離散最適化基礎論 第 13 回
固定パラメータ・アルゴリズムと木幅

岡本 吉央
okamotoy@uec.ac.jp

電気通信大学

2017 年 2 月 10 日

最終更新：2017 年 2 月 10 日　16:45

スケジュール 前半

1. 離散最適化における木分解の役割 (10/7)
- ⋆ 休講 (国内出張) (10/14)
2. 木に対するアルゴリズム設計 (10/23)
3. 道幅と道分解 (10/30)
4. 道分解を用いたアルゴリズム設計 (11/4)
- ⋆ 休講 (海外出張) (11/11)
5. 木分解と木幅 (11/18)
- ⋆ 休講 (調布祭) (11/25)
6. 木幅の性質 (12/2)

スケジュール 後半 (予定)

7	木分解を用いたアルゴリズム設計	(12/9)
8	木分解を用いたアルゴリズム設計：連結性	(12/16)
⋆	休講 (天皇誕生日)	(12/23)
⋆	冬季休業	(12/30)
9	木幅と論理：単項二階論理	(1/6)
⋆	休講 (センター試験準備)	(1/13)
10	木幅と論理：オートマトン	(1/20)
11	木幅と論理：アルゴリズム設計	(1/27)
12	木分解構成アルゴリズム	(2/3)
13	固定パラメータ・アルゴリズムと木幅	(2/10)
⋆	期末試験	(2/17)

主題

離散最適化のトピックの１つとして
グラフの木分解を取り上げ,

- 木分解とは何か？
- 木分解がなぜ役に立つのか？
- 木分解がどう役に立つのか？

について，数理的側面と計算的側面の双方を意識して講義する

なぜ講義で取り扱う？

- 「離散最適化の神髄」だから

この講義のキーワード (と荒っぽい説明)

グラフの木幅	グラフの「木っぽさ」を表す尺度 (の 1 つ)
グラフの木分解	グラフを「木っぽく」表した構造
動的計画法	木分解上の効率的アルゴリズム
オートマトン	動的計画法に基づくアルゴリズムの解釈
Courcelle の定理	上記と論理学に基づく『メタアルゴリズム』

次の主題が有機的に結びつく面白い話題

- グラフ
- アルゴリズム
- オートマトン
- 論理 (特に，有限モデル理論)

ポイント

効率的アルゴリズムが設計できる背景に「美しい数理構造」がある

この講義では，その一端に触れたい

目次

1 まとめ

2 禁止格子定理

3 固定パラメータ・アルゴリズムと木幅

4 今日のまとめ

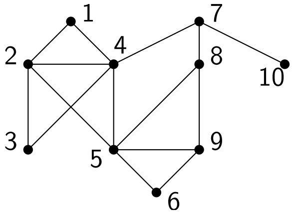
木分解 (tree decomposition) とは？

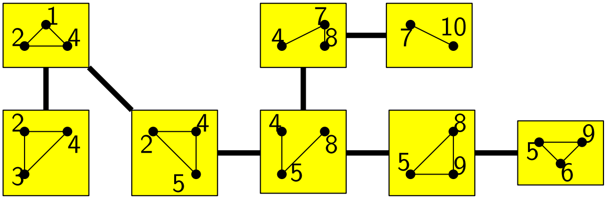
無向グラフ $G = (V, E)$ の木分解 とは木 $\mathcal{T}$ で，

(T1) $\mathcal{T}$ の節点はどれも V の部分集合

(T2) 各辺 $\{u, v\} \in E$ に対して，$u, v \in X$ となる $\mathcal{T}$ の節点 X が存在する

(T3) 各頂点 $v \in V$ に対して，$\mathcal{T}$ の節点で v を含むものは $\mathcal{T}$ の (連結で非空な) 部分木を誘導する





木分解の節点をバッグ (bag) と呼ぶことがある

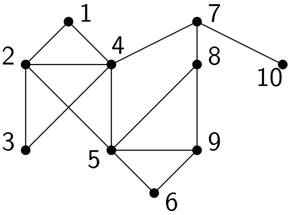
グラフの木幅とは？

- 無向グラフ G の木分解 $\mathcal{T}$ の幅 (width)

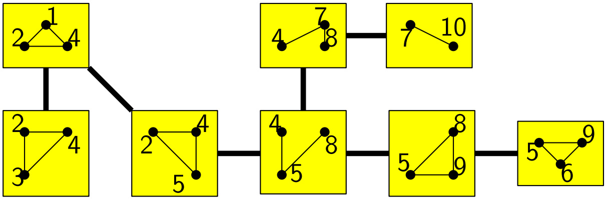
$$\mathtt{tw}(\mathcal{T}) = \max\{|S| - 1 \mid S \text{ は } \mathcal{T} \text{ の節点}\}$$

- 無向グラフ G の木幅 (treewidth)

$$\mathtt{tw}(G) = \min\{\mathtt{tw}(\mathcal{T}) \mid \mathcal{T} \text{ は } G \text{ の木分解}\}$$



$\mathtt{tw}(G) = 2$



Courcelle の定理 (1990)

無向グラフ G と単項二階論理式 ϕ が与えられたとき，
$G \models \phi$ となるか判定することは $O(f(\mathtt{tw}(G), |\phi|)|V|)$ 時間でできる

- ここで，$|\phi|$ は ϕ の長さ (記述長) で，f はある関数
- $\mathtt{tw}(G)$ と $|\phi|$ が定数ならば，この計算量は $O(|V|)$

Courcelle の定理の帰結

G の木幅が定数であれば，単項二階論理式で表現できる性質の判定は
線形時間でできる

重要な点

- 判定したい性質を単項二階論理式で書くだけで，自動的に，
  線形時間アルゴリズムが得られる
- 様々な性質に対するアルゴリズムを包括した「メタアルゴリズム」

単項二階論理式 ϕ が自由変数 $X \subseteq V$ を持つとする

Courcelle の定理 (1990) の系 (Arnborg, Lagergren, Seese '91)

無向グラフ G，単項二階論理式 $\phi(X)$ が与えられたとき，

$$\max\{|S| \mid S \subseteq V, G \models \phi(S)\}$$

という最適化問題は $O(f(\mathtt{tw}(G), |\phi|)|V|)$ 時間で解ける

- ここで，$|\phi|$ は ϕ の長さ (記述長) で，f はある関数
- $\mathtt{tw}(G)$ と $|\phi|$ が定数ならば，この計算量は $O(|V|)$
- 「max」を「min」に変えたバージョンも同じように解ける
- 頂点部分集合ではなく，辺部分集合に対するバージョンも同じように解ける
- 要素数最大化ではなく，重み付き要素数最大化も同じように解ける

今までの考察と Courcelle の定理から次が分かる

G の木幅が定数であるとき，

- 最大独立集合問題は線形時間で解ける
- 最小支配集合問題は線形時間で解ける
- ハミルトン閉路問題は線形時間で解ける

これらは，今までの講義で行ってきた
「動的計画法によるアルゴリズム」でも得られた

注意

「Courcelle の定理から得られるアルゴリズム」では
$O(|V|)$ の定数項が大きくなりがち (f が急激に増加しがち)

今までの考察と Courcelle の定理から次が分かる

G の木幅が定数であるとき，

- 最大独立集合問題は線形時間で解ける
- 最小支配集合問題は線形時間で解ける
- ハミルトン閉路問題は線形時間で解ける

Courcelle の定理：使い方

1. 解きたい問題に現れる性質を単項二階論理式で書いてみる
2. 書ければ，Courcelle の定理が使える
(ここまでで，木幅が定数のときの線形時間アルゴリズムが得られる)
3. しかし，実際のアルゴリズムは動的計画法を用いて設計する
(そうすると，より実用的な線形時間アルゴリズムが得られる)

第 7 回講義より

無向グラフ $G = (V, E)$ の最大独立集合の要素数は，
G の素敵な木分解 $\mathcal{T}$ が与えられていれば，
$O(2^t t^2 |V|)$ 時間で計算できる $(t = \mathtt{tw}(\mathcal{T}))$

アルゴリズムの設計方針

- 素敵な木分解の節点は次の 4 つに分類される
  - 葉，導入節点，忘却節点，結合節点
- そのそれぞれに対して，更新式 (再帰式) を立てる
- 葉の方から順に (ボトムアップに) 計算する

素敵な木分解 (nice tree decomposition) とは？

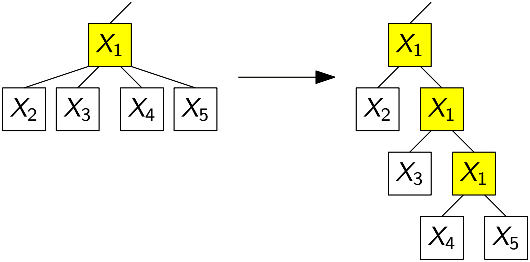
無向グラフ $G = (V, E)$ の木分解 $\mathcal{T}$ が素敵であるとは，
$\mathcal{T}$ の 1 つの節点 X_r を根として $\mathcal{T}$ を根付き木と見なしたときに
次を満たすこと

- $X_r = \emptyset$，かつ，葉である節点 X に対して，$X = \emptyset$
- 各節点の子の数は 2 以下
- 節点 X の子の数が 2 のとき，その子を X', X'' とすると，

$$X = X' = X''$$

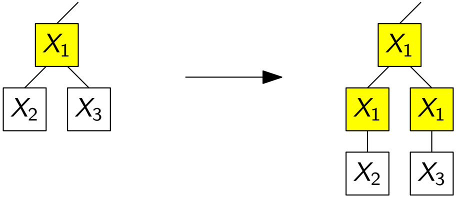
- 節点 X の子の数が 1 のとき，その子を X' とすると，
  次のどちらかが成立
  - ある頂点 $v \notin X'$ が存在して，$X = X' \cup \{v\}$
  - ある頂点 $w \in X'$ が存在して，$X = X' - \{w\}$

木分解から，同じ幅の素敵な木分解が (効率よく) 作れる



子の数が 3 以上のときは，この操作で子の数を 2 にする

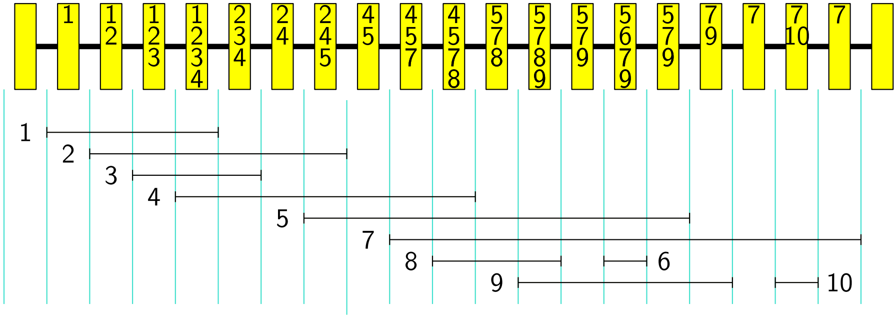
木分解から，同じ幅の素敵な木分解が (効率よく) 作れる



子の数が 2 のとき，親子が同じになるように変形する

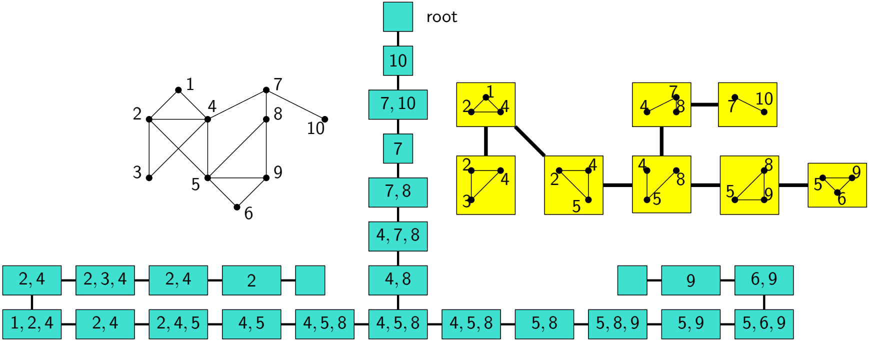
素敵な木分解：差分を減らす

木分解から，同じ幅の素敵な木分解が (効率よく) 作れる

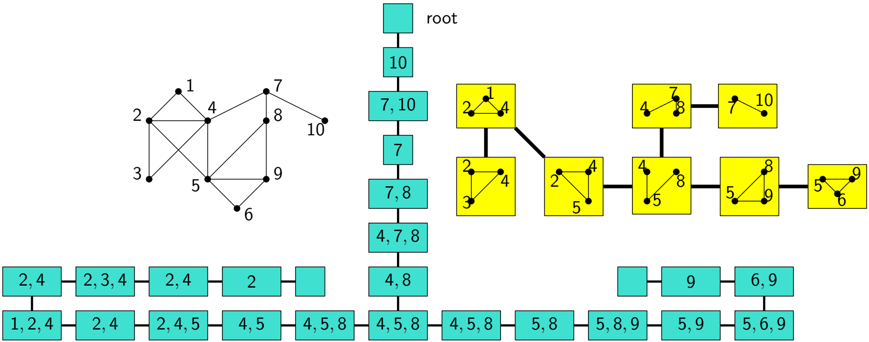


子の数が 1 のところは，素敵な道分解のときと同じように変形する

素敵な木分解：例


root
10
7, 10
7
7, 8
4, 7, 8
4, 8
2, 4
2, 3, 4
2, 4
2
9
6, 9
1, 2, 4
2, 4
2, 4, 5
4, 5
4, 5, 8
4, 5, 8
4, 5, 8
5, 8
5, 8, 9
5, 9
5, 6, 9

素敵な木分解：節点の種類



X の子の数	X の要素数は X の子の要素数より	
2	—	X は結合節点　(join node)
1	大きい	X は導入節点　(introduce node)
1	小さい	X は忘却節点　(forget node)
0	—	X は葉

固定パラメータ・アルゴリズムの研究もされている

定理 (Bodlaender '96)

入力として与えられたグラフ G に対して，$t^{O(t^3)}n$ 時間で次ができる

- $\mathtt{tw}(G) \le t$ ならば，幅 t の木分解を構成する
- $\mathtt{tw}(G) > t$ ならば,「$\mathtt{tw}(G) > t$」であると教えてくれる

ただし，n は G の頂点数

これは Courcelle の定理や動的計画法の文脈で重要

- Courcelle の定理や動的計画法を適用するとき，木分解が必要だから

定理 (Bodlaender '96)

入力として与えられたグラフ G に対して，$t^{O(t^3)}n$ 時間で次ができる

- $\mathtt{tw}(G) \leq t$ ならば，幅 t の木分解を構成する
- $\mathtt{tw}(G) > t$ ならば，「$\mathtt{tw}(G) > t$」であると教えてくれる

ただし，n は G の頂点数

第 7 回講義より

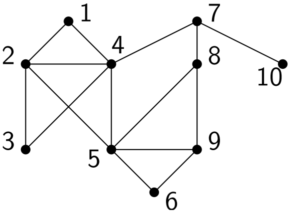
無向グラフ $G = (V, E)$ の最大独立集合の要素数は，
G の素敵な木分解 $\mathcal{T}$ が与えられていれば，
$O(2^t t^2 |V|)$ 時間で計算できる $(t = \mathtt{tw}(\mathcal{T}))$

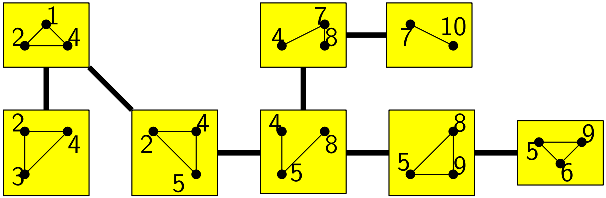
この 2 つをまとめると次がいえる

G の木幅が t であるとき，$t^{O(t^3)}n$ 時間で
G の最大独立集合の要素数が分かる (t を事前に知る必要はない)

疑問

木幅が大きなグラフに対して，ここまでの話は無力なのか？





回答

そうではない (場合がある)

目次

1 まとめ

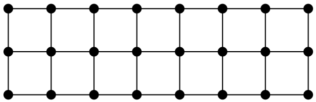
2 禁止格子定理

3 固定パラメータ・アルゴリズムと木幅

4 今日のまとめ

復習：格子とは？

$m \times n$ の格子 (grid) とは次のグラフのこと



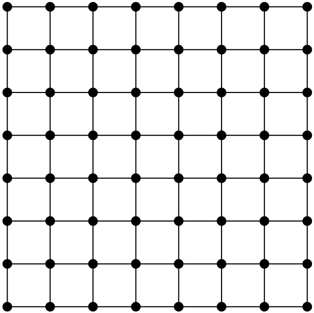
($m = 3,\ n = 8$ の場合)

$\boxed{\text{注意}}$：格子グラフ $\neq$ 格子

- 格子グラフとは格子の部分グラフのこと

格子の木幅

$k \times k$ 格子の木幅は k

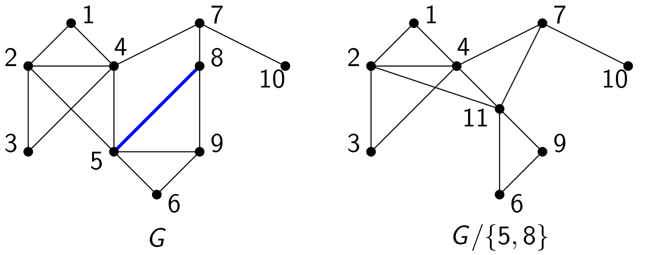


無向グラフ $G = (V, E)$，$e = \{u, v\} \in E$

辺の縮約 (contraction) とは？

頂点 u, v を削除し，新たな頂点 w を追加し，
次のように辺を追加してできるグラフ

$x \in V - \{u, v\}$ が
$\{x, u\} \in E$ または $\{x, v\} \in E$ を満たす　$\Rightarrow$　辺 $\{x, w\}$ を追加



この操作によってできるグラフを G/e と書く

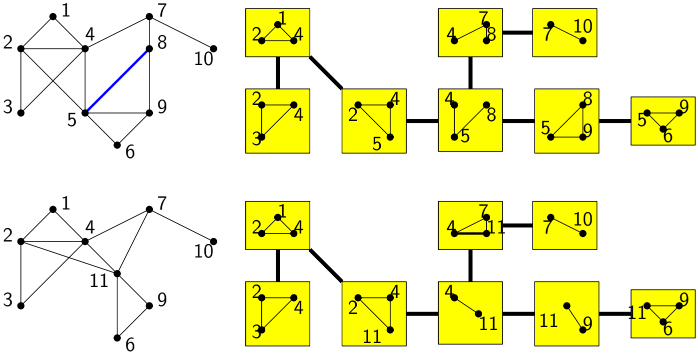
復習：辺の縮約と木幅

無向グラフ $G = (V, E)$，$e = \{u, v\} \in E$

命題：辺の縮約と木幅の関係

$\mathtt{tw}(G/e) \leq \mathtt{tw}(G)$

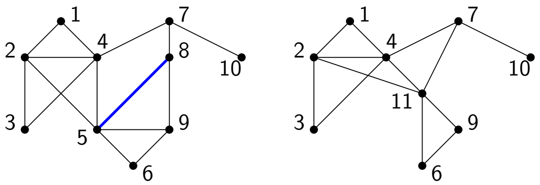
証明に対する直感



無向グラフ G, H

マイナーとは？

H が G のマイナーであるとは，
G の頂点や辺の除去，辺の縮約を繰り返しで H が得られること



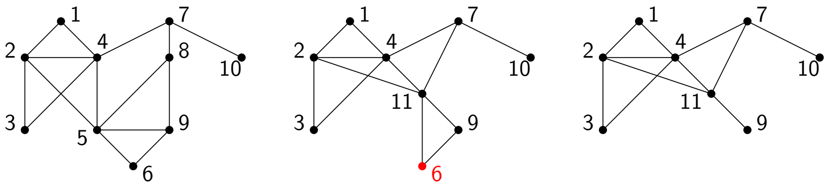
今までの議論の帰結

H が G のマイナー $\Rightarrow \mathtt{tw}(H) \leq \mathtt{tw}(G)$

無向グラフ G, H

マイナーとは？

H が G のマイナーであるとは，
G の頂点や辺の除去，辺の縮約を繰り返しで H が得られること



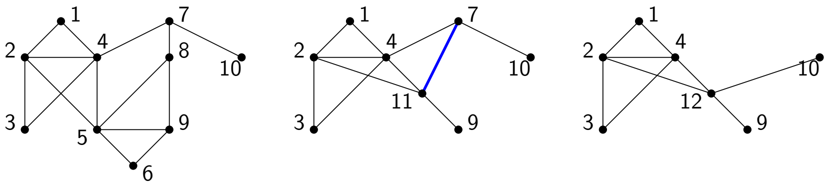
今までの議論の帰結

H が G のマイナー $\Rightarrow \mathtt{tw}(H) \leq \mathtt{tw}(G)$

無向グラフ G, H

マイナーとは？

H が G のマイナーであるとは，
G の頂点や辺の除去，辺の縮約を繰り返しで H が得られること



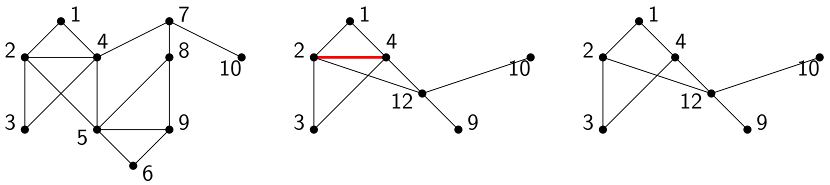
今までの議論の帰結

H が G のマイナー $\Rightarrow \texttt{tw}(H) \leq \texttt{tw}(G)$

無向グラフ G, H

マイナーとは？

H が G のマイナーであるとは，
G の頂点や辺の除去，辺の縮約を繰り返しで H が得られること



今までの議論の帰結

H が G のマイナー $\Rightarrow \mathtt{tw}(H) \leq \mathtt{tw}(G)$

格子マイナーと木幅

事実 1

$k \times k$ 格子の木幅は k

事実 2

H が G のマイナー $\Rightarrow \mathtt{tw}(G) \geq \mathtt{tw}(H)$

帰結

$k \times k$ 格子が G のマイナー $\Rightarrow \mathtt{tw}(G) \geq k$

事実 1

$k \times k$ 格子の木幅は k

事実 2

H が G のマイナー $\Rightarrow \mathtt{tw}(G) \geq \mathtt{tw}(H)$

帰結

$k \times k$ 格子が G のマイナー $\Rightarrow \mathtt{tw}(G) \geq k$

疑問：逆は成り立つか？

次は成り立つか？

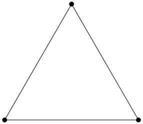
$\mathtt{tw}(G) \geq k \Rightarrow k \times k$ 格子が G のマイナー

回答：成り立たない

次は成り立つか？

$\mathtt{tw}(G) \geq k \Rightarrow k \times k$ 格子が G のマイナー

$\boxed{\text{反例}}$：完全グラフ K_3 を考える

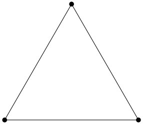


- $\mathtt{tw}(K_3) = 2$
- しかし，2×2 格子は K_3 のマイナーではない

次は成り立つか？

$\mathtt{tw}(G) \geq k \Rightarrow k \times k$ 格子が G のマイナー

$\boxed{\text{反例}}$：完全グラフ K_3 を考える



- $\mathtt{tw}(K_3) = 2$
- しかし，2×2 格子は K_3 のマイナーではない

より一般に，完全グラフ K_{k+1} を考えると

- $\mathtt{tw}(K_{k+1}) = k$
- しかし，$k \times k$ 格子は K_{k+1} のマイナーではない

頂点数を比べると，K_{k+1} の頂点数は $k+1$，$k \times k$ 格子の頂点数は k^2

次は成り立つか？

$\mathtt{tw}(G) \geq k \Rightarrow k \times k$ 格子が G のマイナー

- 回答：成り立たない
- 反例：完全グラフ

次は成り立つか？

$\mathtt{tw}(G) \geq k \Rightarrow k \times k$ 格子が G のマイナー

- 回答：成り立たない
- 反例：完全グラフ

では，次は成り立つか？

$\mathtt{tw}(G) \geq k^2 \Rightarrow k \times k$ 格子が G のマイナー

- 回答：成り立たない
- 反例：ランダムグラフ (Robertson, Seymour, Thomas '94)

次は成り立つか？

$\mathtt{tw}(G) \geq k \Rightarrow k \times k$ 格子が G のマイナー

- 回答：成り立たない
- 反例：完全グラフ

では，次は成り立つか？

$\mathtt{tw}(G) \geq k^2 \Rightarrow k \times k$ 格子が G のマイナー

- 回答：成り立たない
- 反例：ランダムグラフ (Robertson, Seymour, Thomas '94)

この「$\mathtt{tw}(G) \geq k$」,「$\mathtt{tw}(G) \geq k^2$」の右辺をどこまで大きくすれば「$k \times k$ 格子が G のマイナー」という結論を得ることができるのか？

禁止格子定理 (Excluded Grid Theorem)　(Robertson, Seymour '86)

ある関数 $f\colon \mathbb{N} \to \mathbb{N}$ が存在して,

$$\mathtt{tw}(G) \geq f(k) \Rightarrow k \times k \text{ 格子が } G \text{ のマイナー}$$

禁止格子定理 (Excluded Grid Theorem) (Robertson, Seymour '86)

ある関数 $f \colon \mathbb{N} \to \mathbb{N}$ が存在して，

$$\mathtt{tw}(G) \geq f(k) \Rightarrow k \times k \text{ 格子が } G \text{ のマイナー}$$

この f はどれほど大きいのか？ (小さいのか？)

- 大きすぎて分からない (Robertson, Seymour '86)

禁止格子定理 (Excluded Grid Theorem) (Robertson, Seymour '86)

ある関数 $f\colon \mathbb{N} \to \mathbb{N}$ が存在して，

$$\mathtt{tw}(G) \geq f(k) \Rightarrow k \times k \text{ 格子が } G \text{ のマイナー}$$

この f はどれほど大きいのか？ (小さいのか？)

- 大きすぎて分からない (Robertson, Seymour '86)
- $f(k) = 2^{O(k^5)}$ (Robertson, Seymour, Thomas '94)

禁止格子定理 (Excluded Grid Theorem) (Robertson, Seymour '86)

ある関数 $f\colon \mathbb{N} \to \mathbb{N}$ が存在して,

$$\mathtt{tw}(G) \geq f(k) \Rightarrow k \times k \text{ 格子が } G \text{ のマイナー}$$

この f はどれほど大きいのか？ (小さいのか？)

- 大きすぎて分からない (Robertson, Seymour '86)
- $f(k) = 2^{O(k^5)}$ (Robertson, Seymour, Thomas '94)
- $f(k) = 2^{O(k^2 \log k)}$ (Diestel, Jensen, Gorbunov, Thomassen '99)

禁止格子定理 (Excluded Grid Theorem) (Robertson, Seymour '86)

ある関数 $f\colon \mathbb{N} \to \mathbb{N}$ が存在して，

$$\mathtt{tw}(G) \geq f(k) \Rightarrow k \times k \text{ 格子が } G \text{ のマイナー}$$

この f はどれほど大きいのか？ (小さいのか？)

- 大きすぎて分からない (Robertson, Seymour '86)
- $f(k) = 2^{O(k^5)}$ (Robertson, Seymour, Thomas '94)
- $f(k) = 2^{O(k^2 \log k)}$ (Diestel, Jensen, Gorbunov, Thomassen '99)
- $f(k) = 2^{O(k^2/\log k)}$ (Kawarabayashi, Kobayashi '12; Leaf, Seymour '14)

禁止格子定理 (Excluded Grid Theorem) (Robertson, Seymour '86)

ある関数 $f\colon \mathbb{N} \to \mathbb{N}$ が存在して,

$$\mathtt{tw}(G) \geq f(k) \Rightarrow k \times k \text{ 格子が } G \text{ のマイナー}$$

この f はどれほど大きいのか？ (小さいのか？)

- 大きすぎて分からない (Robertson, Seymour '86)
- $f(k) = 2^{O(k^5)}$ (Robertson, Seymour, Thomas '94)
- $f(k) = 2^{O(k^2 \log k)}$ (Diestel, Jensen, Gorbunov, Thomassen '99)
- $f(k) = 2^{O(k^2/\log k)}$ (Kawarabayashi, Kobayashi '12; Leaf, Seymour '14)
- $f(k) = \tilde{O}(k^{98})$ (Chekuri, Chuzhoy '14)

禁止格子定理 (Excluded Grid Theorem) (Robertson, Seymour '86)

ある関数 $f\colon \mathbb{N} \to \mathbb{N}$ が存在して,

$$\mathtt{tw}(G) \geq f(k) \Rightarrow k \times k \text{ 格子が } G \text{ のマイナー}$$

この f はどれほど大きいのか？ (小さいのか？)

- 大きすぎて分からない (Robertson, Seymour '86)
- $f(k) = 2^{O(k^5)}$ (Robertson, Seymour, Thomas '94)
- $f(k) = 2^{O(k^2 \log k)}$ (Diestel, Jensen, Gorbunov, Thomassen '99)
- $f(k) = 2^{O(k^2 / \log k)}$ (Kawarabayashi, Kobayashi '12; Leaf, Seymour '14)
- $f(k) = \tilde{O}(k^{98})$ (Chekuri, Chuzhoy '14)
- $f(k) = \tilde{O}(k^{19})$ (Chuzhoy '15)

禁止格子定理 (Excluded Grid Theorem) (Robertson, Seymour '86)

ある関数 $f\colon \mathbb{N} \to \mathbb{N}$ が存在して,

$$\mathtt{tw}(G) \geq f(k) \Rightarrow k \times k \text{ 格子が } G \text{ のマイナー}$$

この f はどれほど大きいのか？ (小さいのか？)

- 大きすぎて分からない (Robertson, Seymour '86)
- $f(k) = 2^{O(k^5)}$ (Robertson, Seymour, Thomas '94)
- $f(k) = 2^{O(k^2 \log k)}$ (Diestel, Jensen, Gorbunov, Thomassen '99)
- $f(k) = 2^{O(k^2/\log k)}$ (Kawarabayashi, Kobayashi '12; Leaf, Seymour '14)
- $f(k) = \tilde{O}(k^{98})$ (Chekuri, Chuzhoy '14)
- $f(k) = \tilde{O}(k^{19})$ (Chuzhoy '15)

下界

- $f(k) = \Omega(k^2 \log k)$ (Robertson, Seymour, Thomas '94)

予想 (Demaine, Hajiaghayi, Kawarabayashi '09)

$f(k) = \Theta(k^3)$

禁止格子定理 (Excluded Grid Theorem)　(Robertson, Seymour '86)

ある関数 $f\colon \mathbb{N} \to \mathbb{N}$ が存在して，

$$\mathtt{tw}(G) \geq f(k) \Rightarrow k \times k \text{ 格子が } G \text{ のマイナー}$$

禁止格子定理の使い方

グラフ G に対して，ある問題を解こうとするとき

1. $\mathtt{tw}(G) \geq f(k)$ かどうか，判定する
2. $\mathtt{tw}(G) \leq f(k)$ ならば　(つまり，$\mathtt{tw}(G)$ が小さい)
   Courcelle の定理や動的計画法を用いて解く
3. $\mathtt{tw}(G) \geq f(k)$ ならば　(つまり，$\mathtt{tw}(G)$ が大きい)
   $k \times k$ 格子が G のマイナーであることを利用して，解く

具体例を次に見る

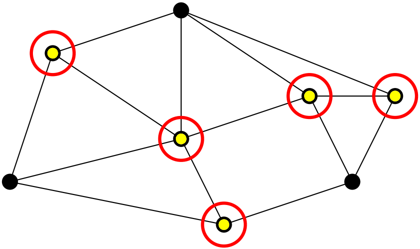
目次

1. まとめ
2. 禁止格子定理
3. 固定パラメータ・アルゴリズムと木幅
4. 今日のまとめ

$G = (V, E)$ 無向グラフ

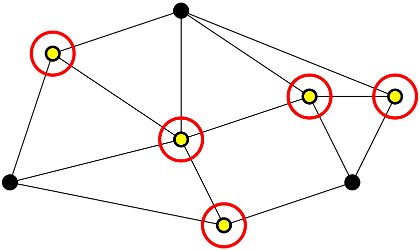
頂点被覆とは？

G の頂点被覆 (vertex cover) とは，
頂点部分集合 $C \subseteq V$ で，G のどの辺も C の頂点に接続するもの



最小頂点被覆問題とは？

- 入力：無向グラフ $G = (V, E)$
- 出力：G の最小頂点被覆 (の要素数)

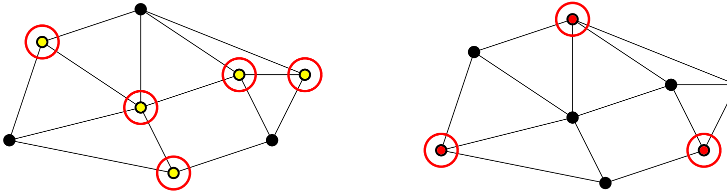


これは NP 困難問題 (解きにくい問題)

無向グラフ $G = (V, E)$

性質

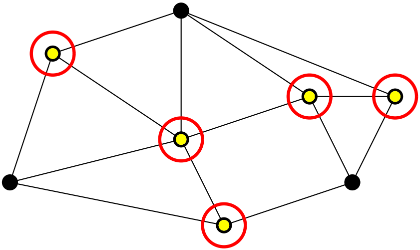
$C \subseteq V$ が G の頂点被覆 $\Leftrightarrow$ $V - C$ が G の独立集合



つまり，最小頂点被覆問題が解ければ，最大独立集合問題も解ける

最小頂点被覆問題 (判定版) とは？

- 入力：無向グラフ $G = (V, E)$，自然数 k
- 出力：G が要素数 k 以下の頂点被覆を持つ $\Rightarrow$ Yes
  そうでない $\Rightarrow$ No



これが解ければ，最小頂点被覆問題が解ける (例えば，二分探索による)

最小頂点被覆問題 (判定版) とは？

- 入力：無向グラフ $G = (V, E)$，自然数 k
- 出力：G が要素数 k 以下の頂点被覆を持つ $\Rightarrow$ Yes
  そうでない $\Rightarrow$ No

禁止格子定理を用いるアルゴリズム

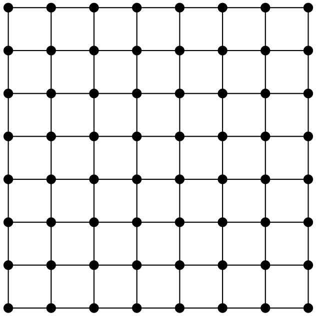
自然数 h は，k を用いてうまく定める (後で解説)

1. $\mathtt{tw}(G) \geq f(h)$ かどうか，判定する
2. $\mathtt{tw}(G) \leq f(h)$ ならば (つまり，$\mathtt{tw}(G)$ が小さい)
   Courcelle の定理や動的計画法を用いて解く
3. $\mathtt{tw}(G) \geq f(h)$ ならば (つまり，$\mathtt{tw}(G)$ が大きい)
   $h \times h$ 格子が G のマイナーであることを利用して，解く

注：「C が頂点被覆である」という性質は単項二階論理を用いて書けるので，Courcelle の定理が使える

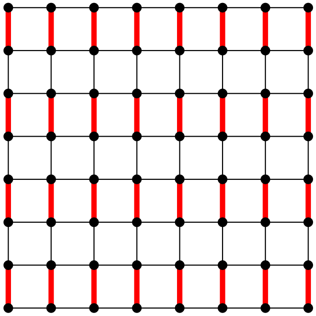
格子と頂点被覆

$h \times h$ 格子が G の部分グラフであるとする



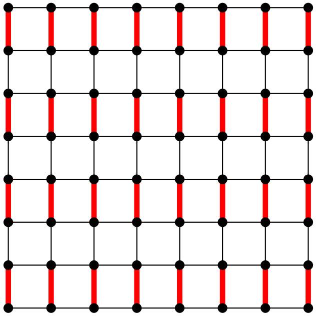
格子と頂点被覆

$h \times h$ 格子が G の部分グラフであるとする



- $h \times h$ 格子は要素数 $h \cdot \lfloor h/2 \rfloor$ のマッチングを持つ

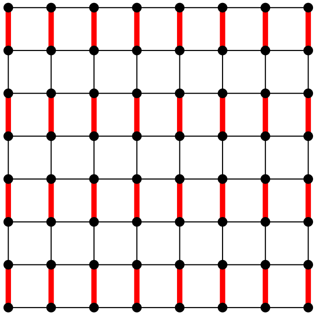
$h \times h$ 格子が G の部分グラフであるとする



- $h \times h$ 格子は要素数 $h \cdot \lfloor h/2 \rfloor$ のマッチングを持つ
- つまり，G の頂点被覆の要素数 $\geq h \cdot \lfloor h/2 \rfloor$

格子と頂点被覆

$h \times h$ 格子が G の部分グラフであるとする



- $h \times h$ 格子は要素数 $h \cdot \lfloor h/2 \rfloor$ のマッチングを持つ
- つまり，G の頂点被覆の要素数 $\geq h \cdot \lfloor h/2 \rfloor$

帰結 1

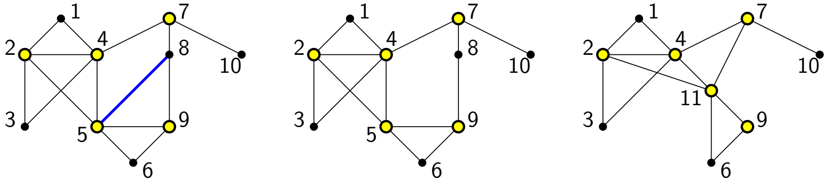
$\lceil \sqrt{2k} + 1 \rceil \times \lceil \sqrt{2k} + 1 \rceil$ 格子が G の部分グラフ $\Rightarrow$
G の最小頂点被覆の頂点数 $> k$

G の 1 辺を除去して H を得ると

$$G \text{ の最小頂点被覆の頂点数} \geq H \text{ の最小頂点被覆の頂点数}$$

G の 1 辺を縮約して H を得ると

$$G \text{ の最小頂点被覆の頂点数} \geq H \text{ の最小頂点被覆の頂点数}$$



G $\quad$ $G - \{5,8\}$ $\quad$ $G/\{5,8\}$

帰結 2

H が G のマイナー，H の最小頂点被覆の頂点数 $\geq k$
$\Rightarrow$ G の最小頂点被覆の頂点数 $\geq k$

格子マイナーと頂点被覆

帰結 1

$\lceil\sqrt{2k}+1\rceil \times \lceil\sqrt{2k}+1\rceil$ 格子が G の部分グラフ $\Rightarrow$
G の最小頂点被覆の頂点数 $> k$

帰結 2

H が G のマイナー，H の最小頂点被覆の頂点数 $\geq k$
$\Rightarrow$ G の最小頂点被覆の頂点数 $\geq k$

格子マイナーと頂点被覆

帰結 1

$\lceil\sqrt{2k}+1\rceil \times \lceil\sqrt{2k}+1\rceil$ 格子が G の部分グラフ $\Rightarrow$
G の最小頂点被覆の頂点数 $> k$

帰結 2

H が G のマイナー，H の最小頂点被覆の頂点数 $\geq k$
$\Rightarrow$ G の最小頂点被覆の頂点数 $\geq k$

帰結 1 と帰結 2 から得られる帰結

$\lceil\sqrt{2k}+1\rceil \times \lceil\sqrt{2k}+1\rceil$ 格子が G のマイナー
$\Rightarrow$ G の最小頂点被覆の頂点数 $> k$

これが使える

最小頂点被覆問題 (判定版) とは？

- 入力：無向グラフ $G = (V, E)$，自然数 k
- 出力：G が要素数 k 以下の頂点被覆を持つ $\Rightarrow$ Yes
  そうでない $\Rightarrow$ No

禁止格子定理を用いるアルゴリズム

$h = \lceil \sqrt{2k} + 1 \rceil$ とする

1. $\mathtt{tw}(G) \geq f(h)$ かどうか，判定する
2. $\mathtt{tw}(G) \leq f(h)$ ならば　(つまり，$\mathtt{tw}(G)$ が小さい)
   Courcelle の定理や動的計画法を用いて解く
3. $\mathtt{tw}(G) \geq f(h)$ ならば　(つまり，$\mathtt{tw}(G)$ が大きい)
   $h \times h$ 格子が G のマイナーであるので，No を出力

これは正しいアルゴリズム

禁止格子定理を用いるアルゴリズム

$h = \lceil \sqrt{2k} + 1 \rceil$ とする $(f(h) = \tilde{O}(h^{19}))$

1. $\mathtt{tw}(G) \geq f(h)$ かどうか，判定する
2. $\mathtt{tw}(G) \leq f(h)$ ならば (つまり，$\mathtt{tw}(G)$ が小さい)
   Courcelle の定理や動的計画法を用いて解く
3. $\mathtt{tw}(G) \geq f(h)$ ならば (つまり，$\mathtt{tw}(G)$ が大きい)
   $h \times h$ 格子が G のマイナーであるので，No を出力

計算量：$n = |V|$ とする

1. Bodlaender のアルゴリズムを使えば，$f(h)^{O(f(h)^3)}n$ 時間
2. 動的計画法を使えば，$O(2^{f(h)}f(h)^2 n)$ 時間
3. 何もせずに No を出力するので，$O(1)$ 時間

$\therefore f(h)^{O(f(h)^3)}n = k^{\tilde{O}(k^{28.5})}n = 2^{\tilde{O}(k^{28.5})}n$ 時間

ポイント

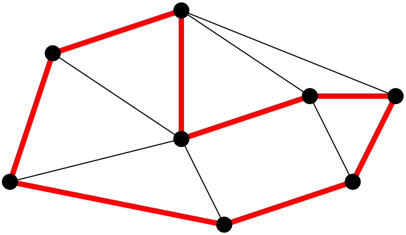
最小頂点被覆問題は次の性質を持っていた

1. 格子に対して，最小頂点被覆の要素数が大きい
2. マイナーの最小頂点被覆の要素数が大きい $\Rightarrow$
   もとのグラフの最小頂点被覆の要素数も大きい
3. 木幅が小さい $\Rightarrow$ 最小頂点被覆問題は効率よく解ける

この 3 つと同じような性質を持っていれば，
禁止格子定理を用いることで，いろいろな問題が解けそう

最長閉路問題とは？

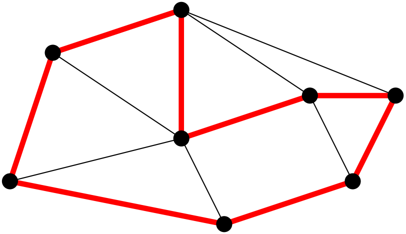
- 入力：無向グラフ $G = (V, E)$
- 出力：G の最長閉路 (の長さ)



これは NP 困難問題 (解きにくい問題)

最長閉路問題 (判定版) とは？

- 入力：無向グラフ $G = (V, E)$，自然数 k
- 出力：G が長さ k 以上の閉路を持つ $\Rightarrow$ Yes
  そうでない $\Rightarrow$ No



これが解ければ，最長閉路問題も解ける (例えば，二分探索による)

最長閉路問題 (判定版) とは？

- 入力：無向グラフ $G = (V, E)$，自然数 k
- 出力：G が長さ k 以上の閉路を持つ $\Rightarrow$ Yes
  そうでない $\Rightarrow$ No

禁止格子定理を用いるアルゴリズム

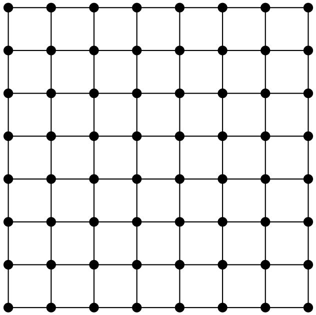
自然数 h は，k を用いてうまく定める (後で解説)

1. $\mathtt{tw}(G) \geq f(h)$ かどうか，判定する
2. $\mathtt{tw}(G) \leq f(h)$ ならば　(つまり，$\mathtt{tw}(G)$ が小さい)
   Courcelle の定理や動的計画法を用いて解く
3. $\mathtt{tw}(G) \geq f(h)$ ならば　(つまり，$\mathtt{tw}(G)$ が大きい)
   $h \times h$ 格子が G のマイナーであることを利用して，解く

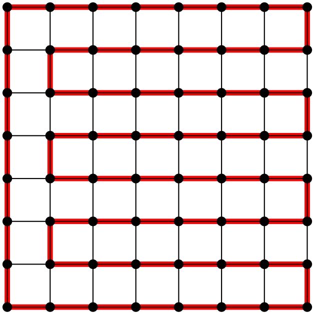
注：「C が閉路である」という性質は単項二階論理を用いて書けるので，
Courcelle の定理が使える

格子と閉路

$h \times h$ 格子が G の部分グラフであるとする

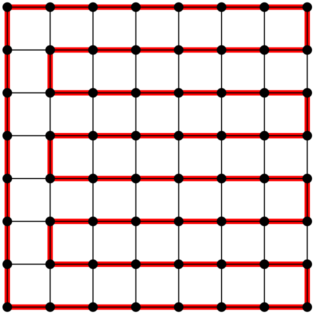


$h \times h$ 格子が G の部分グラフであるとする



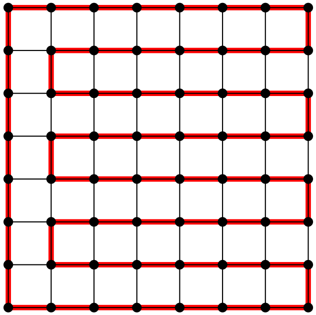
- $h \times h$ 格子は長さ $h(h-1)$ 以上の閉路を持つ

$h \times h$ 格子が G の部分グラフであるとする



- $h \times h$ 格子は長さ $h(h-1)$ 以上の閉路を持つ
- つまり，G の最長閉路の長さ $\geq h(h-1)$

$h \times h$ 格子が G の部分グラフであるとする



- $h \times h$ 格子は長さ $h(h-1)$ 以上の閉路を持つ
- つまり，G の最長閉路の長さ $\geq h(h-1)$

帰結 1

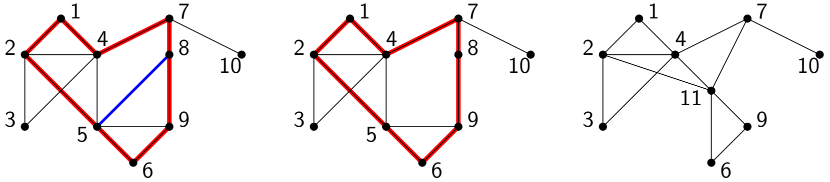
$\lceil \sqrt{k}+1 \rceil \times \lceil \sqrt{k}+1 \rceil$ 格子が G の部分グラフ $\Rightarrow$
G の最長閉路の長さ $\geq k$

G の 1 辺を除去して H を得ると

$$G \text{ の最長閉路の長さ} \geq H \text{ の最長閉路の長さ}$$

G の 1 辺を縮約して H を得ると

$$G \text{ の最長閉路の長さ} \geq H \text{ の最長閉路の長さ}$$



G $\quad$ $G - \{5, 8\}$ $\quad$ $G/\{5, 8\}$

帰結 2

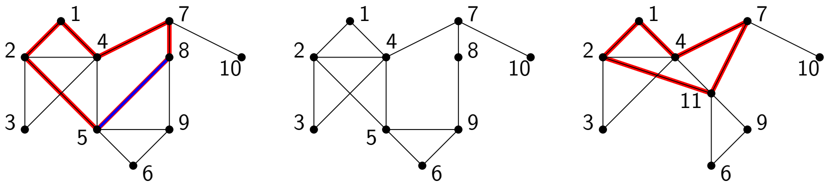
H が G のマイナー，H の最長閉路の長さ $\geq k$
$\Rightarrow$ G の最長閉路の長さ $\geq k$

G の 1 辺を除去して H を得ると

$$G \text{ の最長閉路の長さ} \geq H \text{ の最長閉路の長さ}$$

G の 1 辺を縮約して H を得ると

$$G \text{ の最長閉路の長さ} \geq H \text{ の最長閉路の長さ}$$



G $\qquad$ $G - \{5, 8\}$ $\qquad$ $G/\{5, 8\}$

帰結 2

H が G のマイナー，H の最長閉路の長さ $\geq k$
$\Rightarrow$ G の最長閉路の長さ $\geq k$

帰結 1

$\lceil\sqrt{k}+1\rceil \times \lceil\sqrt{k}+1\rceil$ 格子が G の部分グラフ $\Rightarrow$
G の最長閉路の長さ $\geq k$

帰結 2

H が G のマイナー，H の最長閉路の長さ $\geq k$
$\Rightarrow$ G の最長閉路の長さ $\geq k$

帰結 1

$\lceil\sqrt{k}+1\rceil \times \lceil\sqrt{k}+1\rceil$ 格子が G の部分グラフ $\Rightarrow$
G の最長閉路の長さ $\geq k$

帰結 2

H が G のマイナー，H の最長閉路の長さ $\geq k$
$\Rightarrow$ G の最長閉路の長さ $\geq k$

帰結 1 と帰結 2 から得られる帰結

$\lceil\sqrt{k}+1\rceil \times \lceil\sqrt{k}+1\rceil$ 格子が G のマイナー
$\Rightarrow$ G の最大閉路の長さ $\geq k$

これが使える

最長閉路問題 (判定版) とは？

- 入力：無向グラフ $G = (V, E)$，自然数 k
- 出力：G が長さ k 以上の閉路を持つ $\Rightarrow$ Yes
  そうでない $\Rightarrow$ No

禁止格子定理を用いるアルゴリズム

$h = \lceil \sqrt{k} + 1 \rceil$ とする

1. $\mathtt{tw}(G) \geq f(h)$ かどうか，判定する
2. $\mathtt{tw}(G) \leq f(h)$ ならば　(つまり，$\mathtt{tw}(G)$ が小さい)
   Courcelle の定理や動的計画法を用いて解く
3. $\mathtt{tw}(G) \geq f(h)$ ならば　(つまり，$\mathtt{tw}(G)$ が大きい)
   $h \times h$ 格子が G のマイナーであるので，Yes を出力

これは正しいアルゴリズム (計算量：k が定数 $\Rightarrow O(|V|)$)

注意

「禁止格子定理を用いるアルゴリズム」は実用的ではない

- 禁止格子定理は，効率的アルゴリズムの存在性を教えてくれるだけ
- 実際のアルゴリズムは別の方法を使って設計する

例

- 最小頂点被覆問題 (判定版)：$O(1.2738^k + kn)$ 時間
  (Chen, Kanj, Xia '10)

目次

1. まとめ
2. 禁止格子定理
3. 固定パラメータ・アルゴリズムと木幅
4. 今日のまとめ

今日のまとめ

- 木幅が大きいときは，禁止格子定理を利用できる
- そのとき，格子をマイナーとして持つ場合の性質が重要である

これを突き詰めていくと

- グラフマイナー理論 (Graph Minor Theory)
- 双次元性理論 (Bidimensionality Theory)

へ進んでいく

残った時間の使い方

- 授業評価アンケート
- 小さな紙に感想など書いて提出する ← 重要
  - 内容は何でも OK
  - 匿名で OK

目次

1. まとめ
2. 禁止格子定理
3. 固定パラメータ・アルゴリズムと木幅
4. 今日のまとめ