マッチング問題に対する高速なアルゴリズム

山口 勇太郎

大阪大学 情報科学研究科

第 22 回組合せ最適化セミナー (COSS)　　2025/07/24

最大マッチング問題

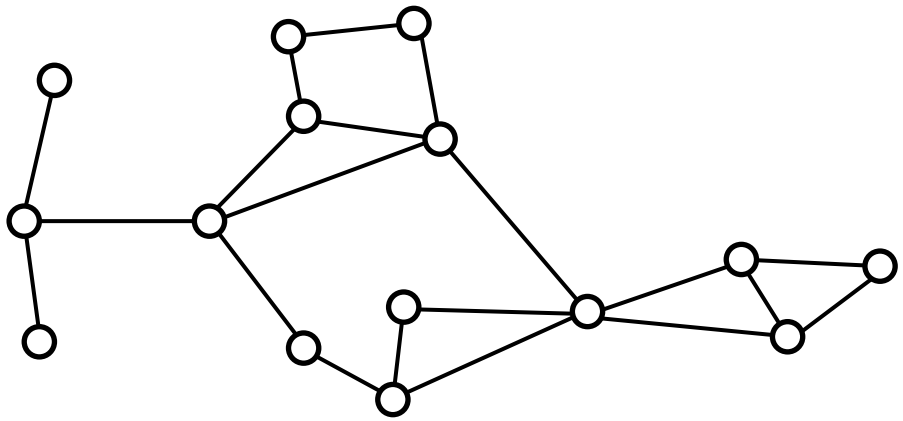
入力: $G = (V, E)$: 無向グラフ（連結）

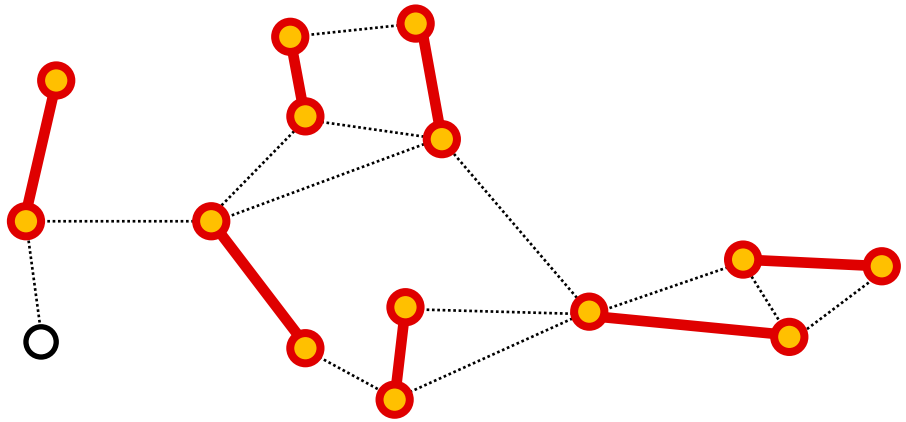
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

頂点を共有しない辺部分集合

$n = |V|,\ m = |E|$

μ: 最大サイズ

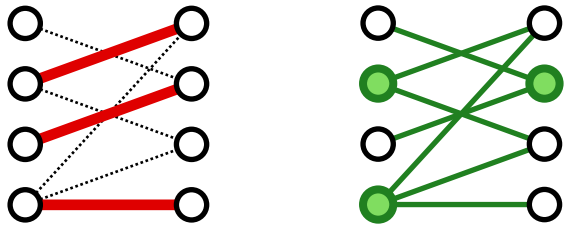




最大マッチング問題（今日やらない話）

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

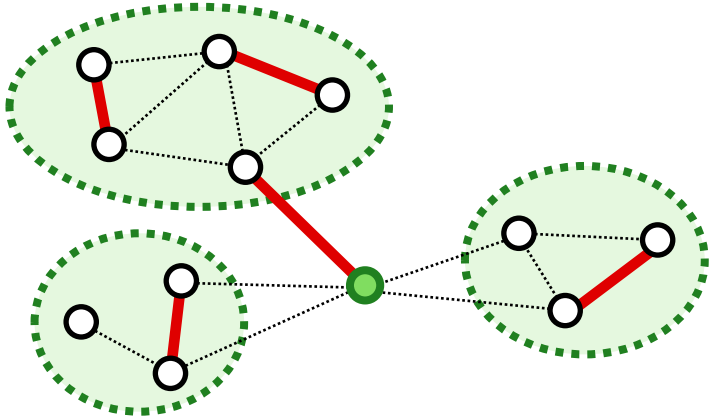
$n = |V|,\ m = |E|$
μ: 最大サイズ



［良い特徴付け（双対定理）］ (cf. Takazawa 2014)

- 2 部グラフ: $\mu = \min\{\,|U| \mid U \subseteq V:$ **頂点被覆** $\}$
  [Kőnig 1931]

全辺に交わる
頂点部分集合



- 一般グラフ: $\mu = \min\left((\bullet\text{の数}) + \sum \left\lfloor \frac{1}{2} |\ \bigcirc\ | \right\rfloor\right)$
  [Tutte 1947 + Berge 1958]

最大マッチング問題（今日やらない話）

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

[辺重み付き版]

- 重み最大化／最小重み完全マッチング (cf. Takazawa 2014, Okamoto 2015)
  指数サイズだが**多項式時間**で解ける線形計画問題 (LP) の代表例
  [Rothvoss 2017] [Edmonds 1965; ...]

[Papadimitriou–Yannakakis 1982]

- 0/1 辺重みで，指定した重みの完全マッチング (Exact Matching)
  乱択多項式時間で解けるが，**決定性多項式時間**アルゴリズムは広く未解決
  [Mulmuley–Vazirani–Vazirani 1987] [Karzanov 1987; Yuster 2012; Galluccio–Loebl 1999; ...]

最大マッチング問題（今日やらない話）

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

[代数的（乱択）アルゴリズム] (cf. Okamoto 2011, Hirai 2019)

G から定まる多項式行列 (Tutte Matrix) の行列式の非零判定に帰着

- m 変数 n 次（以下）の多項式となり，そのまま計算するのは難しい…
- 各変数に値を代入してしまえば，素朴な掃き出し法で $O(n^3)$ 時間で計算可能
- **大きい体上で代入値を乱択**すれば，高確率で以下が成り立つ (Schwartz–Zippel Lemma)
  行列式が**多項式として非零** ⇔ 代入した後の行列式の**値が非零**

最大マッチング問題（今日やる話）

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

[組合せ的（決定性）アルゴリズム]

空集合から始めて，**増加道**に沿って暫定解の改善を繰り返す

- 2 部グラフの場合
  - 素朴にやると，$O(m)$ 時間の増加道探索 (BFS) を μ 回で $O(\mu m)$ 時間 [Kőnig 1931, Ford–Fulkerson 1956]
  - **最短増加道**に注目して工夫すると $O(\sqrt{\mu}m)$ 時間 [Hopcroft–Karp 1973]
  - 約 50 年ぶりの改善で $O(n^{5/3}m^{1/3}\,\mathrm{poly}(\log n))$ 時間 [Chuzhoy–Khanna 2024]

最大マッチング問題（今日やる話）

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

[組合せ的（決定性）アルゴリズム]

空集合から始めて，**増加道**に沿って暫定解の改善を繰り返す

- 一般グラフの場合
  - $O(\mu m)$ 時間の増加道探索（交互 BFS + **花縮約**）を μ 回で $O(\mu^2 m)$ 時間 [Edmonds 1965]
  - 陽な縮約の回避で $O(\mu^2 n)$ 時間など [Balinski 1969; Gabow 1976; ...]
  - **最短増加道**に注目してめちゃくちゃ頑張ると $O(\sqrt{\mu} m)$ 時間 [Micali–Vazirani 1980, Vazirani 2024]

最大マッチング問題（今日やる話）

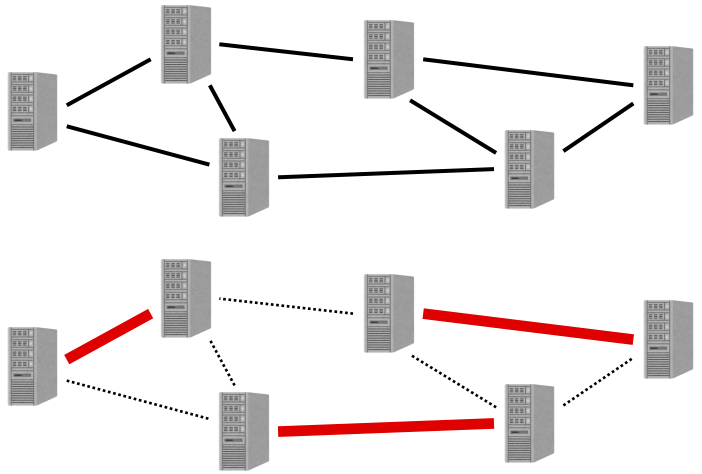
入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

[分散アルゴリズム] (cf. Izumi 2017)

辺に沿った通信で情報共有しながら，ネットワーク上の問題を解く

- 通信タイミング: 同期的／非同期的
- 通信帯域制限: $O(\log n)$／無し
- 計算量評価: 通信ラウンド数／合計通信量



最大マッチング問題（今日やる話）

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

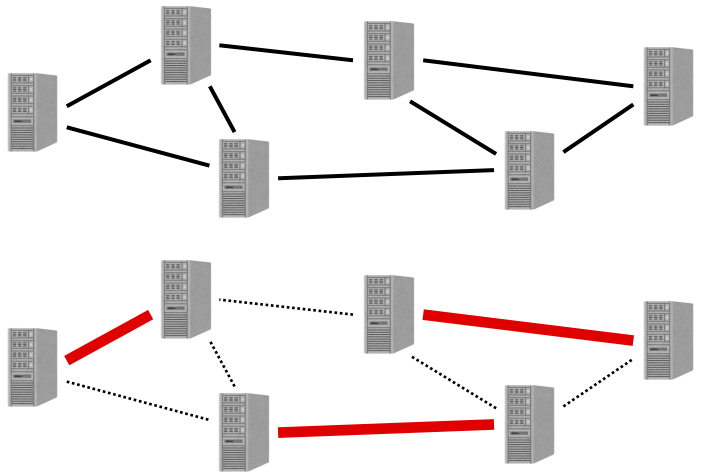
[分散アルゴリズム] (cf. Izumi 2017)

辺に沿った通信で情報共有しながら，ネットワーク上の問題を解く

- 通信タイミング: **同期的**
- 通信帯域制限: $O(\log n)$

CONGEST モデル

- 計算量評価: **通信ラウンド数**



最大マッチング問題（今日やる話）

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

[分散アルゴリズム (CONGEST)]

- 自明な万能アルゴリズムだと $\Theta(m) = O(n^2)$ ラウンド (cf. Izumi 2017)
- 2 部グラフで決定性 $O(\mu \log \mu)$ ラウンド [Ahmadi–Kuhn–Oshman 2018]
- 一般グラフで決定性 $O(\mu^2)$ ラウンド [Ben-Basat–Kawarabayashi–Schwartzman 2019]
- 一般グラフで乱択 $O(\mu^{1.5})$ ラウンド [Kitamura–Izumi 2022]
- 一般グラフで乱択 $O(\mu \log \mu)$ ラウンド（など） [Izumi–Kitamura–Y. 2024]

目次

目次

最大マッチング問題

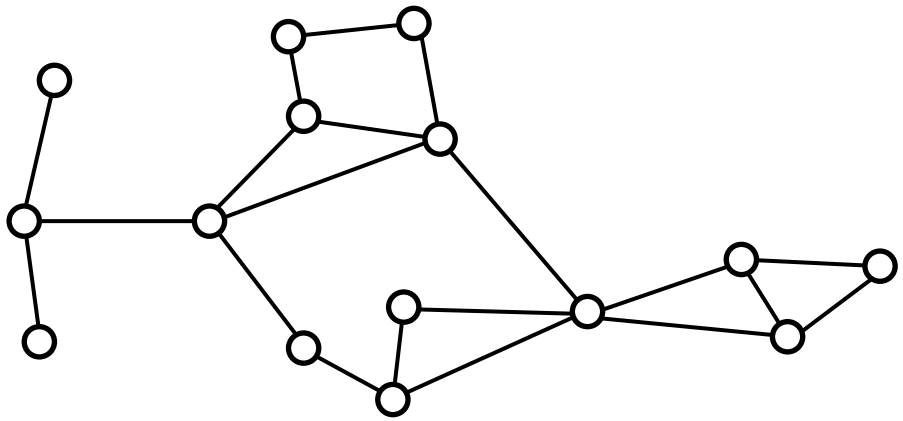
入力: $G = (V, E)$: 無向グラフ（連結）

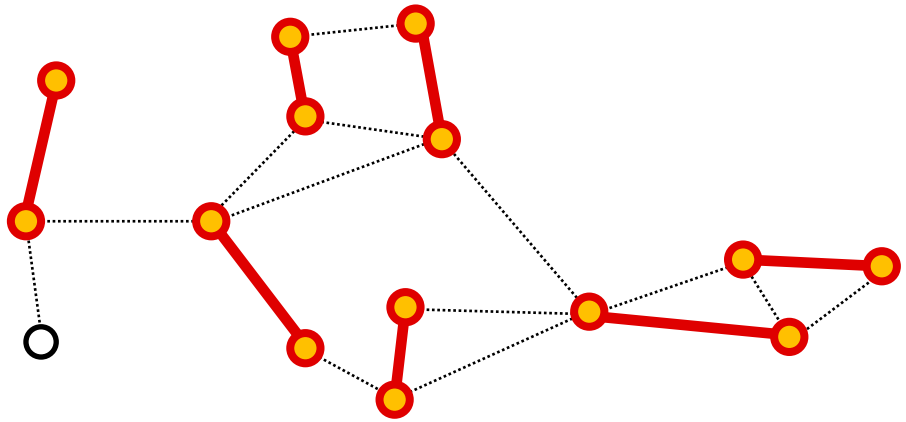
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

頂点を共有しない辺部分集合

$n = |V|,\ m = |E|$

μ: 最大サイズ



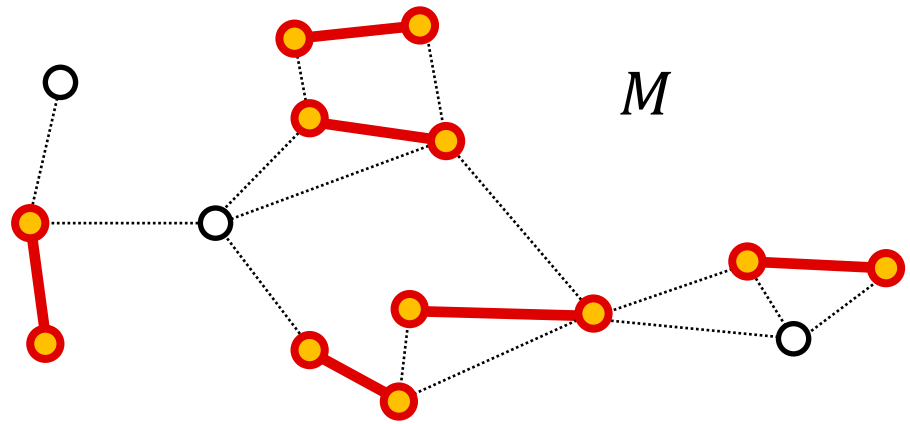


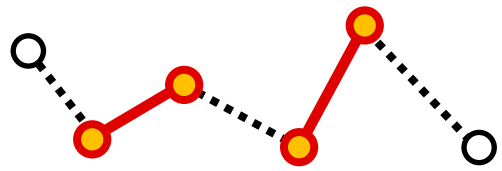
マッチングと増加道

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

補題　マッチング M が最大でない $\Leftrightarrow$ M に関する**増加道**が存在





M と $E \setminus M$ の辺を
交互に通るような
非マッチ頂点間のパス

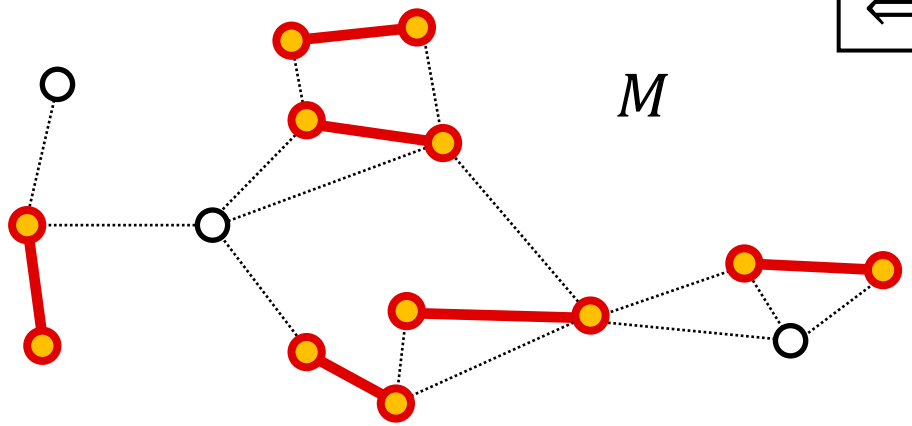
マッチングと増加道

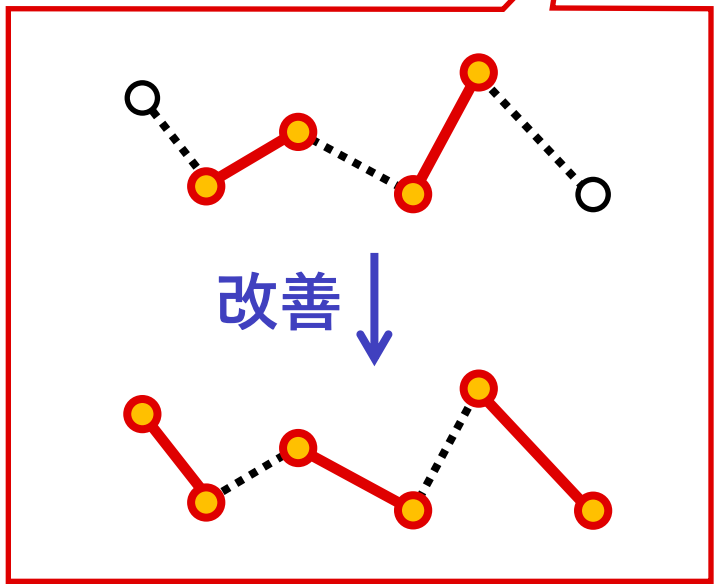
入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

補題　マッチング M が最大でない $\Leftrightarrow$ M に関する**増加道**が存在

$\Leftarrow$ の証明





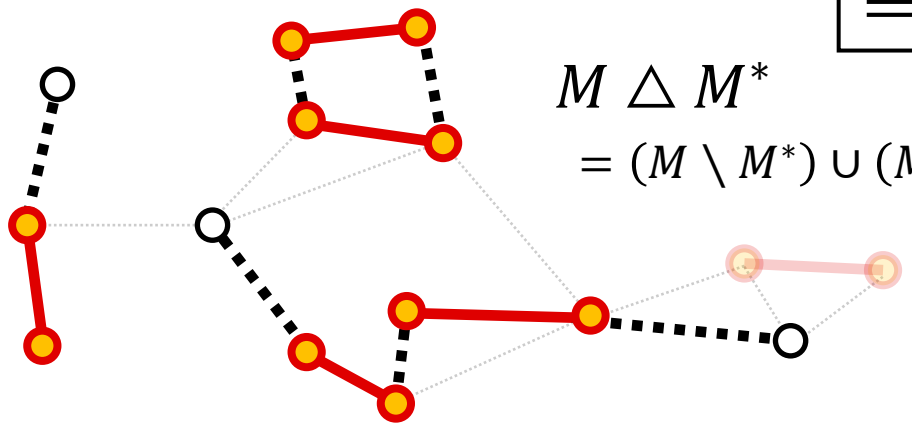
マッチングと増加道

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

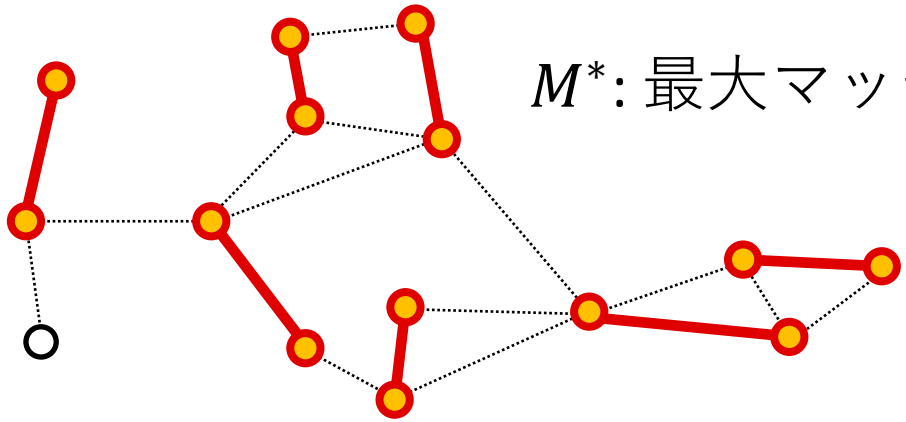
$n = |V|,\ m = |E|$
μ: 最大サイズ

補題　マッチング M が最大でない $\Leftrightarrow$ M に関する**増加道**が存在

$\Rightarrow$ の証明



$M \triangle M^*$
$= (M \setminus M^*) \cup (M^* \setminus M)$



M^*: 最大マッチング

マッチングと増加道

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

<u>補題</u>　マッチング M が最大でない $\Leftrightarrow$ M に関する**増加道**が存在

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

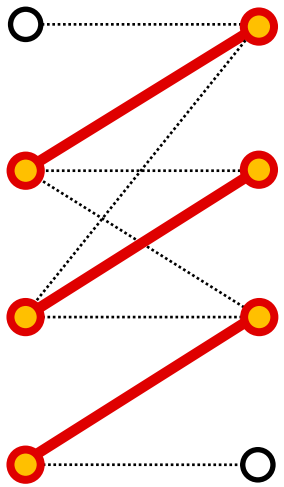
2 部グラフに対する増加道アルゴリズム

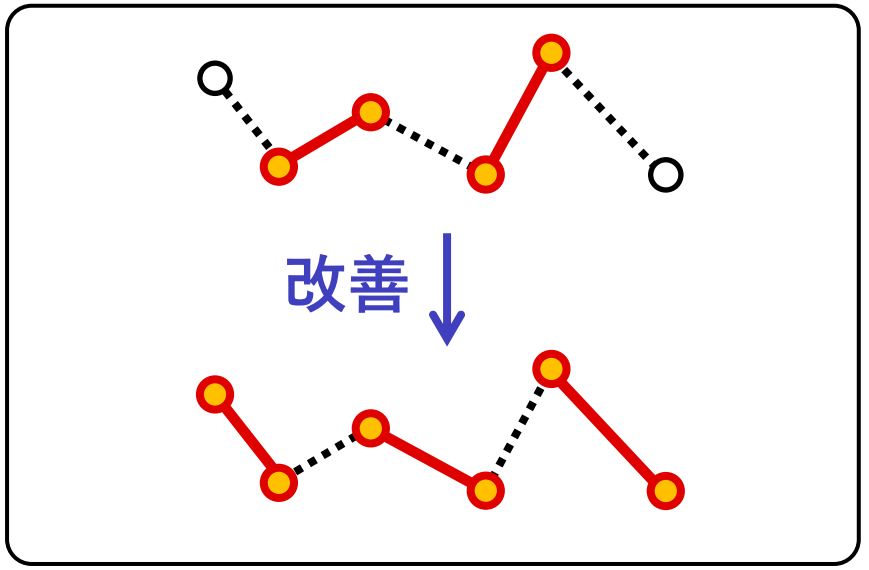
[Kőnig 1931, Ford–Fulkerson 1956]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）





2 部グラフに対する増加道アルゴリズム

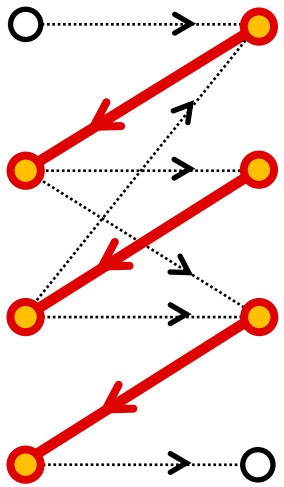
[Kőnig 1931, Ford–Fulkerson 1956]

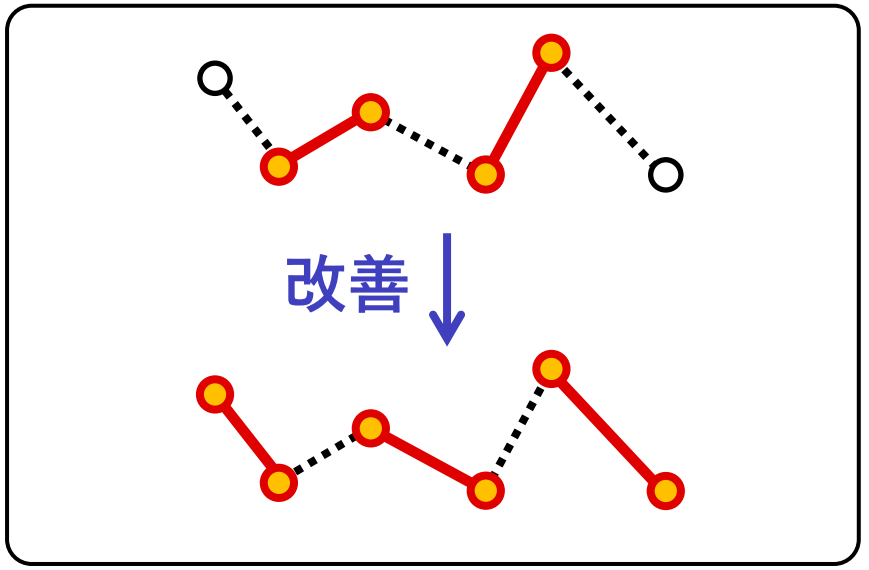
[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

向き付け





2 部グラフに対する増加道アルゴリズム

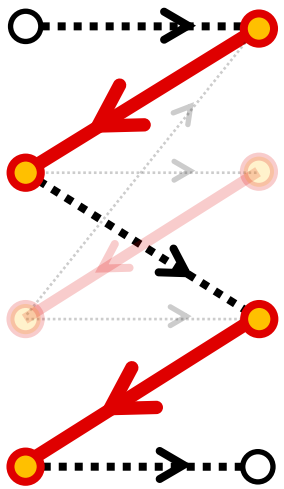
[Kőnig 1931, Ford–Fulkerson 1956]

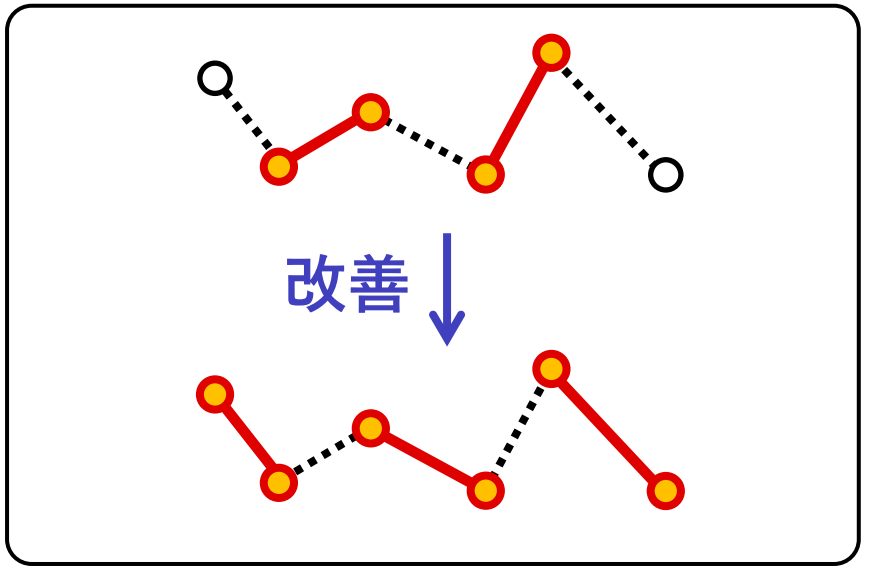
[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\left(\mu \cdot \mathrm{FP}(n, m, \mu)\right)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

到達可能性





2 部グラフに対する増加道アルゴリズム

[Kőnig 1931, Ford–Fulkerson 1956]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

向き付けと BFS (DFS) だけで，増加道が（存在すれば）1 つ見つかる

（見つからなければ最大 ≈ 最大最小定理）

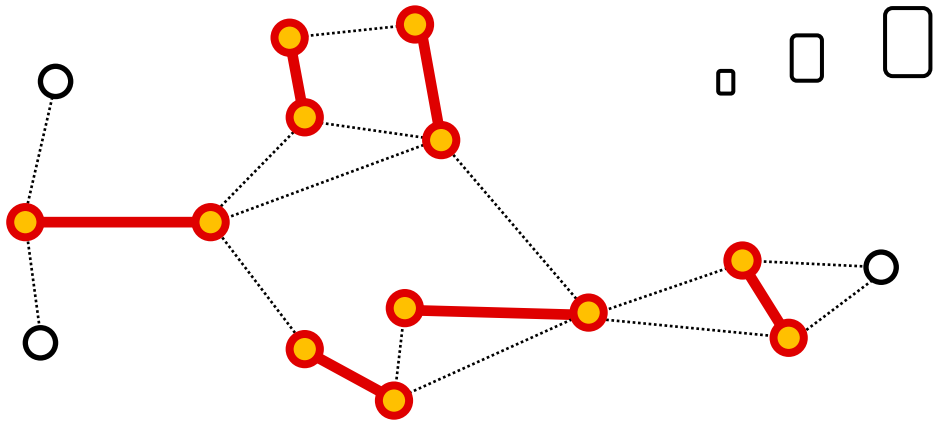
$\mathrm{FP}(n, m, \mu) = \mathrm{O}(m)$ → 全体で $\mathrm{O}(\mu m)$ 時間

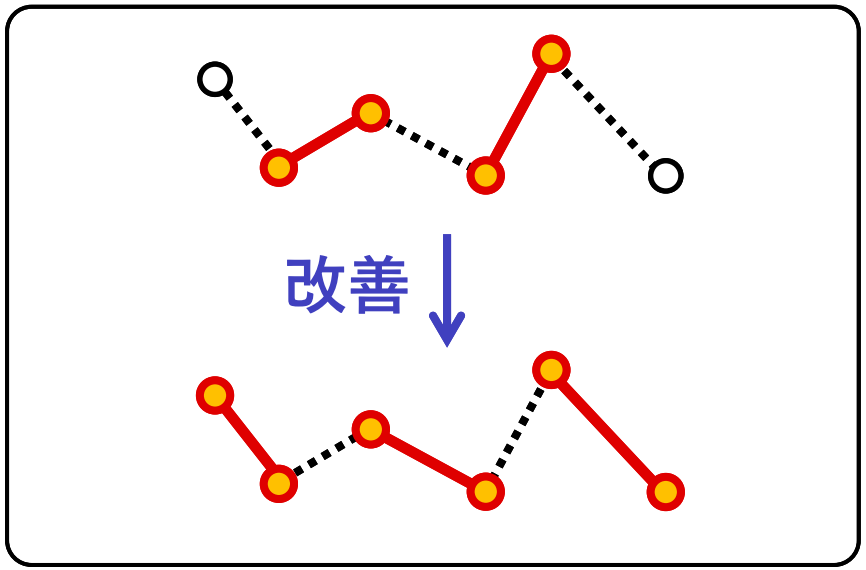
一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）



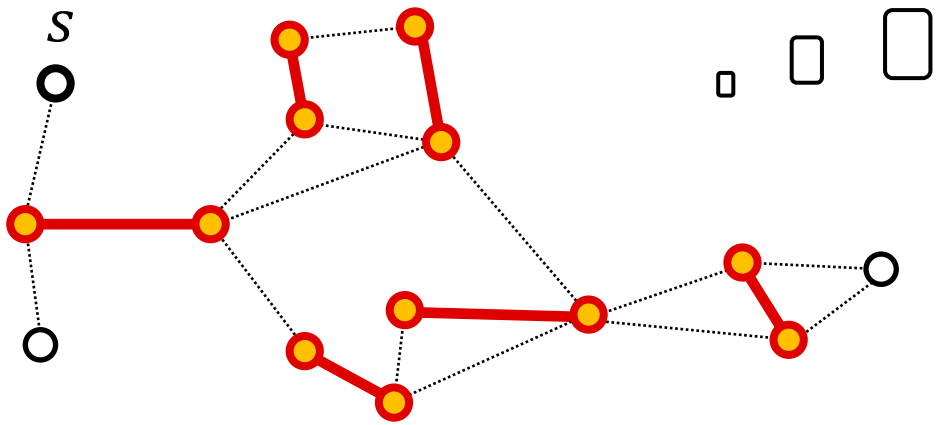


一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

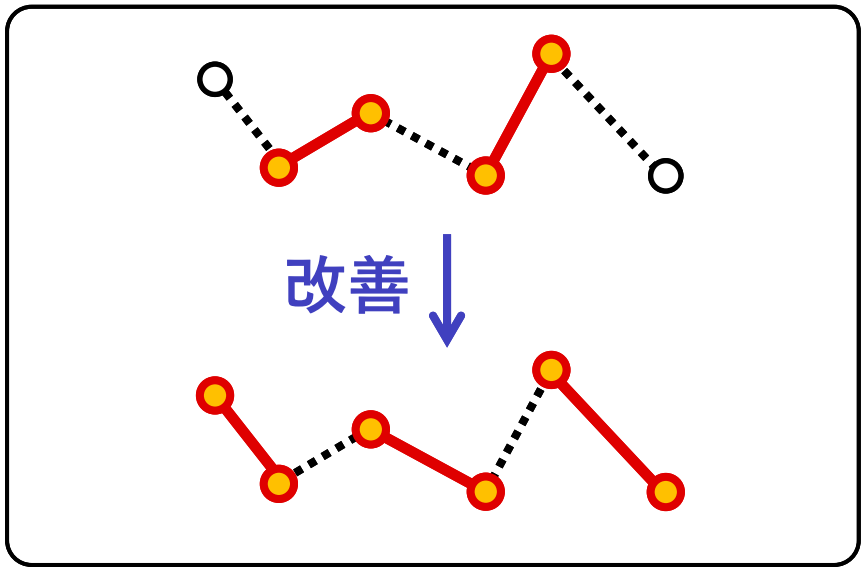
[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\left(\mu \cdot \mathrm{FP}(n, m, \mu)\right)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）



交互 BFS

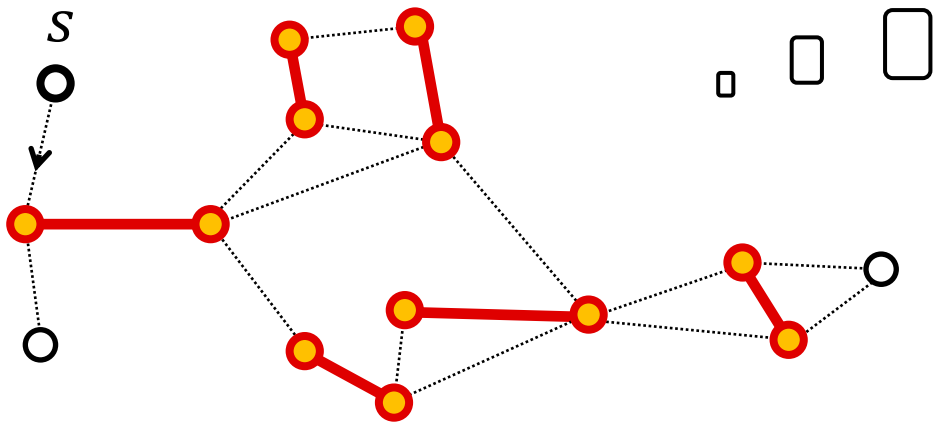


一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

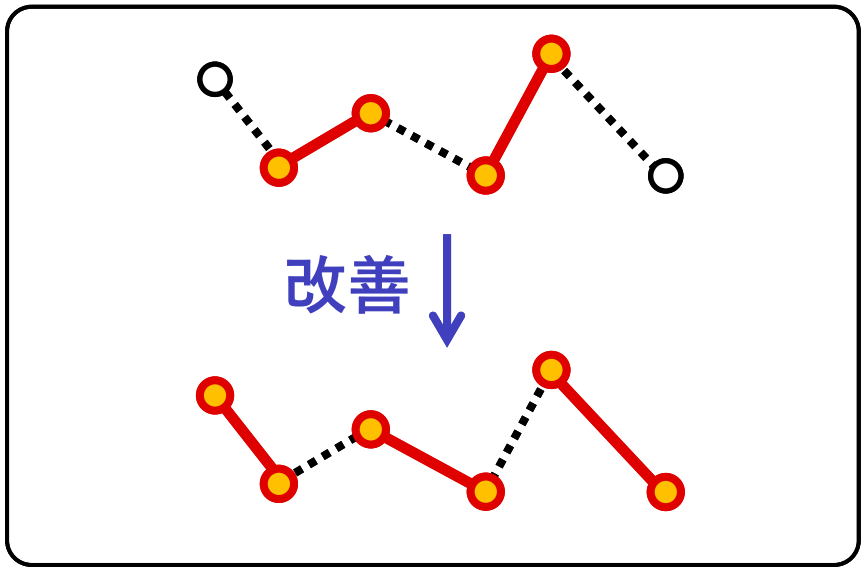
[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）



交互 BFS

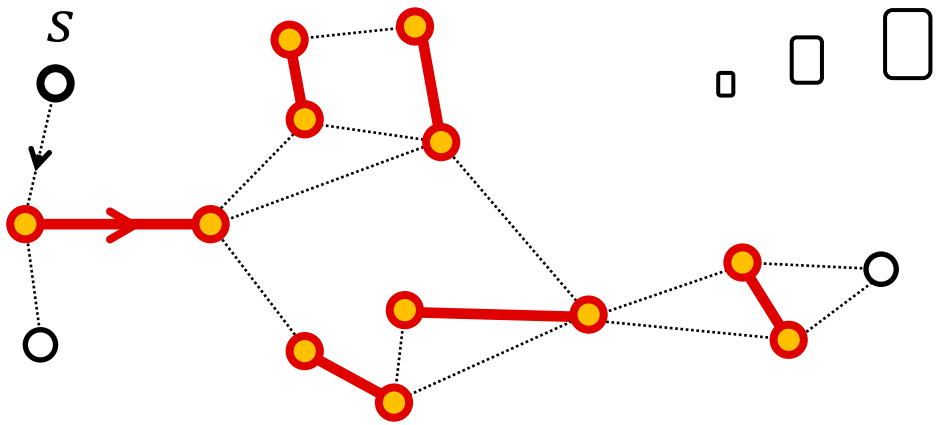


一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

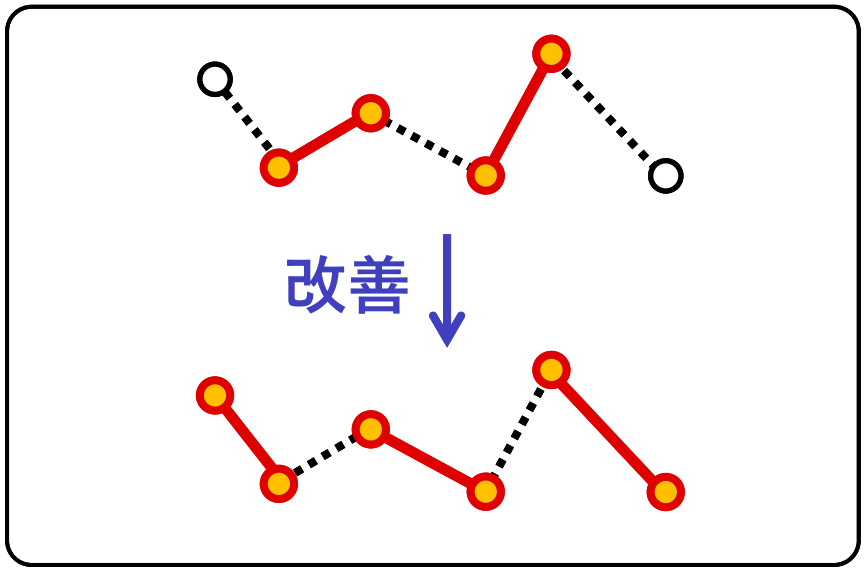
[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\left(\mu \cdot \mathrm{FP}(n, m, \mu)\right)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）



交互 BFS

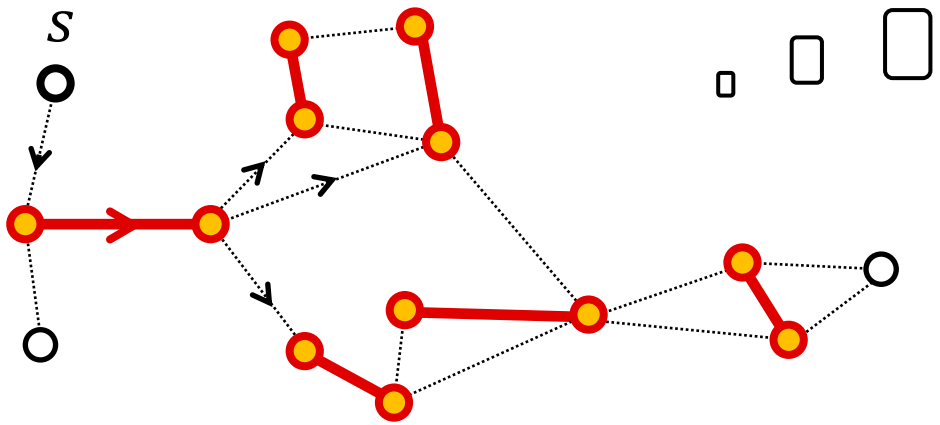


一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

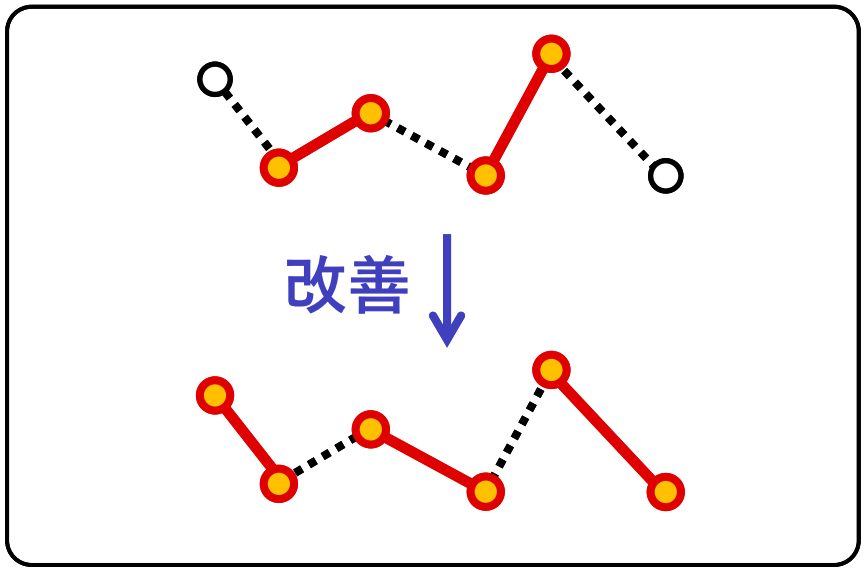
[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\left(\mu \cdot \mathrm{FP}(n, m, \mu)\right)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）



交互 BFS

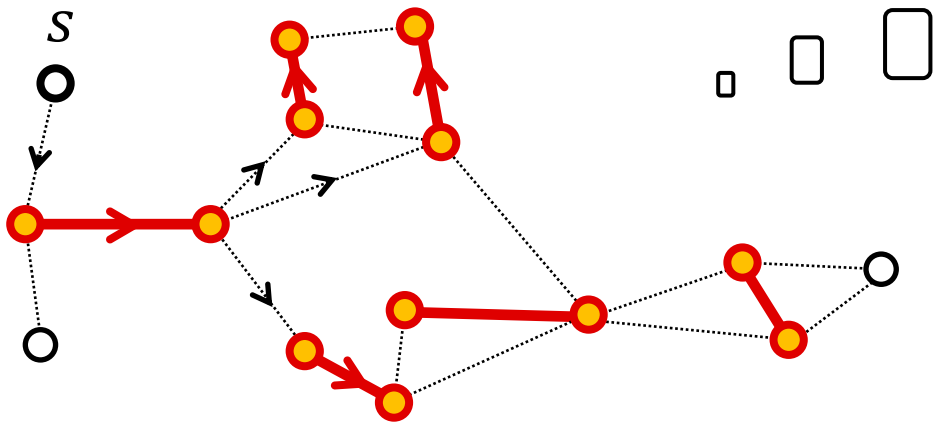


一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

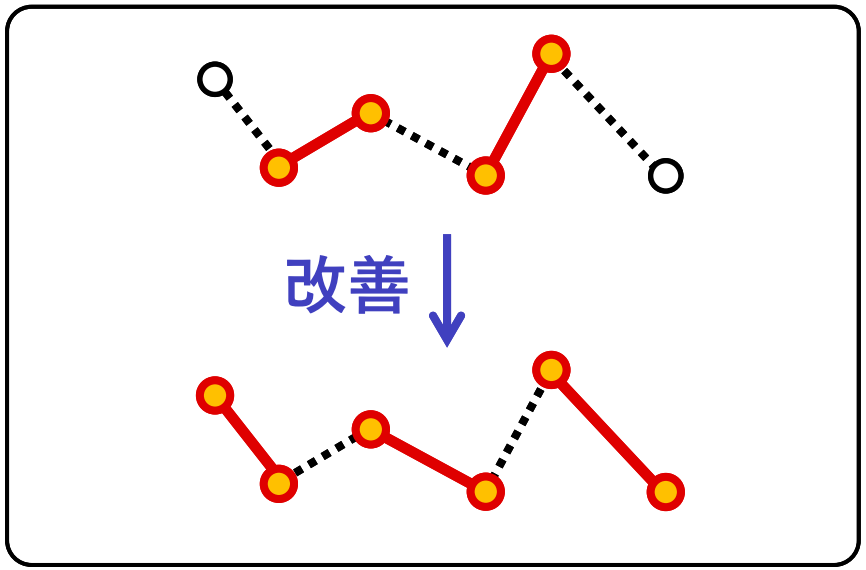
[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\left(\mu \cdot \mathrm{FP}(n, m, \mu)\right)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）



交互 BFS

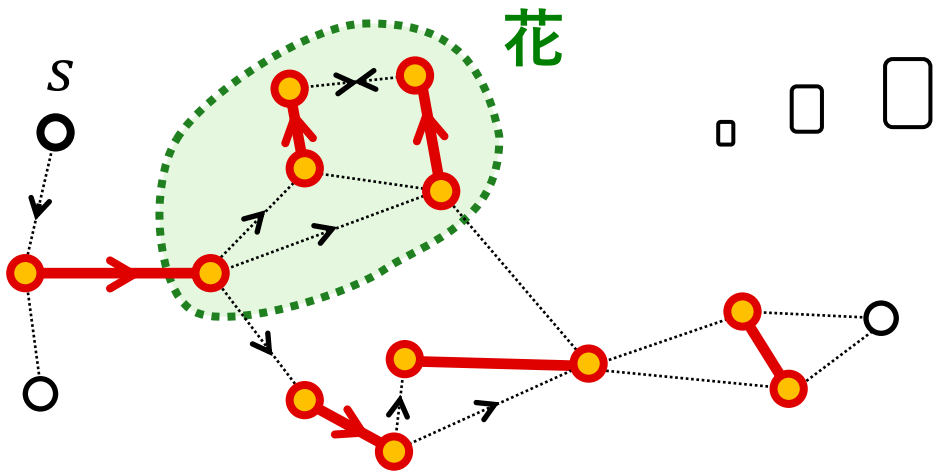


一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

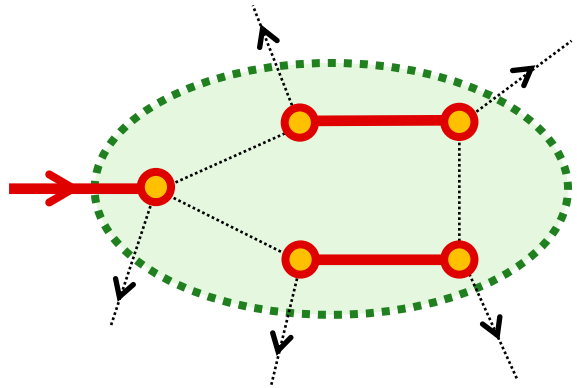
[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）



交互 BFS



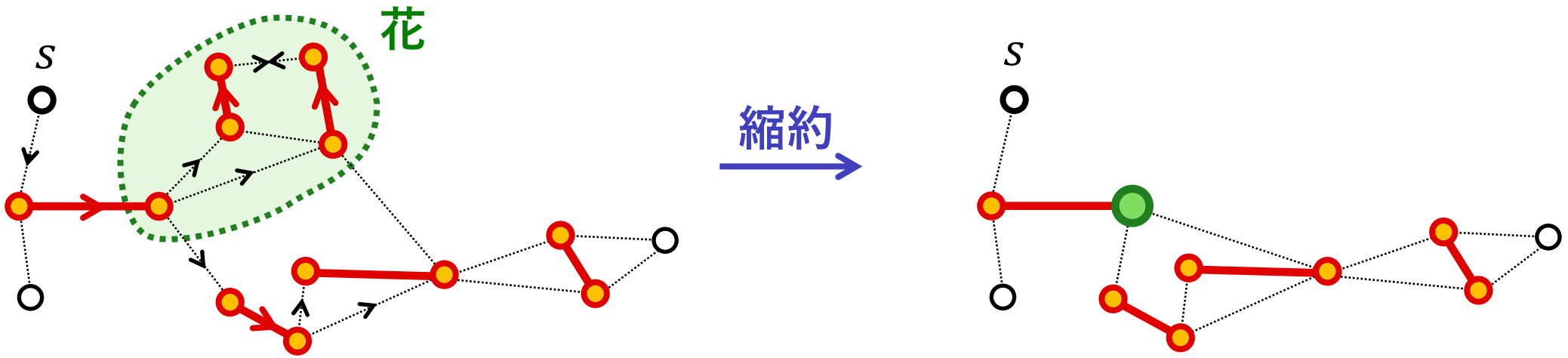
どの頂点からも出られる

一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\left(\mu \cdot \mathrm{FP}(n, m, \mu)\right)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）



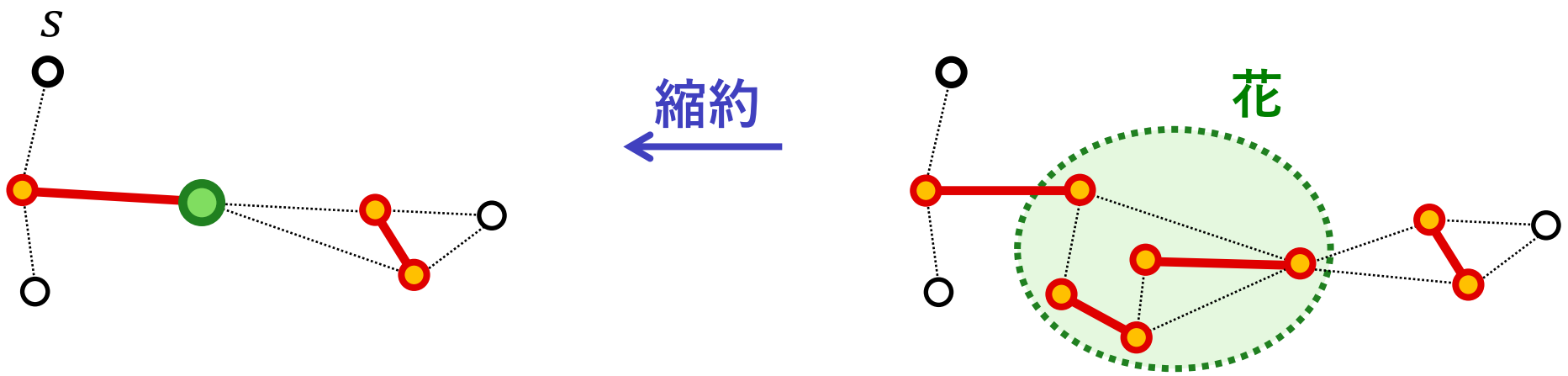
交互 BFS

一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

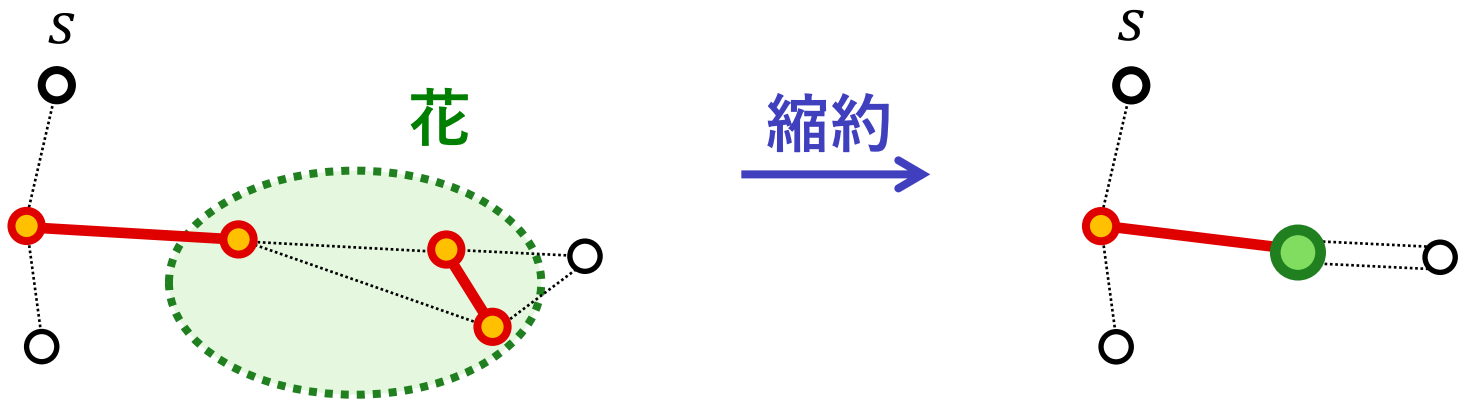


一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

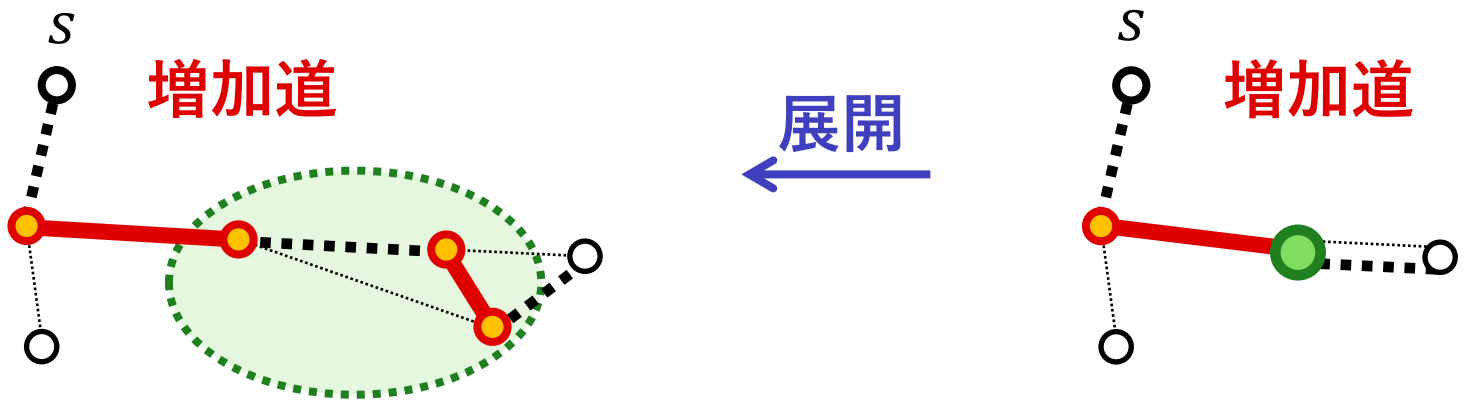


一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\left(\mu \cdot \mathrm{FP}(n, m, \mu)\right)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

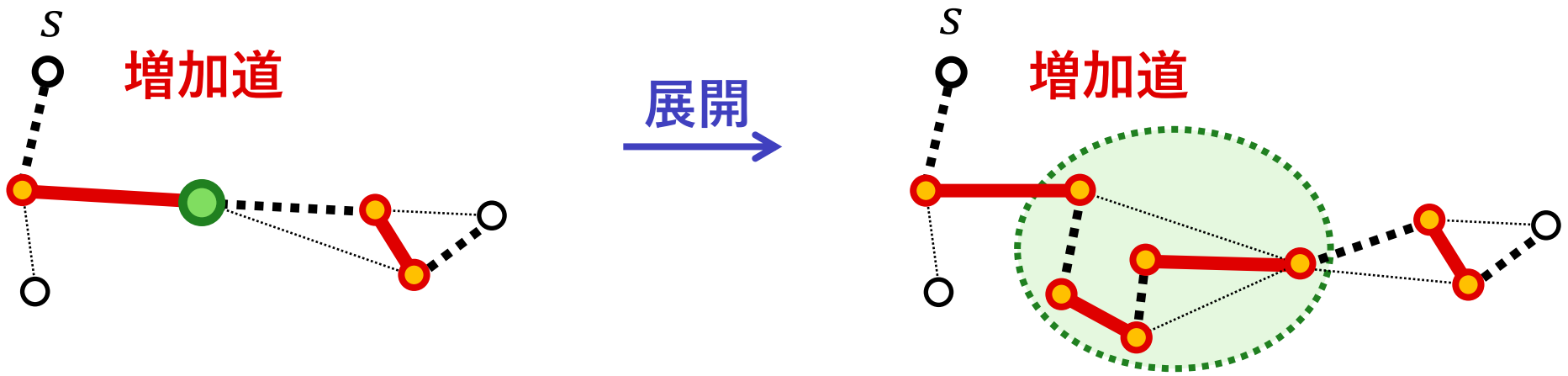


一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

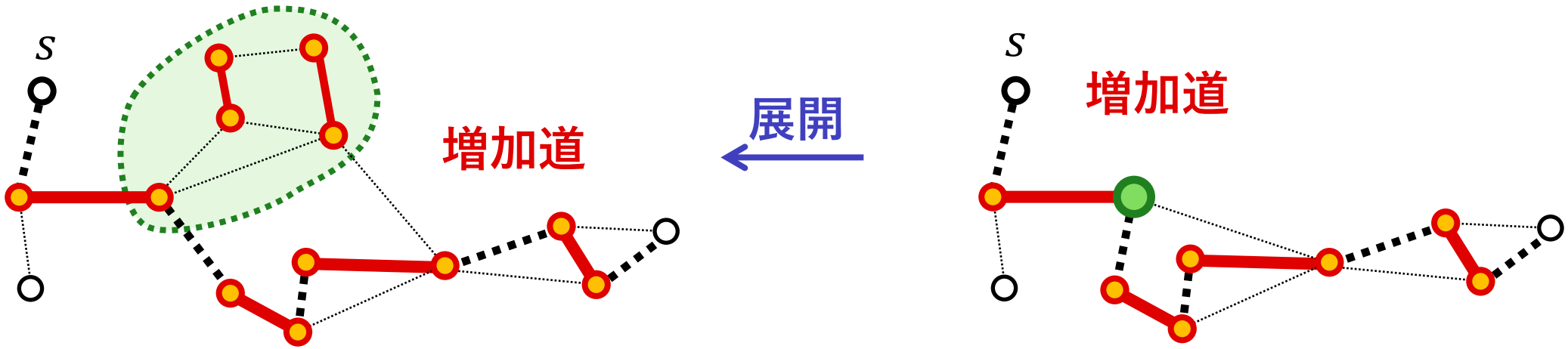


一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\left(\mu \cdot \mathrm{FP}(n, m, \mu)\right)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

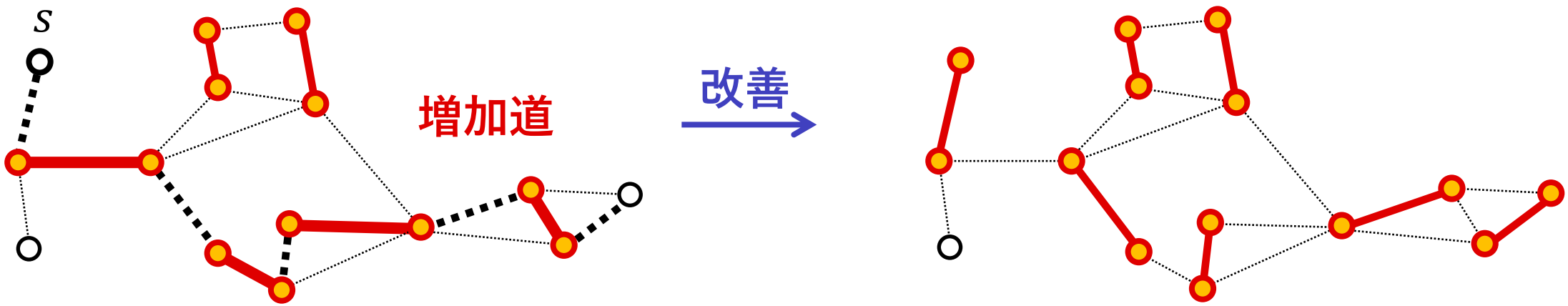


一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）



一般グラフに対する花縮約アルゴリズム [Edmonds 1965]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

- (改善あたりの縮約回数) $\leq \mu$ （各花は M の辺を 1 つ以上含む）
- 縮約，展開，交互 BFS はそれぞれ $\mathrm{O}(m)$ 時間でできる

$\mathrm{FP}(n, m, \mu) = \mathrm{O}(\mu m) \rightarrow$ 全体で $\mathrm{O}(\mu^2 m)$ 時間

演習問題 1. 花縮約の必要性

Q. 素朴に交互 BFS/DFS 的に増加道を探索すると何が困るのか？

各頂点 $v \in V$ を $v^{\mathrm{odd}}, v^{\mathrm{even}}$ に分割し，各辺 $e = \{u, w\} \in E$ を

- $e \notin M$ なら有向辺 $\left(u^{\mathrm{even}}, w^{\mathrm{odd}}\right), \left(w^{\mathrm{even}}, u^{\mathrm{odd}}\right)$
- $e \in M$ なら有向辺 $\left(u^{\mathrm{odd}}, w^{\mathrm{even}}\right), \left(w^{\mathrm{odd}}, u^{\mathrm{even}}\right)$

に置き換えて得られる有向グラフを G_M とする.

(1) G 中の M に関する増加道 P の始点を $s \in V$，終点を $t \in V$ とすると，P は G_M 中で s^{even} から t^{odd} への有向パスに対応することを説明せよ.

(2) 増加道が存在するのに，G_M でそのような有向パスを素朴な BFS/DFS で（木を作る形で）探索するだけでは増加道が見つけられない例を示せ.

(3) (2) の失敗を踏まえて，訪問済み頂点集合を管理するような探索を考える. その計算量（増加道を見つけるために必要な情報）が指数的になる例を示せ.

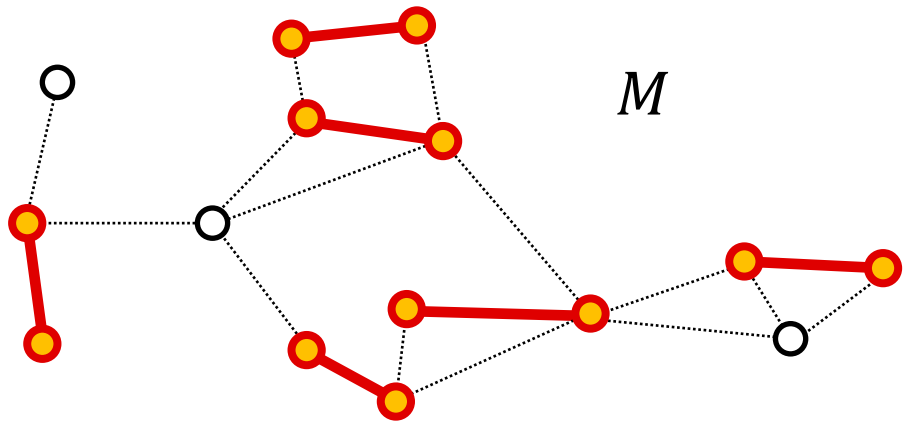
目次

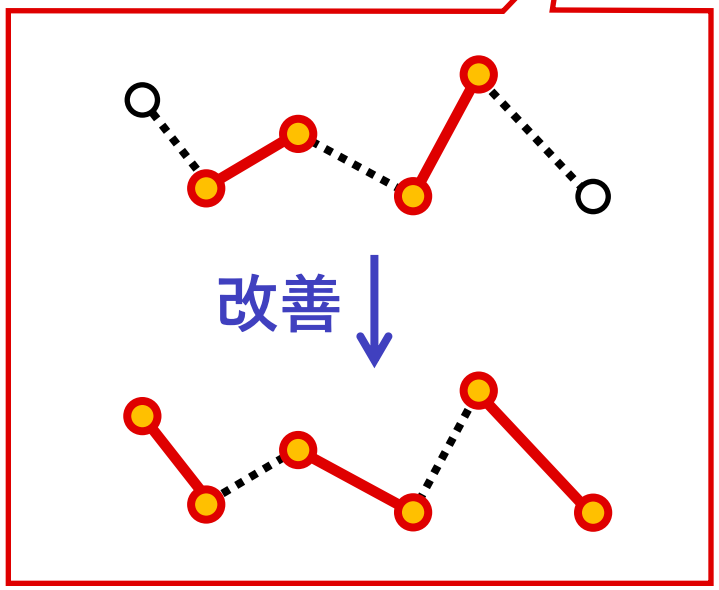
マッチングと増加道

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

補題　マッチング M が最大でない $\Leftrightarrow$ M に関する**増加道**が存在





演習問題 2. 最短増加道の重要性

Q. 証明せよ

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

補題 (1) M がサイズ $\mu - k$ のマッチング $(1 \le k \le \mu)$
$\Longrightarrow$ M に関する増加道であって長さが $\frac{2\mu}{k}$ 未満のものが存在

補題 (2) M: 非最大マッチング, ℓ: M に関する**最短**増加道の長さ
$M \triangle M'$ は**極大な**点素最短増加道（長さが全て ℓ）集合をなす
$\Longrightarrow$ M' に関する長さ ℓ 以下の増加道は存在しない

[Hopcroft–Karp 1973]

演習問題 2. 最短増加道の重要性

Q. 証明せよ

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

補題 (1) M がサイズ $\mu - k$ のマッチング $(1 \leq k \leq \mu)$
$\Longrightarrow$ M に関する増加道であって長さが $\frac{2\mu}{k}$ 未満のものが存在

補題 (2) M: 非最大マッチング, ℓ: M に関する**最短**増加道の長さ
$M \triangle M'$ は**極大な**点素最短増加道（長さが全て ℓ）集合をなす
$\Longrightarrow$ M' に関する長さ ℓ 以下の増加道は存在しない

[Hopcroft–Karp 1973]

最大マッチングを求める高速なアルゴリズム

[高速なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道が存在する限り，
   極大な点素最短増加道集合を見つけてそれで M を更新

- 最短増加道の長さは単調増加

補題　M: 非最大マッチング，ℓ: M に関する**最短**増加道の長さ
$M \triangle M'$ は**極大な**点素最短増加道（長さが全て ℓ）集合をなす
$\Longrightarrow$ M' に関する長さ ℓ 以下の増加道は存在しない

[Hopcroft–Karp 1973]

最大マッチングを求める高速なアルゴリズム

[高速なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道が存在する限り，
   極大な点素最短増加道集合を見つけてそれで M を更新

- 最短増加道の長さは単調増加

<u>補題</u>　M: 非最大マッチング，ℓ: M に関する**最短**増加道の長さ
$M \triangle M'$ は**極大な**点素最短増加道（長さが全て ℓ）集合をなす
$\Longrightarrow$ M' に関する長さ ℓ 以下の増加道は存在しない

[Hopcroft–Karp 1973]

最大マッチングを求める高速なアルゴリズム

[高速なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道が存在する限り，
   極大な点素最短増加道集合を見つけてそれで M を更新

- 最短増加道の長さは単調増加
  → $\left(|M| \leq \mu - \sqrt{\mu} \text{ の間の更新回数}\right) \leq \sqrt{\mu}$

補題　M がサイズ $\mu - k$ のマッチング $(1 \leq k \leq \mu)$

$\Longrightarrow$ M に関する増加道であって長さが $\frac{2\mu}{k}$ 未満のものが存在

[Hopcroft–Karp 1973]

最大マッチングを求める高速なアルゴリズム

[高速なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道が存在する限り，
   極大な点素最短増加道集合を見つけてそれで M を更新

- 最短増加道の長さは単調増加
  → $\left(|M| \leq \mu - \sqrt{\mu} \text{ の間の更新回数}\right) \leq \sqrt{\mu}$
- 明らかに $\left(|M| \geq \mu - \sqrt{\mu} \text{ の間の更新回数}\right) \leq \sqrt{\mu}$

全体で $\mathrm{O}\left(\sqrt{\mu} \cdot \mathrm{FMDSP}(n, m)\right)$ 時間 [Hopcroft–Karp 1973]

（$\mathrm{FMDSP}(\cdot)$: 極大な点素最短増加道集合を見つけるための時間）

$O(\sqrt{\mu}m)$ 時間アルゴリズムの設計

極大な点素最短増加道集合による更新で $O\left(\sqrt{\mu} \cdot \mathrm{FMDSP}(n,m)\right)$ 時間

（$\mathrm{FMDSP}(\cdot)$: 極大な点素最短増加道集合を見つけるための時間）

→ $\mathrm{FMDSP}(n,m) = O(m)$ とできれば十分

- G が 2 部グラフなら結構簡単（向き付け + BFS + DFS でよい）
  [Hopcroft–Karp 1973]
- G が一般グラフだと簡単ではないが可能ではある
  [Micali–Vazirani 1980, Vazirani 2024]

Q. 本質的な違いは何なのか？

A. 最短交互道が **BFS-honesty** という性質を満たすかどうか
（直観的には「最短路の部分路は最短路であってほしい」という性質）

$O(\sqrt{\mu}m)$ 時間アルゴリズムの設計

極大な点素最短増加道集合による更新で $O\left(\sqrt{\mu} \cdot \mathrm{FMDSP}(n,m)\right)$ 時間

（$\mathrm{FMDSP}(\cdot)$: 極大な点素最短増加道集合を見つけるための時間）

→ $\mathrm{FMDSP}(n,m) = O(m)$ とできれば十分

- G が 2 部グラフなら結構簡単（向き付け + BFS + DFS でよい）
  [Hopcroft–Karp 1973]
- G が一般グラフだと簡単ではないが可能ではある
  [Micali–Vazirani 1980, Vazirani 2024]

Q. 本質的な違いは何なのか？

A. 最短交互道が **BFS-honesty** という性質を満たすかどうか
（直観的には「最短路の部分路は最短路であってほしい」という性質）

$O(\sqrt{\mu}m)$ 時間アルゴリズムの設計

極大な点素最短増加道集合による更新で $O\left(\sqrt{\mu} \cdot \mathrm{FMDSP}(n,m)\right)$ 時間

（$\mathrm{FMDSP}(\cdot)$: 極大な点素最短増加道集合を見つけるための時間）

→ $\mathrm{FMDSP}(n,m) = O(m)$ とできれば十分

- G が 2 部グラフなら結構簡単（向き付け + BFS + DFS でよい）
  [Hopcroft–Karp 1973]
- G が一般グラフだと簡単ではないが可能ではある
  [Micali–Vazirani 1980, Vazirani 2024]

Q. 本質的な違いは何なのか？

A. 最短交互道が **BFS-honesty** という性質を満たすかどうか
（直観的には「最短路の部分路は最短路であってほしい」という性質）

目次

最短交互道の BFS-honesty

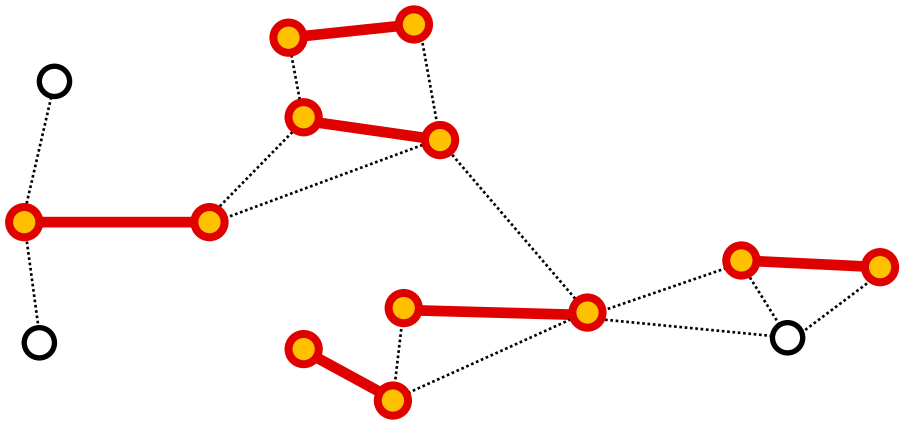
各頂点 $v \in V$ に対し，以下を定義する．

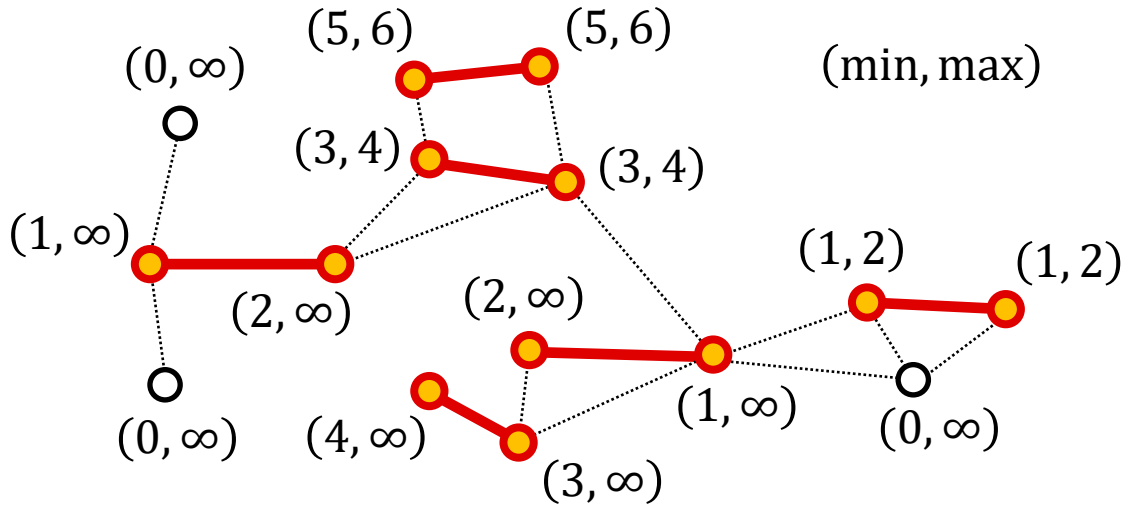
$\mathrm{oddlevel}(v)$: 非マッチ頂点から v への最短**奇数長**交互道の長さ

$\mathrm{evenlevel}(v)$: 非マッチ頂点から v への最短**偶数長**交互道の長さ

$\mathrm{minlevel}(v) = \min\{\mathrm{oddlevel}(v), \mathrm{evenlevel}(v)\}$

$\mathrm{maxlevel}(v) = \max\{\mathrm{oddlevel}(v), \mathrm{evenlevel}(v)\}$





最短交互道の BFS-honesty

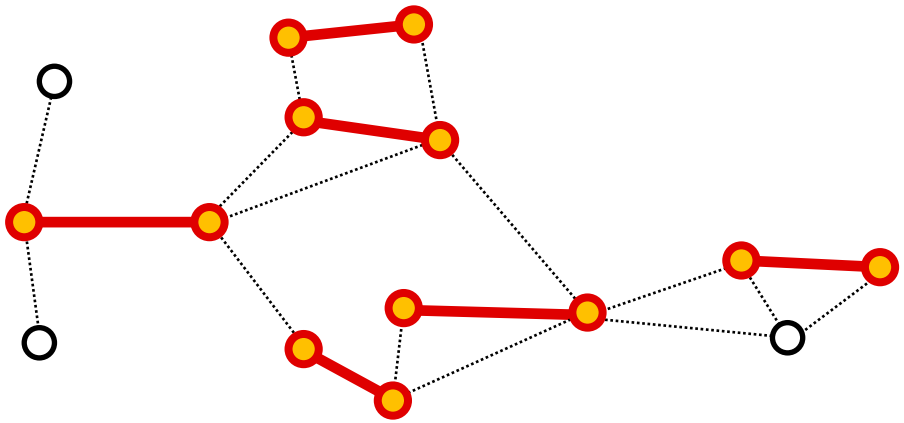
各頂点 $v \in V$ に対し，以下を定義する．

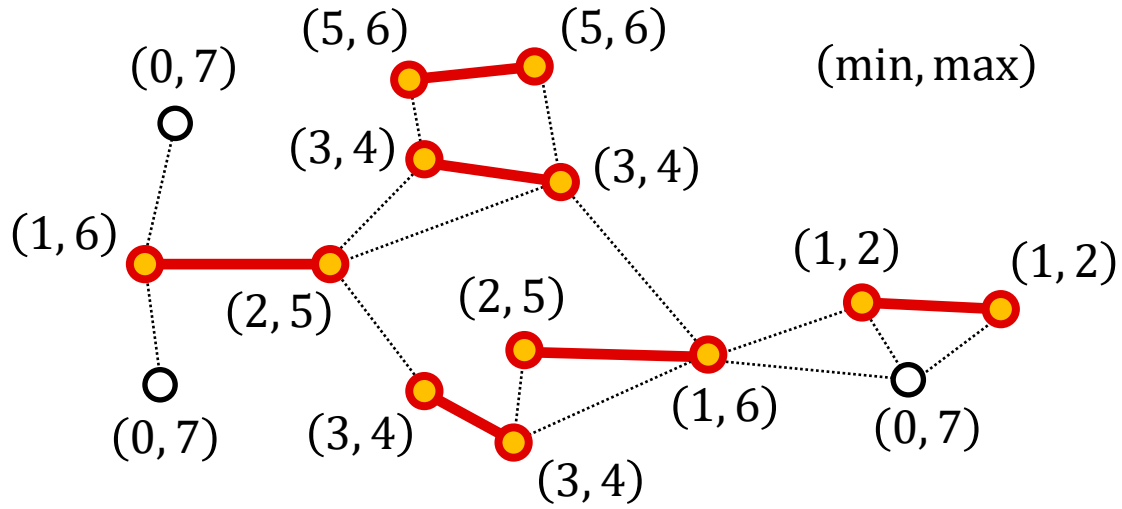
$\mathrm{oddlevel}(v)$: 非マッチ頂点から v への最短**奇数長**交互道の長さ

$\mathrm{evenlevel}(v)$: 非マッチ頂点から v への最短**偶数長**交互道の長さ

$\mathrm{minlevel}(v) = \min\{\mathrm{oddlevel}(v), \mathrm{evenlevel}(v)\}$

$\mathrm{maxlevel}(v) = \max\{\mathrm{oddlevel}(v), \mathrm{evenlevel}(v)\}$





最短交互道の BFS-honesty

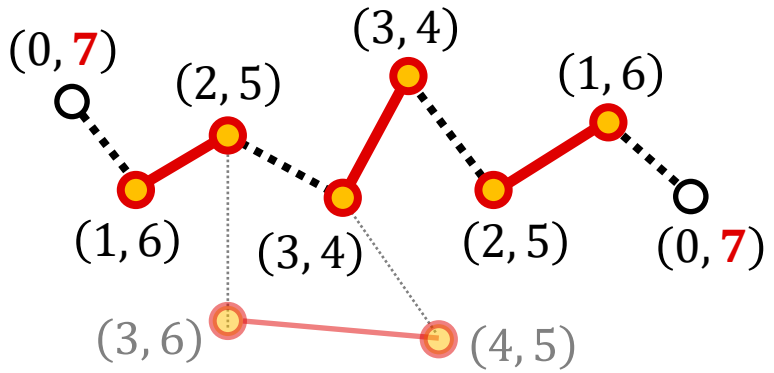
各頂点 $v \in V$ に対し，以下を定義する．

$\mathrm{oddlevel}(v)$: 非マッチ頂点から v への最短**奇数長**交互道の長さ

$\mathrm{evenlevel}(v)$: 非マッチ頂点から v への最短**偶数長**交互道の長さ

$\mathrm{minlevel}(v) = \min\{\mathrm{oddlevel}(v), \mathrm{evenlevel}(v)\}$

$\mathrm{maxlevel}(v) = \max\{\mathrm{oddlevel}(v), \mathrm{evenlevel}(v)\}$



- $\ell = \min\{\,\mathrm{oddlevel}(v) \mid v$: 非マッチ頂点 $\}$
- G が 2 部グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路は常に最短偶奇交互道 (BFS-honest)
- G が一般グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路が最短偶奇交互道とは限らない

最短交互道の BFS-honesty

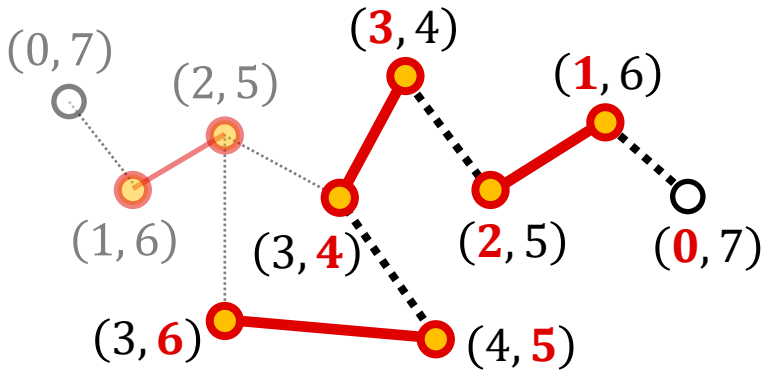
各頂点 $v \in V$ に対し，以下を定義する．

$\mathrm{oddlevel}(v)$: 非マッチ頂点から v への最短**奇数長**交互道の長さ

$\mathrm{evenlevel}(v)$: 非マッチ頂点から v への最短**偶数長**交互道の長さ

$\mathrm{minlevel}(v) = \min\{\mathrm{oddlevel}(v), \mathrm{evenlevel}(v)\}$

$\mathrm{maxlevel}(v) = \max\{\mathrm{oddlevel}(v), \mathrm{evenlevel}(v)\}$



- $\ell = \min\{\, \mathrm{oddlevel}(v) \mid v$: 非マッチ頂点 $\}$
- G が 2 部グラフ

  $\Longrightarrow$ 最短偶奇交互道の先頭部分路は常に最短偶奇交互道 (BFS-honest)

- G が一般グラフ

  $\Longrightarrow$ 最短偶奇交互道の先頭部分路が最短偶奇交互道とは限らない

最短交互道の BFS-honesty

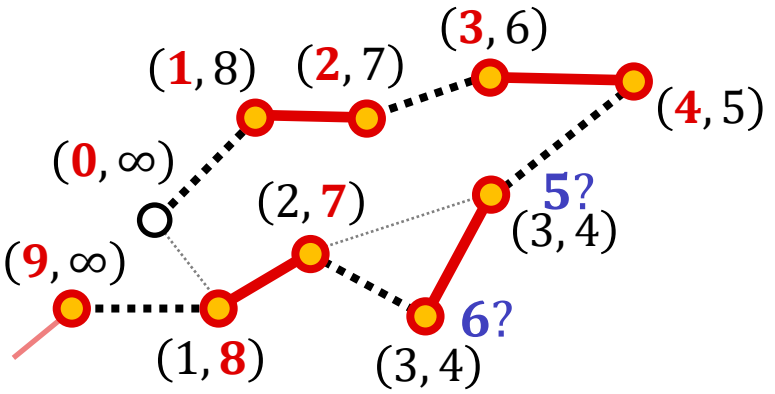
各頂点 $v \in V$ に対し，以下を定義する．

$\mathrm{oddlevel}(v)$: 非マッチ頂点から v への最短**奇数長**交互道の長さ

$\mathrm{evenlevel}(v)$: 非マッチ頂点から v への最短**偶数長**交互道の長さ

$\mathrm{minlevel}(v) = \min\{\mathrm{oddlevel}(v), \mathrm{evenlevel}(v)\}$

$\mathrm{maxlevel}(v) = \max\{\mathrm{oddlevel}(v), \mathrm{evenlevel}(v)\}$



- $\ell = \min\{\, \mathrm{oddlevel}(v) \mid v$: 非マッチ頂点 $\}$
- G が 2 部グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路は常に最短偶奇交互道 (BFS-honest)
- G が一般グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路が最短偶奇交互道とは限らない

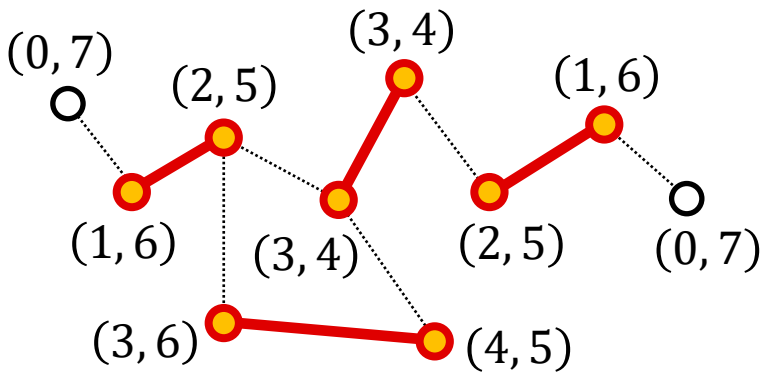
演習問題 3. 最短交互道の BFS-honesty

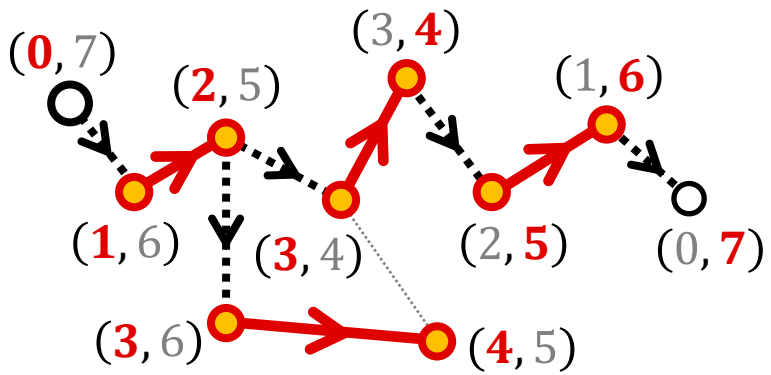
※ ここでは「交互道」は全て非マッチ頂点を始点とするものを指す.

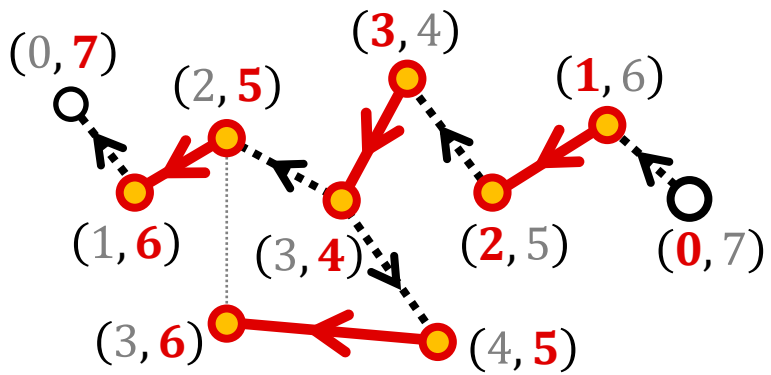
(1) 2 部グラフの場合，最短偶奇交互道が BFS-honesty を満たすことを示せ.
すなわち，任意の最短偶奇交互道 P と任意の頂点 $v \in V(P)$ に対して,
P の v までの先頭部分路は v への最短偶奇交互道であることを示せ.

(2) 一般グラフの場合，最短偶奇交互道は BFS-honesty を任意に破ることを示せ.
すなわち，任意の正整数 k に対して，(なるべく小さいグラフで)
以下の条件を満たす最短偶奇交互道 P と頂点 $v \in V(P)$ が存在する例を示せ.

条件: P の v までの先頭部分路の長さは,
同じ偶奇での v への交互道の長さのうち小さい方から k 番目に入らない.





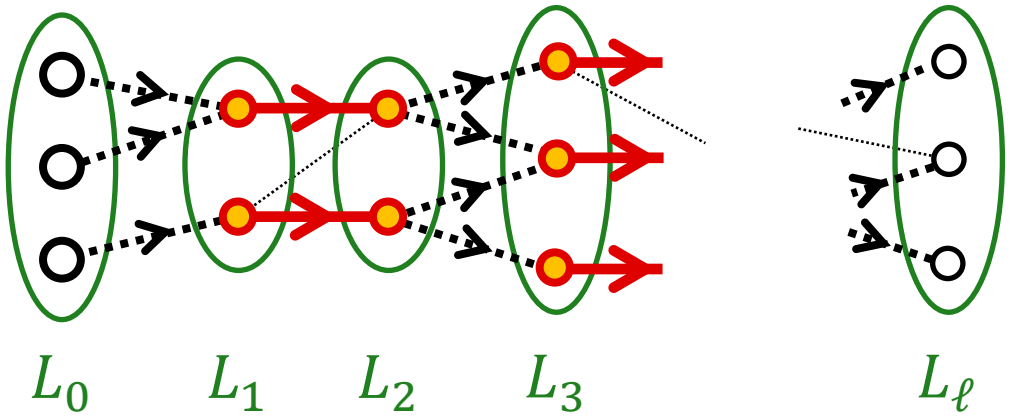


最短偶奇交互道 (Odd/Evenlevel) の計算

- G が 2 部グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路は常に最短偶奇交互道 (BFS-honest)
  - 最短偶奇交互道長 (odd/evenlevel) は全て BFS 的に与えられる
  - 偶奇は 2 部グラフのどちら側から BFS を始めるかのみで決まる
  - 極大な点素最短増加道集合は，BFS の親候補辺からなる DAG 上の DFS で求まる

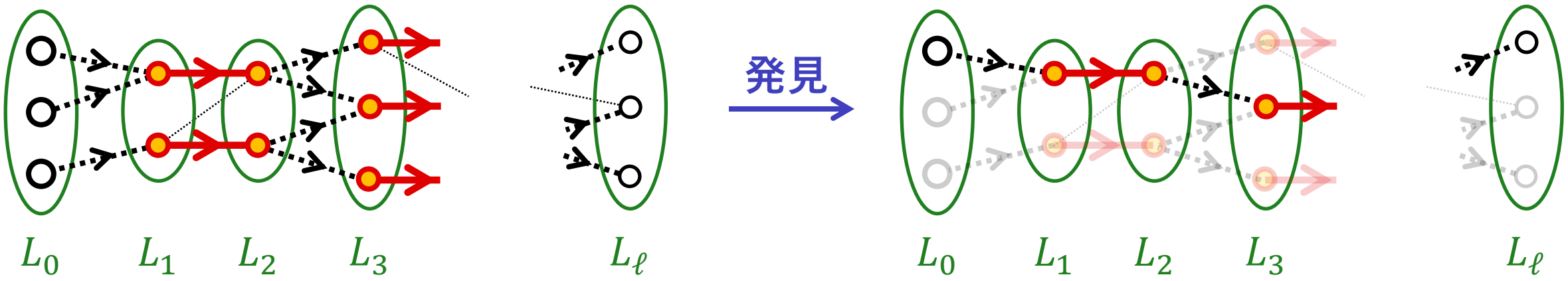
最短偶奇交互道 (Odd/Evenlevel) の計算

- G が 2 部グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路は常に最短偶奇交互道 (BFS-honest)
  - 最短偶奇交互道長 (odd/evenlevel) は全て BFS 的に与えられる
  - 偶奇は 2 部グラフのどちら側から BFS を始めるかのみで決まる
  - 極大な点素最短増加道集合は，BFS の親候補辺からなる DAG 上の DFS で求まる



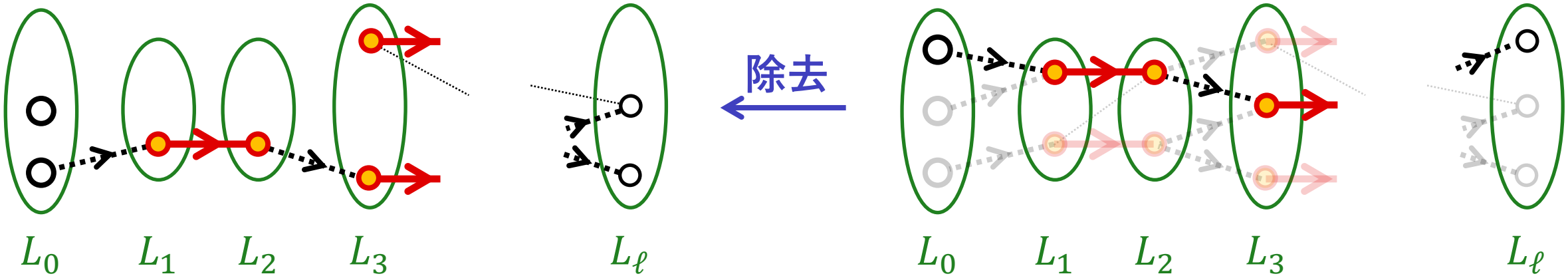
最短偶奇交互道 (Odd/Evenlevel) の計算

- G が 2 部グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路は常に最短偶奇交互道 (BFS-honest)
  - 最短偶奇交互道長 (odd/evenlevel) は全て BFS 的に与えられる
  - 偶奇は 2 部グラフのどちら側から BFS を始めるかのみで決まる
  - 極大な点素最短増加道集合は，BFS の親候補辺からなる DAG 上の DFS で求まる



最短偶奇交互道 (Odd/Evenlevel) の計算

- G が 2 部グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路は常に最短偶奇交互道 (BFS-honest)
  - 最短偶奇交互道長 (odd/evenlevel) は全て BFS 的に与えられる
  - 偶奇は 2 部グラフのどちら側から BFS を始めるかのみで決まる
  - 極大な点素最短増加道集合は，BFS の親候補辺からなる DAG 上の DFS で求まる



最短偶奇交互道 (Odd/Evenlevel) の計算

- G が 2 部グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路は常に最短偶奇交互道 (BFS-honest)
  - 最短偶奇交互道長 (odd/evenlevel) は全て BFS 的に与えられる
  - 偶奇は 2 部グラフのどちら側から BFS を始めるかのみで決まる
  - 極大な点素最短増加道集合は，BFS の親候補辺からなる DAG 上の DFS で求まる

- G が一般グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路が最短偶奇交互道とは限らない
  - 各頂点の minlevel は同様 (BFS-like) に直前の頂点から +1 の形で与えられる
  - 各頂点の maxlevel は誰がどのように与えているのか？ → 橋 (Bridge)

最短偶奇交互道 (Odd/Evenlevel) の計算

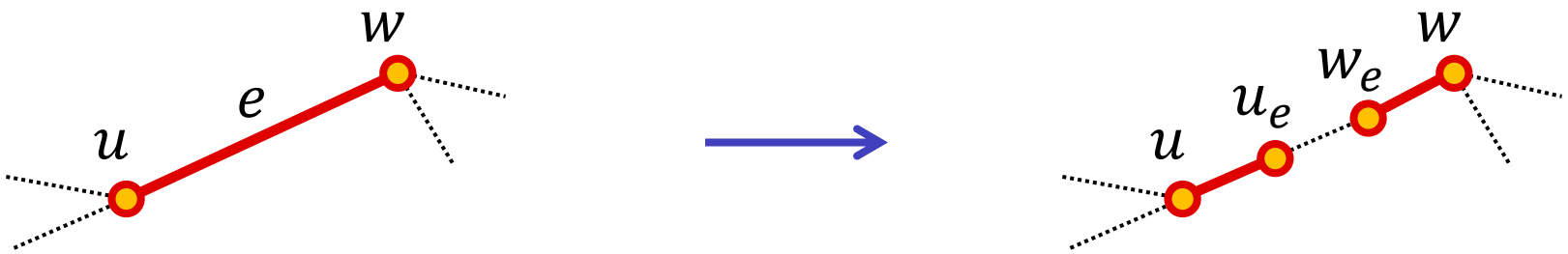
- G が 2 部グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路は常に最短偶奇交互道 (BFS-honest)
  - 最短偶奇交互道長 (odd/evenlevel) は全て BFS 的に与えられる
  - 偶奇は 2 部グラフのどちら側から BFS を始めるかのみで決まる
  - 極大な点素最短増加道集合は，BFS の親候補辺からなる DAG 上の DFS で求まる

- G が一般グラフ
  $\Longrightarrow$ 最短偶奇交互道の先頭部分路が最短偶奇交互道とは限らない
  - 各頂点の minlevel は同様 (BFS-like) に直前の頂点から +1 の形で与えられる
  - 各頂点の maxlevel は誰がどのように与えているのか？ → **橋 (Bridge)**

目次

状況の単純化

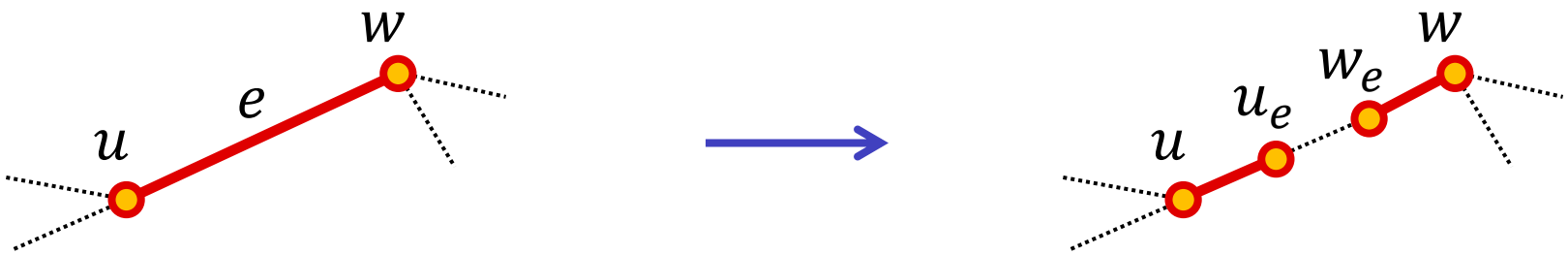
- 各マッチ辺 $e = \{u, w\} \in M$ を長さ 3 のパス (u, u_e, w_e, w) で置き換え，$\{u, u_e\}, \{w_e, w\}$ を新たなマッチ辺とする．



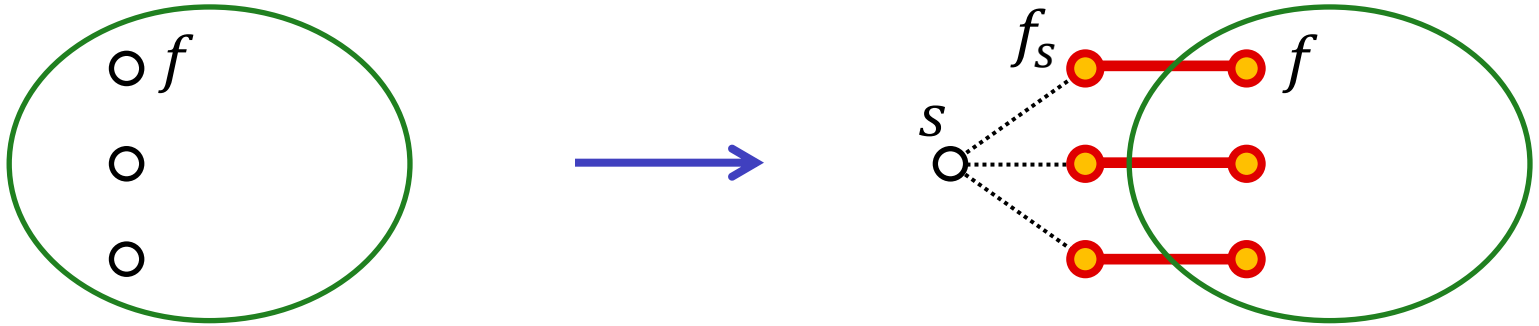
- 超始点 s を追加し，各非マッチ頂点 $f \in V \setminus V(M)$ に対して長さ 2 のパス (s, f_s, f) を追加し，$\{f_s, f\}$ を新たなマッチ辺とする．

状況の単純化

- 各マッチ辺 $e = \{u, w\} \in M$ を長さ 3 のパス (u, u_e, w_e, w) で置き換え，$\{u, u_e\}, \{w_e, w\}$ を新たなマッチ辺とする．



- 超始点 s を追加し，各非マッチ頂点 $f \in V \setminus V(M)$ に対して
長さ 2 のパス (s, f_s, f) を追加し，$\{f_s, f\}$ を新たなマッチ辺とする．



演習問題 4. 単純化の効果

単純化を行った後のグラフとマッチングをそれぞれ $\tilde{G}, \tilde{M}$ とする.

(1) G 中の M に関する長さ k の増加道は,
$\tilde{G}$ 中で s を基点とする長さ $2k+3$ の花と 1 対 1 に対応することを説明せよ.
以降では，元の最短増加道の長さを ℓ として，$\tilde{\ell} := 2\ell + 3$ とする.

(2) $\tilde{G}, \tilde{M}$ に関する（s からの）odd/even/min/maxlevel を考える.
このとき，各マッチ辺は一方の端点の minlevel を与えることを示せ.
すなわち，s から交互道で到達可能な各マッチ辺 $\{u, w\} \in \tilde{M}$ に対して,

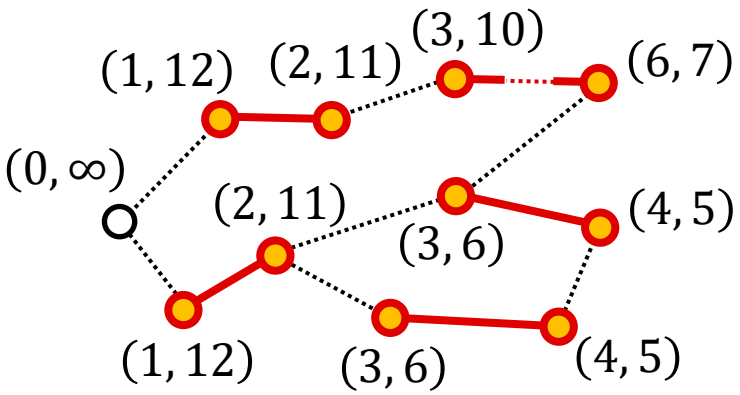
- $\mathrm{minlevel}(u) = \mathrm{evenlevel}(u) = \mathrm{oddlevel}(w) + 1$
- $\mathrm{minlevel}(w) = \mathrm{evenlevel}(w) = \mathrm{oddlevel}(u) + 1$

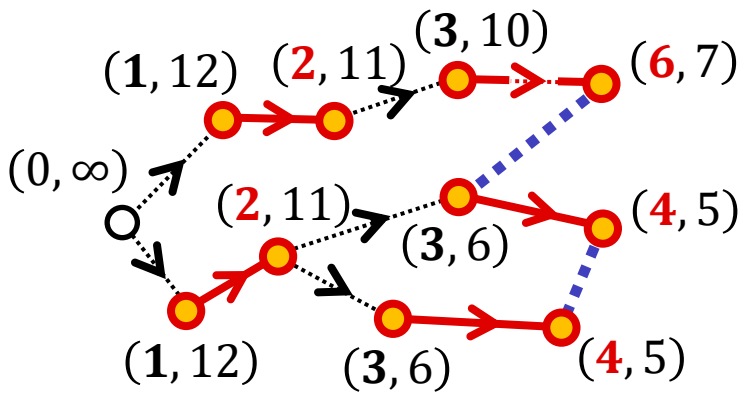
のいずれか一方（のみ）が成り立つことを示せ.

橋 (Bridge) と Tenacity

辺 $e = \{u, w\}$ が**橋 (Bridge)** である $\overset{\text{def}}{\Leftrightarrow}$ e が u, w どちらの minlevel も与えない

- e が非マッチ辺であり，以下をともに満たすことと同値
  - $\mathrm{oddlevel}(u) = \mathrm{maxlevel}(u)$ または $\mathrm{oddlevel}(u) < \mathrm{evenlevel}(w) + 1$
  - $\mathrm{oddlevel}(w) = \mathrm{maxlevel}(w)$ または $\mathrm{oddlevel}(w) < \mathrm{evenlevel}(u) + 1$
- $\mathrm{tenacity}(e) \coloneqq \mathrm{evenlevel}(u) + \mathrm{evenlevel}(w) + 1$ （e: 橋）
- $\mathrm{tenacity}(v) \coloneqq \mathrm{oddlevel}(v) + \mathrm{evenlevel}(v) = \mathrm{minlevel}(v) + \mathrm{maxlevel}(v)$ （$v \in V$）

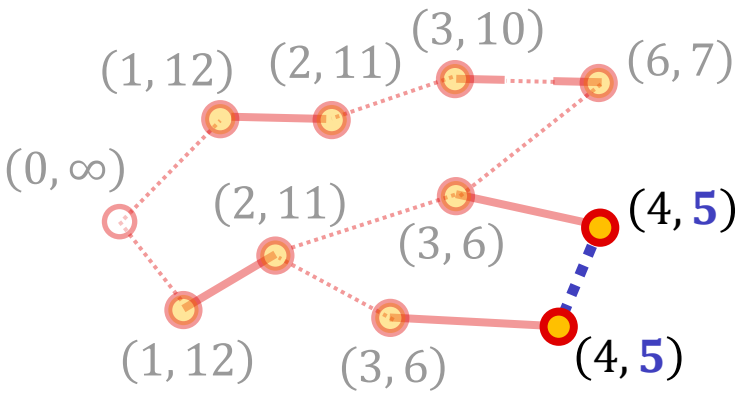


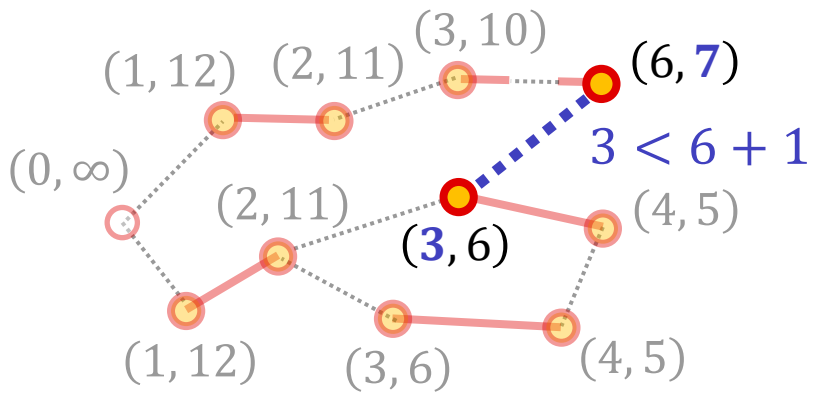


橋 (Bridge) と Tenacity

辺 $e = \{u, w\}$ が**橋 (Bridge)** である $\overset{\text{def}}{\Leftrightarrow}$ e が u, w どちらの minlevel も与えない

- e が非マッチ辺であり，以下をともに満たすことと同値
  - $\text{oddlevel}(u) = \text{maxlevel}(u)$ または $\text{oddlevel}(u) < \text{evenlevel}(w) + 1$
  - $\text{oddlevel}(w) = \text{maxlevel}(w)$ または $\text{oddlevel}(w) < \text{evenlevel}(u) + 1$
- $\text{tenacity}(e) \coloneqq \text{evenlevel}(u) + \text{evenlevel}(w) + 1$ $(e$: 橋$)$
- $\text{tenacity}(v) \coloneqq \text{oddlevel}(v) + \text{evenlevel}(v) = \text{minlevel}(v) + \text{maxlevel}(v)$ $(v \in V)$

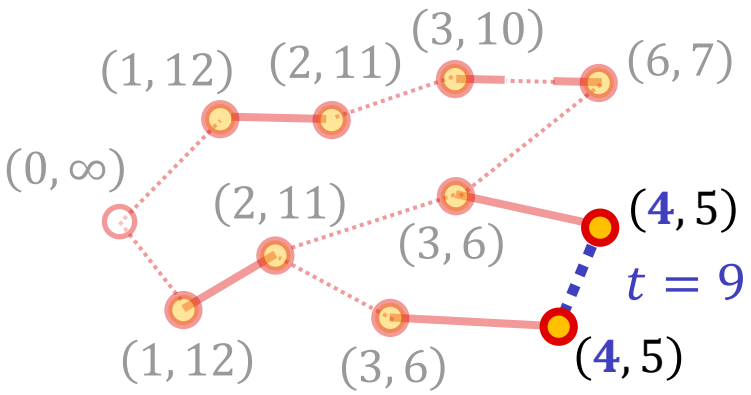


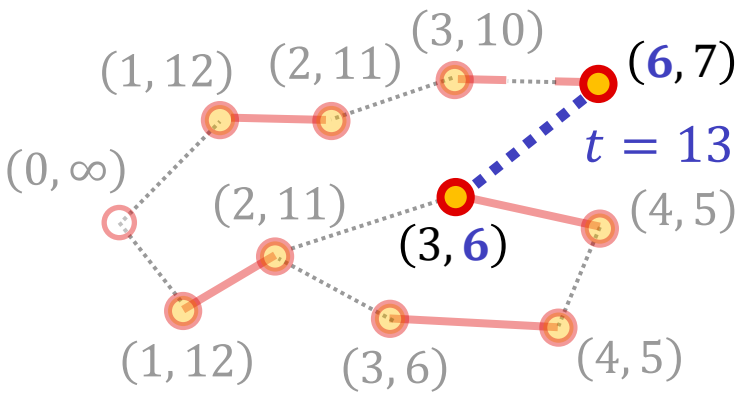


橋 (Bridge) と Tenacity

辺 $e = \{u, w\}$ が**橋 (Bridge)** である $\overset{\text{def}}{\Leftrightarrow}$ e が u, w どちらの minlevel も与えない

- e が非マッチ辺であり，以下をともに満たすことと同値
  - $\mathrm{oddlevel}(u) = \mathrm{maxlevel}(u)$ または $\mathrm{oddlevel}(u) < \mathrm{evenlevel}(w) + 1$
  - $\mathrm{oddlevel}(w) = \mathrm{maxlevel}(w)$ または $\mathrm{oddlevel}(w) < \mathrm{evenlevel}(u) + 1$
- $\mathrm{tenacity}(e) \coloneqq \mathrm{evenlevel}(u) + \mathrm{evenlevel}(w) + 1$ （e: 橋）
- $\mathrm{tenacity}(v) \coloneqq \mathrm{oddlevel}(v) + \mathrm{evenlevel}(v) = \mathrm{minlevel}(v) + \mathrm{maxlevel}(v)$ （$v \in V$）

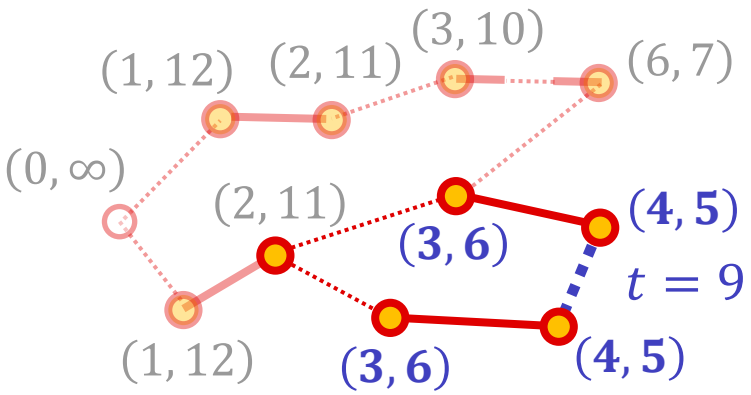


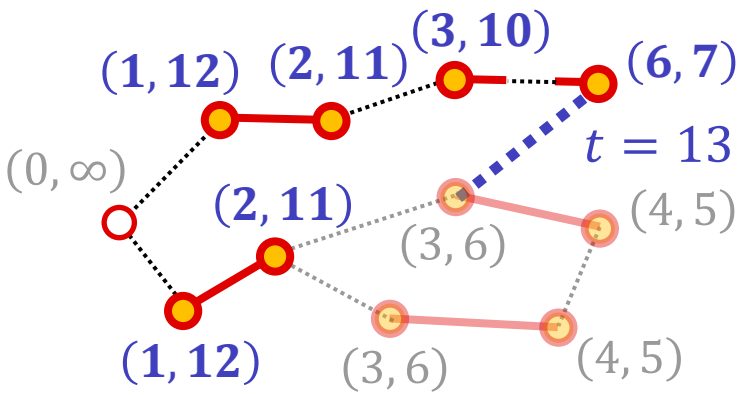


橋 (Bridge) と Tenacity

辺 $e = \{u, w\}$ が**橋 (Bridge)** である $\overset{\text{def}}{\Leftrightarrow}$ e が u, w どちらの minlevel も与えない

- e が非マッチ辺であり，以下をともに満たすことと同値
  - $\text{oddlevel}(u) = \text{maxlevel}(u)$ または $\text{oddlevel}(u) < \text{evenlevel}(w) + 1$
  - $\text{oddlevel}(w) = \text{maxlevel}(w)$ または $\text{oddlevel}(w) < \text{evenlevel}(u) + 1$
- $\text{tenacity}(e) \coloneqq \text{evenlevel}(u) + \text{evenlevel}(w) + 1$ （e: 橋）
- $\text{tenacity}(v) \coloneqq \text{oddlevel}(v) + \text{evenlevel}(v) = \text{minlevel}(v) + \text{maxlevel}(v)$ （$v \in V$）





橋 (Bridge) と Tenacity

辺 $e = \{u, w\}$ が**橋 (Bridge)** である $\overset{\text{def}}{\Leftrightarrow}$ e が u, w どちらの minlevel も与えない

- e が非マッチ辺であり，以下をともに満たすことと同値
  - $\mathrm{oddlevel}(u) = \mathrm{maxlevel}(u)$ または $\mathrm{oddlevel}(u) < \mathrm{evenlevel}(w) + 1$
  - $\mathrm{oddlevel}(w) = \mathrm{maxlevel}(w)$ または $\mathrm{oddlevel}(w) < \mathrm{evenlevel}(u) + 1$
- $\mathrm{tenacity}(e) \coloneqq \mathrm{evenlevel}(u) + \mathrm{evenlevel}(w) + 1$ （e: 橋）
- $\mathrm{tenacity}(v) \coloneqq \mathrm{oddlevel}(v) + \mathrm{evenlevel}(v) = \mathrm{minlevel}(v) + \mathrm{maxlevel}(v)$ （$v \in V$）

補題 $\mathrm{tenacity}(v) = t$ の頂点 v への maxlevel パスは，
$\mathrm{tenacity}(e) = t$ の橋 e を通り，そのような橋は唯一である．

$\mathrm{tenacity}(v) = t$ の頂点 v の maxlevel は $\mathrm{tenacity}(e) = t$ の橋 e が与える！

演習問題 5. 最短偶奇交互道の構造定理

tenacity としてあり得る最小値を $t_{\min}$ とする.

$t_v = t_{\min}$ のとき，以下の補題の主張が成り立つことを示せ.

ヒント: (1) 先頭部分路の共通部分が最長になるような P_v を取る
(2), (3) 先に P_v を固定してから Q_v を任意に取って背理法

<u>補題</u> $t_v \coloneqq \mathrm{tenacity}(v) < +\infty$, P_v, Q_v: 頂点 v への min/maxlevel パス

(1) Q_v は $\mathrm{tenacity}(e) = t_v$ の橋 e を通り，そのような橋は唯一

(2) P_v, Q_v 上で $\mathrm{tenacity}(b) > t_v$ となる最後の頂点 b はパスに依らず共通

(3) P_v, Q_v は，s から b への minlevel = evenlevel パスと，b から v への非マッチ辺で始まる偶奇交互道のうち最短のものに分解できる

一般グラフに対する $O(\sqrt{\mu}m)$ 時間アルゴリズム

[Micali–Vazirani 1980][Vazirani 2024]

$O(m)$ 時間で以下をやって，極大な点素最短増加道集合を見つける．

Procedure MIN: 各頂点の minlevel を minlevel $i = 1, 2, \ldots, \frac{\tilde{\ell}-1}{2}$ の順に求める．

Procedure MAX: 各頂点の maxlevel を tenacity $t = 3, 5, \ldots, \tilde{\ell}$ の順に求める．

2 つの手続きを i と $\frac{t}{2}$ の昇順になるように交互に行う．

一般グラフに対する $O(\sqrt{\mu}m)$ 時間アルゴリズム

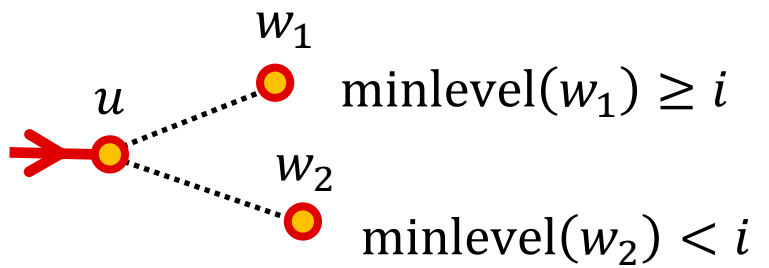
[Micali–Vazirani 1980][Vazirani 2024]

$O(m)$ 時間で以下をやって，極大な点素最短増加道集合を見つける．

Procedure MIN: 各頂点の minlevel を minlevel $i = 1, 2, \ldots, \frac{\tilde{\ell}-1}{2}$ の順に求める．

$\mathrm{odd/evenlevel}(u) = i - 1$ であるような頂点 u と拡張可能な辺 $\{u, w\}$ に対して，
$\mathrm{minlevel}(w) \geq i$ であれば，$\mathrm{minlevel}(w) \leftarrow i$ として u を w の**親 (Predecessor)**とする．

$\mathrm{oddlevel}(u) = i - 1$



$\mathrm{evenlevel}(u) = i - 1$

一般グラフに対する $O(\sqrt{\mu}m)$ 時間アルゴリズム

[Micali–Vazirani 1980][Vazirani 2024]

$O(m)$ 時間で以下をやって，極大な点素最短増加道集合を見つける．

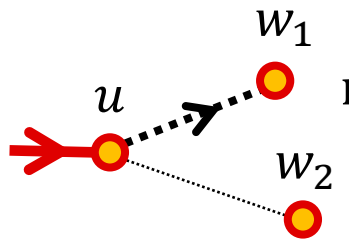
Procedure MIN: 各頂点の minlevel を minlevel $i = 1, 2, \ldots, \frac{\tilde{\ell}-1}{2}$ の順に求める．

$\mathrm{odd/evenlevel}(u) = i - 1$ であるような頂点 u と拡張可能な辺 $\{u, w\}$ に対して，

$\mathrm{minlevel}(w) \geq i$ であれば，$\mathrm{minlevel}(w) \leftarrow i$ として u を w の**親 (Predecessor)**とする．

$\mathrm{minlevel}(w) = i$

$\mathrm{oddlevel}(u) = i - 1$



$\mathrm{minlevel}(w_1) = i$

$\mathrm{minlevel}(w_2) < i$

$\mathrm{evenlevel}(u) = i - 1$

一般グラフに対する $O(\sqrt{\mu}m)$ 時間アルゴリズム

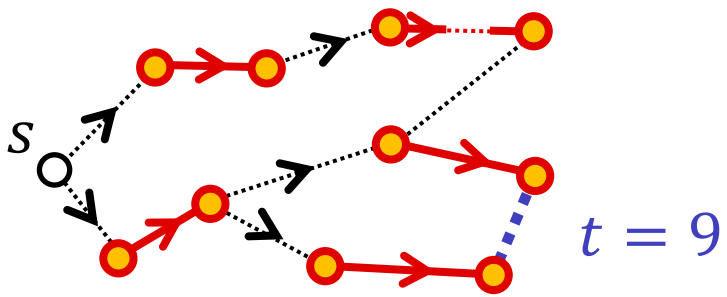
[Micali–Vazirani 1980][Vazirani 2024]

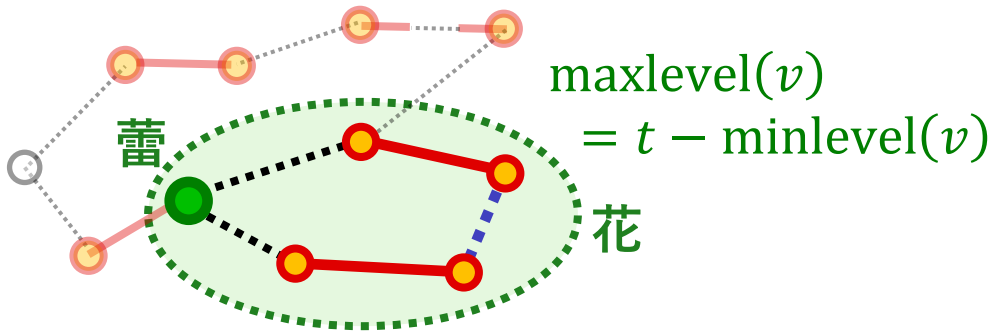
$O(m)$ 時間で以下をやって，極大な点素最短増加道集合を見つける．

Procedure MIN: 各頂点の minlevel を minlevel $i = 1, 2, \ldots, \frac{\tilde{\ell}-1}{2}$ の順に求める．

Procedure MAX: 各頂点の maxlevel を tenacity $t = 3, 5, \ldots, \tilde{\ell}$ の順に求める．

$\mathrm{tenacity}(e) = t$ であるような各橋 $e = \{u, w\}$ から親を辿って DFS 的に遡り，
花 (Blossom) とその蕾 (Bud)（$t = \tilde{\ell}$ のときは**最短増加道**の可能性有り）を見つける．
増加道が見つかったら，使えなくなった頂点を再帰的にグラフから取り除く．





一般グラフに対する $O(\sqrt{\mu}m)$ 時間アルゴリズム

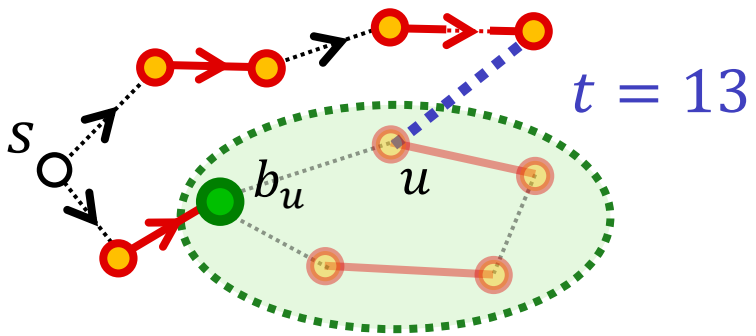
[Micali–Vazirani 1980][Vazirani 2024]

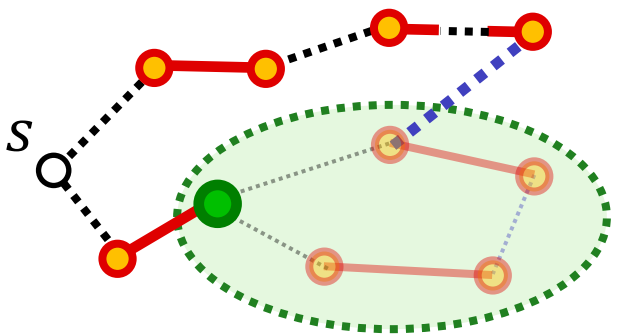
$O(m)$ 時間で以下をやって，極大な点素最短増加道集合を見つける．

Procedure MIN: 各頂点の minlevel を minlevel $i = 1, 2, \ldots, \frac{\tilde{\ell}-1}{2}$ の順に求める．

Procedure MAX: 各頂点の maxlevel を tenacity $t = 3, 5, \ldots, \tilde{\ell}$ の順に求める．

$\mathrm{tenacity}(e) = t$ であるような各橋 $e = \{u, w\}$ から親を辿って DFS 的に遡り，
花 (Blossom) とその**蕾 (Bud)**（$t = \tilde{\ell}$ のときは**最短増加道**の可能性有り）を見つける．
増加道が見つかったら，使えなくなった頂点を再帰的にグラフから取り除く．





一般グラフに対する $O(\sqrt{\mu}m)$ 時間アルゴリズム

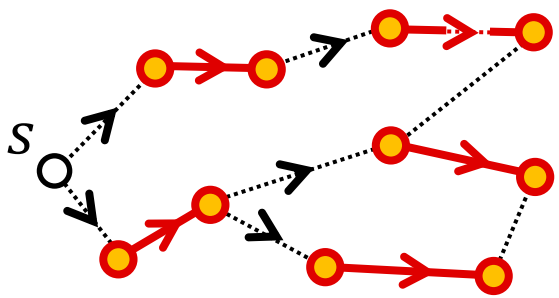
[Micali–Vazirani 1980][Vazirani 2024]

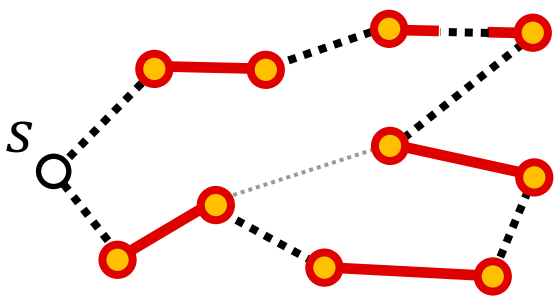
$O(m)$ 時間で以下をやって，極大な点素最短増加道集合を見つける．

Procedure MIN: 各頂点の minlevel を minlevel $i = 1, 2, \ldots, \frac{\tilde{\ell}-1}{2}$ の順に求める．

Procedure MAX: 各頂点の maxlevel を tenacity $t = 3, 5, \ldots, \tilde{\ell}$ の順に求める．

$\mathrm{tenacity}(e) = t$ であるような各橋 $e = \{u, w\}$ から親を辿って DFS 的に遡り，
花 (Blossom) とその**蕾 (Bud)**（$t = \tilde{\ell}$ のときは**最短増加道**の可能性有り）を見つける．
増加道が見つかったら，使えなくなった頂点を再帰的にグラフから取り除く．





一般グラフに対する $O(\sqrt{\mu}m)$ 時間アルゴリズム

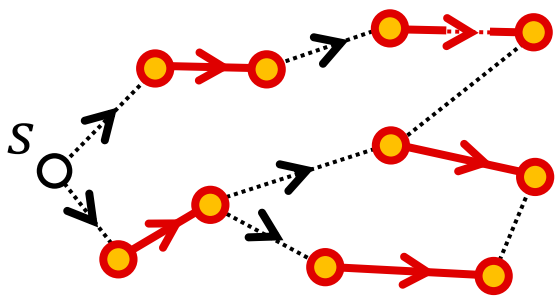
[Micali–Vazirani 1980][Vazirani 2024]

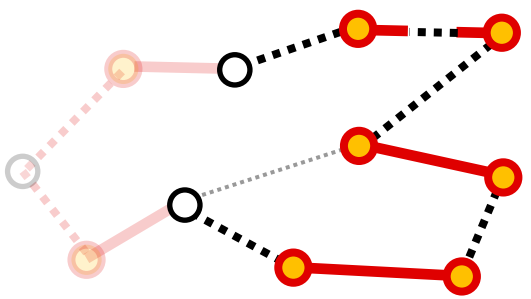
$O(m)$ 時間で以下をやって，極大な点素最短増加道集合を見つける．

Procedure MIN: 各頂点の minlevel を minlevel $i = 1, 2, \ldots, \frac{\tilde{\ell}-1}{2}$ の順に求める．

Procedure MAX: 各頂点の maxlevel を tenacity $t = 3, 5, \ldots, \tilde{\ell}$ の順に求める．

$\mathrm{tenacity}(e) = t$ であるような各橋 $e = \{u, w\}$ から親を辿って DFS 的に遡り，
花 (Blossom) とその**蕾 (Bud)**（$t = \tilde{\ell}$ のときは**最短増加道**の可能性有り）を見つける．
増加道が見つかったら，使えなくなった頂点を再帰的にグラフから取り除く．





一般グラフに対する $O(\sqrt{\mu}m)$ 時間アルゴリズム

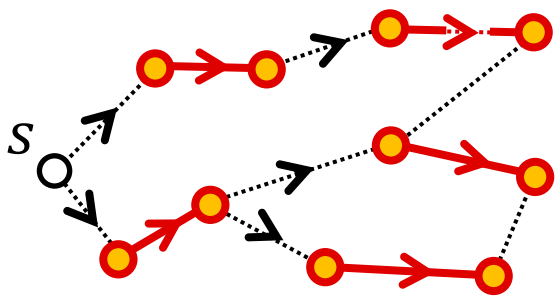
[Micali–Vazirani 1980][Vazirani 2024]

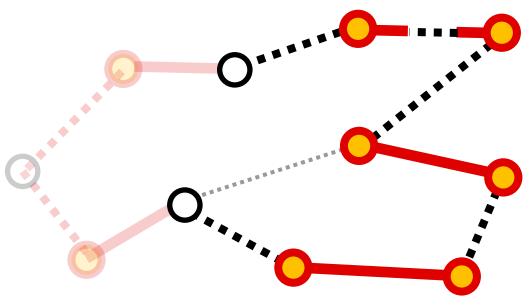
$O(m)$ 時間で以下をやって，極大な点素最短増加道集合を見つける．

Procedure MIN: 各頂点の minlevel を minlevel $i = 1, 2, \ldots, \frac{\tilde{\ell}-1}{2}$ の順に求める．

Procedure MAX: 各頂点の maxlevel を tenacity $t = 3, 5, \ldots, \tilde{\ell}$ の順に求める．

$\mathrm{tenacity}(e) = t$ であるような各橋 $e = \{u, w\}$ から親を辿って DFS 的に遡り，
花 (Blossom) とその**蕾 (Bud)**（$t = \tilde{\ell}$ のときは**最短増加道**の可能性有り）を見つける．
増加道が見つかったら，使えなくなった頂点を再帰的にグラフから取り除く．





一般グラフに対する $O(\sqrt{\mu}m)$ 時間アルゴリズム

[Micali–Vazirani 1980][Vazirani 2024]

$O(m)$ 時間で以下をやって，極大な点素最短増加道集合を見つける．

Procedure MIN: 各頂点の minlevel を minlevel $i = 1, 2, \ldots, \frac{\tilde{\ell}-1}{2}$ の順に求める．

Procedure MAX: 各頂点の maxlevel を tenacity $t = 3, 5, \ldots, \tilde{\ell}$ の順に求める．

2 つの手続きを i と $\frac{t}{2}$ の昇順になるように交互に行う．

[補足]

- min/maxlevel を正しく計算するためには 2 つの手続きの同期が本質的に重要
- 花と蕾の管理（仮想的な縮約）には Union-Find (Disjoint Set Forest) が必要
- 最も非自明なのは最短増加道を再帰的に復元する部分（花の入れ子構造が複雑）

  → **構造定理** $(+\alpha)$ が，再帰処理の正当性と最終的な増加道集合の極大性を保証

一般グラフに対する $O(\sqrt{\mu}m)$ 時間アルゴリズム

[Micali–Vazirani 1980][Vazirani 2024]

補題 $t_v := \mathrm{tenacity}(v) < +\infty$, P_v, Q_v: 頂点 v への min/maxlevel パス

(1) Q_v は $\mathrm{tenacity}(e) = t_v$ の橋 e を通り，そのような橋は唯一

(2) P_v, Q_v 上で $\mathrm{tenacity}(b) > t_v$ となる最後の頂点 b はパスに依らず共通

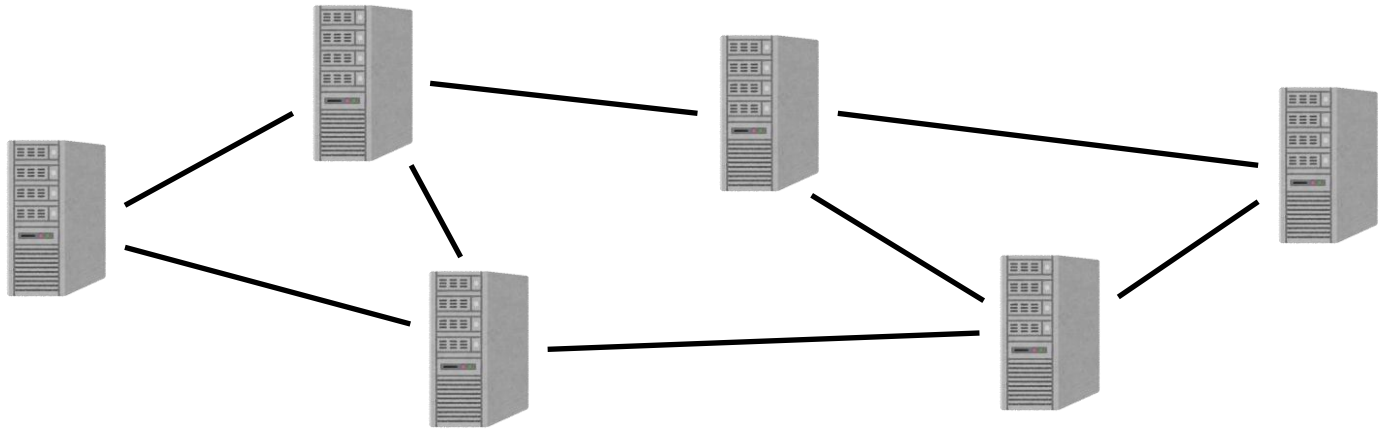
(3) P_v, Q_v は，s から b への minlevel = evenlevel パスと，b から v への非マッチ辺で始まる偶奇交互道のうち最短のものに分解できる

- min/maxlevel を正しく計算するためには 2 つの手続きの同期が本質的に重要
- 花と蕾の管理（仮想的な縮約）には Union-Find (Disjoint Set Forest) が必要
- 最も非自明なのは最短増加道を再帰的に復元する部分（花の入れ子構造が複雑）
  → **構造定理** $(+\alpha)$ が，再帰処理の正当性と最終的な増加道集合の極大性を保証

目次

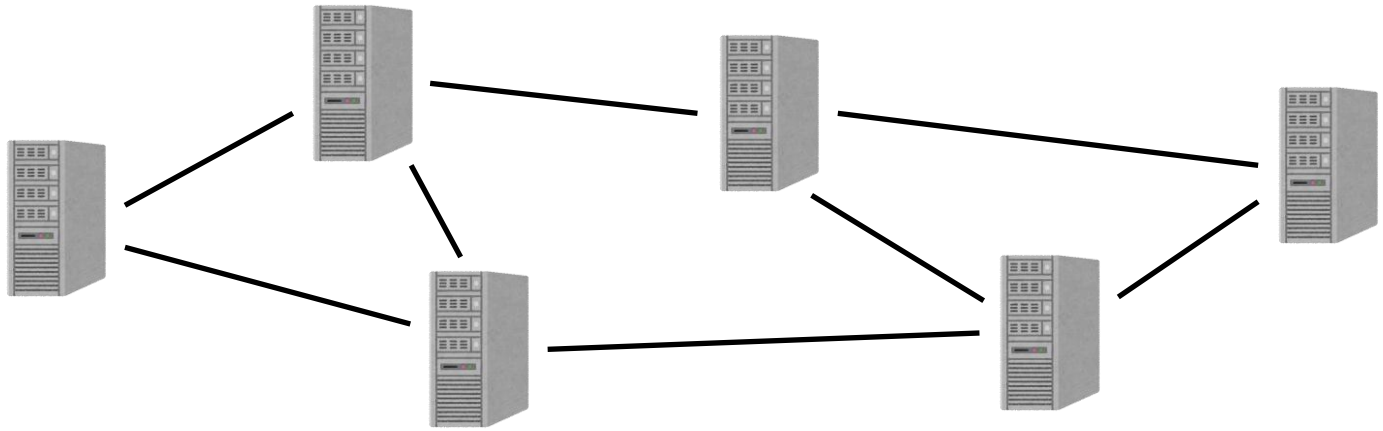
分散計算モデル

- コンピュータ (計算端末) がネットワーク構造 (グラフ) をなす
- 各端末は十分な計算能力を持っている
- 各端末は周囲の局所的な情報 (自身と近傍) のみを知っている
- 各端末は各接続辺を通じてメッセージを送り合うことができる



分散計算モデル CONGEST

- コンピュータ（計算端末）がネットワーク構造（グラフ）をなす
- 各端末は十分な計算能力を持っている
- 各端末は周囲の局所的な情報（自身と近傍）のみを知っている
- 各端末は各接続辺を通じて**メッセージ**を送り合うことができる
  （ネットワーク全体で同期されたタイミングで，各 $O(\log n)$ ビット）



問題意識と自明な上界

- コンピュータ (計算端末) がネットワーク構造 (グラフ) をなす
- 各端末は十分な計算能力を持っている
- 各端末は周囲の局所的な情報 (自身と近傍) のみを知っている
- 各端末は各接続辺を通じて**メッセージ**を送り合うことができる
  (ネットワーク全体で同期されたタイミングで，各 $O(\log n)$ ビット)

Q. グラフ上の問題を解くために**何ラウンドの通信**が必要か？

A. リーダーを決めてグラフの全情報をそこに集めて計算すれば，
大抵の問題は $\Theta(m) = O(n^2)$ ラウンドで解けてしまう．

→ **どの程度改善できるか？**

問題意識と自明な上界

- コンピュータ (計算端末) がネットワーク構造 (グラフ) をなす
- 各端末は十分な計算能力を持っている
- 各端末は周囲の局所的な情報 (自身と近傍) のみを知っている
- 各端末は各接続辺を通じて**メッセージ**を送り合うことができる
  (ネットワーク全体で同期されたタイミングで，各 $O(\log n)$ ビット)

Q. グラフ上の問題を解くために**何ラウンドの通信**が必要か？

A. リーダーを決めてグラフの全情報をそこに集めて計算すれば,
大抵の問題は $\Theta(m) = O(n^2)$ ラウンドで解けてしまう.

→ どの程度改善できるか？

最大マッチング問題 in CONGEST モデル（再掲）

入力: $G = (V, E)$: 無向グラフ（連結）
目標: 最大サイズの**マッチング** $M \subseteq E$ の発見

$n = |V|,\ m = |E|$
μ: 最大サイズ

[分散アルゴリズム (CONGEST)]

- 自明な万能アルゴリズムだと $\Theta(m) = O(n^2)$ ラウンド (cf. Izumi 2017)
- 2 部グラフで決定性 $O(\mu \log \mu)$ ラウンド [Ahmadi–Kuhn–Oshman 2018]
- 一般グラフで決定性 $O(\mu^2)$ ラウンド [Ben-Basat–Kawarabayashi–Schwartzman 2019]
- 一般グラフで乱択 $O(\mu^{1.5})$ ラウンド [Kitamura–Izumi 2022]
- 一般グラフで乱択 $O(\mu \log \mu)$ ラウンド（など） [Izumi–Kitamura–Y. 2024]

$O(\mu \log \mu)$ ラウンドへの道 [Izumi–Kitamura–Y. 2024]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $O\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

- $\mathrm{FP}(n, m, \mu)$ を均し $O(\log \mu)$ で抑えれば十分
- **Hopcroft–Karp の解析**から，$\mathrm{FP}(n, m, \mu) = O(\ell)$ が達成できれば十分
- **Ahmadi–Kuhn の最大マッチング検証アルゴリズム**のおかげで，
  状況をかなり限定しても大丈夫

$O(\mu \log \mu)$ ラウンドへの道 [Izumi–Kitamura–Y. 2024]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $O\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

- $\mathrm{FP}(n, m, \mu)$ を均し $O(\log \mu)$ で抑えれば十分
- **Hopcroft–Karp の解析**から，$\mathrm{FP}(n, m, \mu) = O(\ell)$ が達成できれば十分

補題　M がサイズ $\mu - k$ のマッチング $(1 \le k \le \mu)$

$\Longrightarrow$ M に関する増加道であって長さが $\frac{2\mu}{k}$ 未満のものが存在

$O(\mu \log \mu)$ ラウンドへの道 [Izumi–Kitamura–Y. 2024]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $O\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

- $\mathrm{FP}(n, m, \mu)$ を均し $O(\log \mu)$ で抑えれば十分
- **Hopcroft–Karp の解析**から，$\mathrm{FP}(n, m, \mu) = O(\ell)$ が達成できれば十分
- **Ahmadi–Kuhn の最大マッチング検証アルゴリズム**のおかげで，
  状況をかなり限定しても大丈夫

最大マッチング検証アルゴリズム [Ahmadi–Kuhn 2020]

入力: $G = (V, E)$: 無向グラフ，$M \subseteq E$: マッチング

目標: 以下の正しい方を実行する．

- M が最大マッチングであると結論付ける．
- M に関する**最短増加道の端点のペア**を 1 つ見つけ，
  一方の端点からの ℓ 以下の odd/evenlevel を全て求める．

定理 この問題は乱択 CONGEST アルゴリズムによって解ける．
計算ラウンド数は前者の場合 $O(\mu)$，後者の場合 $O(\ell)$ である．

"We hope that our algorithm constitutes a significant step towards developing a CONGEST algorithm to *compute* a maximum matching in time $\tilde{O}(s^*)$, where s^* is the size of a maximum matching."

$O(\mu \cdot \mathrm{polylog}(\mu))$ ラウンドという意味

目次

$\mathrm{O}(\mu \log \mu)$ ラウンドへの道 [Izumi–Kitamura–Y. 2024]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $\mathrm{O}\big(\mu \cdot \mathrm{FP}(n, m, \mu)\big)$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

- $\mathrm{FP}(n, m, \mu)$ を均し $\mathrm{O}(\log \mu)$ で抑えれば十分
- **Hopcroft–Karp の解析**から，$\mathrm{FP}(n, m, \mu) = \mathrm{O}(\ell)$ が達成できれば十分
- **Ahmadi–Kuhn の最大マッチング検証アルゴリズム**のおかげで，
  最短増加道の一方の端点 s からの odd/evenlevel が既知と仮定できる

$O(\mu^{1.5})$ ラウンドアルゴリズム [Kitamura–Izumi 2022]

[素朴なアルゴリズム]

1. $M \leftarrow \emptyset$
2. M に関する増加道 P が見つかる限り，$M \leftarrow M \triangle P$ と更新

全体で $O(\mu \cdot \mathrm{FP}(n, m, \mu))$ 時間（$\mathrm{FP}(\cdot)$: 増加道を 1 つ見つけるための時間）

- $\mathrm{FP}(n, m, \mu)$ を均し $O(\sqrt{\mu})$ で抑えれば十分
- **Hopcroft–Karp の解析**から，$\mathrm{FP}(n, m, \mu) = O(\min\{\ell^2, \mu\})$ で十分
- **Ahmadi–Kuhn の最大マッチング検証アルゴリズム**のおかげで，
  最短増加道の一方の端点 s からの odd/evenlevel が既知と仮定できる

演習問題 6. 均し計算量解析

Hopcroft–Karp の解析を利用して，
素朴な増加道アルゴリズムの計算量に関して以下の主張を示せ．

(1) $\mathrm{FP}(n, m, \mu) = \mathrm{O}(\ell)$ のとき，全体の計算量は $\mathrm{O}(\mu \log \mu)$ である．

(2) $\mathrm{FP}(n, m, \mu) = \mathrm{O}(\min\{\ell^2, \mu\})$ のとき，全体の計算量は $\mathrm{O}(\mu^{1.5})$ である．

補題　M がサイズ $\mu - k$ のマッチング $(1 \le k \le \mu)$

$\Longrightarrow$ M に関する増加道であって長さが $\frac{2\mu}{k}$ 未満のものが存在

[Hopcroft–Karp 1973]

$O(\min\{\ell^2, \mu\})$ ラウンドの増加道発見 [Kitamura–Izumi 2022]

どちらも **Ahmadi–Kuhn の最大マッチング検証アルゴリズム**を活用

- $O(\ell^2)$ ラウンドは簡単
  終点側の端点から，親としてあり得る頂点を決め打って遡るのを繰り返すだけ．
- $O(\mu)$ ラウンドは**疎な部分グラフ**の抽出が肝
  - 辺の数は $O(\mu)$
  - 非マッチ頂点 s からの交互道による到達可能性（偶奇）を全て保持
  - 特に，増加道を保持

  → この部分グラフで「リーダーに集めて解く」自明アルゴリズムをやればよい．

  Edmonds の花縮約アルゴリズムの正当性から，そのような部分グラフが存在！

$O(\min\{\ell^2, \mu\})$ ラウンドの増加道発見 [Kitamura–Izumi 2022]

どちらも **Ahmadi–Kuhn の最大マッチング検証アルゴリズム**を活用

- $O(\ell^2)$ ラウンドは簡単
  終点側の端点から，親としてあり得る頂点を決め打って遡るのを繰り返すだけ.
- $O(\mu)$ ラウンドは**疎な部分グラフ**の抽出が肝
  - 辺の数は $O(\mu)$
  - 非マッチ頂点 s からの交互道による到達可能性 (偶奇) を全て保持
  - 特に，増加道を保持

  → この部分グラフで「リーダーに集めて解く」自明アルゴリズムをやればよい.

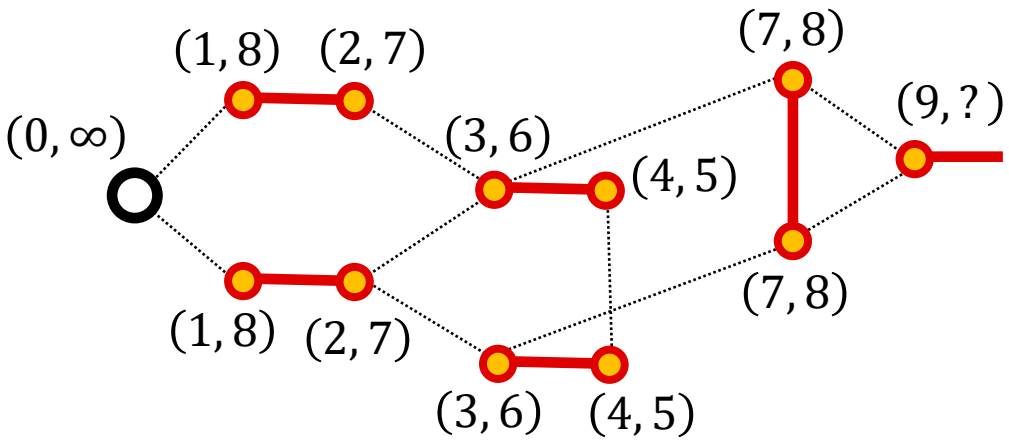
Edmonds の花縮約アルゴリズムの正当性から，そのような部分グラフが存在！

交互ベース木 (ABT) と最小外向辺 (MOE)

[Kitamura–Izumi 2022]

- 辺の数は $O(\mu)$
- 非マッチ頂点 s からの交互道による到達可能性 (偶奇) を全て保持
- 特に，増加道を保持

Edmonds の花縮約アルゴリズムの正当性から，そのような部分グラフが存在！



Ahmadi–Kuhn の出力

各頂点は 2 本の辺を以下のように選ぶ

- minlevel を与える**親辺** (任意に 1 つ固定)
- **最小外向辺** (Minimum Outgoing Edge; **MOE**):
  - **部分木の内外を跨ぐ**非木辺
  - 整合する odd/evenlevel の**最大値を最小化**

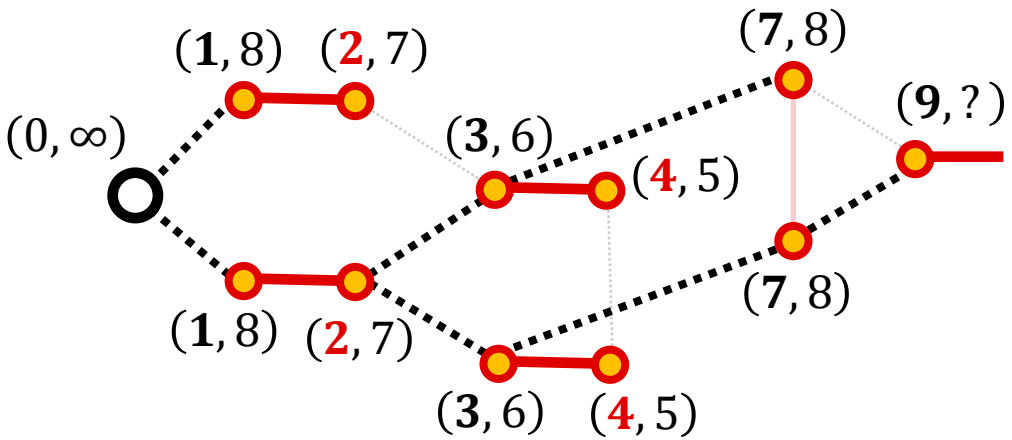
この疎な部分グラフは所望の性質を満たす！

交互ベース木 (ABT) と最小外向辺 (MOE)

[Kitamura–Izumi 2022]

- 辺の数は $O(\mu)$
- 非マッチ頂点 s からの交互道による到達可能性 (偶奇) を全て保持
- 特に，増加道を保持

Edmonds の花縮約アルゴリズムの正当性から，そのような部分グラフが存在！



交互ベース木 (Alternating Base Tree)

各頂点は 2 本の辺を以下のように選ぶ

- minlevel を与える**親辺** (任意に 1 つ固定)
- **最小外向辺** (Minimum Outgoing Edge; **MOE**):
  - **部分木の内外を跨ぐ**非木辺
  - 整合する odd/evenlevel の**最大値を最小化**

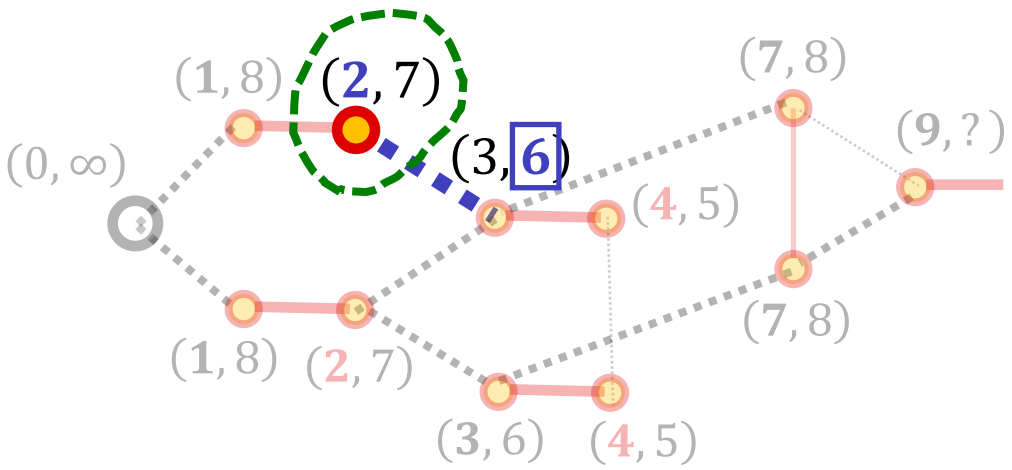
この疎な部分グラフは所望の性質を満たす！

交互ベース木 (ABT) と最小外向辺 (MOE)

[Kitamura–Izumi 2022]

- 辺の数は $O(\mu)$
- 非マッチ頂点 s からの交互道による到達可能性 (偶奇) を全て保持
- 特に，増加道を保持

Edmonds の花縮約アルゴリズムの正当性から，そのような部分グラフが存在！



ABT + **MOE**

各頂点は 2 本の辺を以下のように選ぶ

- minlevel を与える**親辺** (任意に 1 つ固定)
- **最小外向辺** (Minimum Outgoing Edge; **MOE**):
  - **部分木の内外を跨ぐ**非木辺
  - 整合する odd/evenlevel の**最大値を最小化**

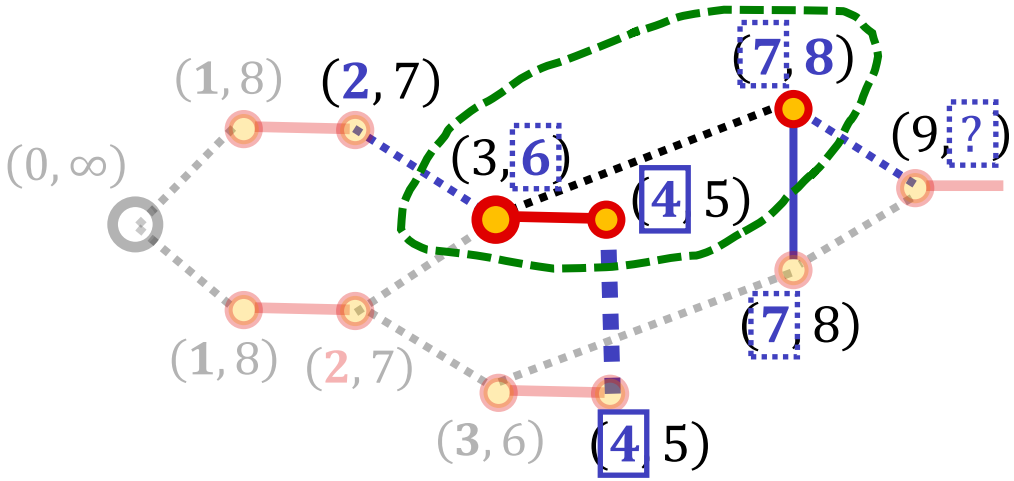
この疎な部分グラフは所望の性質を満たす！

交互ベース木 (ABT) と最小外向辺 (MOE)

[Kitamura–Izumi 2022]

- 辺の数は $O(\mu)$
- 非マッチ頂点 s からの交互道による到達可能性 (偶奇) を全て保持
- 特に，増加道を保持

Edmonds の花縮約アルゴリズムの正当性から，そのような部分グラフが存在！



ABT + **MOE**

各頂点は 2 本の辺を以下のように選ぶ

- minlevel を与える**親辺** (任意に 1 つ固定)
- **最小外向辺** (Minimum Outgoing Edge; **MOE**):
  - **部分木の内外を跨ぐ**非木辺
  - 整合する odd/evenlevel の**最大値を最小化**

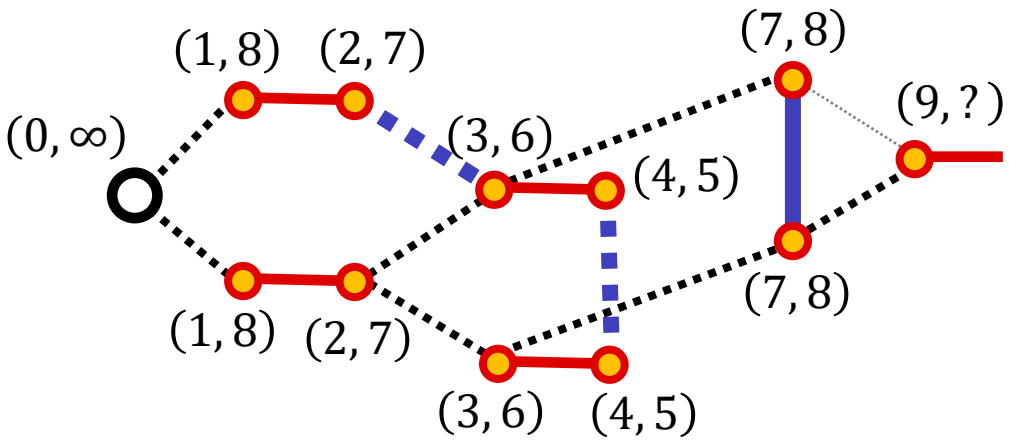
この疎な部分グラフは所望の性質を満たす！

交互ベース木 (ABT) と最小外向辺 (MOE)

[Kitamura–Izumi 2022]

- 辺の数は $O(\mu)$
- 非マッチ頂点 s からの交互道による到達可能性 (偶奇) を全て保持
- 特に，増加道を保持

Edmonds の花縮約アルゴリズムの正当性から，そのような部分グラフが存在！



ABT + **MOE**

各頂点は 2 本の辺を以下のように選ぶ

- minlevel を与える**親辺** (任意に 1 つ固定)
- **最小外向辺** (Minimum Outgoing Edge; **MOE**):
  - **部分木の内外を跨ぐ**非木辺
  - 整合する odd/evenlevel の**最大値を最小化**

この疎な部分グラフは所望の性質を満たす！

目次

$O(\min\{\ell^2, \mu\})$ ラウンドの増加道発見 [Kitamura–Izumi 2022]

どちらも **Ahmadi–Kuhn の最大マッチング検証アルゴリズム**を活用

- $O(\ell^2)$ ラウンドは簡単
  終点側の端点から，親としてあり得る頂点を決め打って遡るのを繰り返すだけ.
- $O(\mu)$ ラウンドは**疎な部分グラフ**の抽出が肝
  - 辺の数は $O(\mu)$
  - 非マッチ頂点 s からの交互道による到達可能性 (偶奇) を全て保持
  - 特に，増加道を保持

  → この部分グラフで「リーダーに集めて解く」自明アルゴリズムをやればよい.

Edmonds の花縮約アルゴリズムの正当性から，そのような部分グラフが存在！

$O(\ell)$ ラウンドにするには？ [Izumi–Kitamura–Y. 2024]

どちらも **Ahmadi–Kuhn の最大マッチング検証アルゴリズム**を活用

- $O(\ell^2)$ ラウンドは簡単
  終点側の端点から，親としてあり得る頂点を決め打って遡るのを繰り返すだけ．

- $O(\mu)$ ラウンドは**疎な部分グラフ**の抽出が肝
  - 辺の数は $O(\mu)$
  - 非マッチ頂点 s からの交互道による**到達可能性（偶奇）**を全て保持
  - 特に，**増加道（いくらでも長くなり得る）**を保持

  → この部分グラフで「リーダーに集めて解く」自明アルゴリズムをやればよい．

Edmonds の花縮約アルゴリズムの正当性から，そのような部分グラフが存在！

$O(\ell)$ ラウンドの増加道発見 [Izumi–Kitamura–Y. 2024]

どちらも **Ahmadi–Kuhn の最大マッチング検証アルゴリズム**を活用

- $O(\ell^2)$ ラウンドは簡単
  終点側の端点から，親としてあり得る頂点を決め打って遡るのを繰り返すだけ．

- $O(\ell)$ ラウンドは**最短増加道を持つ疎な部分グラフ**の抽出が肝
  - $\mathrm{tenacity}(v) = \mathrm{oddlevel}(v) + \mathrm{evenlevel}(v) \leq \ell$ を満たす頂点 v のみからなる
  - 必要な頂点間の**最短偶奇交互道**を 1 本以上保持
  - 特に，**最短増加道**を保持

→ この部分グラフ上で再帰的に最短偶奇交互道を復元すればよい．

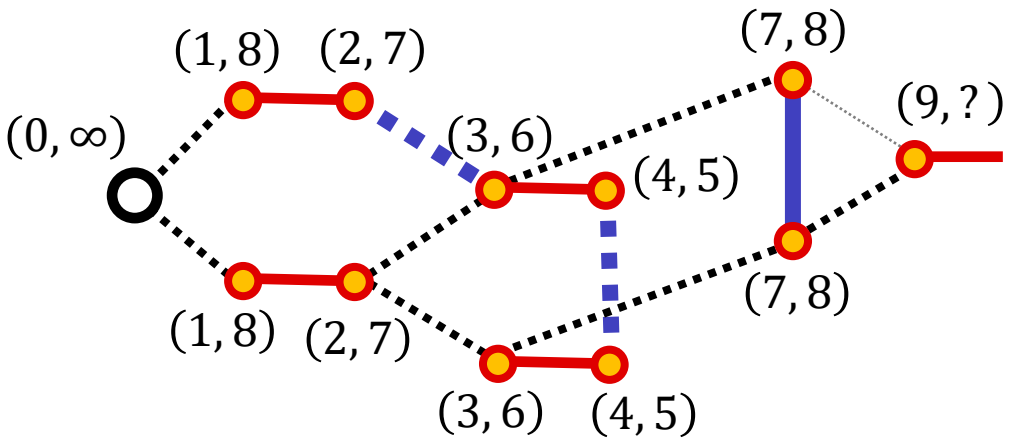
Micali–Vazirani のアルゴリズムのアイデアを基にそのような部分グラフを構成！

交互ベース木 (ABT) と最小外向辺 (MOE)

[Kitamura–Izumi 2022]

- 辺の数は $O(\mu)$
- 非マッチ頂点 s からの交互道による到達可能性 (偶奇) を全て保持
- 特に，増加道を保持 **いくらでも長くなり得る！**

Edmonds の花縮約アルゴリズムの正当性から，そのような部分グラフが存在！



ABT + MOE

各頂点は 2 本の辺を以下のように選ぶ

- minlevel を与える**親辺** (任意に 1 つ固定)
- **最小外向辺** (Minimum Outgoing Edge; **MOE**):
  - **部分木の内外を跨ぐ**非木辺
  - 整合する odd/evenlevel の**最大値を最小化**

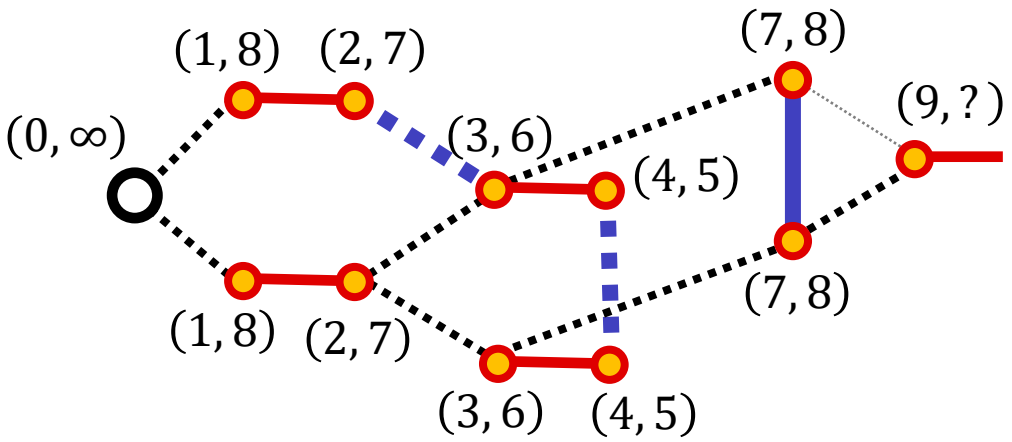
この疎な部分グラフは所望の性質を満たす！

交互ベース木 (ABT) と最小外向辺 (MOE)・改

[Izumi–Kitamura–Y. 2024]

- $\mathrm{tenacity}(v) = \mathrm{oddlevel}(v) + \mathrm{evenlevel}(v) \leq \ell$ を満たす頂点 v のみからなる
- 必要な頂点間の**最短偶奇交互道**を 1 本以上保持
- 特に，**最短増加道**を保持

Micali–Vazirani のアルゴリズムのアイデアを基にそのような部分グラフを構成！



ABT + **MOE**

各頂点は 2 本の辺を以下のように選ぶ

- minlevel を与える**親辺** (任意に 1 つ固定)
- **最小外向辺** (Minimum Outgoing Edge; **MOE**):
  - **部分木の内外を跨ぐ**非木辺
  - 整合レベルの **(和, 最大値) を辞書順最小化**

この疎な部分グラフは所望の性質を満たす！

演習問題 7. 最小外向辺とその改良の必要性

増加道が存在するとし，マッチ辺を細分する単純化 (のみ) を行った状況を考える.

最短増加道の一方の端点 s からの odd/even/min/maxlevel が既知であるとき,
各頂点 $v \in V \setminus \{s\}$ について $\mathrm{minlevel}(v)$ を与える親辺を任意に 1 つ選ぶことで
構成される根付き木を**交互ベース木 (ABT)** と呼ぶ.

各頂点について，ABT における部分木の内外を跨ぐ非木辺を**外向辺**と呼び,
各頂点について外向辺を 1 本選んで ABT に追加して得られるグラフを H とする.

(1) 一般に，H が増加道を含むとは限らないことを示せ.

(2) 外向辺のうち，両端の evenlevel の最大値が最小の辺を選ぶとする.
　　このとき，H が**最短**増加道を含むとは限らないことを示せ.

$O(\ell)$ ラウンドの増加道発見 [Izumi–Kitamura–Y. 2024]

再帰的に u から w への最短偶奇 (α) 交互道を見つけるアルゴリズム

α = even ⇔ マッチ辺で終わる

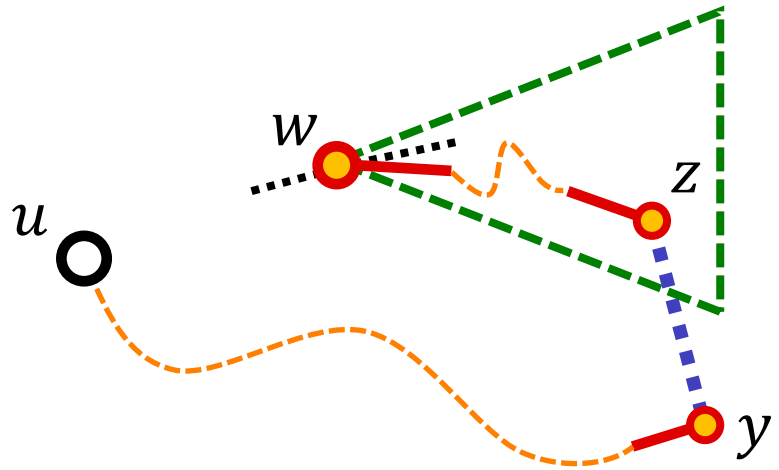
$\underline{\mathrm{FindPath}(u, w, \alpha)}$

0. $u = w$ のとき空パスを出力して停止
1. α が w の minlevel 側の偶奇であれば，w の親辺を $\{v, w\}$ として $\mathrm{FindPath}(u, v, \bar{\alpha}) \circ (v, w)$ を出力して停止
2. α が w の maxlevel 側の偶奇であれば，w の MOE を $\{y, z\}$ として $\mathrm{FindPath}(u, y, \mathrm{even}) \circ (y, z) \circ \overline{\mathrm{FindPath}(w, z, \mathrm{even})}$ を出力して停止
（単純化を仮定すると，マッチ辺は全て ABT に含まれる）

$O(\ell)$ ラウンドの増加道発見 [Izumi–Kitamura–Y. 2024]

再帰的に u から w への最短偶奇 (α) 交互道を見つけるアルゴリズム

$\alpha = \text{even} \Leftrightarrow$ マッチ辺で終わる



2. α が w の maxlevel 側の偶奇であれば，w の MOE を $\{y, z\}$ として $\text{FindPath}(u, y, \text{even}) \circ (y, z) \circ \overline{\text{FindPath}(w, z, \text{even})}$ を出力して停止
（単純化を仮定すると，マッチ辺は全て ABT に含まれる）

$O(\ell)$ ラウンドの増加道発見 [Izumi–Kitamura–Y. 2024]

$\underline{\mathrm{FindPath}(u, w, \alpha)}$

0. $u = w$ のとき空パスを出力して停止
1. α が w の minlevel 側の偶奇であれば，w の親辺を $\{v, w\}$ として
   $\mathrm{FindPath}(u, v, \bar{\alpha}) \circ (v, w)$ を出力して停止
2. α が w の maxlevel 側の偶奇であれば，w の MOE を $\{y, z\}$ として
   $\mathrm{FindPath}(u, y, \mathrm{even}) \circ (y, z) \circ \overline{\mathrm{FindPath}(w, z, \mathrm{even})}$ を出力して停止
   （単純化を仮定すると，マッチ辺は全て ABT に含まれる）

定理 $\mathrm{FindPath}(u, w, \alpha)$ は，出力サイズに関する線形ラウンドで
u から w に偶奇 α で到達する最短交互道を計算する．

$O(\ell)$ ラウンドの増加道発見 [Izumi–Kitamura–Y. 2024]

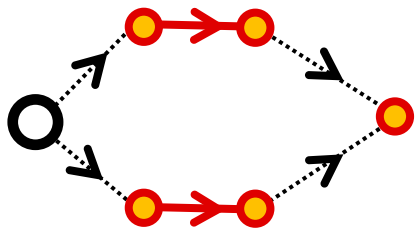
定理 $\mathrm{FindPath}(u, w, \alpha)$ は，出力サイズに関する線形ラウンドで u から w に偶奇 α で到達する最短交互道を計算する．

- Ahmadi–Kuhn で見つけた最短増加道の両端点を s, t として，$\mathrm{FindPath}(s, t, \mathrm{odd})$ は最短増加道を $O(\ell)$ ラウンドで計算する
- MOE で分割して再帰した場合の計算ラウンド数が出力サイズの線形で抑えられるのは少し非自明
- Micali–Vazirani で 1 本見つける手続きと似ているが少し違う（MOE で分割して再帰するか，橋で分割して再帰するか）

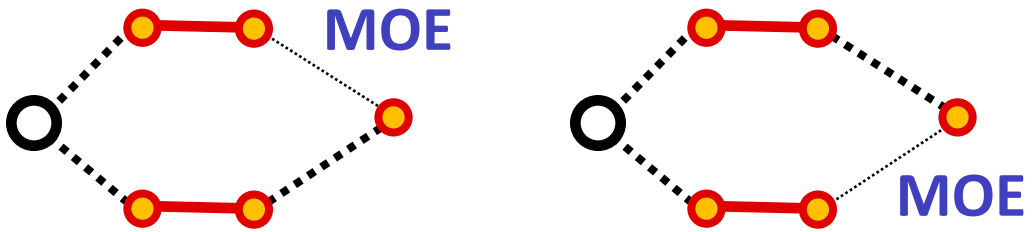
アルゴリズムに関する補遺 [Izumi–Kitamura–Y. 2025+]

Micali–Vazirani で 1 本見つける手続きと似ているが少し違う
(MOE で分割して再帰するか，橋で分割して再帰するか)

- 必要な橋は全て MOE になるが，MOE は MV における親辺も含む
  - 再帰回数は MV より多い
  - 記憶しておく情報と再帰処理で考えることは MV より少ない



MV の親辺



ABT 候補

アルゴリズムに関する補遺 [Izumi–Kitamura–Y. 2025+]

Micali–Vazirani で 1 本見つける手続きと似ているが少し違う
(MOE で分割して再帰するか，橋で分割して再帰するか)

- 必要な橋は全て MOE になるが，MOE は MV における親辺も含む
  - 再帰回数は MV より多い
  - 記憶しておく情報と再帰処理で考えることは MV より少ない
- MV は min/maxlevel と最短増加道を同じ枠組みで計算しているが，IKY は min/maxlevel → ABT/MOE → 最短増加道と段階を分けている
  - この方針で $O(\sqrt{\mu}m)$ 時間アルゴリズム (と証明) を簡素化できるのでは？

目次

参考文献 1

M. Ahmadi and F. Kuhn: **Distributed maximum matching verification in CONGEST**. *DISC 2020*.

M. Ahmadi, F. Kuhn, and R. Oshman: **Distributed approximate maximum matching in the CONGEST model**. *DISC 2018*.

R. Ben-Basat, K. Kawarabayashi, and G. Schwartzman: **Parameterized distributed algorithms**. *DISC 2018*.

J. Edmonds: **Paths, trees, and flowers**. *Canadian Journal of Mathematics*, 1965.

L. R. Ford, Jr., and D. R. Fulkerson: **Maximal flow through a network**. *Canadian Journal of Mathematics*, 1956.

J. E. Hopcroft and R. M. Karp: **An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs**. *SICOMP*, 1973.

T. Izumi, N. Kitamura, and Y. Yamaguchi: **A nearly linear-time distributed algorithm for exact maximum matching**. *SODA 2024*.

N. Kitamura and T. Izumi. **A subquadratic-time distributed algorithm for exact maximum matching**. *IEICE Trans. on Information and Systems*, 2022.

D. Kőnig: **Graphok és matrixok**, *Mathematikai és Fizikai Lapok*, 1931.

S. Micali and V. V. Vazirani: **An $\mathbf{O}(\sqrt{V}E)$ algorithm for finding maximum matching in general graphs**. *FOCS 1980*.

V. V. Vazirani: **A theory of alternating paths and blossoms from the perspective of minimum length**. *Mathematics of Operations Research*, 2024.

参考文献 2（やらなかった話の分）

M. L. Balinski: **Labelling to obtain a maximum matching**. In *Combinatorial Mathematics and Its Applications*. 1969.

C. Berge: **Sur le couplage maximum d'un graphe**. *Comptes Rendus Hebdomadaires des Séances de l'Académie de Sciences,* 1958.

J. Chuzhoy and S. Khanna: **A faster combinatorial algorithm for maximum bipartite matching**. *SODA 2024*.

H. N. Gabow: **An efficient implementation of Edmonds' algorithm for maximum matching on graphs**. *J. ACM*, 1976.

A. Galluccio and M. Loebl: **On the theory of Pfaffian orientations. I. Perfect matchings and permanents**. *Electronic Journal of Combinatorics*, 1999.

A. V. Goldberg and A. V. Karzanov: **Maximum skew-symmetric flows and matchings**. *Mathematical Programming*, 2004.

A. V. Karzanov: **Maximum matching of given weight in complete and complete bipartite graphs**. *Cybernetics*, 1987.

K. Mulmuley, U. V. Vazirani, and V. V. Vazirani: **Matching is as easy as matrix inversion**. *Combinatorica*, 1987.

C. H. Papadimitriou and M. Yannakakis: **The complexity of restricted spanning tree problems**. *J. ACM*, 1982.

T. Rothvoss: **The matching polytope has exponential extension complexity**. *J. ACM*, 2017.

W. T. Tutte: **The factorization of linear graphs**. *The Journal of the London Mathematical Society*, 1947.

R. Yuster: **Almost exact matching**. *Algorithmica*, 2012.