# Constructive algorithm for path-width of matroids

Jisu Jeong (Dept. of Math, KAIST)

joint work with

Eun Jung Kim (CNRS / Univ. Paris-Dauphine), Sang-il Oum (KAIST)

SIAM-DM16, Atlanta, USA

June 9, 2016

- **Input :** F-representable matroid

- Matroid (E, $I$) = a ground set E + the independent sets $I$
  1. $\emptyset \in I$
  2. $X \subset Y,\ Y \in I \Rightarrow X \in I$
  3. $X, Y \in I, |X| < |Y| \Rightarrow \exists y \in Y\ s.t.\ X \cup \{y\} \in I$

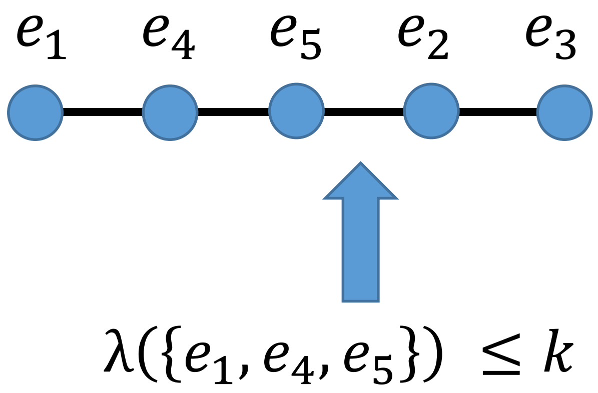
- A matroid (E, $I$) is **F-representable** if

  E = a set of vectors over F,

  $X \in I$ if vectors in X are linearly independent.

- **Input :** F-representable matroid (or $n$ vectors), an integer $k$
- **Goal :** to find a permutation $e_1, e_2, \dots, e_n$ of an elments in $E$

  such that for every $i$, $\lambda(\{e_1, e_2, \dots, e_i\}) \le k$




- Here, λ is a connectivity function of a matroid.
- λ(X) = rank(X) + rank(E-X) - rank(E)

  = dim(<X>) + dim(<E-X>) - dim(<E>)

  = dim(<X> ∩ <E-X>)

- a path-decomposition of width at most $k$,

  or a linear layout of path-width at most $k$

- History

**Bodlaender and Kloks (1996)**

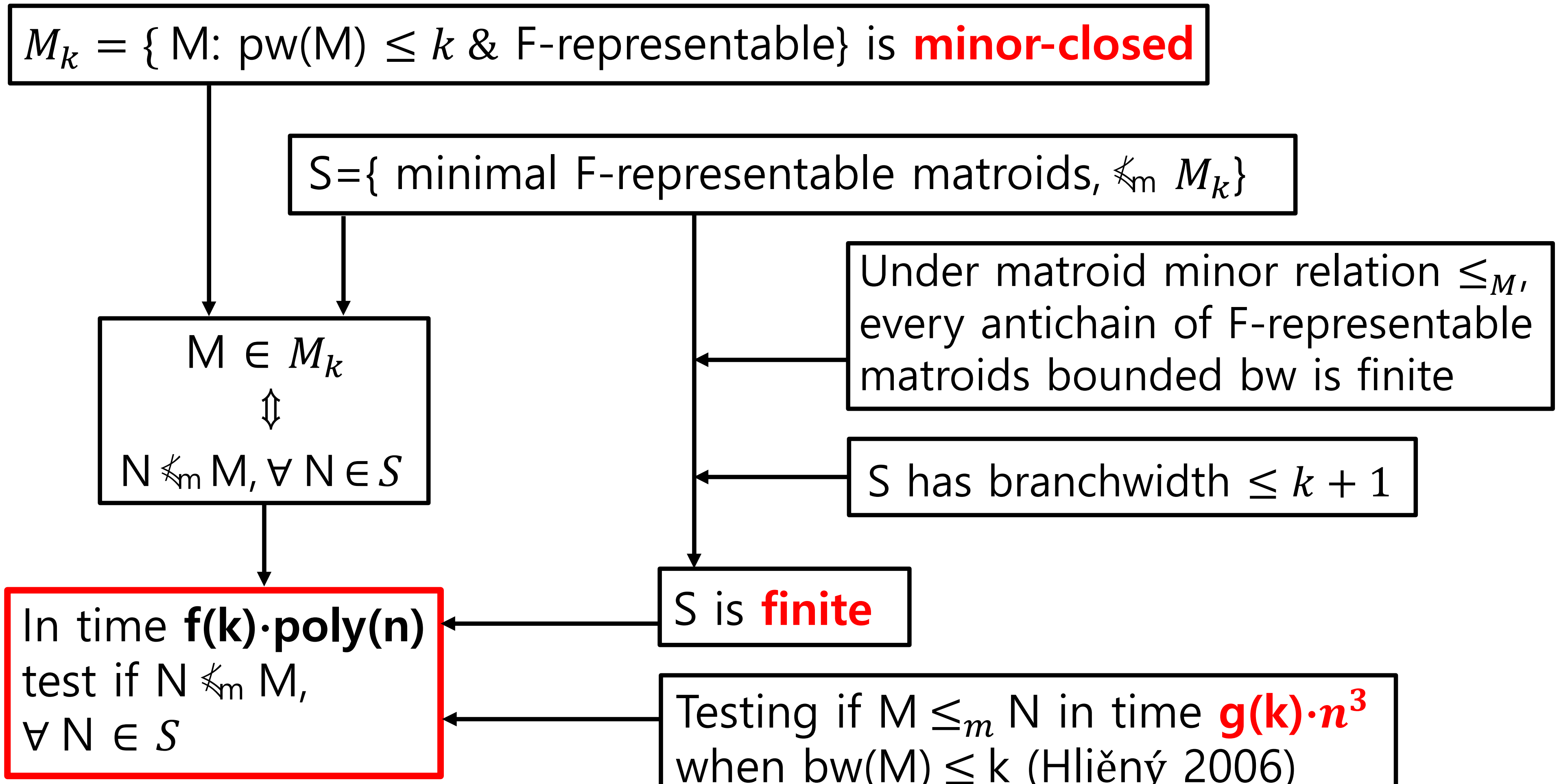
**Input :** a graph $G$, an integer $k$
**Output :** a tree-decomposition of $G$ of width at most $k$
**Time :** FPT $= f(k) \cdot poly(n)$

**Bodlaender and Thilikos (1997)**

**Input :** a graph $G$, an integer $k$
**Output :** a branch-decomposition of $G$ of width at most $k$
**Time :** FPT

- Thus, it is natural to ask a matroid path-width.

# Decision FPT algorithm for an F-representable matroid path-width $\leq k$



- Our results (SODA16)

**Constructive algorithm for path-width of matroids**

**Input :** a matroid (F-representable), an integer $k$
**Output :** a linear layout of path-width $\leq k$ if it exists
**Time :** FPT $= f(k) \cdot n^3$

- Note that this problem is NP-complete [Kashyap 2008].
- F is a *finite* field.

- Our results (SODA16)

**Constructive algorithm for path-width of matroids**

**Input :** a $n$-element matroid (F-representable) with its branch-decomposition of width $\theta$, an integer $k$
**Output :** a linear layout of path-width $\leq k$ if it exists
**Time :** $poly(\theta, |F|, k) \cdot |F|^{12\theta^2} \cdot 2^{1510\theta k} \cdot n$

To have branch-decomposition, we can use

- iterative compression, or
- Hliněný-Oum (2008) algorithm

## Constructive algorithm for path-width of vectors

**Input :** $n$ vectors over F, an integer $k$
**Output :** a permutation $v_1, v_2, \dots, v_n$ of $n$ vectors satisfying that for all $i$,

$$\dim\langle v_1, v_2, \dots, v_i\rangle \cap \langle v_{i+1}, v_{i+2}, \dots, v_n\rangle \le k.$$

**Time :** FPT $= f(k) \cdot n^3$

$$\begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array}$$
$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{array}\right)$$

3

$$\begin{array}{cccccc} v_1 & v_6 & v_2 & v_4 & v_5 & v_3 \end{array}$$
$$\left(\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right)$$

1

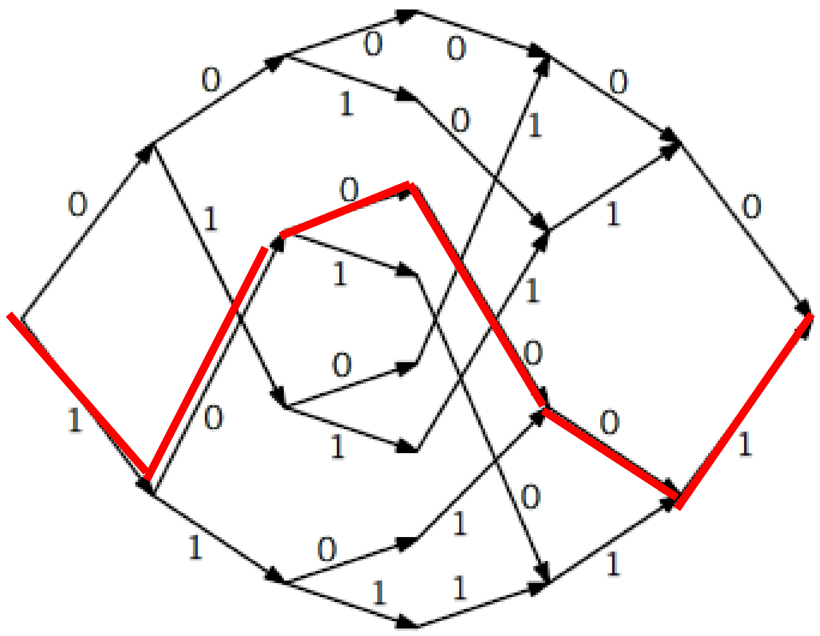
- Application to coding theory

Consider the linear code C that is generated by (100001), (010100), and (001010).

The generator matrix is $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$.

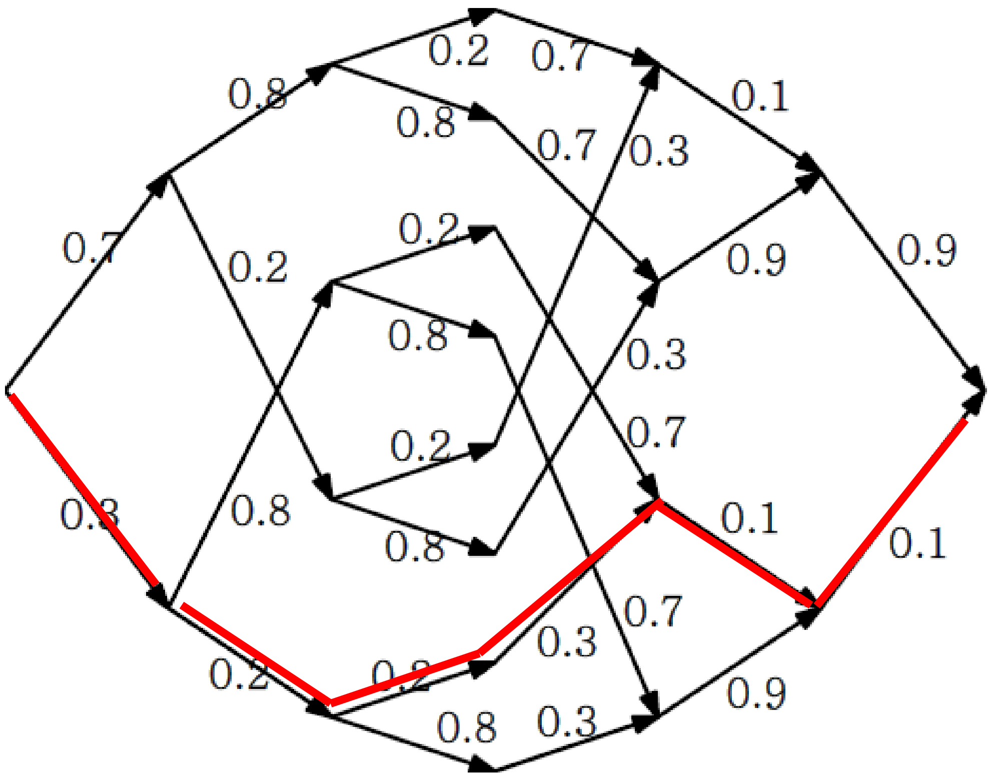
Codewords

{(000000), **(100001)**, (010100), (001010), (110101), (101011), (011110), (111111)}



trellis

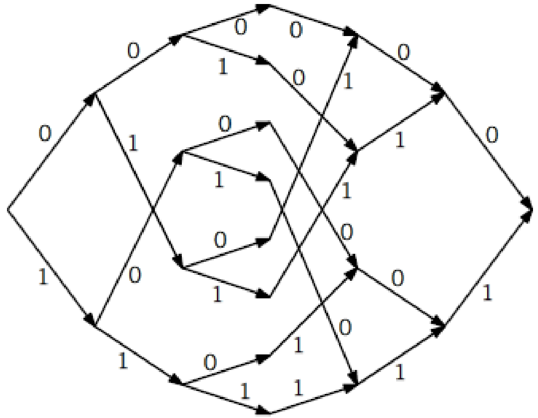
Decode using

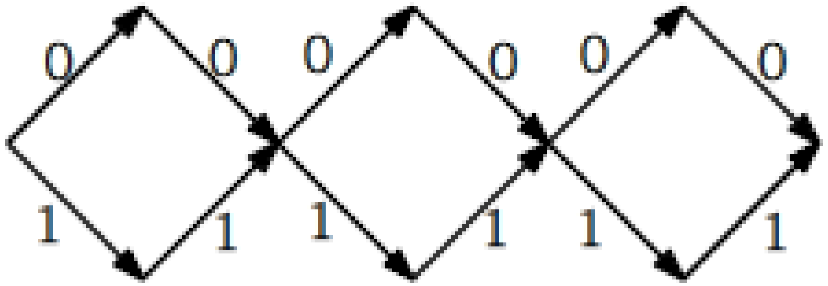



(110101)

(0.7 0.8 0.2 0.7 0.1 0.9)

{(000000), (100001), (010100), (001010), (110101), (101011), (011110), (111111)}





Want to make a better (thinner) trellis

Permute the columns of

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

$\pi = (1)(4)(5)(2,3,6)$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

# Proof ideas

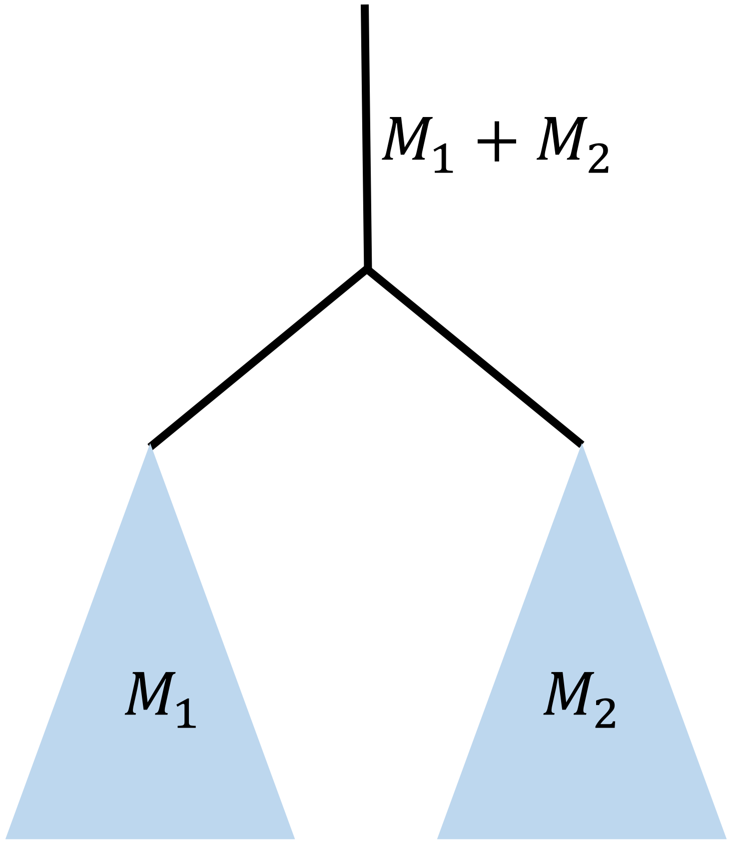
1. Dynamic programming  2. Typical sequences  3. Subspace analysis (linear algebra)

**Constructive algorithm for path-width of matroids**

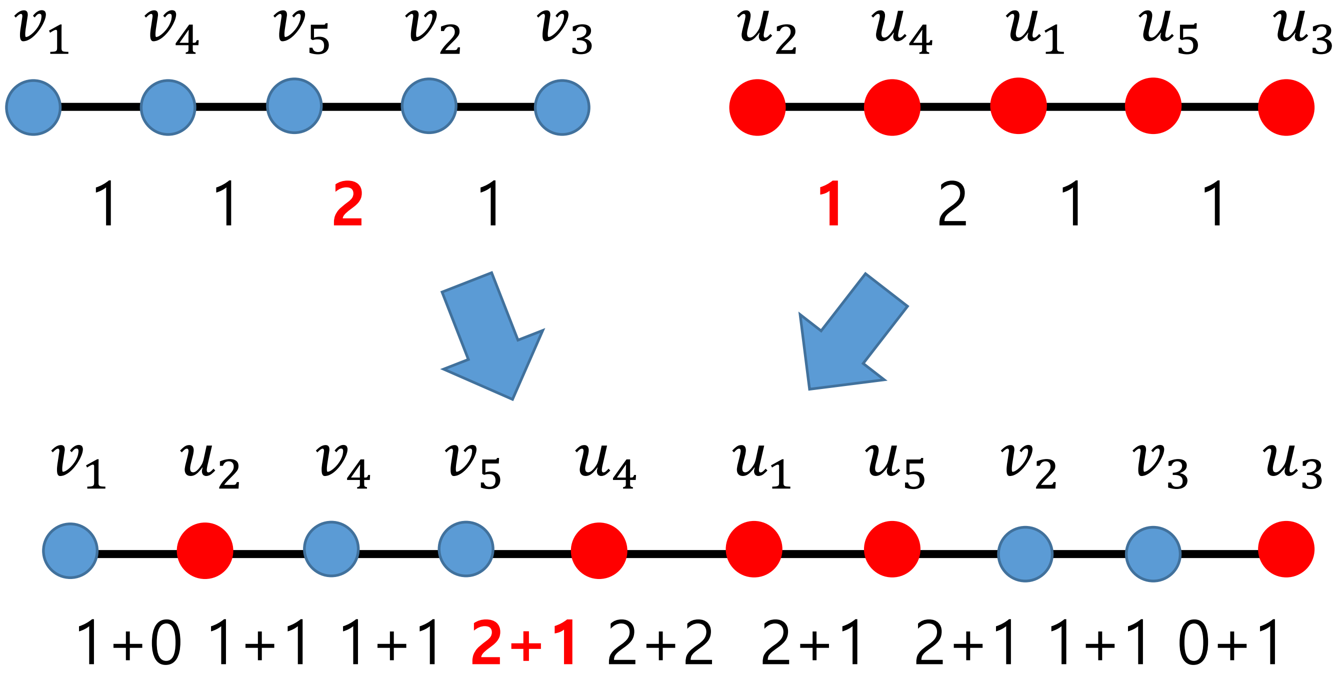
**Input :** a $n$-element matroid (F-representable) with its branch-decomposition of width θ, an integer $k$
**Output :** a linear layout of path-width $\leq k$ if it exists
**Time :** $poly(\theta, |F|, k) \cdot |F|^{12\theta^2} \cdot 2^{151\theta k} \cdot n$

# Proof ideas

1. Dynamic programming



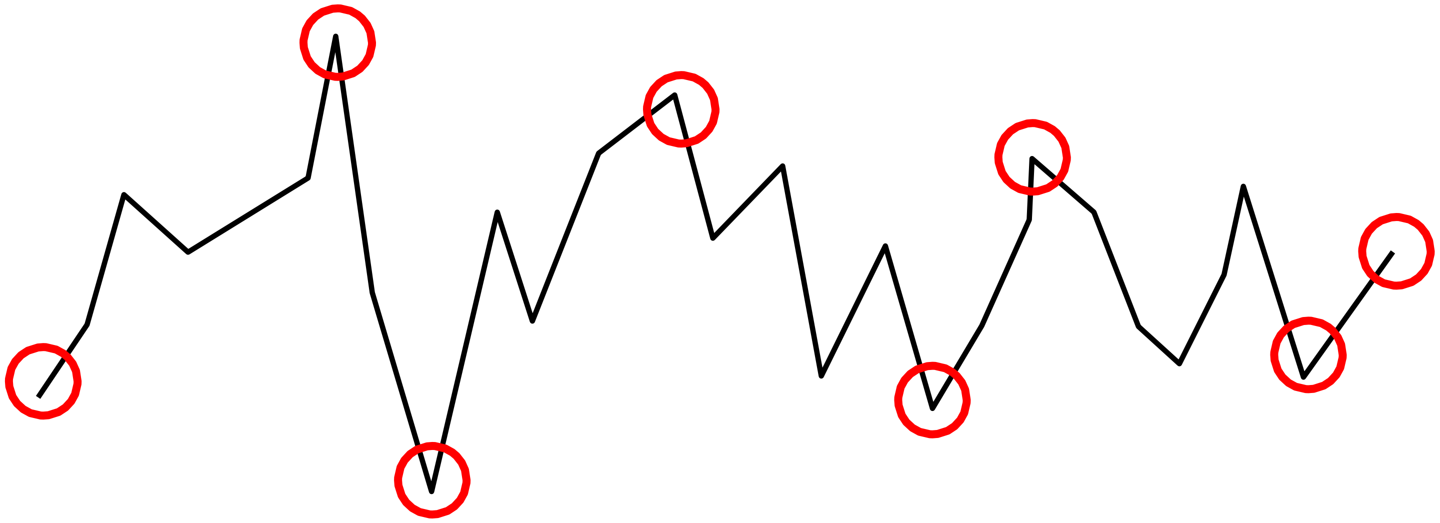

given branch-decomposition




Too huge to store

# Proof ideas

2. Typical sequences



3 6 9 0 5 4 8 5 6 2     3 9 0 5 4 8 5 6 2     3 9 0 8 5 6 2     3 9 0 8 2

**Lemma** (Bodlaender and Kloks, 1996).
There are at most $\frac{8}{3}2^{2k}$ distinct typical sequences consisting of $\{0,1,\ldots,k\}$.

# Proof ideas

3. Subspace analysis (linear algebra)

Let $X, Y, Z$ be subspaces of $F^r$.

1. If $X \subset Y$, then
$$\dim X - \dim X \cap Z \leq \dim Y - \dim Y \cap Z.$$

2. If $(X + Z) \cap (Y + Z) = Z$, then
$$(X \cap Z) + (Y \cap Z) = (X + Y) \cap Z.$$

Thus, our algorithm works correctly.

- Difficulties

1. A branch-decomposition of a matroid is given
   instead of a tree-decomposition of a graph.

2. For correctness,
   we need some properties (equalities) on subspaces.

- Difficulties

1. A branch-decomposition of a matroid is given instead of a tree-decomposition of a graph.

   A tree-decomposition has a bag (vertices).
   Our boundary is a subspace $\langle X \rangle \cap \langle E\text{-}X \rangle$.

   Recall that $\lambda(X) = \dim(\langle X \rangle \cap \langle E\text{-}X \rangle)$

- Actually, we proved a more general statement.
  \`vectors' → \`subspaces'  (vector = 1-dimensional subspace)

**Constructive algorithm for path-width of vectors**

**Input :** $n$ vectors over F, an integer $k$
**Output :** a permutation $v_1, v_2, \ldots, v_n$ of $n$ vectors satisfying that for all $i$,

$$\dim\langle v_1, v_2, \ldots, v_i\rangle \cap \langle v_{i+1}, v_{i+2}, \ldots, v_n\rangle \leq k.$$

**Time :** FPT $= f(k) \cdot n^3$

- Actually, we proved a more general statement.
  `vectors' → `subspaces'  (vector = 1-dimensional subspace)

**Constructive algorithm for path-width of subspaces**

**Input :** $n$ subspaces over F, an integer $k$
**Output :** a permutation $V_1, V_2, \dots, V_n$ of $n$ subspaces satisfying that for all $i$,

$$\dim\langle V_1, V_2, \dots, V_i\rangle \cap \langle V_{i+1}, V_{i+2}, \dots, V_n\rangle \leq k.$$

**Time :** FPT $= f(k) \cdot n^3$

## Constructive algorithm for path-width of subspaces

**Input :** $n$ subspaces over F, an integer $k$
**Output :** a permutation $V_1, V_2, \ldots, V_n$ of $n$ subspaces satisfying that for all $i$,

$$\dim\langle V_1, V_2, \ldots, V_i\rangle \cap \langle V_{i+1}, V_{i+2}, \ldots, V_n\rangle \leq k.$$

**Time :** FPT $= f(k) \cdot n^3$

- Let G be a graph and $e_1, e_2, \ldots, e_n$ be the standard basis.
- Let $$v_i = \sum_{j \text{ is adjacent to } i \text{ in } G} e_j.$$
- Let $V_i = < v_i, e_i >$.

# Constructive algorithm for linear rank-width of graphs

**Input :** a graph, an integer $k$
**Output :** a linear layout of **linear rank-width** at most $k$
**Time :** FPT $= f(k) \cdot n^3$

- Let G be a graph and $e_1, e_2, \ldots, e_n$ be the standard basis.
- Let $$v_i = \sum_{j \text{ is adjacent to } i \text{ in } G} e_j.$$
- Let $V_i = \langle v_i, e_i \rangle$.

# Our results (summary)

**Constructive algorithm for path-width of n vectors**

**Input :** n vectors
**Output :** a linear layout of path-width $\leq k$ if it exists
**Time :** $f(k) \cdot n^3$

# Our results

**Constructive algorithm for path-width of n subspaces**

**Input :** n subspaces
**Output :** a linear layout of path-width $\leq k$ if it exists
**Time :** $f(k) \cdot n^3$

$V_1, V_2, \cdots, V_n$
where $V_i$ =span($v_1, v_2, \cdots, v_j$)

**Constructive algorithm for path-width of matroids**

**Input :** matroid(F-representable)
**Output :** a linear layout of path-width $\leq k$ if it exists
**Time :** $f(k) \cdot n^3$

Matroid represented by
$v_1, v_2, \cdots, v_n$

# Our results

**Constructive algorithm for path-width of n subspaces**

**Input :** n subspaces
**Output :** a linear layout of path-width $\leq k$ if it exists
**Time :** $f(k) \cdot n^3$

**Constructive algorithm for trellis-width of linear codes**

**Input :** linear code
**Output :** a linear layout of trellis-width $\leq k$ if it exists
**Time :** $f(k) \cdot n^3$

**Constructive algorithm for path-width of matroids**

**Input :** matroid(F-representable)
**Output :** a linear layout of path-width $\leq k$ if it exists
**Time :** $f(k) \cdot n^3$

**Constructive algorithm for *linear rank-width* of graphs**

**Input :** graph
**Output :** a linear layout of linear rank-width $\leq k$ if it exists
**Time :** $f(k) \cdot n^3$

Linear code whose generator matrix is
$$(v_1 \quad v_2 \quad \cdots \quad v_n)$$

**Rank-width** is a width-parameter introduced by Oum and Seymour, which is equivalent to **clique-width**.

# Our results

**Constructive algorithm for path-width of n subspaces**

**Input :** n subspaces
**Output :** a linear layout of path-width $\leq k$ if it exists
**Time :** $f(k) \cdot n^3$

**Constructive algorithm for trellis-width of linear codes**

**Input :** linear code
**Output :** a linear layout of trellis-width $\leq k$ if it exists
**Time :** $f(k) \cdot n^3$

**Constructive algorithm for path-width of matroids**

**Input :** matroid(F-representable)
**Output :** a linear layout of path-width $\leq k$ if it exists
**Time :** $f(k) \cdot n^3$

**Constructive algorithm for linear rank-width of graphs**

**Input :** graph
**Output :** a linear layout of linear rank-width $\leq k$ if it exists
**Time :** $f(k) \cdot n^3$

**Exact algorithm for path-width of matroids**

**Input :** matroid (F-representable, **bounded branch-width**)
**Output :** path-width of given matroid
**Time :** poly(n)

**Exact algorithm for linear rank-width of graphs**

**Input :** graph of **bounded rw**
**Output :** linear rank-width of given graph
**Time :** poly(n)

**Approximation algorithm for linear clique-width of graphs**

**Input :** graph
**Output :** linear $(2^k + 1)$-expression of given graph
**Time :** $f(k) \cdot n^3$

| Constructive algorithm for path-width of n subspaces |
| --- |
| **Input :** n subspaces<br>**Output :** a linear layout of path-width $\leq k$ if it exists<br>**Time :** $f(k) \cdot n^3$ |

| Constructive algorithm for path-width of matroids |
| --- |
| **Input :** matroid(F-representable)<br>**Output :** a linear layout of path-width $\leq k$ if it exists<br>**Time :** $f(k) \cdot n^3$ |

## Further questions

1. FPT algorithms for path-width of **general** matroids
2. Can $O(n^3)$ factor in the running time improved?
   for example, $O(n^w)$? (w=matrix multiplication exponent)