

# Universal Graphs (& Implicit Representation)

*Cyril Gavoille*

LABORATOIRE  
BORDELAIS  
DE RECHERCHE  
EN INFORMATIQUE

**LaBRI** - *University of Bordeaux*

Journées de Combinatoire de Bordeaux  
@Home on BBB  
Feb. 1-4, 2021

# Induced-Universal Graphs

[Babai, Chung, Erdős, Graham, Spencer '82]

A graph  $U$  is an **induced-universal** graph for the family  $\mathcal{F}$  if every graph of  $\mathcal{F}$  is isomorphic to an *induced* subgraph of  $U$ .



**Goal:** to find such a universal graph for a given family  $\mathcal{F}_n$  with few vertices (w.r.t.  $n$ )

# Representation of a Graph



adjacency list

1	2,5
2	1,3
<b>3</b>	2,4,5
4	3
5	1,3

matrix

	1	2	<b>3</b>	4	5
1	0	1	0	0	1
2	1	0	1	0	0
<b>3</b>	0	1	0	1	1
4	0	0	1	0	0
5	1	0	1	0	0

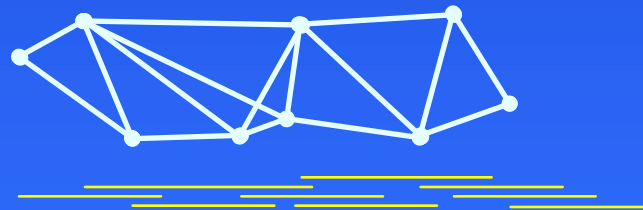
1 node = 1 pointer in the data-structure  
(it does not carry any specific information)

# Implicit Representation

To associate with the nodes more information, typically adjacency, and to remove the data-structure.

Interval graphs:  $u \mapsto I(u) \subseteq [1, 2n]$

Edges:  $u-v \Leftrightarrow I(u) \cap I(v) \neq \emptyset$



Compact representation:  $O(\log n)$  bits/node

Possibly time  $O(n)$  algorithms vs.  $O(n+m)$



# Labelling Schemes

$P$  = a graph property defined on pairs of nodes

$F$  = a graph family

A  $P$ -labelling scheme for  $F$  is a pair  $(\lambda, f)$  such that  $\forall G \in F, \forall u, v \in V(G)$ :

- [labelling]  $\lambda(u, G)$  is a binary string
- [decoder]  $f(\lambda(u, G), \lambda(v, G)) = P(u, v, G)$

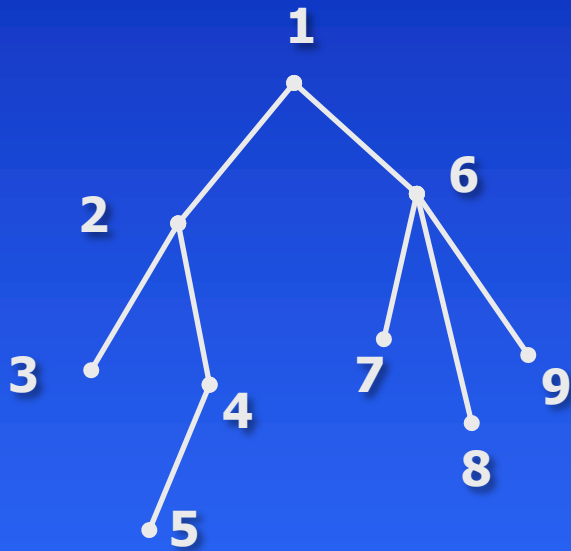
**Goal:** to minimize the maximum label size

**In this talk:**  $P(u, v, G)$  is TRUE  $\Leftrightarrow uv \in E(G)$

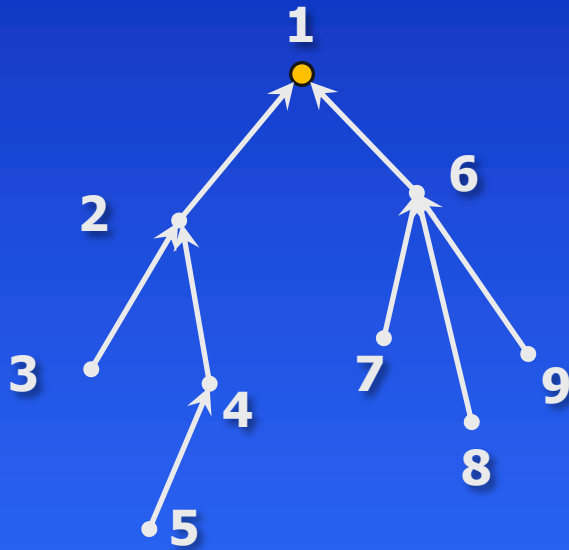
# Basic Example: Trees



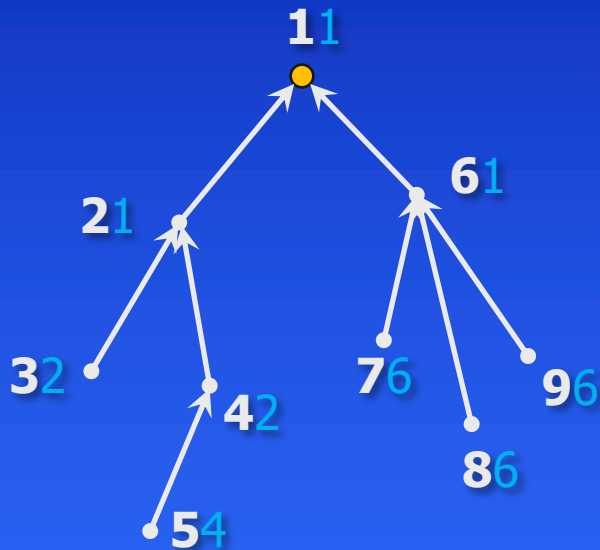
# Basic Example: Trees



# Basic Example: Trees



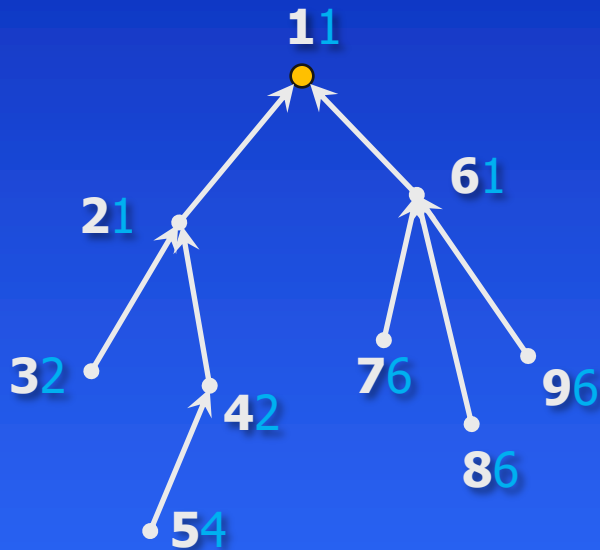
# Basic Example: Trees



$\lambda(u, T) = (u, \text{parent}(u))$  or  $(u, u)$

$f(uv, xy) = (v = x$  or  $u = y)$

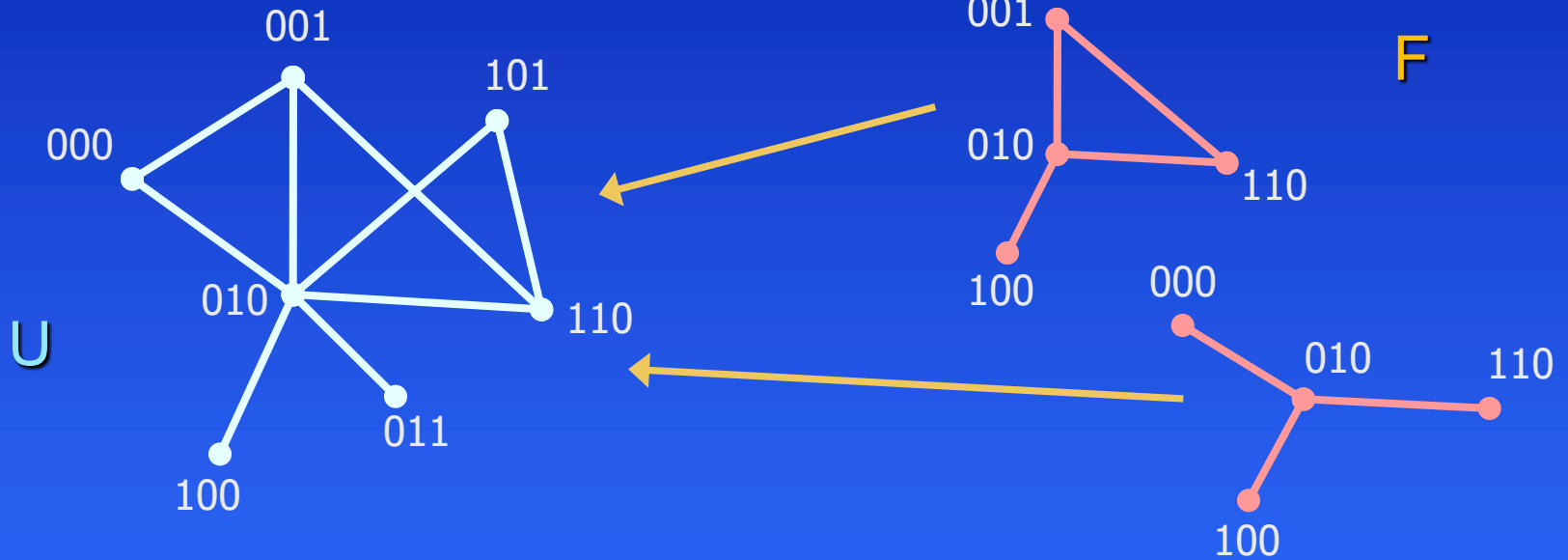
# Basic Example: Trees



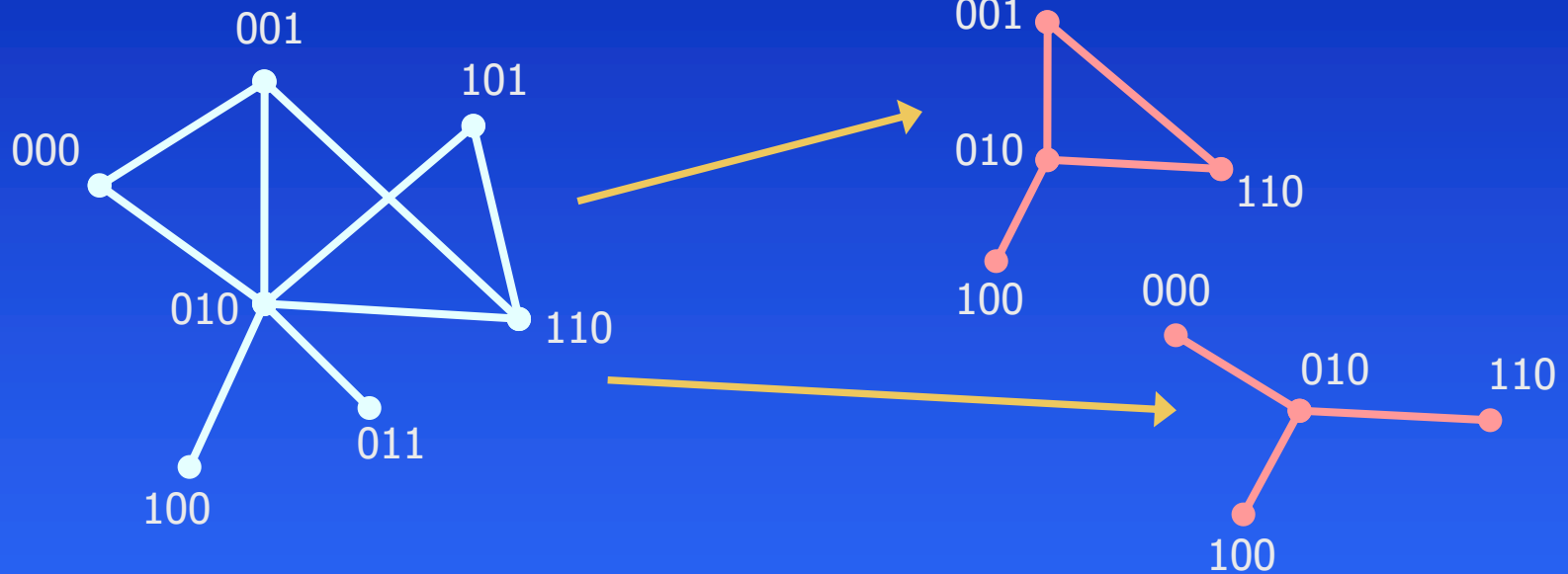
$$\lambda(u, T) = (u, \text{parent}(u)) \text{ or } (u, u)$$
$$f(uv, xy) = (v = x \text{ or } u = y)$$

For trees with  $n$  nodes:  $\sim 2 \log n$  bits/node  
(the constant does matter [Abiteboul et al. - SICOMP '06])

# Universal & Label Graphs



# Universal & Label Graphs



induced-universal graph  $U$

graphs of  $F$

$c \log n$ -bit labelling  $\Leftrightarrow$  induced-universal graph of  $n^c$  nodes



# Universal Graphs for Trees (for $n=6$ nodes)

From the  $(u, \text{parent}(u))$  labelling  
 $\Rightarrow$  induced-universal graph of  $n^2=36$  nodes



Using DFS for T:  $(u, v)$   
 $\Rightarrow u > v$  or  $u = v = 1$

# Universal Graphs for Trees (for $n=6$ nodes)

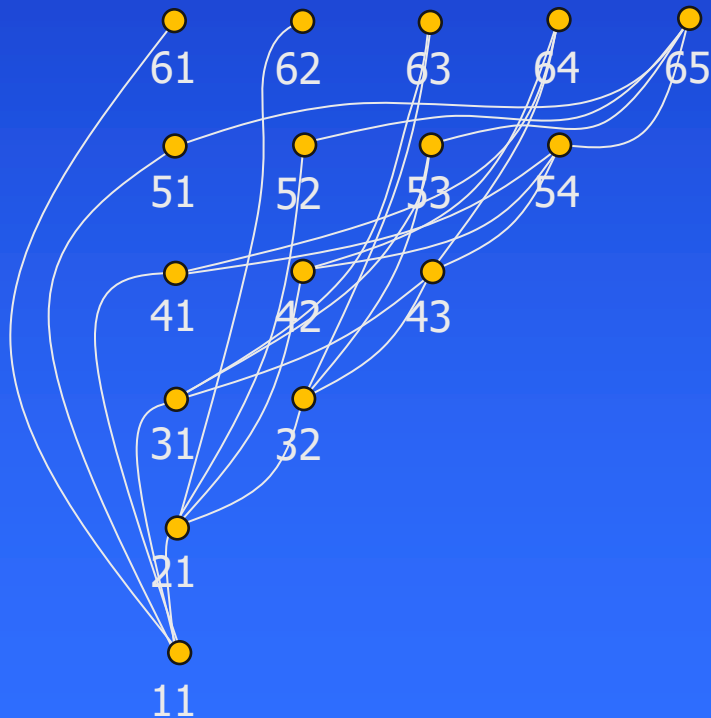
From the  $(u, \text{parent}(u))$  labelling  
 $\Rightarrow$  induced-universal graph of  $n^2=36$  nodes



Using DFS for T:  $(u, v)$   
 $\Rightarrow u > v$  or  $u = v = 1$   
 $\Rightarrow n(n-1)/2 + 1 = 16$  nodes

# Universal Graphs for Trees (for $n=6$ nodes)

From the  $(u, \text{parent}(u))$  labelling  
 $\Rightarrow$  induced-universal graph of  $n^2=36$  nodes



Using DFS for  $T: (u, v)$   
 $\Rightarrow u > v$  or  $u = v = 1$   
 $\Rightarrow n(n-1)/2 + 1 = 16$  nodes

# Universal Graphs for Trees (universal trees)

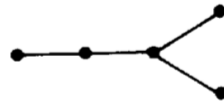
$n = 1$ :



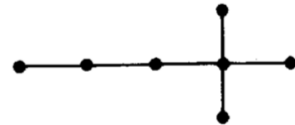
$n = 2$ :



$n = 3$ :



$n = 4$ :



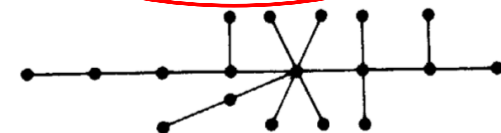
$n = 5$ :



$n = 6$ :



$n = 7$ :

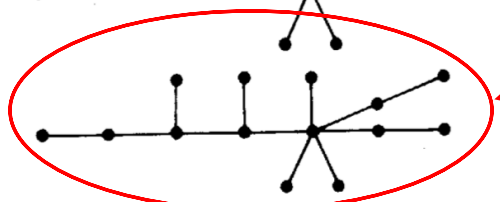


ON GRAPHS WHICH CONTAIN ALL SMALL TREES, II.

F.R.K. CHUNG — R.L. GRAHAM — N. PIPPENGER

COLLOQUIA MATHEMATICA SOCIETATIS JÁNOS BOLYAI  
18. COMBINATORICS, KESZTHELY (HUNGARY), 1976.

14 nodes

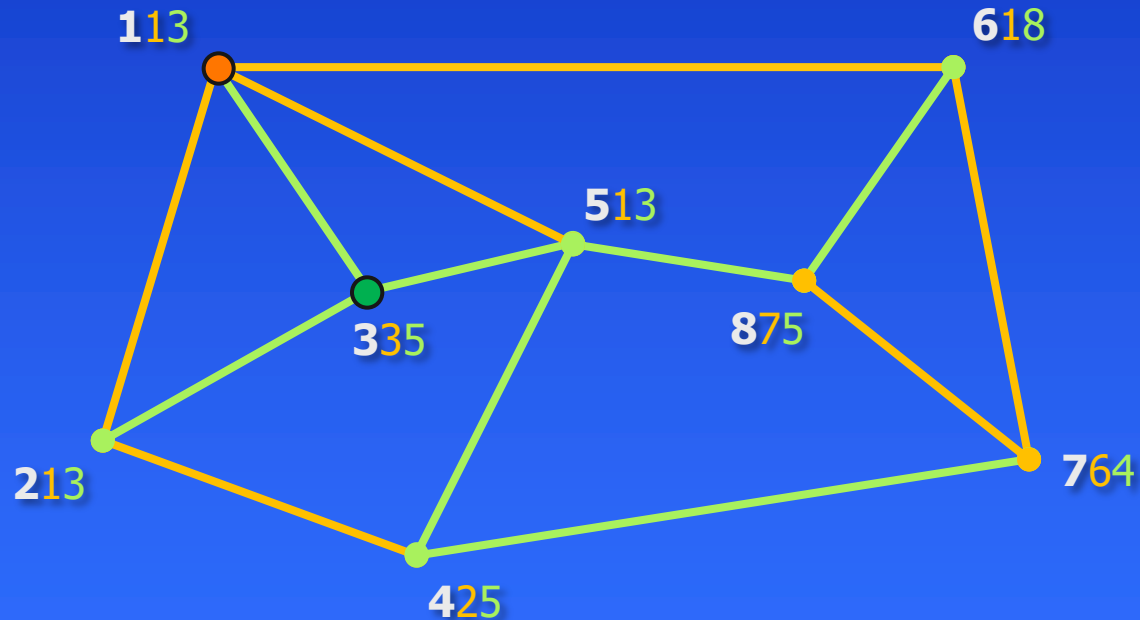


Ex: Degree at most 2

[ How to get  $\log n + O(1)$  bit labels? ]

# Labelling Schemes for Planar Graphs

Edge partition: combining schemes



Arboricity- $k$  graphs:  $(k+1)\log n$  bits  
 $\Rightarrow$  Planar ( $k=3$ ): **4** $\log n$  bits [KNR - STOC'88]

# Better Labelling Schemes

For trees:  $\log n + O(\log^* n)$ ,  $\log n + O(1)$

[Alstrup,Rauhe – FOCS'02]

[Alstrup,Dahlgaard,B.T.Knudsen – FOCS'15 & JACM'17]

⇒ Arboricity- $k$ :  $k \log n + O(1)$

⇒ Planar: **3** $\log n + O(1)$

For treewidth- $k$ :  $\log n + O(k \log \log n)$

[G.,Labourel – ESA'07]

⇒ Planar & Minor-free: **2** $\log n + O(\log \log n)$

# Better: Bonamy, G., Pilipczuk – SODA'20

For planar & bounded genus:  $\frac{4}{3} \log n + O(\log \log n)$

⇒

**Induced-universal graph of  $n^{4/3+o(1)}$  nodes for  $n$ -node planar graphs**  
(and bounded genus graphs)

Labelling the nodes is polynomial  
Decoding adjacency takes constant time



# Improved Bound: Dujmovic, Esperet, G., Joret, Micek, Morin – FOCS'20

**Theorem.** The family of  $n$ -vertex subgraphs of  $H \boxtimes P$  has a labelling scheme with  $\log n + o(\log n)$  bit labels, where  $H$  has bounded treewidth and  $P$  is a path.

⇒

**Induced-universal graph of  $n^{1+o(1)}$   
nodes for  $n$ -node planar graphs**  
(and many other graph families)

Labelling the nodes takes  $O(n \log n)$  time  
Decoding adjacency takes  $\sim \sqrt{\log n}$  time

# Strong Product

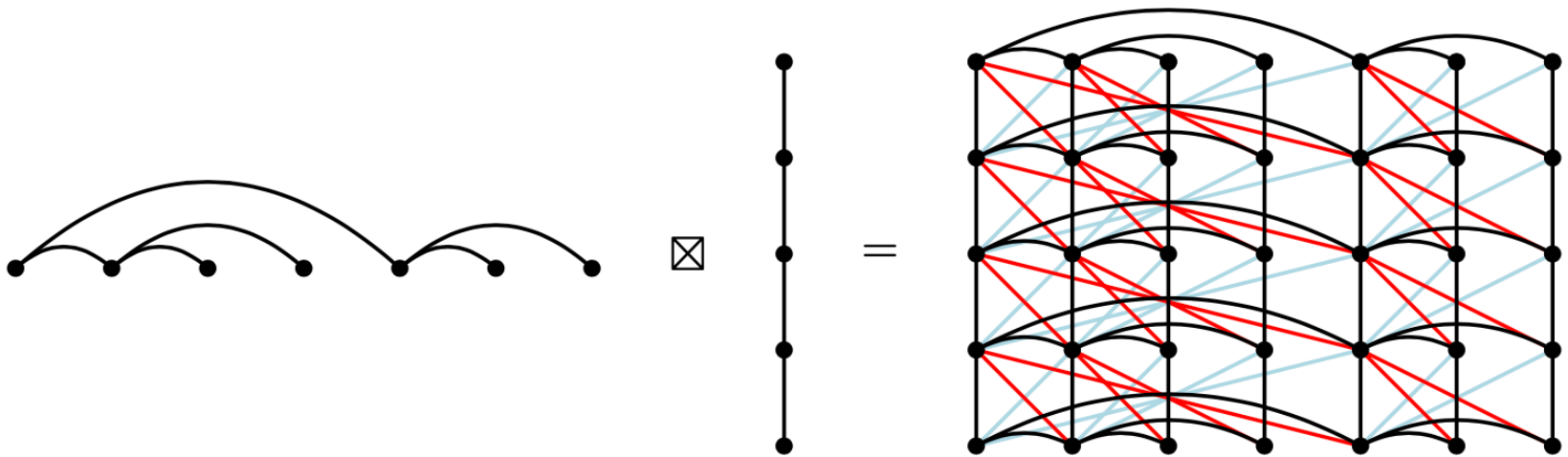


Figure 1: The strong product  $H \boxtimes P$  of a tree  $H$  and a path  $P$ .

# Special Case

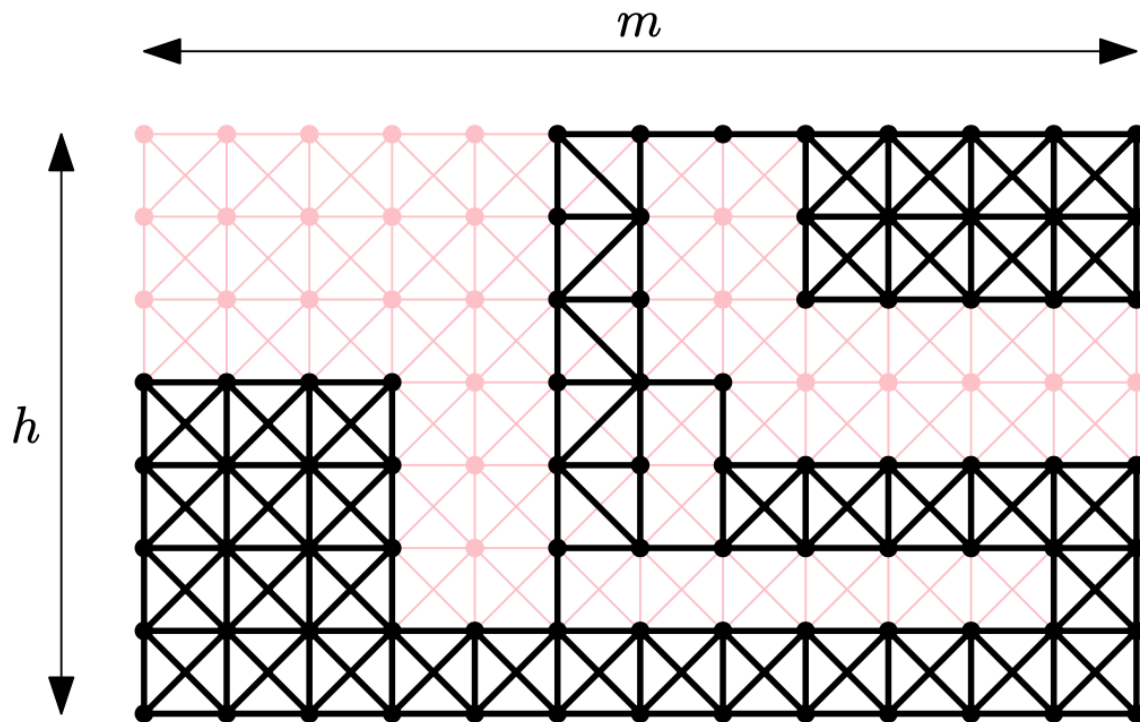


Figure 5: The special case where  $G$  is a subgraph of  $P_1 \boxtimes P_2$ .

# Open Problems

1. Improve the 2nd order term,  $\log n + O(1)$ ?
2. Extend to minor-free graphs
3. Improve to  $\log n + \theta(k)$  for treewidth- $k$
4. Prove lower bounds for planar or minor-free

Best lower bound for planar:  $\log n + \Omega(1)$  ☹️

No families with  $n! 2^{O(n)}$  labelled graphs like trees, planar, bounded genus, bounded treewidth, minor-free (hereditary)... is known to require labels of  $\log n + \omega(1)$  bits.