

グラフとネットワーク 第5回
マッチング：モデル化

岡本 吉央
okamotoy@uec.ac.jp

電気通信大学

2017年5月8日

最終更新：2017年5月5日 16:20

スケジュール 前半 (予定)

- | | | |
|---|--------------|--------|
| 1 | グラフの定義と次数：数理 | (4/10) |
| 2 | 道と閉路：数理 | (4/17) |
| 3 | 木：数理 | (4/24) |
| 4 | マッチング：数理 | (5/1) |
| 5 | マッチング：モデル化 | (5/8) |
| 6 | 最大流：数理 | (5/15) |
| 7 | 最大流：モデル化 (1) | (5/22) |
| 8 | 最大流：モデル化 (2) | (5/29) |
| 9 | 連結性：数理とモデル化 | (6/5) |
| | ● 中間試験 | (6/12) |

注意：予定の変更もありうる

スケジュール 後半 (予定)

- | | | |
|----|---------------|--------|
| 10 | 彩色：数理 | (6/19) |
| 11 | 彩色：モデル化 | (6/26) |
| * | 休講 (出張) | (7/3) |
| 12 | 平面グラフ：数理 | (7/10) |
| * | 休講 (海の日) | (7/17) |
| 13 | 平面グラフ：モデル化 | (7/24) |
| 14 | 予備日 (たぶんやらない) | (7/31) |
| ● | 期末試験 | (8/7) |

注意：予定の変更もありうる

主題

離散最適化の入門として、次を概説する

- ▶ グラフとネットワークを用いた数理モデル化
- ▶ アルゴリズム的解法の背後にある数理

キャッチフレーズ：「本当の離散数学がここから始まる」

達成目標

以下の4項目をすべて達成すること

- 1 グラフやネットワークに関する用語を正しく使うことができる
- 2 現実世界の諸問題をグラフやネットワークで表現し、数理モデルを構築できる
- 3 アルゴリズム的解法の背後にある数理、特に、最小最大定理の重要性を説明でき、それを用いて最適性の証明ができる
- 4 グラフとネットワークに関する簡単な数学的事実を証明できる

格言

- ▶ 「応用」ということばの意味は非常にあいまい
 - ▶ 「応用」というときは、学問の階層を意識する
-
- ▶ 研究には段階がある
 - ▶ 基礎研究の応用先は別の基礎研究かもしれない
 - ▶ 基礎研究の応用先は応用研究かもしれない
 - ▶ 応用研究の応用先は別の応用研究かもしれない
 - ▶ 応用研究の応用先は基礎研究かもしれない
 - ▶ 「応用」は実世界応用を意味しないかもしれない

今日の目標

最大マッチングを使って問題を解決する例を見る

- ▶ 除雪車の運行計画問題 (最小費用オイラーグラフ化)
- ▶ スリザーの必勝戦略

目次

- ① 最大マッチングと最大重みマッチング
- ② 除雪車の運行計画問題
- ③ スリザーの必勝戦略
- ④ 今日のまとめ

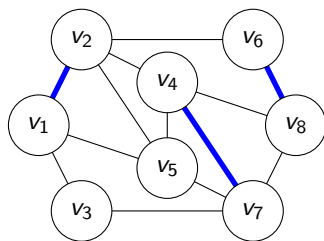
グラフにおけるマッチング

無向グラフ $G = (V, E)$

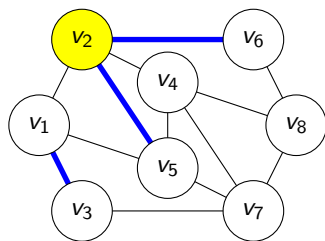
定義：マッチングとは？

(復習)

G の **マッチング** とは辺部分集合 $M \subseteq E$ で、
 M のどの2辺も同じ頂点に接続しないもの



$\{\{v_1, v_2\}, \{v_4, v_7\}, \{v_6, v_8\}\}$ は
 マッチングである



$\{\{v_1, v_3\}, \{v_2, v_5\}, \{v_2, v_6\}\}$ は
 マッチングではない

マッチングの辺 $e \in M$ は e の端点を **飽和** する

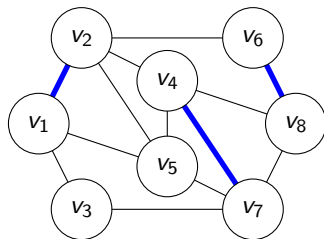
最大マッチング

無向グラフ $G = (V, E)$

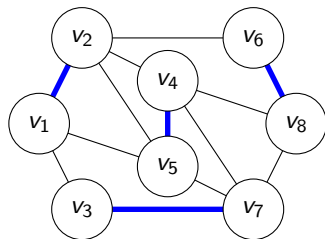
定義：最大マッチングとは？

(復習)

G の**最大マッチング**とは G のマッチング $M \subseteq E$ で、
 G の任意のマッチング M' に対して $|M| \geq |M'|$ を満たすもの



最大マッチングではない



最大マッチングである

最大重みマッチング

無向グラフ $G = (V, E)$

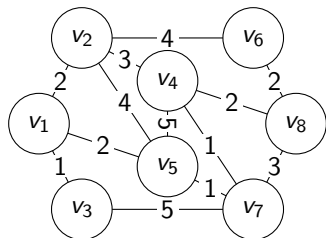
各辺 $e \in E$ に対する非負重み $w(e) \geq 0$ (辺重み関数 $w: E \rightarrow \mathbb{R}$)

定義：最大重みマッチングとは？

w に関する G の最大重みマッチングとは

G のマッチング $M \subseteq E$ で、

G の任意のマッチング M' に対して $\sum_{e \in M} w(e) \geq \sum_{e \in M'} w(e)$ を満たすもの



以後, $w(M) = \sum_{e \in M} w(e)$ と書く

最大重みマッチング

無向グラフ $G = (V, E)$

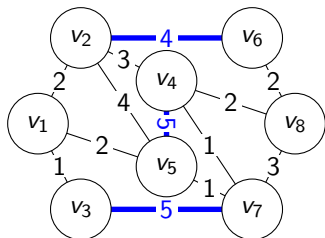
各辺 $e \in E$ に対する非負重み $w(e) \geq 0$ (辺重み関数 $w: E \rightarrow \mathbb{R}$)

定義：最大重みマッチングとは？

w に関する G の最大重みマッチングとは

G のマッチング $M \subseteq E$ で、

G の任意のマッチング M' に対して $\sum_{e \in M} w(e) \geq \sum_{e \in M'} w(e)$ を満たすもの



以後, $w(M) = \sum_{e \in M} w(e)$ と書く

完全グラフにおける最大重みマッチング

次は正しい

(演習問題)

完全グラフ $G = (V, E)$, 非負辺重み関数 $w: E \rightarrow \mathbb{R}$ に対して
 G のある最大マッチングは w に関する G の最大重みマッチングである

次が正しいとは限らない

(演習問題)

完全グラフ $G = (V, E)$, 非負辺重み関数 $w: E \rightarrow \mathbb{R}$ に対して
 G の任意の最大マッチングは w に関する G の最大重みマッチングである

次が正しいとは限らない

(演習問題)

無向グラフ $G = (V, E)$, 非負辺重み関数 $w: E \rightarrow \mathbb{R}$ に対して
 w に関する G の任意の最大重みマッチングは G の最大マッチングである

最大重みマッチング問題

最大重みマッチング問題とは？

- ▶ 入力：無向グラフ $G = (V, E)$, 非負辺重み関数 $w: E \rightarrow \mathbb{R}$
- ▶ 出力： G のマッチングで、重みが最大のもの

事実

最大重みマッチング問題は効率よく解くことができる (Edmonds, '65)

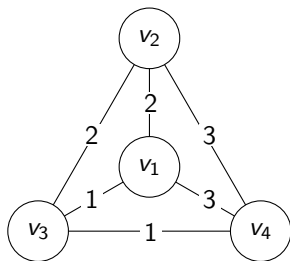
- ▶ Edmonds のアルゴリズムは、増加道を見つける手続きをサブルーチンとして用いている

「効率よく」 = $|V|$ と $|E|$ に関する多項式時間で

詳しくは『アルゴリズム論第一』, 『アルゴリズム論第二』, 『計算理論』で

最小重み完全マッチング

完全グラフ $G = (V, E)$, $|V|$ は偶数, 非負辺重み関数 $w: E \rightarrow \mathbb{R}$



重みが最小の**完全**マッチングを見つけるには, どうすればよいか?

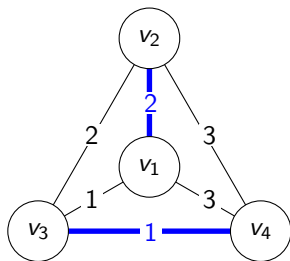
定義: 完全マッチングとは?

(復習)

G の**完全マッチング**とは G のマッチング $M \subseteq E$ で,
 G の任意の頂点に M のある辺が接続しているもの

最小重み完全マッチング

完全グラフ $G = (V, E)$, $|V|$ は偶数, 非負辺重み関数 $w: E \rightarrow \mathbb{R}$



重みが最小の**完全**マッチングを見つけるには, どうすればよいか?

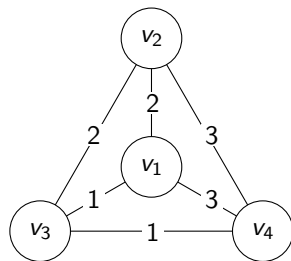
定義: 完全マッチングとは?

(復習)

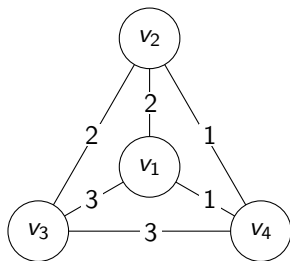
G の**完全マッチング**とは G のマッチング $M \subseteq E$ で,
 G の任意の頂点に M のある辺が接続しているもの

最小重み完全マッチング問題を最大重みマッチング問題に帰着

完全グラフ $G = (V, E)$, $|V|$ は偶数, 非負辺重み関数 $w: E \rightarrow \mathbb{R}$



辺重み: $w(e)$



$w'(e) = \max\{w(f) \mid f \in E\} - w(e) + 1$

最小重み完全マッチング問題を最大重みマッチング問題に帰着

完全グラフ $G = (V, E)$, $|V|$ は偶数, 非負辺重み関数 $w: E \rightarrow \mathbb{R}$

演習問題

任意の辺 $e \in E$ に対して, $w'(e) = \max\{w(f) \mid f \in E\} - w(e) + 1$ としたとき

M が w に関する G の最小重み完全マッチングである \Leftrightarrow
 M が w' に関する G の最大重みマッチングである

つまり, 最小重み完全マッチングを見つけるためには,
最大重みマッチング問題が解ければよい

(帰着)

概要

今日の目標

最大マッチングを使って問題を解決する例を見る

- ▶ 除雪車の運行計画問題 (最小費用オイラーグラフ化)
↪ 完全グラフにおける最小重み完全マッチング
- ▶ スリザーの必勝戦略
↪ 最大マッチング

目次

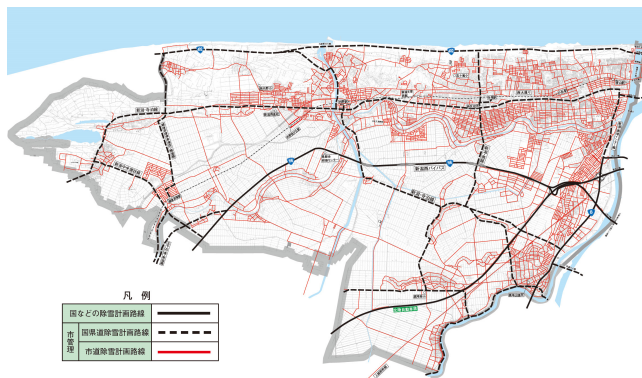
- ① 最大マッチングと最大重みマッチング
- ② 除雪車の運行計画問題
- ③ スリザーの必勝戦略
- ④ 今日のまとめ

除雪車の運行計画問題

除雪車の運行計画問題

除雪車を効率よく運行するルートを決定したい

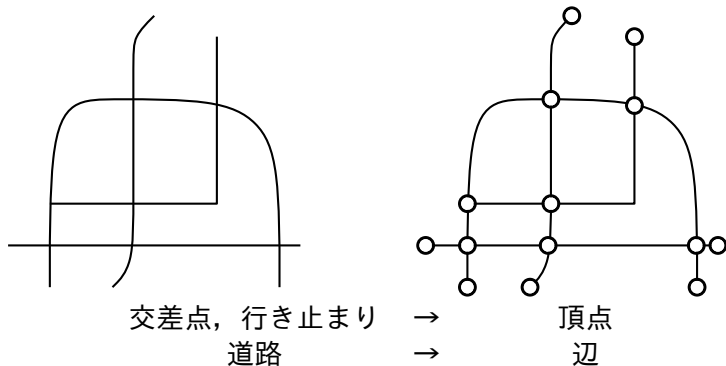
以下、除雪車が1台だけの場合を考える



https://www.city.niigata.lg.jp/nishi/kohoshi/pr/h24/nishi_1202/nishi_136_2.html

交通網のモデル化

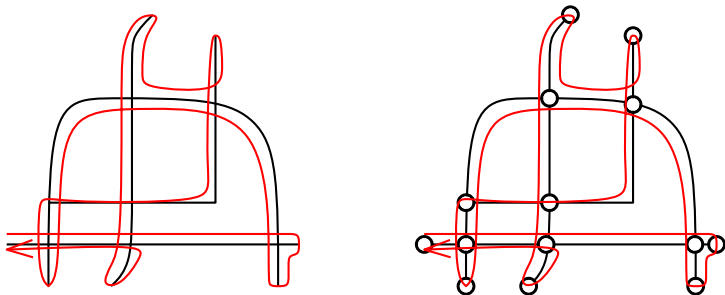
道路ネットワークをグラフとしてモデル化



問題によっては有向グラフを使うこともある

除雪車が行わなくてはならないこと

すべての辺を最低1回は通って、元の場所に戻る



二度以上通っている辺には「無駄」がある
 ⇨ 「無駄」を最小化したい

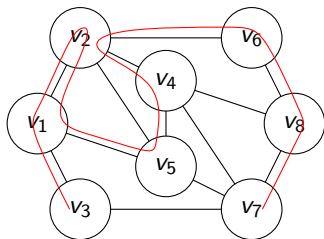
グラフにおける歩道

無向グラフ $G = (V, E)$

定義：歩道とは？

G における歩道とは、頂点の列 v_1, v_2, \dots, v_k で
 任意の $i \in \{1, \dots, k-1\}$ に対して、 $\{v_i, v_{i+1}\} \in E$ であるもの

直感的には「同じ頂点や辺を二度以上通ってよい道」



$v_3, v_1, v_2, v_1, v_5, v_4, v_2, v_6, v_8, v_7$ はこのグラフにおける歩道

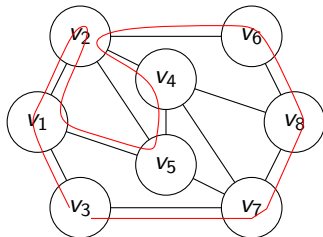
グラフにおける回路

無向グラフ $G = (V, E)$

定義：回路とは？

G における回路 (または閉歩道) とは、歩道 v_1, v_2, \dots, v_k で $v_1 = v_k$ を満たすもののこと

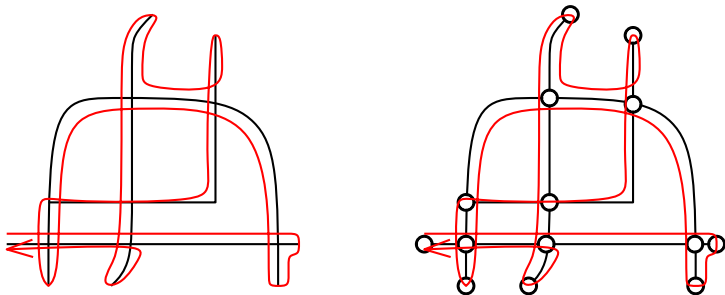
直感的には「同じ頂点や辺を二度以上通ってよい閉路」



$v_3, v_1, v_2, v_1, v_5, v_4, v_2, v_6, v_8, v_7, v_3$ はこのグラフにおける回路

除雪車が行わなくてはならないこと

すべての辺を最低1回は通って、元の場所に戻る



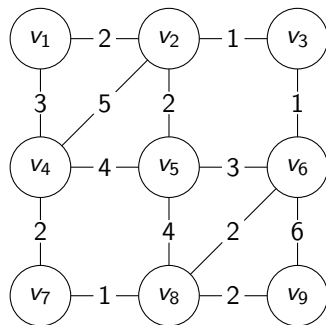
二度以上通っている辺には「無駄」がある
 ⇨ 「無駄」を最小化したい

行いたいこと

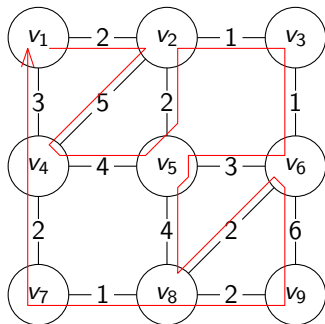
すべての辺を通る回路で、「長さ」が最小のものを見つけたい

長さ = 辺の重み (長さ) の和

例題 1



例題 1



無駄のない除雪計画

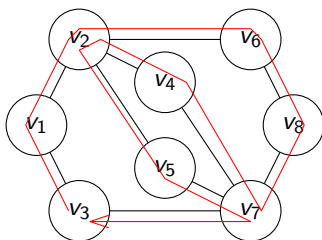
(すべての辺をちょうど一度ずつ通る回路が存在)

オイラー回路

無向グラフ $G = (V, E)$

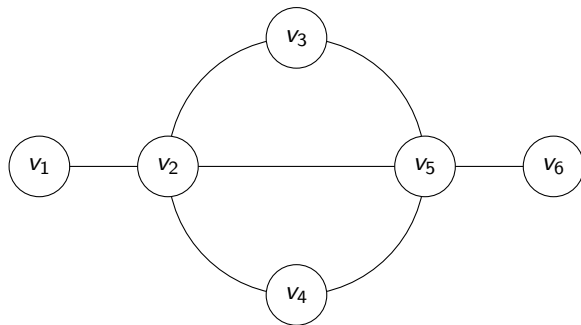
定義：オイラー回路とは？

G における **オイラー回路**とは、回路 v_1, v_2, \dots, v_k で G のすべての辺をちょうど一度ずつ通るもの



オイラー回路を持たないグラフ

次のグラフはオイラー回路を持たない

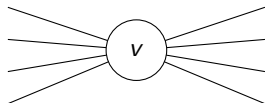


どんなグラフがオイラー回路を持ち、
どんなグラフがオイラー回路を持たないのだろうか？

オイラー回路を持つための必要条件

$G = (V, E)$ がオイラー回路 C を持つと仮定する

- ▶ 任意の頂点 $v \in V$ に着目する
- ▶ C は v に接続する辺をすべてちょうど一度ずつ通る
- ▶ その通り方を見てもみる
- ▶ C において v が登場する前と後の辺が対になっている
- ▶ すなわち、 v に接続する辺の数は偶数

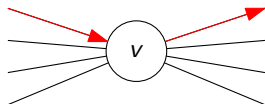


つまり、 v の次数は偶数

オイラー回路を持つための必要条件

$G = (V, E)$ がオイラー回路 C を持つと仮定する

- ▶ 任意の頂点 $v \in V$ に着目する
- ▶ C は v に接続する辺をすべてちょうど一度ずつ通る
- ▶ その通り方を見ても
- ▶ C において v が登場する前と後の辺が対になっている
- ▶ すなわち、 v に接続する辺の数は偶数

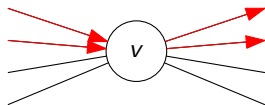


つまり、 v の次数は偶数

オイラー回路を持つための必要条件

$G = (V, E)$ がオイラー回路 C を持つと仮定する

- ▶ 任意の頂点 $v \in V$ に着目する
- ▶ C は v に接続する辺をすべてちょうど一度ずつ通る
- ▶ その通り方を見てもみる
- ▶ C において v が登場する前と後の辺が対になっている
- ▶ すなわち、 v に接続する辺の数は偶数

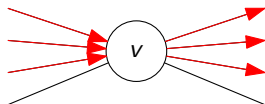


つまり、 v の次数は偶数

オイラー回路を持つための必要条件

$G = (V, E)$ がオイラー回路 C を持つと仮定する

- ▶ 任意の頂点 $v \in V$ に着目する
- ▶ C は v に接続する辺をすべてちょうど一度ずつ通る
- ▶ その通り方を見てもみる
- ▶ C において v が登場する前と後の辺が対になっている
- ▶ すなわち、 v に接続する辺の数は偶数

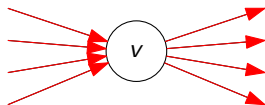


つまり、 v の次数は偶数

オイラー回路を持つための必要条件

$G = (V, E)$ がオイラー回路 C を持つと仮定する

- ▶ 任意の頂点 $v \in V$ に着目する
- ▶ C は v に接続する辺をすべてちょうど一度ずつ通る
- ▶ その通り方を見てもみる
- ▶ C において v が登場する前と後の辺が対になっている
- ▶ すなわち、 v に接続する辺の数は偶数



つまり、 v の次数は偶数

オイラー回路を持つための必要十分条件

連結 無向グラフ $G = (V, E)$

オイラー回路を持つための必要十分条件

G がオイラー回路を持つ $\Leftrightarrow G$ の任意の頂点の次数が偶数

「 \Rightarrow 」の証明：演習問題 (前ページの内容がヒント)

「 \Leftarrow 」の証明の概略：辺数に関する帰納法

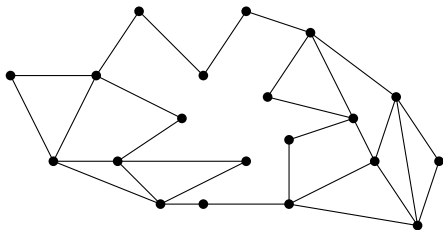
- ▶ $|E| = 0$ のときを考えると、 G は辺数 0 のオイラー回路を持つ
- ▶ 任意の $k \geq 0$ を考える (注：累積帰納法)
- ▶ 辺数 k 以下の任意の連結無向グラフ G' に対して、
 G' の任意の頂点の次数が偶数であるならば、
 G' がオイラー回路を持つと仮定する

オイラー回路を持つための必要十分条件 (続 1)

証明すること

辺数 $k + 1$ の任意の連結無向グラフ $G = (V, E)$ に対して,
 G の任意の頂点の次数が偶数であるならば,
 G がオイラー回路を持つ

- ▶ G が連結であり, 任意の頂点次数が偶数であることを仮定
- ▶ G は連結なので, 任意の頂点の次数は 2 以上

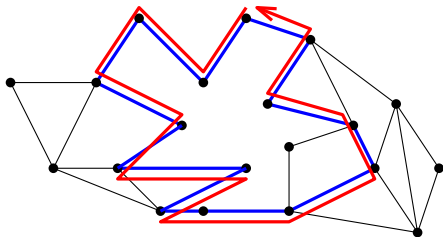


オイラー回路を持つための必要十分条件 (続 1)

証明すること

辺数 $k + 1$ の任意の連結無向グラフ $G = (V, E)$ に対して、
 G の任意の頂点の次数が偶数であるならば、
 G がオイラー回路を持つ

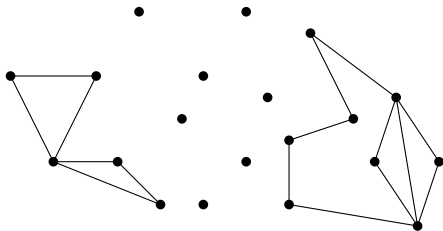
- ▶ G が連結であり、任意の頂点次数が偶数であることを仮定
- ▶ G は連結なので、任意の頂点の次数は 2 以上
- ▶ 演習問題 2.6 より、 G は閉路を含む (それを C とする)



オイラー回路を持つための必要十分条件 (続 2)

G から C の辺をすべて取り除いたグラフを考える (\tilde{G} とする)

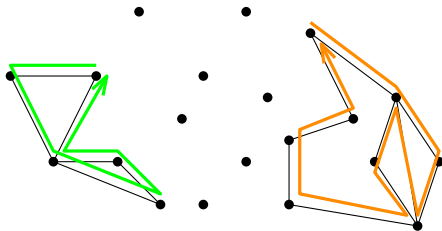
- ▶ C における各頂点の次数は 2 で, G の各頂点の次数は偶数なので, \tilde{G} の各頂点の次数も偶数
- ▶ \tilde{G} の連結成分を $\tilde{G}_1, \tilde{G}_2, \dots, \tilde{G}_k$ とする
- ▶ 各 \tilde{G}_i の辺数は G の辺数未満
- ▶ したがって, \tilde{G}_i はオイラー回路を含む (C_i とする)



オイラー回路を持つための必要十分条件 (続 2)

G から C の辺をすべて取り除いたグラフを考える (\tilde{G} とする)

- ▶ C における各頂点の次数は 2 で, G の各頂点の次数は偶数なので, \tilde{G} の各頂点の次数も偶数
- ▶ \tilde{G} の連結成分を $\tilde{G}_1, \tilde{G}_2, \dots, \tilde{G}_k$ とする
- ▶ 各 \tilde{G}_i の辺数は G の辺数未満
- ▶ したがって, \tilde{G}_i はオイラー回路を含む (C_i とする)

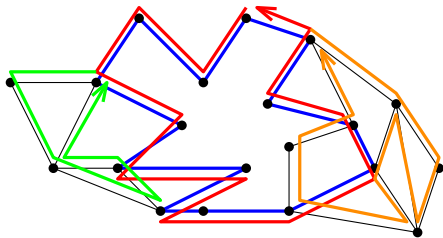


オイラー回路を持つための必要十分条件 (続 3)

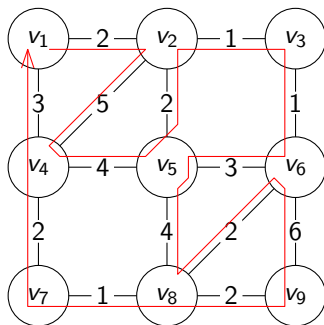
G から C の辺をすべて取り除いたグラフを考える (\tilde{G} とする)

- ▶ C_1, \dots, C_k と C を組み合わせることで,
 G のオイラー回路を構成できる

(詳細は演習問題)



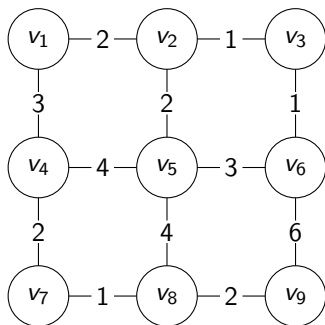
例題 1 (再び)



各頂点の次数が偶数

- ▶ ∴ オイラー回路が存在
- ▶ ∴ 無駄のない除雪が可能

例題 2

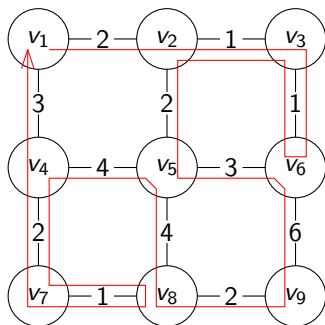


次数が奇数である頂点が存在

- ▶ ∴ オイラー回路が存在しない
- ▶ ∴ 無駄のない除雪が可能ではない
- ▶ ⇨ 無駄を最小化したい

「仮想的な道路」を付け加えて、オイラー回路があるようにする

例題 2

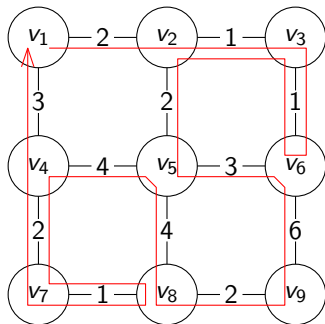


次数が奇数である頂点が存在

- ▶ ∴ オイラー回路が存在しない
- ▶ ∴ 無駄のない除雪が可能ではない
- ▶ ⇨ 無駄を最小化したい

「仮想的な道路」を付け加えて、オイラー回路があるようにする

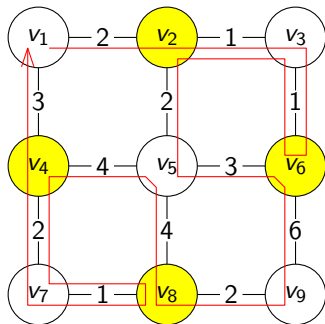
例題 2 : 仮想的な道路を追加



- ▶ 次数が奇数である頂点間に辺を引いて，すべての頂点の次数が偶数であるようにする
- ▶ 次数が奇数である頂点の数は偶数 (演習問題)
- ▶ ∴ これは必ず可能 !!

疑問：無駄を抑えるには，どのように辺を引けばよいか？

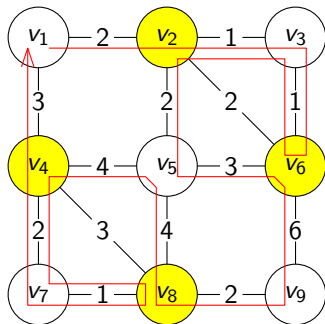
例題 2 : 仮想的な道路を追加



- ▶ 次数が奇数である頂点間に辺を引いて、すべての頂点の次数が偶数であるようにする
- ▶ 次数が奇数である頂点の数は偶数 (演習問題)
- ▶ ∴ これは必ず可能 !!

疑問：無駄を抑えるには、どのように辺を引けばよいか？

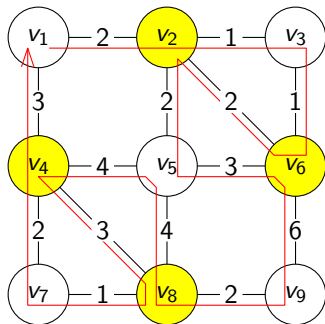
例題 2 : 仮想的な道路を追加



- ▶ 次数が奇数である頂点間に辺を引いて，すべての頂点の次数が偶数であるようにする
- ▶ 次数が奇数である頂点の数は偶数 (演習問題)
- ▶ ∴ これは必ず可能 !!

疑問：無駄を抑えるには，どのように辺を引けばよいか？

例題 2 : 仮想的な道路を追加

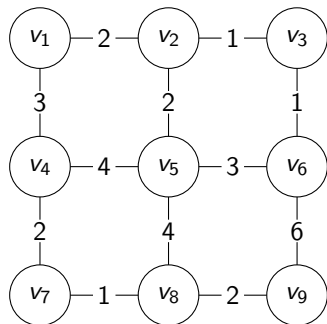


- ▶ 次数が奇数である頂点間に辺を引いて、すべての頂点の次数が偶数であるようにする
- ▶ 次数が奇数である頂点の数は偶数 (演習問題)
- ▶ ∴ これは必ず可能 !!

疑問：無駄を抑えるには、どのように辺を引けばよいか？

無駄を抑える辺の引き方

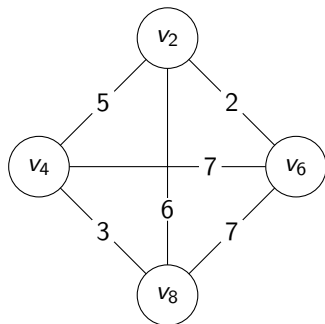
次数が奇数である頂点の間で、最も効率的な経路の長さを考える



	v_2	v_4	v_6	v_8
v_2	-	5	2	6
v_4		-	7	3
v_6			-	7
v_8				-

無駄を抑える辺の引き方 \rightsquigarrow 最小重み完全マッチング：モデル化

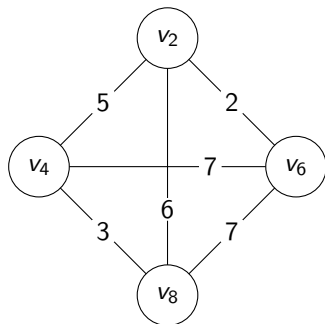
次のような完全グラフと非負辺重みを考える



	v_2	v_4	v_6	v_8
v_2	-	5	2	6
v_4		-	7	3
v_6			-	7
v_8				-

無駄を抑える辺の引き方 \rightsquigarrow 最小重み完全マッチング：モデル化

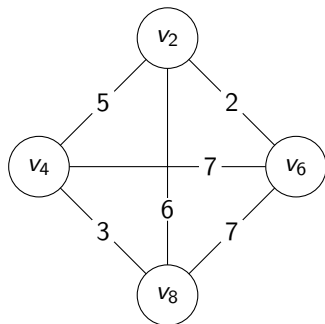
各頂点は元のグラフにおいて次数が奇数である頂点 (注：頂点数は偶数)



	v_2	v_4	v_6	v_8
v_2	-	5	2	6
v_4		-	7	3
v_6			-	7
v_8				-

無駄を抑える辺の引き方 \rightsquigarrow 最小重み完全マッチング：モデル化

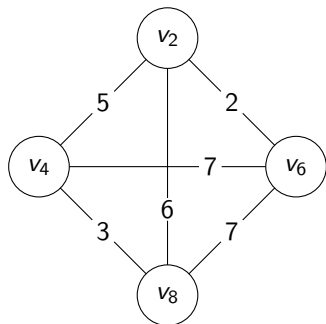
辺の重みは、対応する 2 頂点間の最短経路長



	v_2	v_4	v_6	v_8
v_2	-	5	2	6
v_4		-	7	3
v_6			-	7
v_8				-

無駄を抑える辺の引き方 \rightsquigarrow 最小重み完全マッチング：モデル化

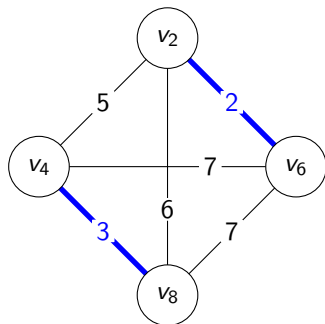
求めたいものは、最小重み完全マッチング



	v_2	v_4	v_6	v_8
v_2	-	5	2	6
v_4		-	7	3
v_6			-	7
v_8				-

無駄を抑える辺の引き方 \rightsquigarrow 最小重み完全マッチング：求めた結果

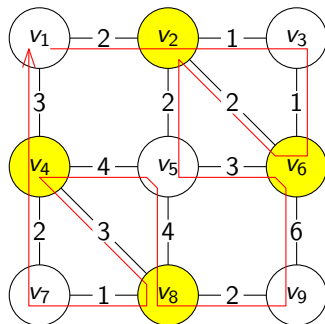
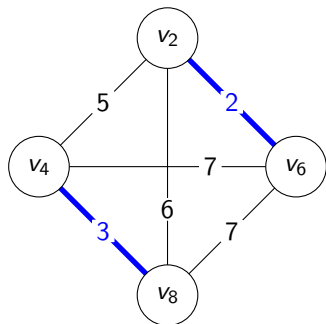
次のマッチングは最小重み完全マッチング



	v_2	v_4	v_6	v_8
v_2	-	5	2	6
v_4		-	7	3
v_6			-	7
v_8				-

無駄を抑える辺の引き方 \rightsquigarrow 最小重み完全マッチング：求めた結果

次のマッチングは最小重み完全マッチング



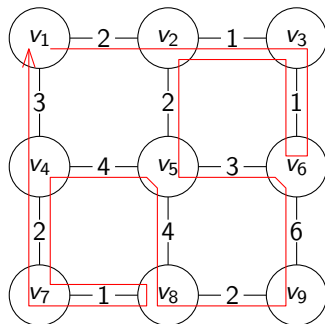
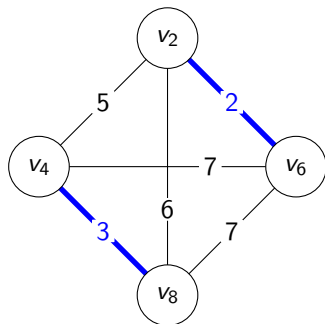
格言

モデルに対する解を元の問題の文脈で必ず解釈し直す

これで解けた

無駄を抑える辺の引き方 \rightsquigarrow 最小重み完全マッチング：求めた結果

次のマッチングは最小重み完全マッチング



格言

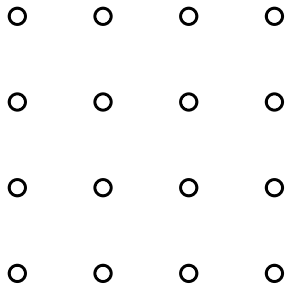
モデルに対する解を元の問題の文脈で必ず解釈し直す

これで解けた

目次

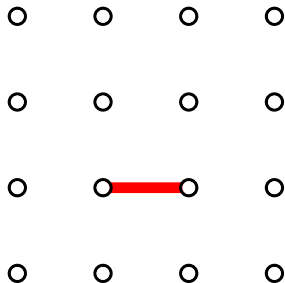
- ① 最大マッチングと最大重みマッチング
- ② 除雪車の運行計画問題
- ③ **スリザーの必勝戦略**
- ④ 今日のまとめ

格子上で遊ぶゲーム



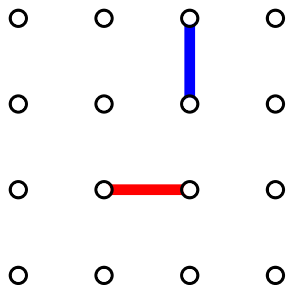
たてよこに隣りあう点を交互に結ぶ

格子上で遊ぶゲーム



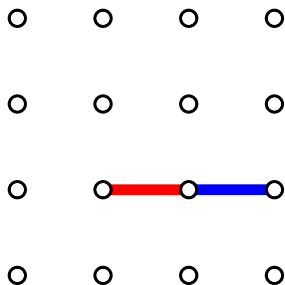
たてよこに隣りあう点を交互に結ぶ

格子上で遊ぶゲーム



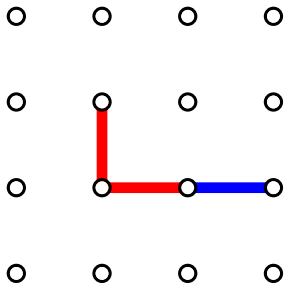
必ず道になるように結ばないといけない

格子上で遊ぶゲーム



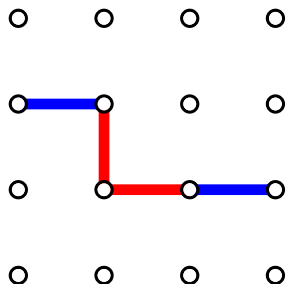
必ず道になるように結ばないといけない

格子上で遊ぶゲーム



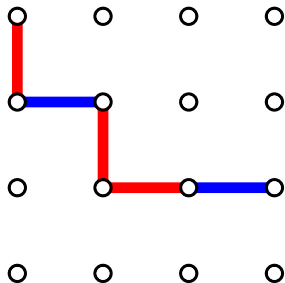
結べなくなった方の負け

格子上で遊ぶゲーム



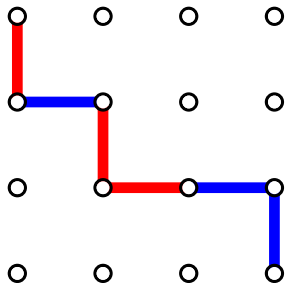
結べなくなった方の負け

格子上で遊ぶゲーム



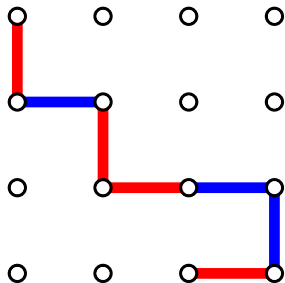
結べなくなった方の負け

格子上で遊ぶゲーム



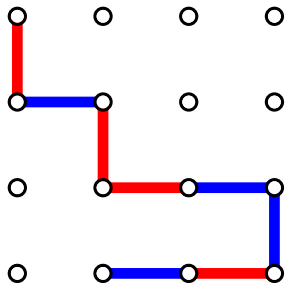
結べなくなった方の負け

格子上で遊ぶゲーム



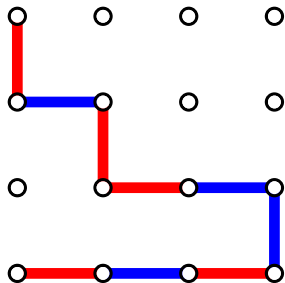
結べなくなった方の負け

格子上で遊ぶゲーム



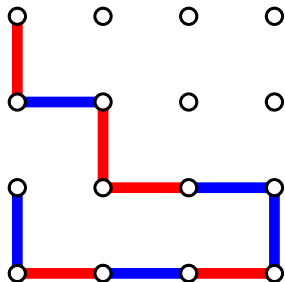
結べなくなった方の負け

格子上で遊ぶゲーム



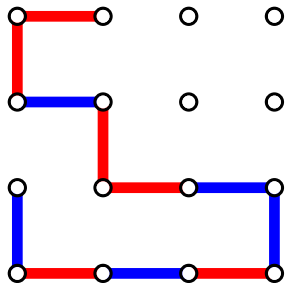
結べなくなった方の負け

格子上で遊ぶゲーム



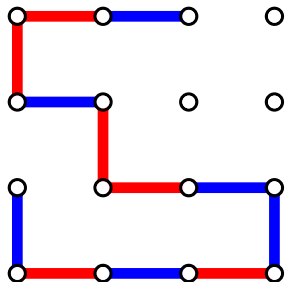
結べなくなった方の負け

格子上で遊ぶゲーム



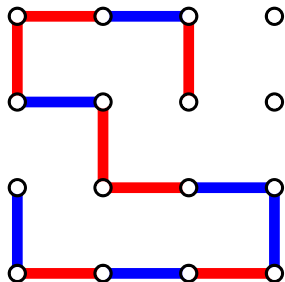
結べなくなった方の負け

格子上で遊ぶゲーム



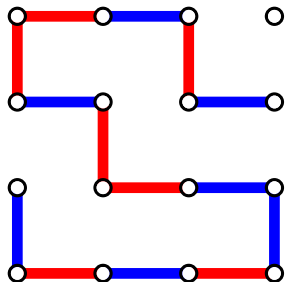
結べなくなった方の負け

格子上で遊ぶゲーム



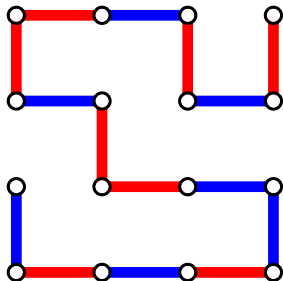
結べなくなった方の負け

格子上で遊ぶゲーム



結べなくなった方の負け

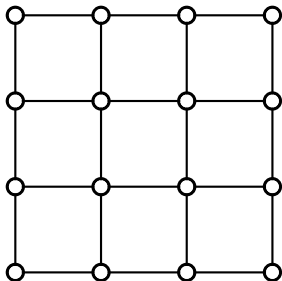
格子上で遊ぶゲーム



結べなくなった方の負け

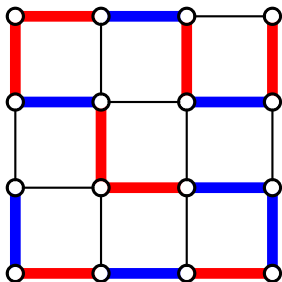
無向グラフ上のスリザー

スリザーを無向グラフの辺を選ぶゲームだと見なす



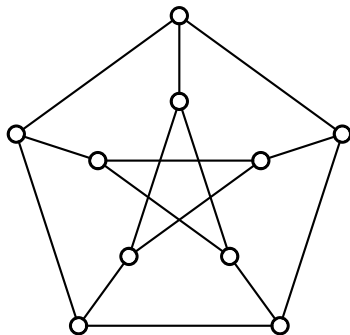
無向グラフ上のスリザー

スリザーを無向グラフの辺を選ぶゲームだと見なす



無向グラフ上のスリザー：別の例

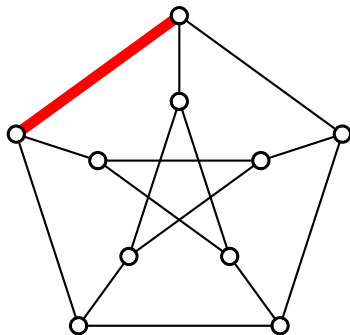
スリザーを無向グラフの辺を選ぶゲームだと見なす



このグラフはペテルセン・グラフと呼ばれる有名な無向グラフ

無向グラフ上のスリザー：別の例

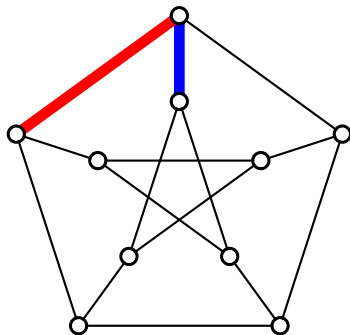
スリザーを無向グラフの辺を選ぶゲームだと見なす



このグラフはペテルセン・グラフと呼ばれる有名な無向グラフ

無向グラフ上のスリザー：別の例

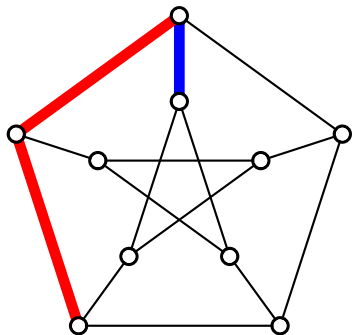
スリザーを無向グラフの辺を選ぶゲームだと見なす



このグラフはペテルセン・グラフと呼ばれる有名な無向グラフ

無向グラフ上のスリザー：別の例

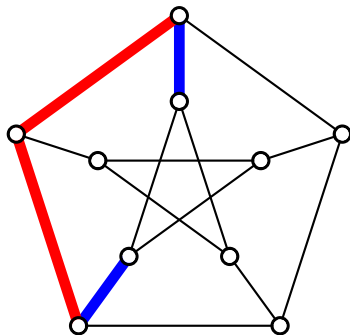
スリザーを無向グラフの辺を選ぶゲームだと見なす



このグラフはペテルセン・グラフと呼ばれる有名な無向グラフ

無向グラフ上のスリザー：別の例

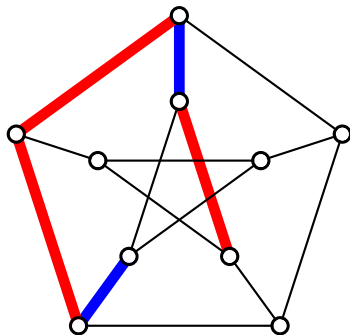
スリザーを無向グラフの辺を選ぶゲームだと見なす



このグラフはペテルセン・グラフと呼ばれる有名な無向グラフ

無向グラフ上のスリザー：別の例

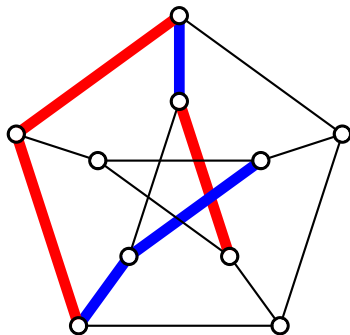
スリザーを無向グラフの辺を選ぶゲームだと見なす



このグラフはペテルセン・グラフと呼ばれる有名な無向グラフ

無向グラフ上のスリザー：別の例

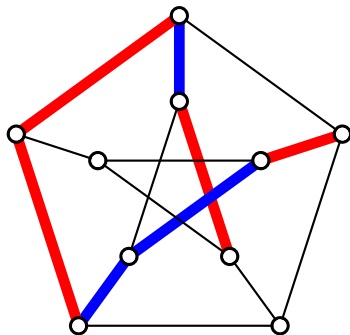
スリザーを無向グラフの辺を選ぶゲームだと見なす



このグラフはペテルセン・グラフと呼ばれる有名な無向グラフ

無向グラフ上のスリザー：別の例

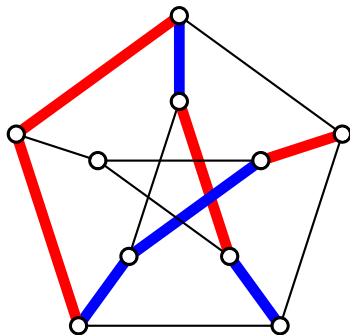
スリザーを無向グラフの辺を選ぶゲームだと見なす



このグラフはペテルセン・グラフと呼ばれる有名な無向グラフ

無向グラフ上のスリザー：別の例

スリザーを無向グラフの辺を選ぶゲームだと見なす



このグラフはペテルセン・グラフと呼ばれる有名な無向グラフ

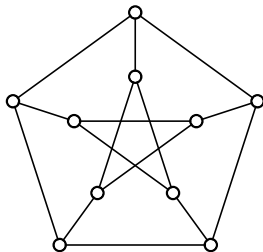
先手必勝の場合

無向グラフ $G = (V, E)$ 上のスリザーを考える

先手必勝の場合

G が完全マッチングを持つ \Rightarrow 先手必勝

先手必勝：先手と後手が最善を尽くしたとき、先手が必ず勝てる



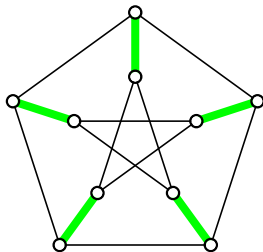
先手必勝の場合

無向グラフ $G = (V, E)$ 上のスリザーを考える

先手必勝の場合

G が完全マッチングを持つ \Rightarrow 先手必勝

先手必勝：先手と後手が最善を尽くしたとき、先手が必ず勝てる



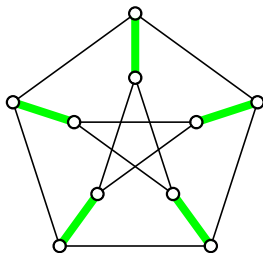
先手必勝の場合

無向グラフ $G = (V, E)$ 上のスリザーを考える

先手必勝の場合

G が完全マッチングを持つ \Rightarrow 先手必勝

先手必勝：先手と後手が最善を尽くしたとき、先手が必ず勝てる



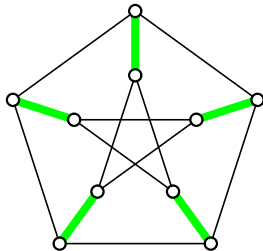
つまり、ペテルセン・グラフ上のスリザーは先手必勝

G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (1)

▶ M を G の完全マッチングとする

次のような先手の戦略を考える

- 1 初手は M の任意の辺を選ぶ
- 2 2 手目以降は選べる M の辺を選ぶ

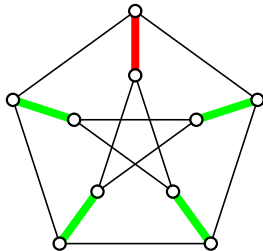


G が完全マッチングを持つ \Rightarrow 先手必勝：証明 (1)

▶ M を G の完全マッチングとする

次のような先手の戦略を考える

- 1 初手は M の任意の辺を選ぶ
- 2 2 手目以降は選べる M の辺を選ぶ

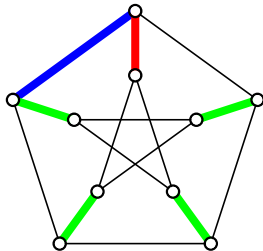


G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (1)

▶ M を G の完全マッチングとする

次のような先手の戦略を考える

- 1 初手は M の任意の辺を選ぶ
- 2 2 手目以降は選べる M の辺を選ぶ

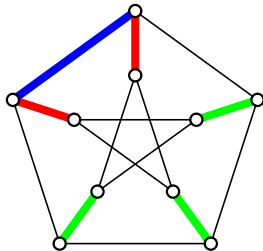


G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (1)

▶ M を G の完全マッチングとする

次のような先手の戦略を考える

- 1 初手は M の任意の辺を選ぶ
- 2 2 手目以降は選べる M の辺を選ぶ

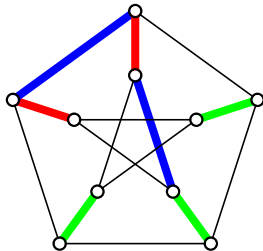


G が完全マッチングを持つ \Rightarrow 先手必勝：証明 (1)

▶ M を G の完全マッチングとする

次のような先手の戦略を考える

- 1 初手は M の任意の辺を選ぶ
- 2 2 手目以降は選べる M の辺を選ぶ

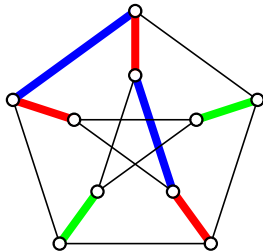


G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (1)

▶ M を G の完全マッチングとする

次のような先手の戦略を考える

- 1 初手は M の任意の辺を選ぶ
- 2 2 手目以降は選べる M の辺を選ぶ

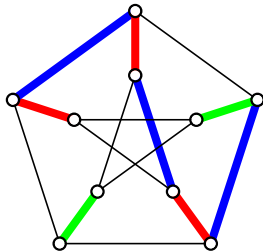


G が完全マッチングを持つ \Rightarrow 先手必勝：証明 (1)

▶ M を G の完全マッチングとする

次のような先手の戦略を考える

- 1 初手は M の任意の辺を選ぶ
- 2 2 手目以降は選べる M の辺を選ぶ

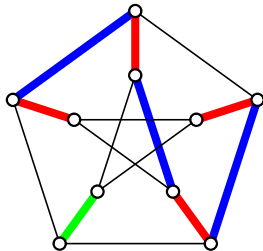


G が完全マッチングを持つ \Rightarrow 先手必勝：証明 (1)

▶ M を G の完全マッチングとする

次のような先手の戦略を考える

- 1 初手は M の任意の辺を選ぶ
- 2 2 手目以降は選べる M の辺を選ぶ

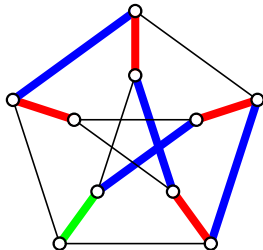


G が完全マッチングを持つ \Rightarrow 先手必勝：証明 (1)

▶ M を G の完全マッチングとする

次のような先手の戦略を考える

- 1 初手は M の任意の辺を選ぶ
- 2 2 手目以降は選べる M の辺を選ぶ

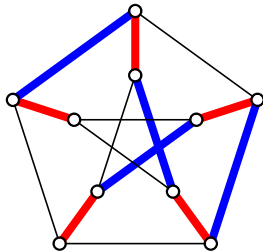


G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (1)

▶ M を G の完全マッチングとする

次のような先手の戦略を考える

- 1 初手は M の任意の辺を選ぶ
- 2 2 手目以降は選べる M の辺を選ぶ



G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (2)

- ▶ M を G の完全マッチングとする

次のような先手の戦略を考える

- 1 初手は M の任意の辺を選ぶ
- 2 2 手目以降は選べる M の辺を選ぶ

重要な性質

上の戦略に従って、先手は必ず M の辺を選べる

なぜ選べるのか？

G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (3)

なぜ重要な性質が成り立つのか?: 不変条件

先手が選んだ後,
構成している道は M に関する交互道で, 端辺は必ず M の辺

不変条件が成り立つことの証明 : 手数に関する帰納法

- ▶ まず, 初手では成り立っている
- ▶ 先手が k 個目の辺を選んだ直後を考える (不変条件の成立を仮定)
- ▶ 構成している道の端点を u, v とする
- ▶ M はマッチングなので, 帰納法の仮定より, 後手は M の辺を選べない
- ▶ 後手が選ぶ k 個目の辺を $\{u, w\} \notin M$ とする



G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (3)

なぜ重要な性質が成り立つのか?: 不変条件

先手が選んだ後,
構成している道は M に関する交互道で, 端辺は必ず M の辺

不変条件が成り立つことの証明 : 手数に関する帰納法

- ▶ まず, 初手では成り立っている
- ▶ 先手が k 個目の辺を選んだ直後を考える (不変条件の成立を仮定)
- ▶ 構成している道の端点を u, v とする
- ▶ M はマッチングなので, 帰納法の仮定より, 後手は M の辺を選べない
- ▶ 後手が選ぶ k 個目の辺を $\{u, w\} \notin M$ とする



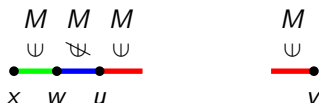
G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (4)

なぜ重要な性質が成り立つのか?: 不変条件

先手が選んだ後,
構成している道は M に関する交互道で, 端辺は必ず M の辺

不変条件が成り立つことの証明 (続き):

- ▶ M が完全マッチングなので, w を飽和する M の辺 $\{w, x\}$ が存在し, それはまだ選ばれていない
- ▶ 帰納法の仮定より, 構成している道は M に関する交互道なので, x は構成している道の頂点ではない
- ▶ つまり, 先手は $\{w, x\}$ を選べばよい □



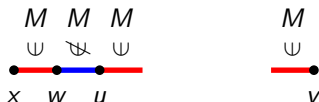
G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (4)

なぜ重要な性質が成り立つのか?: 不変条件

先手が選んだ後,
構成している道は M に関する交互道で, 端辺は必ず M の辺

不変条件が成り立つことの証明 (続き):

- ▶ M が完全マッチングなので, w を飽和する M の辺 $\{w, x\}$ が存在し, それはまだ選ばれていない
- ▶ 帰納法の仮定より, 構成している道は M に関する交互道なので, x は構成している道の頂点ではない
- ▶ つまり, 先手は $\{w, x\}$ を選べばよい □



G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (5)

先手必勝性の証明 :

- ▶ 以上の議論より, 先手は必ず辺を選べるので, 先手は負けない
- ▶ つまり, 先手は必ず勝てる



G が完全マッチングを持つ \Rightarrow 先手必勝 : 証明 (5)

先手必勝性の証明 :

- ▶ 以上の議論より, 先手は必ず辺を選べるので, 先手は負けない
- ▶ つまり, 先手は必ず勝てる □

格言

アルゴリズムとプログラムの正当性の鍵は**不変条件**

- ▶ プログラムは時々刻々と「状態」を変える「動的システム」
- ▶ 時間的に変わらない性質が設計や解析の要となる

無向グラフ上のスリザー：まとめ

無向グラフ $G = (V, E)$ 上のスリザーを考える

先手必勝の場合

G が完全マッチングを持つ \Rightarrow 先手必勝

G がこの条件を満たさない場合はどうなのか？

- ▶ 与えられた無向グラフ G に対して、
それ上のスリザーが先手必勝か後手必勝か判定する
効率よいアルゴリズムが知られている

(Anderson '73)

目次

- ① 最大マッチングと最大重みマッチング
- ② 除雪車の運行計画問題
- ③ スリザーの必勝戦略
- ④ 今日のまとめ

概要

今日の目標

最大マッチングを使って問題を解決する例を見る

- ▶ 除雪車の運行計画問題 (最小費用オイラーグラフ化)
- ▶ スリザーの必勝戦略

残った時間の使い方

- ▶ 演習問題をやる
 - ▶ 相談推奨 (ひとりでやらない)
- ▶ 質問をする
 - ▶ 教員と TA は巡回
- ▶ 退室時, 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 最大マッチングと最大重みマッチング
- ② 除雪車の運行計画問題
- ③ スリザーの必勝戦略
- ④ 今日のまとめ