Product structure theory with applications

VIDA DUJMOVIĆ
U. Ottawa

Oxford discrete math & probability seminar
Feb 9, 2020

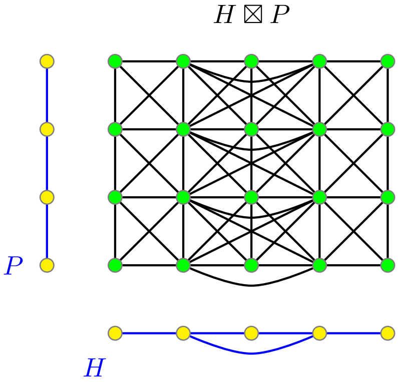
APPLICATIONS

- queue layouts [1992] & 3D grid drawings [2002]
- non-repetitive graph colourings [2002]
- p-centered colourings
- compact encodings [1988]
- universal graphs
- clustered colourings
- vertex ranking

⋮

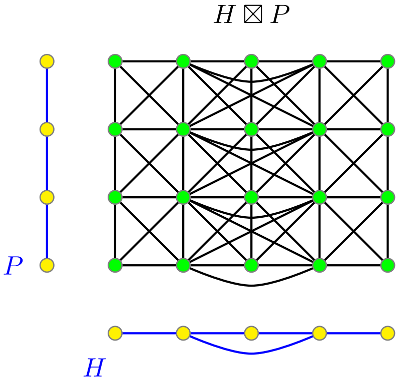
simple proofs + better bounds

STRONG GRAPH PRODUCT ⊠



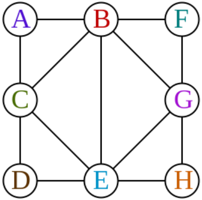
structure of planar graphs

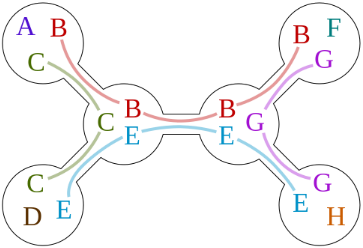
theorem [Dujmović, Joret, Micek, Morin, Ueckerdt, Wood'19]
every planar graph is a subgraph of $H \boxtimes P$
for some graph H with tree-width $\leqslant 8$ and some path P



treewidth

a tree decomposition represents each vertex as a subtree of a tree T so that the subtrees of adjacent vertices intersect in T



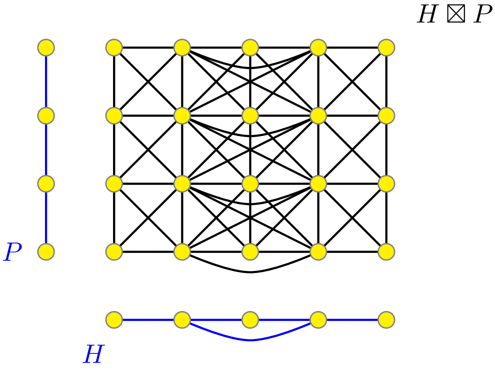


width := maximum bag size -1

treewidth tw(G) :=
min width of tree decomposition of G

[Images courtesy of Wikipedia]

theorem [Dujmović, Joret, Micek, Morin, Ueckerdt, Wood'19]
every planar graph is a subgraph of $H \boxtimes P$ where $\mathrm{tw}(H) \leqslant 8$



what is it good for?

APPLICATIONS

- queue layouts [1992] & 3D grid drawings [2002]
- non-repetitive graph colourings [2002]
- p-centered colourings
- compact encodings [1988]
- universal graphs
- clustered colourings
- vertex ranking ✓

⋮

simple proofs + better bounds

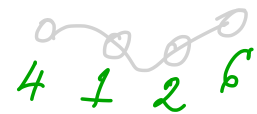
(vertex) ℓ-ranking

Def

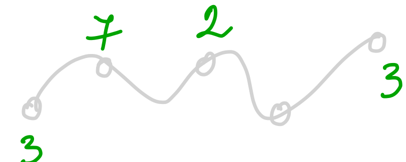
$\varphi : V(G) \to \mathbb{N}$ (vertex colouring) s.t.

every path $u_1, \dots, u_p$ of length at most ℓ (# edges)

- either $\varphi(u_1) \neq \varphi(u_p)$



- $\exists u_i$ s.t. $\varphi(u_i) > \varphi(u_1)$



$\chi_\ell(G)$

- proper colouring
- star colouring
- $\ell=2$ us-colouring
- $\ell=\infty$ vertex ranking

Previous work

fixed ℓ

	UPPER	LOWER
trees	$\Theta(\log n / \log\log\log n)$	$\Omega(\log n / \log\log\log n)$
planar	$\Theta(\log n)$	$\Omega(\log n / \log\log\log n)$

gap

[Karpas, Neiman, Smorodinsky 2015]

– Contrast with $\chi_\infty(G)$ and centered colouring

Previous work

	UPPER	LOWER
trees	$\Theta(\lg n / \lg\lg n)$	$\Omega(\lg n / \lg\lg n)$
planar	$\Theta(\lg n)$	$\Omega(\lg n / \lg\lg n)$

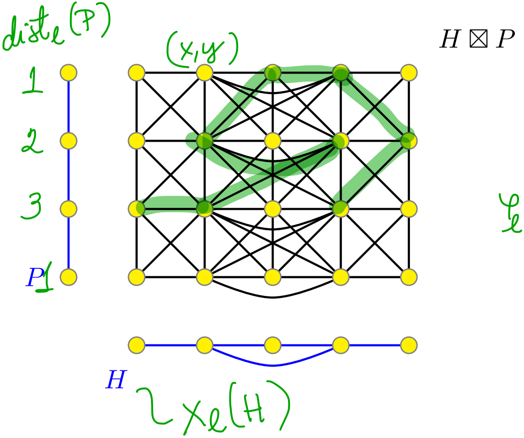
[Karpas, Neiman, Smorodinsky 2015]

gap

Th: [Bose, D., Javarsineh, Morin 2020]

$\forall$ fixed $\ell \geq 2$, every n-vertex planar graph G has $\chi_\ell(G) = \Theta\left(\frac{\lg n}{\lg\lg\lg n}\right)$ → asymptotically optimal

theorem [Dujmović, Joret, Micek, Morin, Ueckerdt, Wood'19]
every planar graph is a subgraph of $H \boxtimes P$ where $\mathrm{tw}(H) \leqslant 8$



Colour vertices of $H \times P$:

$\varphi_\ell(x,y) = (\varphi_\ell(y), \mathrm{dist}_\ell(x))$

what is it good for?

ℓ-vertex ranking of products

Th: [Bose, D., Javarsineh, Morin 2020]
For any two graphs H and J,

$$\chi_\ell(H \boxtimes J) = \underbrace{\chi_\ell(H)}_{?} \cdot \underbrace{\mathrm{dist}_\ell(J)}_{3 \text{ for } J \text{ path}}$$

Th: $\forall \ell \geq 2$, and $t \geq 1$ every graph H of simple treewidth t has

$$\chi_\ell(H) = \Theta\left(\frac{\lg n}{\underbrace{\lg \cdots \lg}_{t \text{ times}} n}\right)$$

→ asymptotically optimal

improving the bound

theorem [Dujmović, Joret, Micek, Morin, Ueckerdt, Wood'19]
every planar graph is a subgraph of $H \boxtimes P \boxtimes K_3$ s.t $stw(H) \leqslant 3$

planar

simple treewidth

$$\chi_\ell(H \boxtimes J) = \chi_\ell(H) \cdot dist_\ell(J)$$

$$\Theta\left(\frac{\lg}{\lg\lg\lg n}\right) \qquad P \boxtimes K_3 \quad \Theta(1)$$

APPLICATIONS

- queue layouts [1992] & 3D grid drawings [2002]
- non-repetitive graph colourings [2002]
- p-centered colourings
- compact encodings [1988] ✓
- universal graphs
- clustered colourings
- vertex ranking

⋮

simple proofs + better bounds

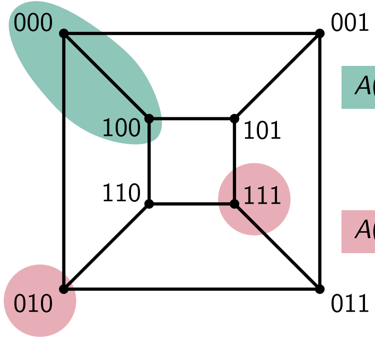
Asymptotically Optimal Adjacency-Labelling for Planar Graphs (and Beyond)

Vida Dujmović, Louis Esperet, Cyril Gavoille, Gwenaël Joret, Piotr Micek, and Pat Morin

Adjacency Labelling

Labelling and Testing for a graph G

- Adjacency tester $A : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$
- Labelling function $\ell : V(G) \to \{0,1\}^*$
- (G, ℓ) *works with* A if, for each $v, w \in V(G)$, $A(\ell(v), \ell(w)) = \begin{cases} 1 & \text{if } vw \in E(G) \\ 0 & \text{if } vw \notin E(G) \end{cases}$



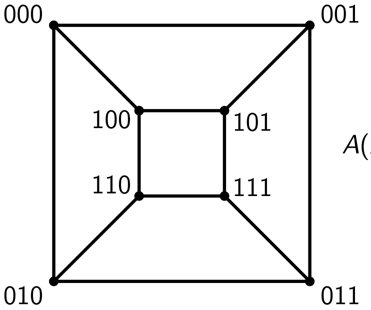
$A(000, 001) = 1$

$A(010, 111) = 0$

Adjacency Labelling

Labelling schemes for graph families

- A graph family $\mathcal{G}$ has an $f(n)$-bit labelling scheme if
  - there exists $A : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$
  - for each n-vertex $G \in \mathcal{G}$ there exists $\ell : V(G) \to \{0,1\}^{f(n)}$ such that (G, ℓ) works with A



$$A(x,y) := \begin{cases} 1 & \text{if } \|x-y\| = 1 \\ 0 & \text{otherwise} \end{cases}$$

[Kannan, Naor, Rudich] 1988

- **Theorem** A graph family $\mathcal{G}$ has an $f(n)$-bit labelling scheme iff for each $n \in \mathbb{N}$ there exists a graph U_n such that
  - $|V(U_n)| = 2^{f(n)}$
  - For each n-vertex $G \in \mathcal{G}$, G appears as an induced subgraph of U_n
- Proof: $V(U_n) := \{0,1\}^{f(n)}$, $E(U_n) := \{vw : A(v,w) = 1\}$.
- U_n is called an *(induced) universal* graph for n-vertex members of $\mathcal{G}$.

Previous Results

Trees and Bounded Treewidth Graphs

Trees

$f(n)$	$\lvert U_n \rvert$	Reference
$\log n + \log\log n + O(1)$	$O(n \log n)$	Chung (1990)
$\log n + O(\log^* n)$	$n2^{O(\log^* n)}$	Alstrup-Rauhe (2006)
$\log n + O(1)$	$O(n)$	Alstrup-Dahlgaard-Knudsen (2017)

Bounded treewidth graphs

$\log n + o(\log n)$	$n^{1+o(1)}$	Gavoille-Labourel (2007)

Planar Graphs

$f(n)$	$\lvert U_n \rvert$	Reference	Techniques
$6 \log n$	n^6	Muller (1988)	5-degeneracy
$4 \log n$	n^4	Kannan-Naor-Rudich (1988)	3-orientation
$3 \log n + O(1)$	$O(n^3)$	—	arboricity + trees
$2 \log n + o(\log n)$	$n^{2+o(1)}$	Gavoille-Labourel (2007)	2-outerplanar decomp.
$\frac{4}{3} \log n + o(\log n)$	$n^{\frac{4}{3}+o(1)}$	Bonamy-Gavoille-Pilipczuk (2020)	product structure

The best known results for several subclasses of planar graphs are reported in Table 1.

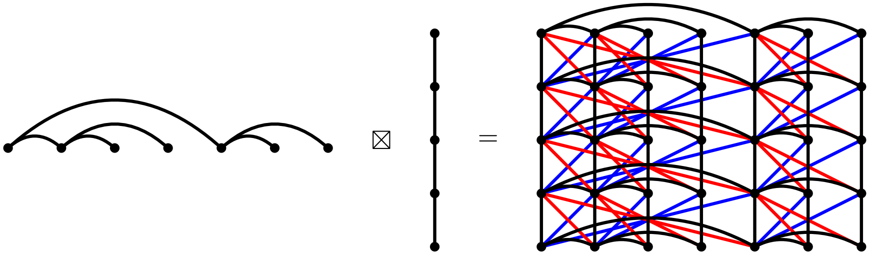
Graph classes (with n vertices)	Upper bound (label length in bits)	References
maximum degree-2	$\log n + O(1)$	[AABTKS16, But09, ELO08]
caterpillars	$\log n + O(1)$	[BGL06]
bounded degree trees	$\log n + O(1)$	[Chu90]
bounded depth trees	$\log n + O(1)$	[FK10a]
trees	$\log n + O(1)$	[ADBTK17]
bounded degree outerplanar	$\log n + O(1)$	[Chu90, AR14]
outerplanar	$\log n + O(\log\log n)$	[GL07]
bounded treewidth planar	$\log n + O(\log\log n)$	[GL07]
maximum degree-4 planar	$\frac{3}{2}\log n + O(\log\log n)$	[AR14]
bounded degree planar	$2\log n + O(1)$	[Chu90]
planar	$2\log n + O(\log\log n)$	[GL07]
planar	$\frac{4}{3}\log n + O(\log\log n)$	[BGP '20]

Planar Graphs

$f(n)$	$\|U_n\|$	Reference	Techniques
$6\log n$	n^6	Muller (1988)	5-degeneracy
$4\log n$	n^4	Kannan-Naor-Rudich (1988)	3-orientation
$3\log n + O(1)$	$O(n^3)$	—	arboricity + trees
$2\log n + o(\log n)$	$n^{2+o(1)}$	Gavoille-Labourel (2007)	2-outerplanar decomp.
$\frac{4}{3}\log n + o(\log n)$	$n^{\frac{4}{3}+o(1)}$	Bonamy-Gavoille-Pilipczuk (2020)	product structure

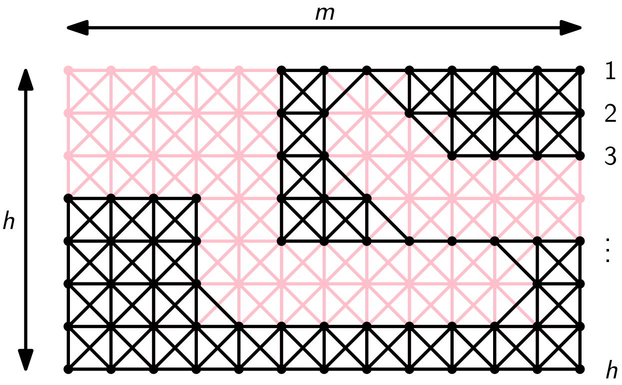
- **Theorem 1:** The family of planar graphs has a $(\log n + o(\log n))$-bit adjacency labelling scheme.
- **Corollary 1:** For each n, there is a graph U_n with $n^{1+o(1)}$ vertices is (induced) universal for n-vertex planar graphs.

The Product Structure Theorem



- **Main Result (Here):** Let $\mathcal{G}_t$ be the family of graphs such that for each $G \in \mathcal{G}$, there exists a graph H of treewidth at most t and a path P such that G a subgraph of $H \boxtimes P$. Then, for any fixed t, $\mathcal{G}$ has a $(\log n + o(\log n))$-bit labelling scheme.
- **Applications:** bounded genus graphs, apex-minor free graphs, bounded-degree graphs from minor-closed families, k-planar graphs for constant k

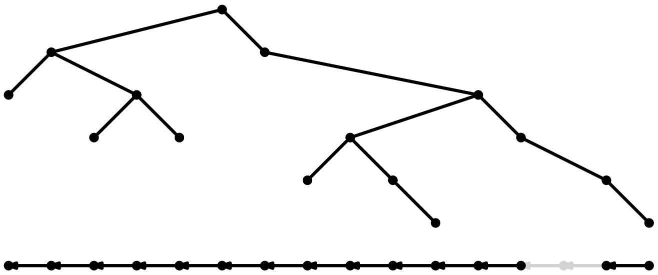
Warm-up: Induced Subgraphs of $P \boxtimes P$



- The label for a vertex $v = (x, i)$ has three main parts:
  - A *row label* of length $\log n - \log n_i + o(\log n)$ defined by i
  - A *column label* of length $\log n_i + o(\log n)$ defined by (G_i, x)
  - A *transition label* of length $o(\log n)$ defined by (G_i, G_{i+1}, x)

Binary Search Trees

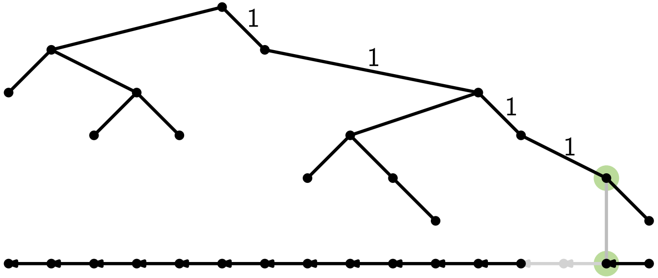
Any binary search tree can label a path



- Label length: $\mathrm{depth}_T(v) + O(\log \mathrm{height}(T)) = \mathrm{depth}_T(v) + O(\log \log n)$ bits

Binary Search Trees

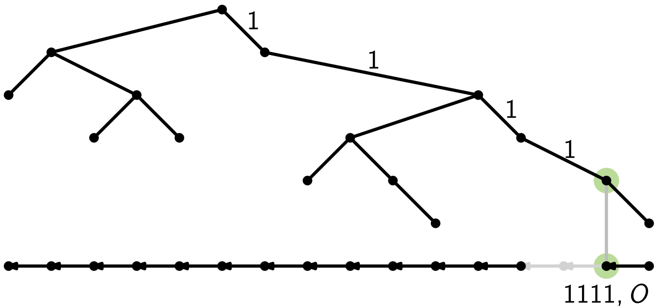
Any binary search tree can label a path



- Label length: $\mathrm{depth}_T(v) + O(\log \mathrm{height}(T)) = \mathrm{depth}_T(v) + O(\log \log n)$ bits

Binary Search Trees

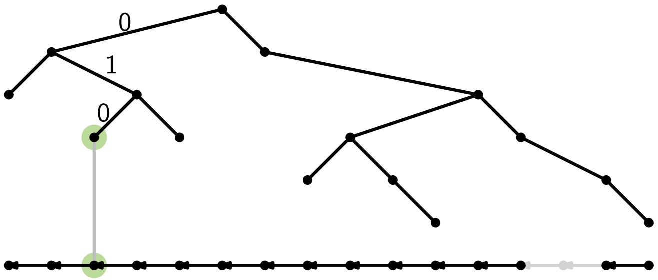
Any binary search tree can label a path



- Label length: $\mathrm{depth}_T(v) + O(\log \mathrm{height}(T)) = \mathrm{depth}_T(v) + O(\log \log n)$ bits

Binary Search Trees

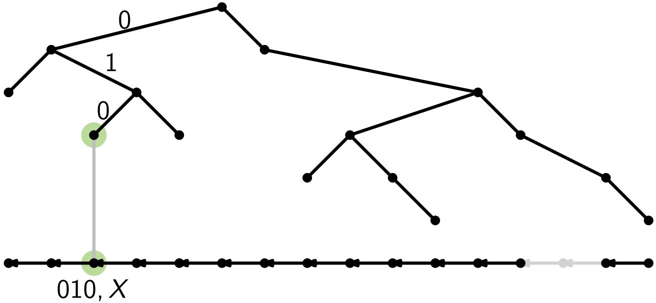
Any binary search tree can label a path



- Label length: $\mathrm{depth}_T(v) + O(\log \mathrm{height}(T)) = \mathrm{depth}_T(v) + O(\log \log n)$ bits

Binary Search Trees

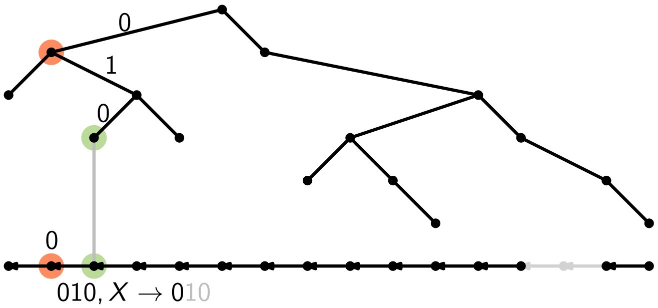
Any binary search tree can label a path



- Label length: $\mathrm{depth}_T(v) + O(\log \mathrm{height}(T)) = \mathrm{depth}_T(v) + O(\log \log n)$ bits

Binary Search Trees

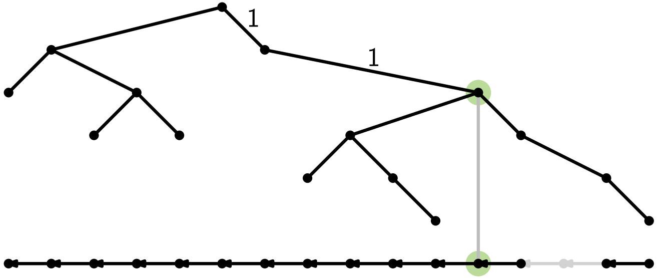
Any binary search tree can label a path



- Label length: $\mathrm{depth}_T(v) + O(\log \mathrm{height}(T)) = \mathrm{depth}_T(v) + O(\log \log n)$ bits

Binary Search Trees

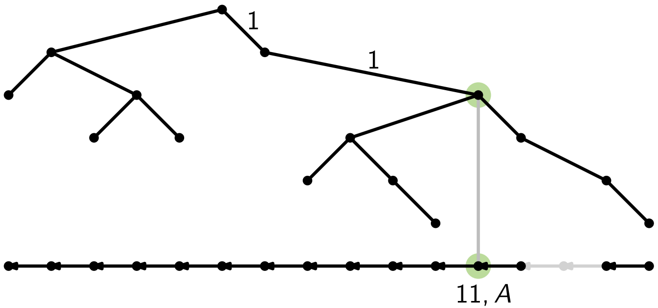
Any binary search tree can label a path



- Label length: $\mathrm{depth}_T(v) + O(\log \mathrm{height}(T)) = \mathrm{depth}_T(v) + O(\log \log n)$ bits

Binary Search Trees

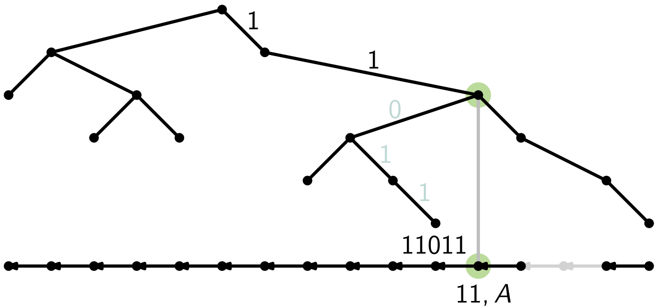
Any binary search tree can label a path



- Label length: $\mathrm{depth}_T(v) + O(\log \mathrm{height}(T)) = \mathrm{depth}_T(v) + O(\log \log n)$ bits

Binary Search Trees

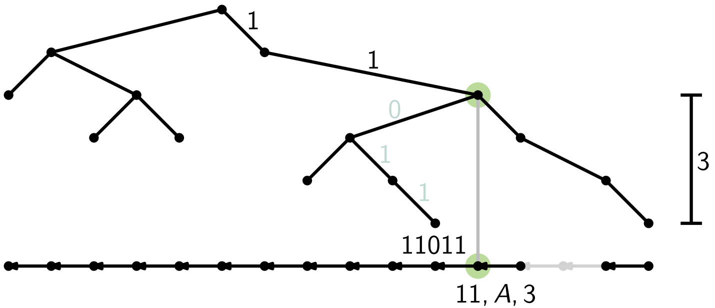
Any binary search tree can label a path



- Label length: $\mathrm{depth}_T(v) + O(\log \mathrm{height}(T)) = \mathrm{depth}_T(v) + O(\log \log n)$ bits

Binary Search Trees

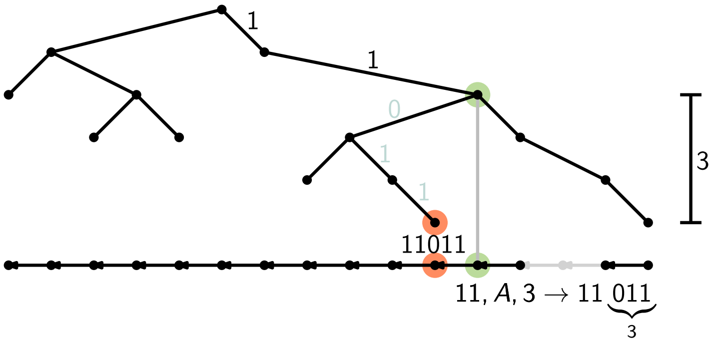
Any binary search tree can label a path



- Label length: $\mathrm{depth}_T(v) + O(\log \mathrm{height}(T)) = \mathrm{depth}_T(v) + O(\log \log n)$ bits

Binary Search Trees

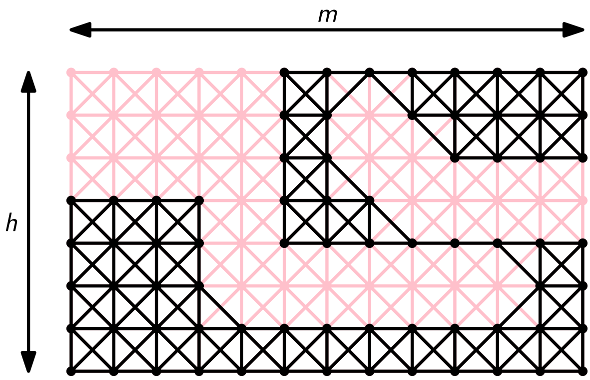
Any binary search tree can label a path



- Label length: $\mathrm{depth}_T(v) + O(\log \mathrm{height}(T)) = \mathrm{depth}_T(v) + O(\log \log n)$ bits

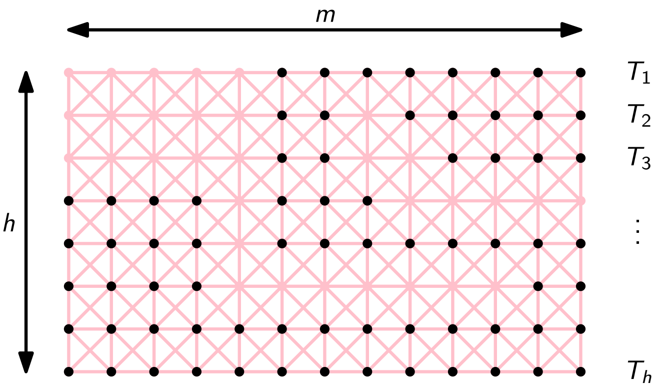
The Column Code Trees

Bulk Trees



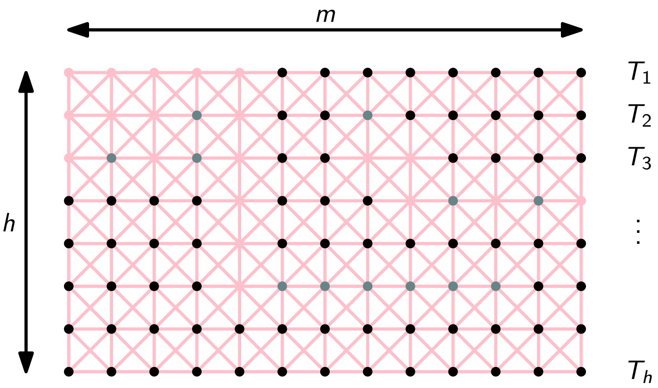
The Column Code Trees

Bulk Trees



The Column Code Trees

Bulk Trees



- $\sum_{i=1}^{h} |T_i| = O(n)$

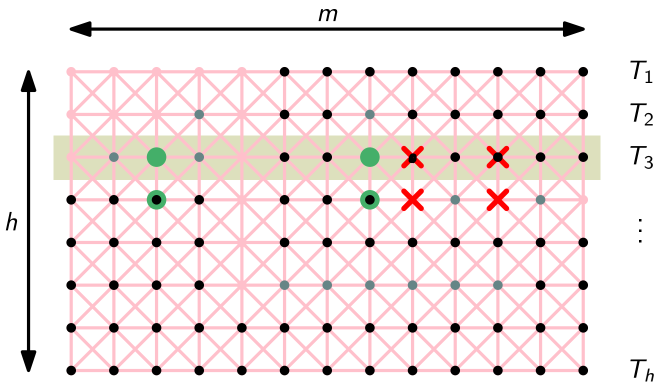
The Row Code Tree

Biased binary search trees

- $\sum_{i=1}^{h} |T_i| = O(n)$
- Make a biased binary search tree T with $V(T) := \{1, \ldots, h\}$ using $\mathrm{weight}(i) = |T_i|$
  - Then $\mathrm{depth}_T(i) \le \log n - \log |T_i| + O(1)$
- Row i label length is $\log n - \log |T_i| + O(\log \log n)$

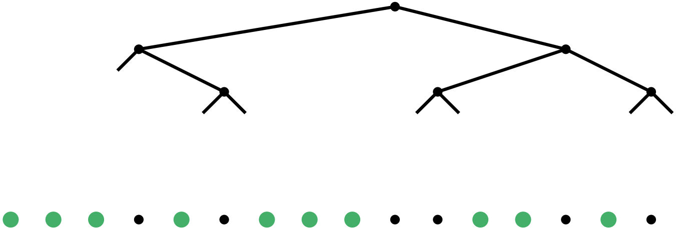
The Column Code Tree

Bulk Trees



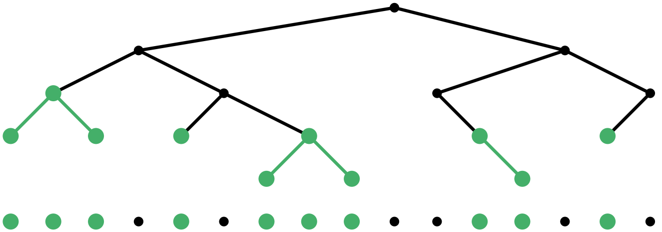
Bulk Insertion

Bulk Trees



Bulk Insertion

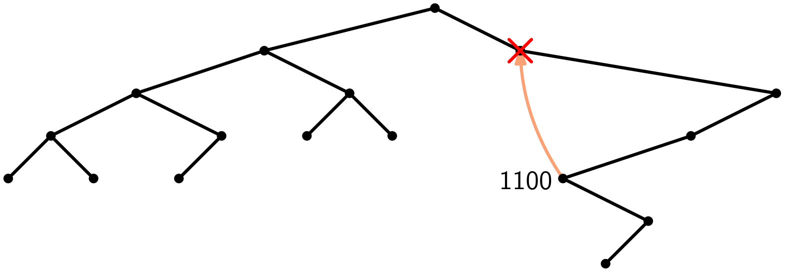
Bulk Trees



- $h(T_{i+1}) \leq h(T_i) + c$

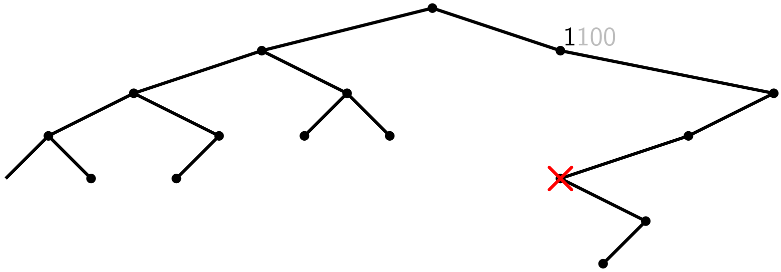
Bulk Deletion

Bulk Trees



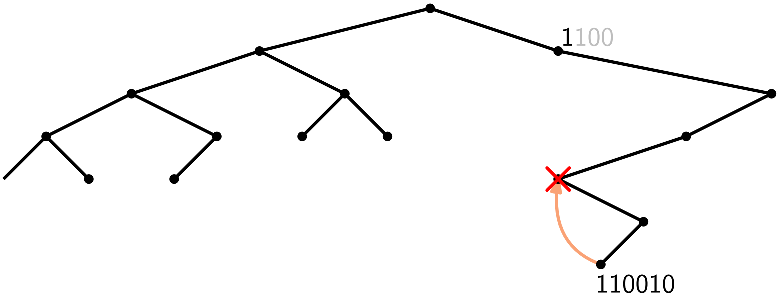
Bulk Deletion

Bulk Trees



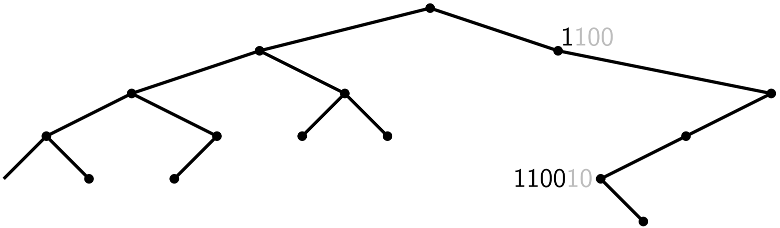
Bulk Deletion

Bulk Trees



Bulk Deletion

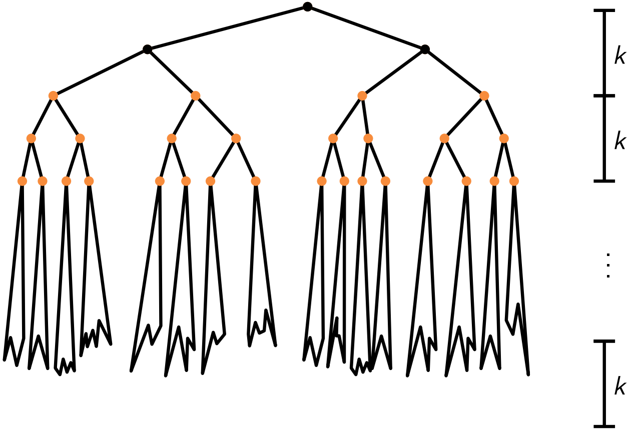
Bulk Trees



- $h(T_{i+1}) \leq h(T_i)$
- $\log |T_i| \leq \log |T_{i+1}| + c$

Rebalancing

balance(θ)



- Ensures that $h(T_i) \leq \log |T_i| + O(\frac{1}{k} \log |T_i|)$

Summary

- transition code length: $O(k \log \log n)$
- $h(T_i) \leq \log |T_i| + O(\frac{1}{k} \log |T_i|)$
- label length: $h(T_i) + O(k \log \log n) \leq \log |T_i| + O(\frac{1}{k} \log n + k \log \log n)$
- optimize k: $h(T_i) + O(\sqrt{\log n \log \log n})$

Missing Pieces

- Subgraphs of $H \boxtimes P$ not $P \boxtimes P$
  - H has pathwidth $O(\log n)$
  - H is contained in an *interval graph* H^+ of thickness $O(\log n)$
  - Use an *interval tree*
- Subgraphs of $H \boxtimes P$ not *induced* subgraphs
  - Use d-orientation of $H \boxtimes P$
  - Additional bit-vector of length d indicates which induced edges are actually present

bits:

$\log n + \Theta(\sqrt{\log n \cdot \log\log n})$

summary of generalizations of product structure

graph class	subgraph of	$\operatorname{tw}(H) \leqslant$
planar	$H \boxtimes P$	8
planar	$H \boxtimes P \boxtimes K_3$	3
genus g	$(K_{2g} + H) \boxtimes P$	8
genus g	$H \boxtimes P \boxtimes K_{\max\{2g,3\}}$	4
apex-minor-free	$H \boxtimes P$	$O(1)$
minor-free	$H \boxtimes P$	$O(\Delta)$
minor-free	clique sums of $(H_i \boxtimes P) + K_a$	$O(1)$
1-planar	$H \boxtimes P$	479
1-planar	$H \boxtimes P \boxtimes K_{30}$	3
k-planar	$H \boxtimes P$	$O(k^5)$
(g, k)-planar	$H \boxtimes P$	$O(gk^6)$
k-th power of genus g	$H \boxtimes P$	$O(gk^8\Delta^k)$
(g, d)-map graph	$H \boxtimes P$	$O(gd^2)$

MINOR-FREE

- queue layouts [1992] & 3D grid drawings [2002]
- non-repetitive graph colourings [2002]

extends to all proper minor-free

- compact encodings [1988] ?

best known:

$2 \cdot \log n + o(\log n)$

the end