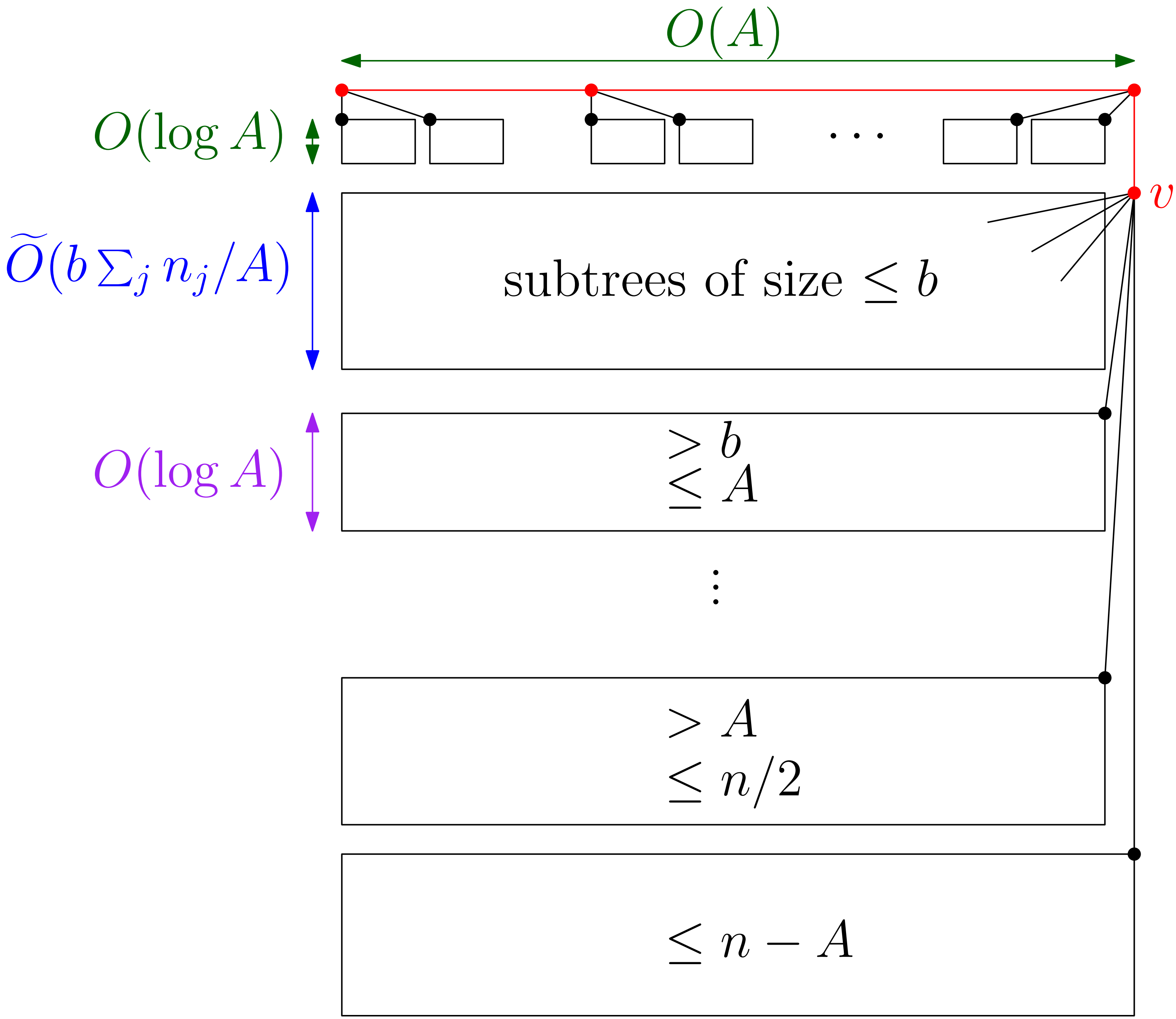
Last Ex: general trees [new]



Last Ex: general trees [new]



Last Ex: general trees [new]



Last Ex: general trees [new]

$$W(n) = O(A + \log n)$$
$$H(n) = \widetilde{O}(\lceil n/A \rceil \log A + bn/A + (n/b) \log A)$$

Last Ex: general trees [new]

$$
\begin{aligned}
W(n) &= O(A + \log n) \\
H(n) &= \widetilde{O}(\lceil n/A \rceil \log A + bn/A + (n/b) \log A)
\end{aligned}
$$

set $b \approx \sqrt{A}$, $A = \log n$

$\Rightarrow \boxed{\widetilde{O}(n\sqrt{\log n})}$ area

Last Ex: general trees [new]

$$W(n) = O(A + \log n)$$
$$H(n) = \widetilde{O}(\lceil n/A \rceil \log A + bn/A + (n/b) \log A)$$

set $b \approx \sqrt{A}$, $A = \log n$

$\Rightarrow$ $\boxed{\widetilde{O}(n\sqrt{\log n})}$ area

bootstrap again $\Rightarrow$ $\boxed{\widetilde{O}(n \log^{1/3} n)}$ area

Last Ex: general trees [new]

$$
\begin{aligned}
W(n) &= O(A + \log n) \\
H(n) &= \widetilde{O}(\lceil n/A \rceil \log A + bn/A + (n/b) \log A)
\end{aligned}
$$

set $b \approx \sqrt{A}$, $A = \log n$

$\Rightarrow$ $\boxed{\widetilde{O}(n\sqrt{\log n})}$ area

bootstrap again $\Rightarrow$ $\boxed{\widetilde{O}(n \log^{1/4} n)}$ area

Last Ex: general trees [new]

$$W(n) = O(A + \log n)$$
$$H(n) = \widetilde{O}(\lceil n/A \rceil \log A + bn/A + (n/b) \log A)$$

set $b \approx \sqrt{A}$, $A = \log n$

$\Rightarrow$ $\boxed{\widetilde{O}(n\sqrt{\log n})}$ area

bootstrap again $\Rightarrow$ $\boxed{\widetilde{O}(n \log^{1/5} n)}$ area

Last Ex: general trees [new]

$$W(n) = O(A + \log n)$$
$$H(n) = \widetilde{O}(\lceil n/A \rceil \log A + bn/A + (n/b) \log A)$$

set $b \approx \sqrt{A}$, $A = \log n$

$\Rightarrow$ $\boxed{\widetilde{O}(n\sqrt{\log n})}$ area

bootstrap again $\Rightarrow$ $\boxed{O(n \log^{\varepsilon} n)}$ area

Last Ex: general trees [new]

$$W(n) = O(A + \log n)$$
$$H(n) = \widetilde{O}(\lceil n/A \rceil \log A + bn/A + (n/b) \log A)$$

set $b \approx \sqrt{A}$, $A = \log n$

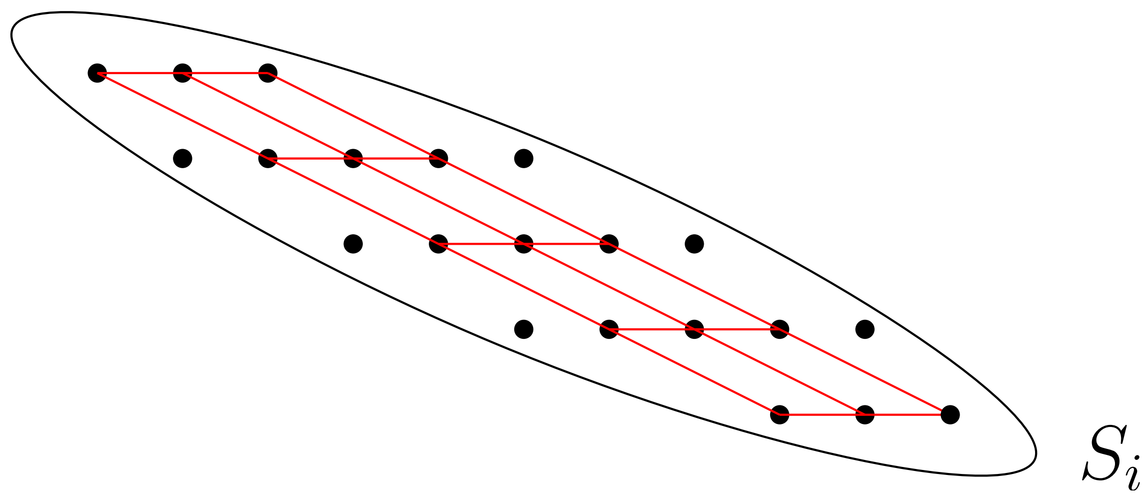
$\Rightarrow$ $\boxed{\widetilde{O}(n\sqrt{\log n})}$ area

bootstrap again $\Rightarrow$ $\boxed{n2^{\widetilde{O}(\sqrt{\log \log n})}}$ area

Last Ex: general trees [new]

Geometry Lemma (needed for bootstrapping):

If S_i is a 2D convex body containing n_i lattice points, then S_i contains an $\Omega(B) \times \Omega(n_i/B)$ grid after some affine transformation for some B



Many Other Open Problems. . .

Many Other Open Problems. . .

ternary tree, orthogonal [Frati'07]:

$O(n^{\log_3 2}) = O(n^{0.631})$ width

$\Omega(n^{0.438})$ width

ternary tree, octilinear, strict upw., *ordered* [Lee'17]:

$O(n^{0.68})$ width (via double recurrence technique)

$\Omega(n^{0.411})$ width

Many Other Open Problems. . .

ternary tree, orthogonal [Frati'07]:

$O(n^{\log_3 2}) = O(n^{0.631})$ width

$\Omega(n^{0.438})$ width

ternary tree, octilinear, strict upw., *ordered* [Lee'17]:

$O(n^{0.68})$ width (via double recurrence technique)

$\Omega(n^{0.411})$ width

(open: best exponent?)

THE END