離散構造とアルゴリズム
(2-2) アルゴリズムの解析

高橋篤司
東京工業大学 工学院 情報通信系

アルゴリズムの解析

- 正当性
  - 問題$\Pi = \left(I, Q(x)\right)$を解くアルゴリズム$A$
    - すべての問題例で（正しい）解を生成するか？
- 計算複雑度
  - 時間計算量
    - どのくらいの時間（基本操作）が必要か？
  - 空間計算量
    - どのくらいのメモリが必要か？
- 最適性
  - 理論的な下限を達成しているか？

2-2（1）問題

- 問題(problem) $\Pi = \left(I, Q(x)\right)$
  - 入力集合Iと質問$Q(x)$の組
  - 問題例（instance）：入力$s(\in I)$に対応する具体的な問題

- 例
  - 問題
    - 2次方程式の根を求めよ
      - $ax^2 + bx + c = 0$ s.t $a, b, c \in \mathbb{N}$, $a \neq 0$
  - 入力
    - $a = 2, b = 3, c = 5$
  - 問題例
    - 2次方程式$2x^2 + 3x + 5 = 0$の根を求めよ

  - アルゴリズム
    - $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

部分問題

- 問題(problem) $\Pi = \left(I, Q(x)\right)$
  - 入力集合Iと質問$Q(x)$の組
  - 問題例（instance）：入力$s (\in I)$に対応する具体的な問題

- 部分問題$\Pi' = \left(I', Q(x)\right)$
  - $\Pi = \left(I, Q(x)\right)$の部分問題
  - $I' \subseteq I$

➢ 例
  - I'：連結グラフ，I：グラフ
  - $I' = \mathbb{N}, I = \mathbb{Z}$

問題

- 判定問題
  - 答えが「Yes」か「No」である問題
    - 例：Gはハミルトングラフか
- 探索問題
  - 「Yes」である証拠を示す問題
    - 例：GがハミルトングラフならばGのハミルトン閉路を一つ示せ
- 最適化問題
  - ある値を最適化する問題
    - 例：Gの長さが最大の閉路を一つ示せ（その長さを示せ）

問題:2次方程式

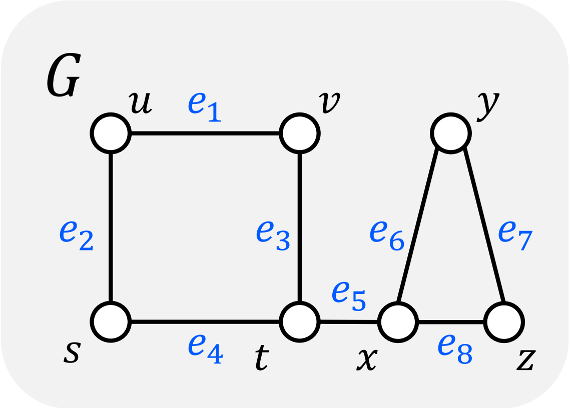
- QUADRATIC EQUATION
  - 入力
    - $a, b, c \in \mathbb{N}, \quad a \neq 0$
  - 質問
    - $ax^2 + bx + c = 0$の根を一つ求めよ

- QUADRATIC EQUATION（判定）
  - 入力
    - $a, b, c \in \mathbb{N}, \quad a \neq 0$
  - 質問
    - $ax^2 + bx + c = 0$ は実数解をもつか?

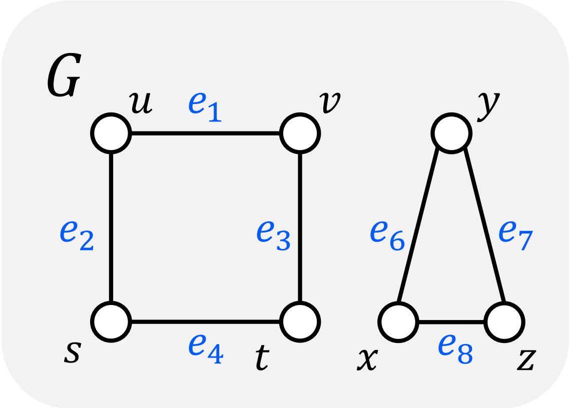
- QUADRATIC EQUATION（最適化）
  - 入力
    - $a, b, c \in \mathbb{N}, \quad a \neq 0$
  - 質問
    - $ax^2 + bx + c = 0$の最大の根を一つ求めよ

問題：連結グラフ

- CONNECTED GRAPH （判定）
  - 入力
    - Graph G
  - 質問
    - Gは連結か？



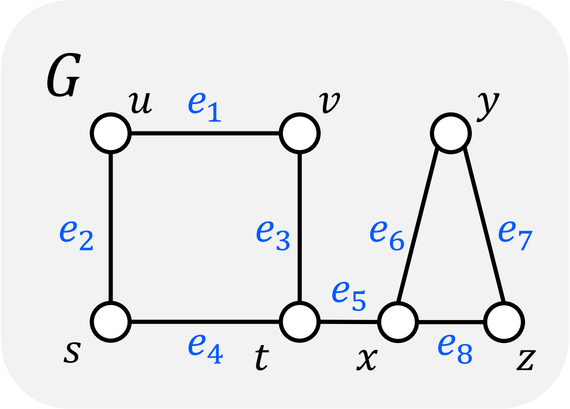
連結(connected)



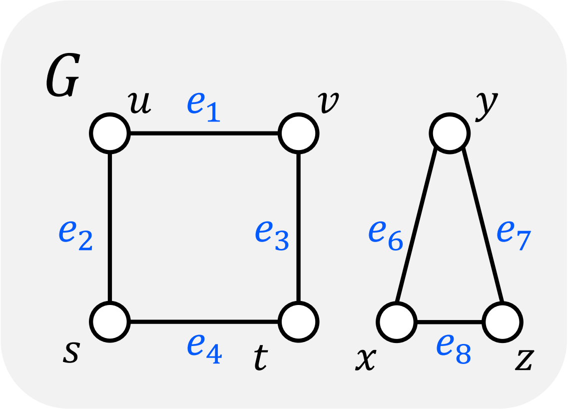
非連結(disconnected)

問題：連結グラフ

- CONNECTED GRAPH（探索）
  - 入力
    - Graph G
  - 質問
    - Gが連結なら，Gの全域木を一つ示せ



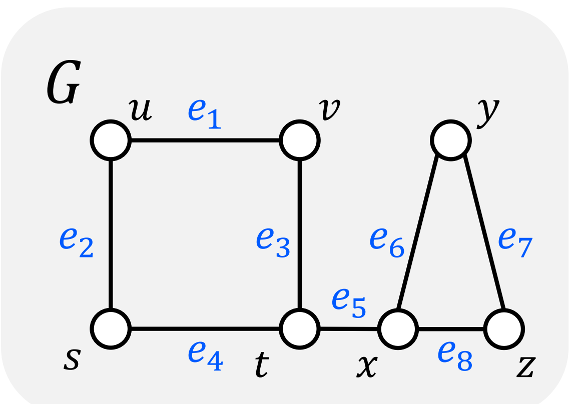
連結(connected)



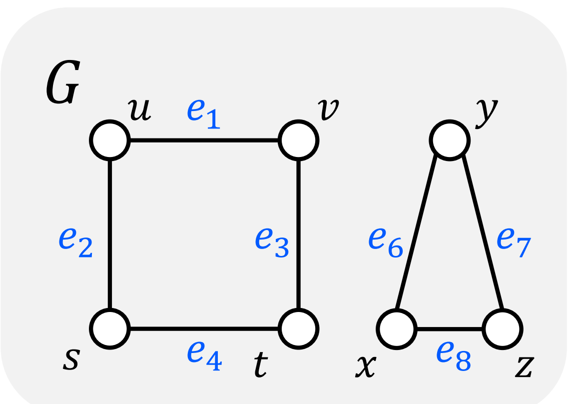
非連結(disconnected)

問題：連結グラフ

- CONNECTED GRAPH（最適化）
  - 入力
    - Graph G
  - 質問
    - Gの最大な連結部分グラフを一つ示せ



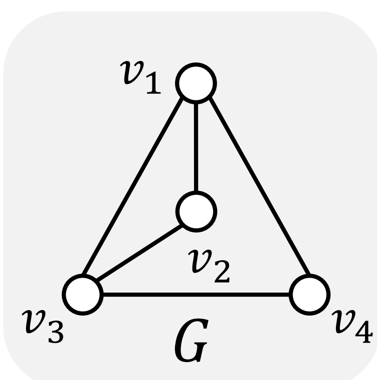
連結(connected)

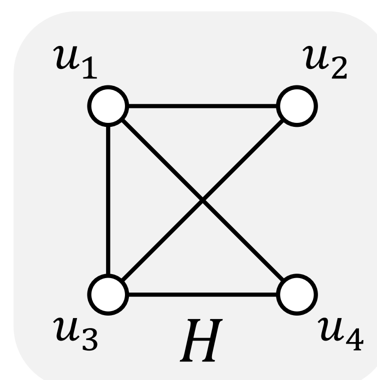


非連結(disconnected)

問題：グラフ同型

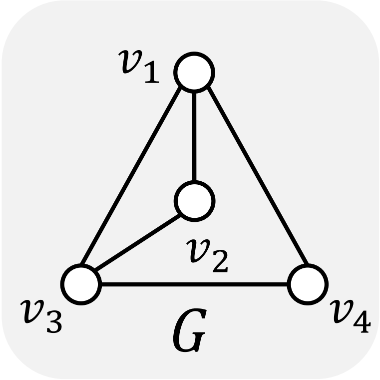
- GRAPH ISOMORPHISM（判定）
  - 入力
    - Graphs G and H
  - 質問
    - GとHは同型か？



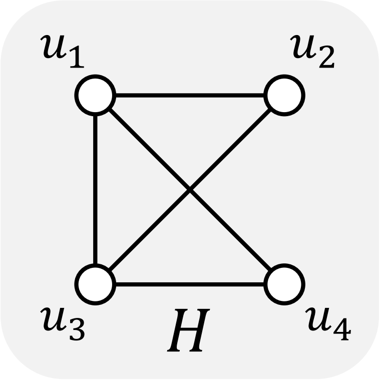


問題：グラフ同型

- GRAPH ISOMORPHISM（探索）
  - 入力
    - Graphs G and H
  - 質問
    - GとHが同型ならば，同型写像$\phi: V(G) \to V(H)$を一つ示せ

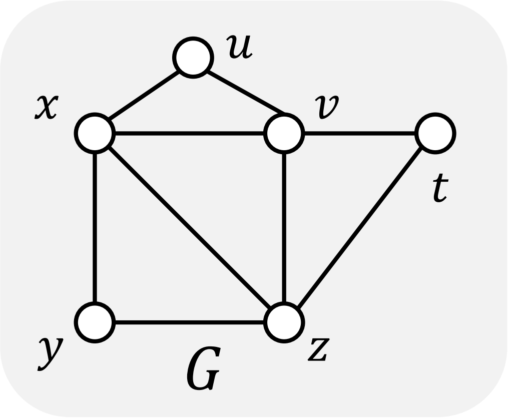


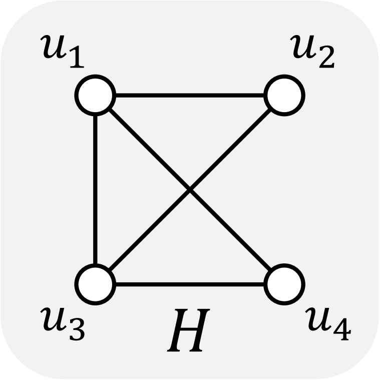
$\phi(v_1)=u_1$
$\phi(v_2)=u_2$
$\phi(v_3)=u_3$
$\phi(v_4)=u_4$



問題:グラフ同型

- GRAPH ISOMORPHISM（最適化）
  - 入力
    - Graphs G and H
  - 質問
    - Hのある部分グラフと同型である最大なGの部分グラフを一つ示せ

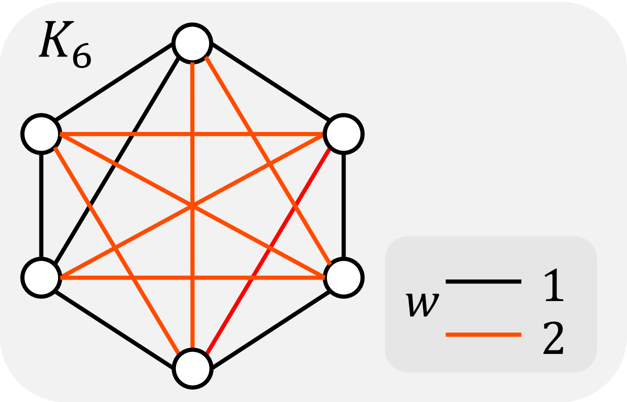




巡回セールスマン

■ 巡回セールスマン問題

- 入力：完全グラフK_n, 重み関数$w: E(K_n) \to \mathbb{R}^+$
- 質問：ネットワーク(K_n, w)の重み最小のハミルトン閉路を一つ示せ



■ 巡回セールスマン判定問題(TS)

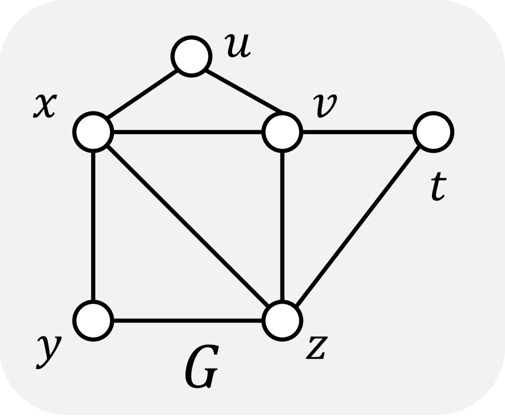
- 入力：完全グラフK_n, 重み関数$w: E(K_n) \to \mathbb{R}^+$, 非負実数$r \in \mathbb{R}^+$
- 質問：ネットワーク(K_n, w)に重みr以下のハミルトン閉路が存在するか

■ 巡回セールスマン探索問題

- 入力：完全グラフK_n, 重み関数$w: E(K_n) \to \mathbb{R}^+$, 非負実数$r \in \mathbb{R}^+$
- 質問：ネットワーク(K_n, w)に重みr以下のハミルトン閉路が存在するならば, それを一つ示せ

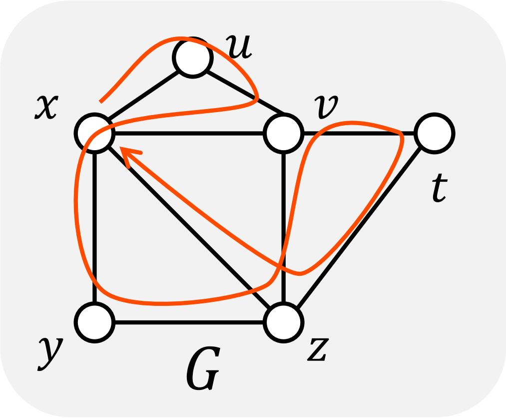
問題：オイラーグラフ

- EULERIAN GRAPH（判定）
  - 入力
    - Graph G
  - 質問
    - Gはオイラーグラフか?



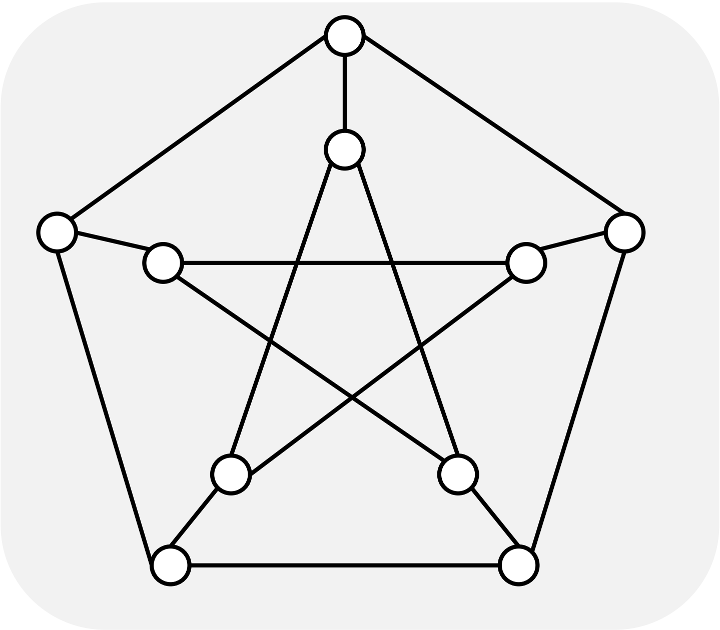
問題：オイラーグラフ

- EULERIAN GRAPH（探索）
  - 入力
    - Graph G
  - 質問
    - Gがオイラーグラフならば，オイラー閉トレイルを一つ示せ

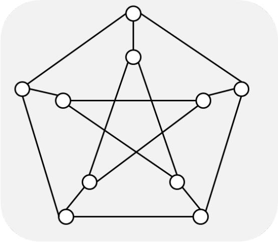


問題：オイラーグラフ

- EULERIAN GRAPH（最適化）
  - 入力
    - Graph G
  - 質問
    - Gの長さが最大の閉トレイルを一つ示せ



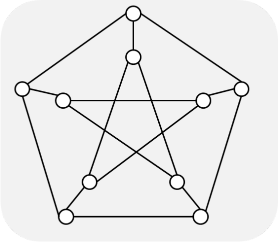
オイラーグラフ



- オイラーグラフ判定問題(EG)
  - 入力：グラフG
  - 質問：Gはオイラーグラフか

- オイラー閉トレイル問題
  - 入力：グラフG
  - 質問：Gがオイラーグラフならば，オイラー閉トレイルを一つ示せ
    - 証拠の例（オイラーグラフ）
      - Gのオイラー閉トレイル
      - 点の次数がすべて偶数である，かつ，Gの全域木
- 最大閉トレイル問題
  - 入力：グラフG
  - 質問：Gの長さが最大である閉トレイルを一つ示せ
    - 最適化する値の例
      - 閉トレイルの長さ
      - オイラーグラフであるGの部分グラフの辺数

ハミルトングラフ

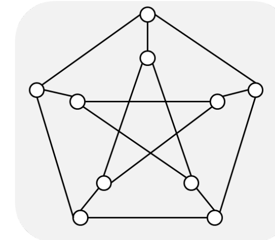


- Π_1：ハミルトングラフ判定問題(HG)
  - 入力：グラフG
  - 質問：Gはハミルトングラフか

- Π_2：ハミルトン閉路問題
  - 入力：グラフG
  - 質問：Gがハミルトングラフならば，ハミルトン閉路を一つ示せ
    - 証拠の例(ハミルトングラフ)
      - Gのハミルトン閉路

- Π_3：最大閉路問題
  - 入力：グラフG
  - 質問：Gの長さが最大である閉路を一つ示せ
    - 最適化する値の例
      - 閉路の長さ
      - ハミルトングラフであるGの部分グラフの辺数

判定問題, 探索問題, 最適化問題



- Π_1：ハミルトングラフ判定問題(HG)
  - 質問：Gはハミルトングラフか
- Π_2：ハミルトン閉路問題
  - 質問：Gがハミルトングラフならば, ハミルトン閉路を一つ示せ
- Π_3：最大閉路問題
  - 質問：Gの長さが最大である閉路を一つ示せ

➢ 最適化問題Π_3が解ける $\Longrightarrow$ 探索問題Π_2が解ける
- 示された最大閉路の長さが点数と等しければ, ハミルトン閉路

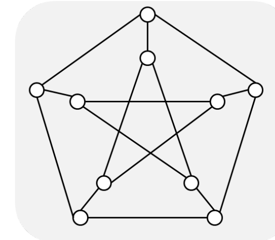
✓ $\Pi_2 \lesssim \Pi_3$ ： 最適化問題は探索問題よりも易しくはない

➢ 探索問題Π_2が解ける $\Longrightarrow$ 判定問題Π_1が解ける
- ハミルトン閉路が示されたら, 判定問題の答えは”Yes”

✓ $\Pi_1 \lesssim \Pi_2$ ： 探索問題は判定問題よりも易しくはない

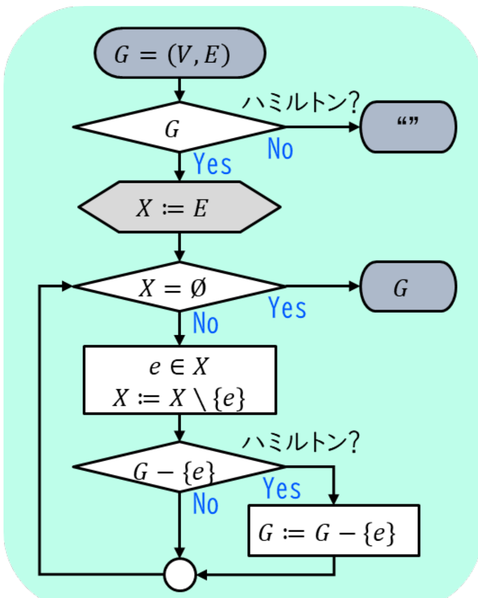
Π_1	$\lesssim$	Π_2	$\lesssim$	Π_3
解ける		解ける		解ける
解けない		解けない		解けない

ハミルトン閉路探索

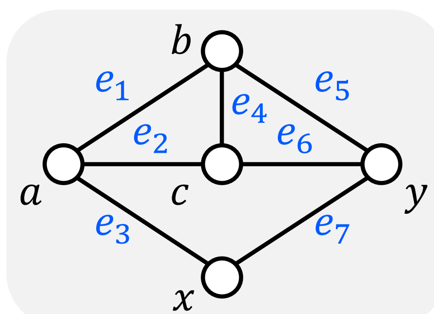


- Π_1：ハミルトングラフ判定問題(HG)
  - 質問：Gはハミルトングラフか
- Π_2：ハミルトン閉路問題
  - 質問：Gがハミルトングラフならば，ハミルトン閉路を一つ示せ

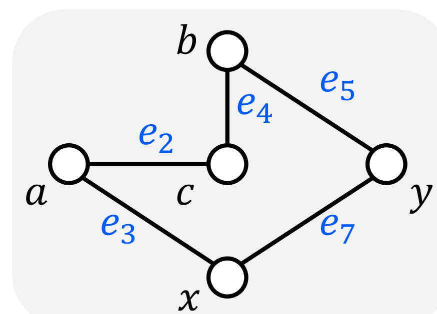
✓ ハミルトン閉路
- ハミルトングラフ判定を用いて得られる
  - $|E(G)|+1$回



- 辺を取り除いたときハミルトングラフ
  - のまま $\Longrightarrow$ その辺は不要
  - でない $\Longrightarrow$ その辺はキープ

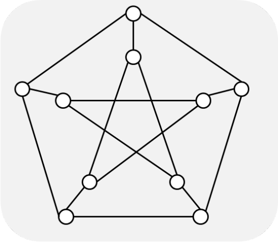


G



Gのハミルトン閉路

判定問題, 探索問題, 最適化問題



- Π_1：ハミルトングラフ判定問題(HG)
  - 質問：Gはハミルトングラフか
- Π_2：ハミルトン閉路問題
  - 質問：Gがハミルトングラフならば, ハミルトン閉路を一つ示せ
- Π_3：最大閉路問題
  - 質問：Gの長さが最大である閉路を一つ示せ

➢ 探索問題Π_2が解ける $\Leftarrow$ 判定問題Π_1が解ける

✓ $\Pi_2 \lesssim \Pi_1$ ： 判定問題は探索問題よりも易しくはない

➢ 探索問題Π_2が解ける $\Rightarrow$ 判定問題Π_1が解ける

- ハミルトン閉路が示されたら, 判定問題の答えは" Yes"

✓ $\Pi_1 \lesssim \Pi_2$ ： 探索問題は判定問題よりも易しくはない

Π_1	$\lesssim$	Π_2	$\lesssim$	Π_3
解ける		解ける		解ける
解けない		解けない		解けない

2-2（2）アルゴリズムの解析

- 問題$\Pi = \big(I, Q(x)\big)$に対するアルゴリズムA
- 実行時間
  - 計算機の性能によって異なる
  - 入力の大きさによって異なる
  - 得意, 不得意な入力によって異なる

➢ 時間的な性能を入力の大きさの関数として表現

✓ 最悪の場合で評価

- アルゴリズムAの時間計算量
  - $T_A(n) = \max\{t_A(s) | s \in I_n\}$
    - $t_A(s)$： 入力sに対する基本操作の最大数
    - I_n： 大きさnのすべての入力からなるIの部分集合

2-2（3）多項式時間アルゴリズム

- 問題が解けたってどういうこと
  - 問題$\Pi = \big(I, Q(x)\big)$に対するアルゴリズム
    - 多項式時間アルゴリズムが存在

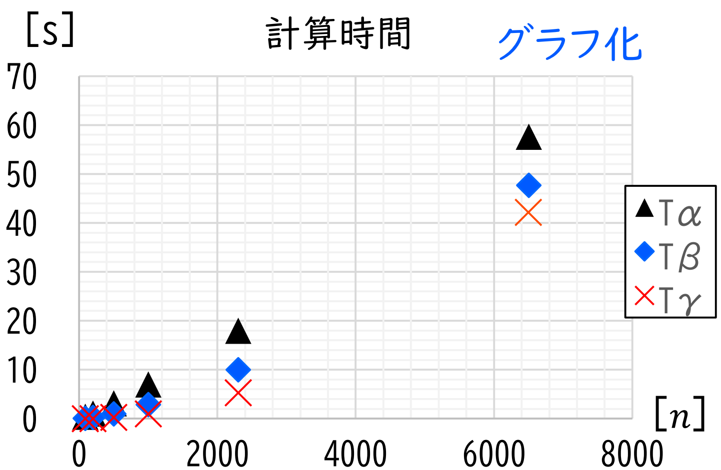
どのアルゴリズムがよいか？

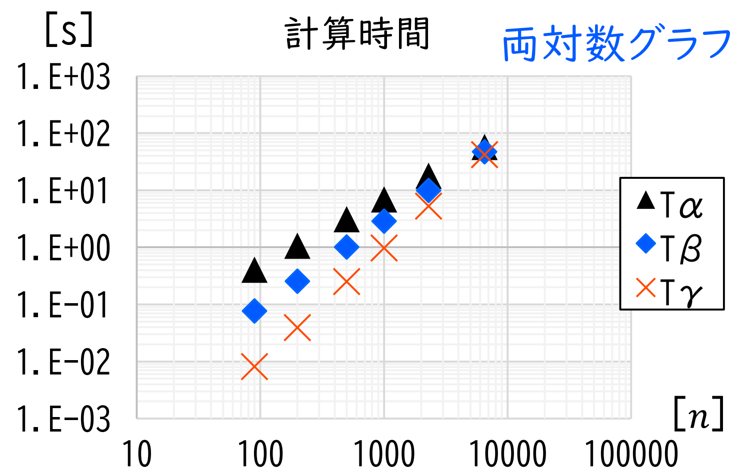
- アルゴリズムα, β, γ
  - データ数nと計算時間$T[s]$の関係から計算複雑度を求める

n	T_α	T_β	T_γ
90	0.408986	0.077623	0.00818
200	1.070139	0.257348	0.039474
500	3.138024	1.01738	0.254517
1000	6.976048	2.877602	0.983591
2300	17.97954	10.03766	5.302225
6500	57.63132	47.68819	42.128

n	T_α	T_β	T_γ
90	0.41	0.08	0.01
200	1.07	0.26	0.04
500	3.14	1.02	0.25
1000	6.98	2.88	0.98
2300	17.98	10.04	5.30
6500	57.63	47.69	42.13

わかりやすさを考慮した表



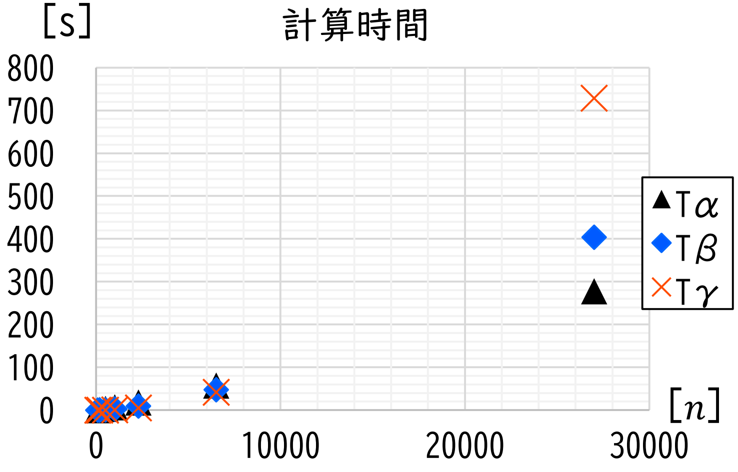


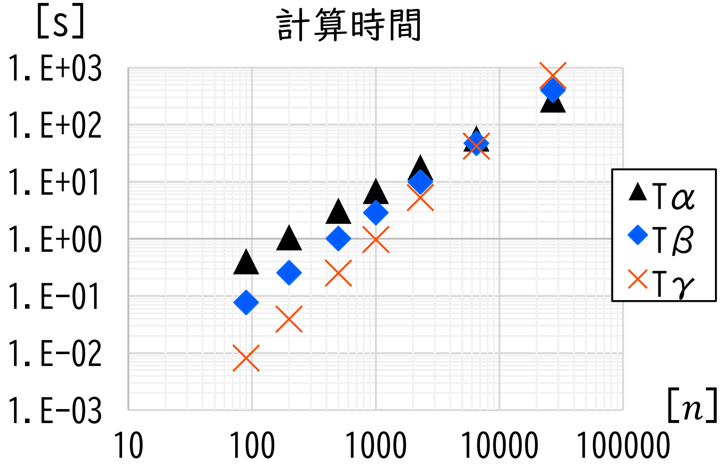
どのアルゴリズムがよいか？

- アルゴリズムα, β, γ
  - データ数nと計算時間$T[s]$の関係から計算複雑度を求める

n	T_α	T_β	T_γ
90	0.408986	0.077623	0.00818
200	1.070139	0.257348	0.039474
500	3.138024	1.01738	0.254517
1000	6.976048	2.877602	0.983591
2300	17.97954	10.03766	5.302225
6500	57.63132	47.68819	42.128
27000	278.2207	403.7262	729.143

n	T_α	T_β	T_γ
90	0.41	0.08	0.01
200	1.07	0.26	0.04
500	3.14	1.02	0.25
1000	6.98	2.88	0.98
2300	17.98	10.04	5.30
6500	57.63	47.69	42.13
27000	278.22	403.73	729.14

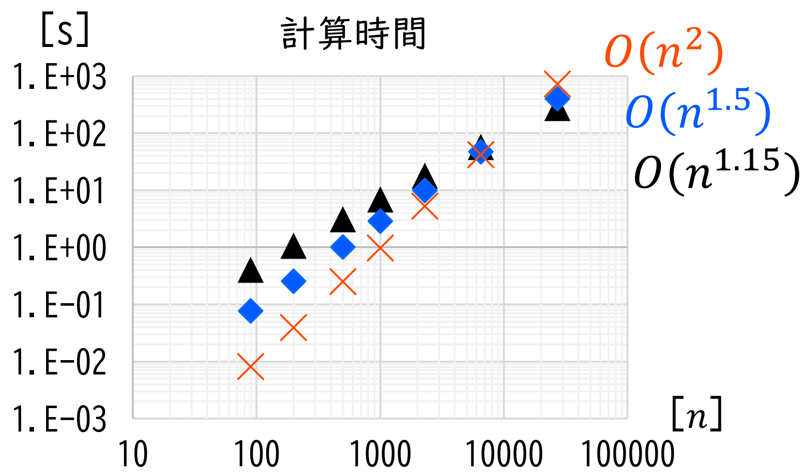




どのアルゴリズムがよいか？

- アルゴリズムα, β, γ
  - データ数nと計算時間$T[s]$の関係から計算複雑度を求める
  - $T = an^b + c$
    - $\log T \approx b \log n + \log a$
    - $Y \approx bX + \log a$
  - $T_\alpha \approx 700 n \log_2 n \times 10^{-6}$
    - $\log T_\alpha \approx 1.15 \log n - 2.60$
  - $T_\beta \approx 91 n\sqrt{n} \times 10^{-6}$
    - $\log T_\beta \approx 1.50 \log n - 4.04$
  - $T_\gamma \approx n^2 \times 10^{-6}$
    - $\log T_\gamma \approx 2.00 \log n - 6.00$

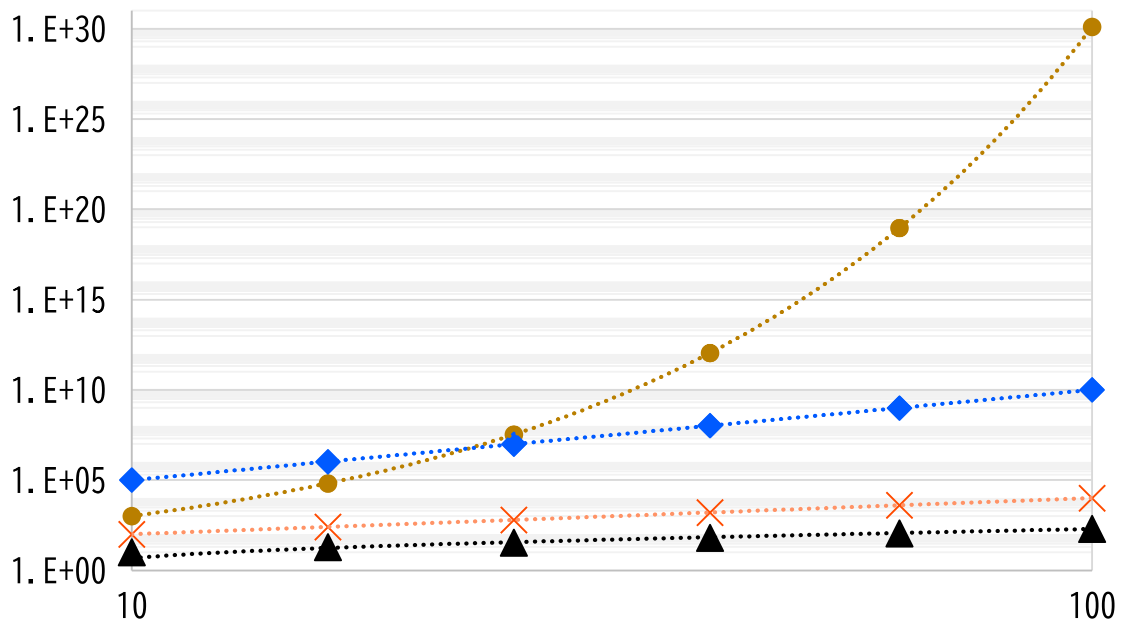
n	T_α	T_β	T_γ
90	0.41	0.08	0.01
200	1.07	0.26	0.04
500	3.14	1.02	0.25
1000	6.98	2.88	0.98
2300	17.98	10.04	5.30
6500	57.63	47.69	42.13
27000	278.22	403.73	729.14



多項式オーダーと指数オーダー

- 両対数グラフ
  - 多項式：直線
    - 傾き：オーダー
  - 指数：曲線

n	2^n	n^5	n^2	$n\log_{10} n$
10	1,024	100,000	100	10
16	65,536	1,048,576	256	19
25	33,554,432	9,765,625	625	34
40	1.10E+12	102,400,000	1,600	64
63	9.22E+18	992,436,543	3,969	113
100	1.27E+30	1.00E+10	10,000	200



何秒使えるか？

- 1分 = 60 s = 6.0×10^{1}s
- 1時間 = $3,600$ s = 3.6×10^{3}s
- 1日 = $86,400$ s = 8.64×10^{4}s
- 1か月(30日) = $2,592,000$ s = 2.592×10^{6}s
- 1年(365日) = $31,536,000$ s = 3.1536×10^{7}s
- 10年 = $315,360,000$ s = 3.1536×10^{8}s
- 100億年 = 3.1536×10^{17}s
- 宇宙年齢 ≒ 138億年 ≒ 4.35×10^{17}s

- $2^{60} \approx 1.15 \times 10^{18}$
- $20! \approx 2.43 \times 10^{18}$

問題が解けたってどういうこと

- 解き方が分かった ⇒ 解けた，とは言えない
  - ある数が素数かどうか判定するには?
    - 小さい整数から順に割り算する
- 効率のよい解き方が分かる ⇒ 解けた!!
  - 素数かどうか効率よく判定できるのか?
  - 問題サイズnの多項式時間で解けるか?
    - 2^n時間，$n!$時間かかったらダメ

- 計算機科学の未解決問題
  - $\mathbf{P} \neq \mathbf{NP}$?
  - 解き方が分かったら，効率のよい解き方が存在するか

計算複雑度

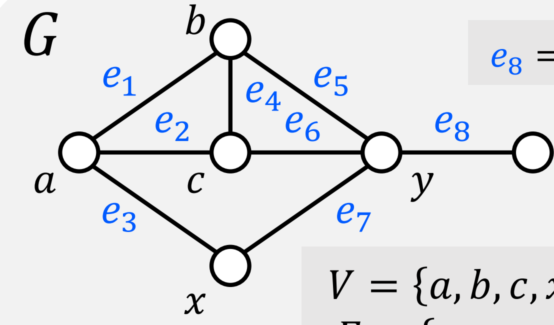
- アルゴリズムAの時間計算量による評価
  - $T_A(n) = O(n)$
    - 線形
      - 最適アルゴリズム
        - 入力をすべて読む必要がある場合
  - $T_A(n) = O(n^k)$
    - 多項式：問題が解けた
      - 大規模データ
        - $O(n^2)$では苦しい
        - $O(n \log n)$以下が望まれる
  - $T_A(n) = O(k^n)$
    - 指数：全探索などの場合
      - 超小規模なデータのみ

2-2（4）グラフの大きさ

- 次数 $\deg_G(v)$ ： 点vに接続する辺の数
- **隣接行列** $A(G)$ ［点×点］ （$[n \times n]$）
  - 点と点の隣接関係　　（無向グラフ ⇒ 対称行列）

$$
\begin{array}{c|cccccc|c}
 & a & b & c & x & y & z & \text{次数} \\
\hline
a & 0 & 1 & 1 & 1 & 0 & 0 & 3 \\
b & 1 & 0 & 1 & 0 & 1 & 0 & 3 \\
c & 1 & 1 & 0 & 0 & 1 & 0 & 3 \\
x & 1 & 0 & 0 & 0 & 1 & 0 & 2 \\
y & 0 & 1 & 1 & 1 & 0 & 1 & 4 \\
z & 0 & 0 & 0 & 0 & 1 & 0 & 1
\end{array}
$$

行の「1」の和：次数
列の「1」の和：次数



$e_8 = (y, z)$

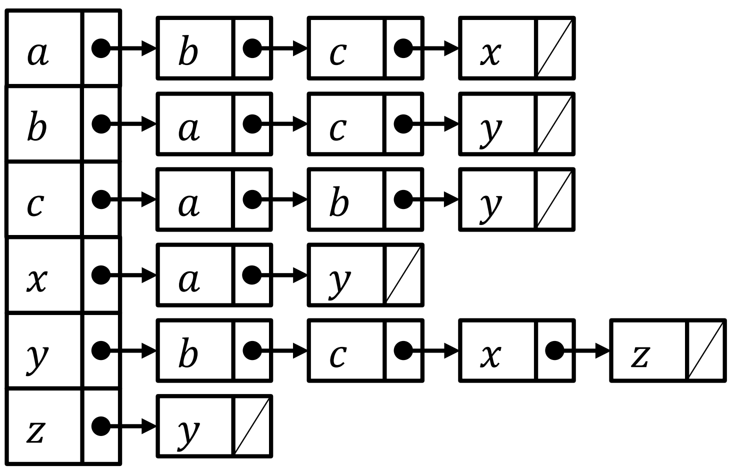
$V = \{a, b, c, x, y, z\}$
$E = \{e_1, e_2, \dots, e_8\}$

- **接続行列** $B(G)$ ［点×辺］ （$[n \times m]$）
  - 点と辺の接続関係

$$
\begin{array}{c|cccccccc|c}
 & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & \text{次数} \\
\hline
a & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 3 \\
b & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 3 \\
c & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 3 \\
x & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 2 \\
y & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 4 \\
z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}
$$

行の「1」の和：次数

隣接リスト

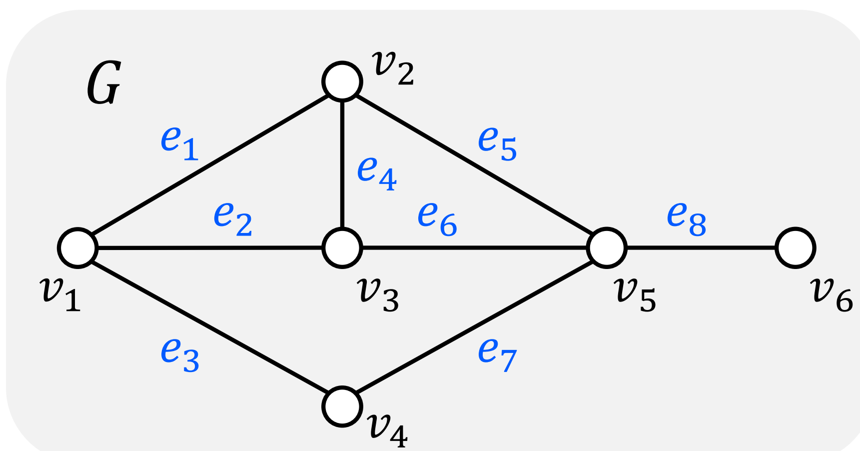


辺数が少ないグラフの効率よい表現

隣接行列 (Adjacency Matrix)

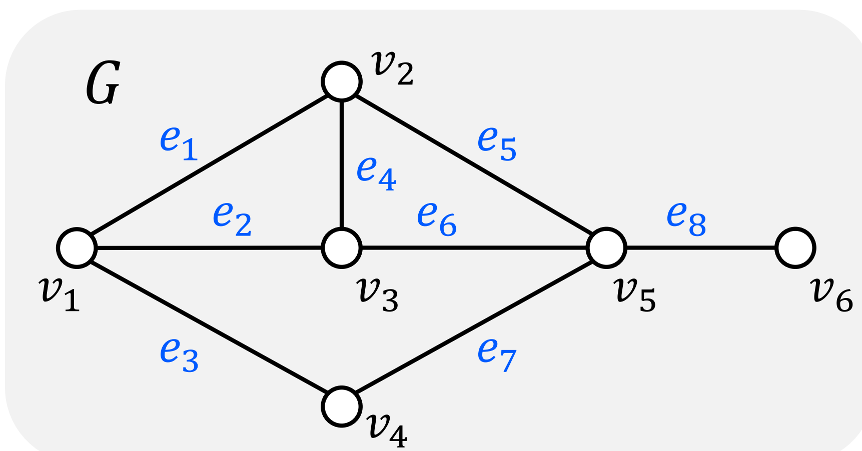
- Computer Representation of Graphs
  - Adjacency Matrix

- $$A(G) = \begin{array}{cc} \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} & \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \\ \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} & \end{array}$$



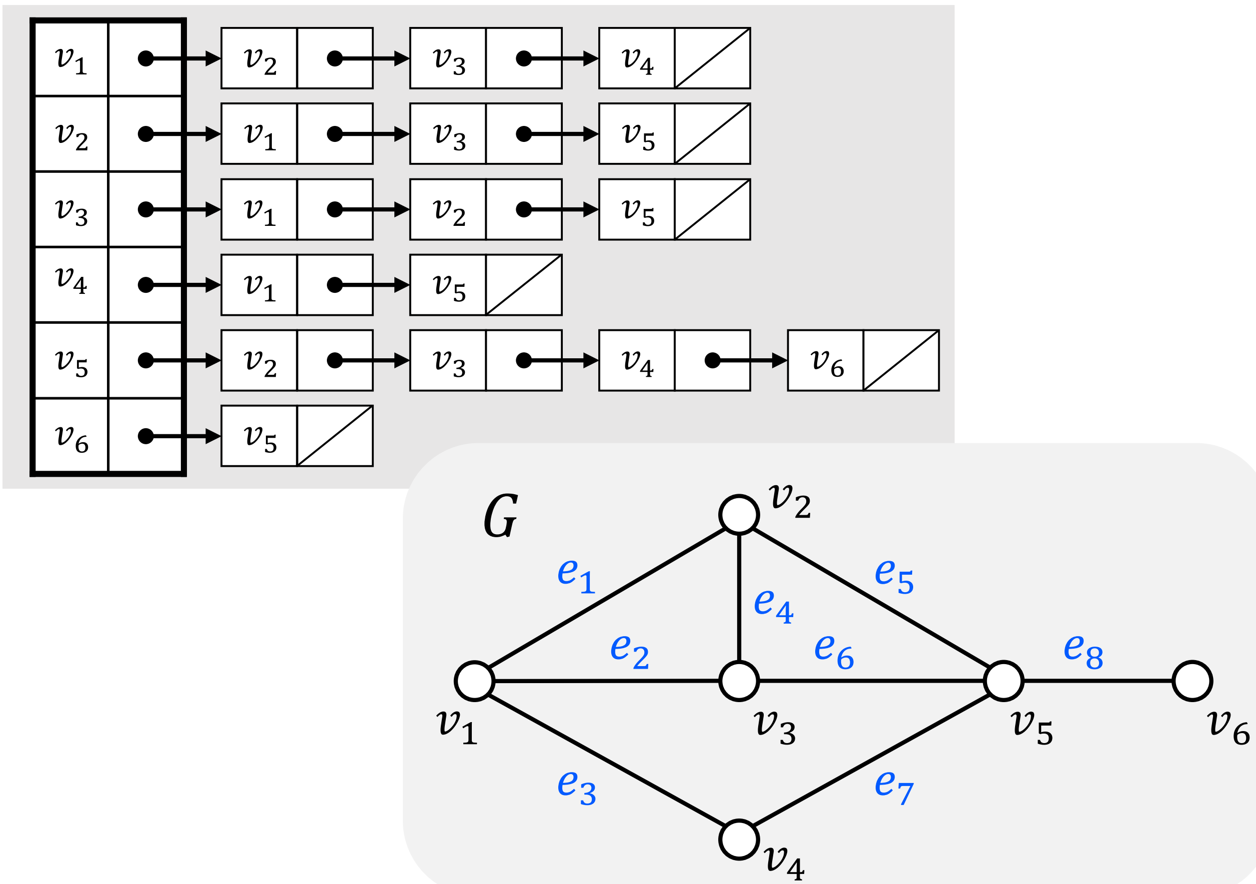
接続行列 (Incidence Matrix)

- Computer Representation of Graphs
  - Incidence Matrix
    - $$B(G) = \begin{array}{c} \\ \end{array}\begin{array}{cc} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} & \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \\ \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} & \end{array}$$



隣接リスト (Adjacency List)

- Computer Representation of Graphs
  - Adjacency List



グラフの表現

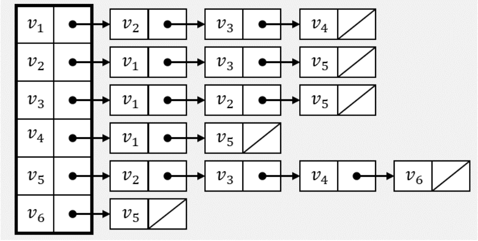
■ 定理 (Theorem 2.2) グラフG

（点数$n = |V(G)|$, 辺数$m = |E(G)|$）

1. 隣接行列$A(G)$の大きさ
   - ■ $\Theta(n^2)$
2. 接続行列$B(G)$の大きさ
   - ■ $\Theta(nm)$
3. 隣接リストの大きさ
   - ■ $\Theta(n+m)$

$$A(G) = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$B(G) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



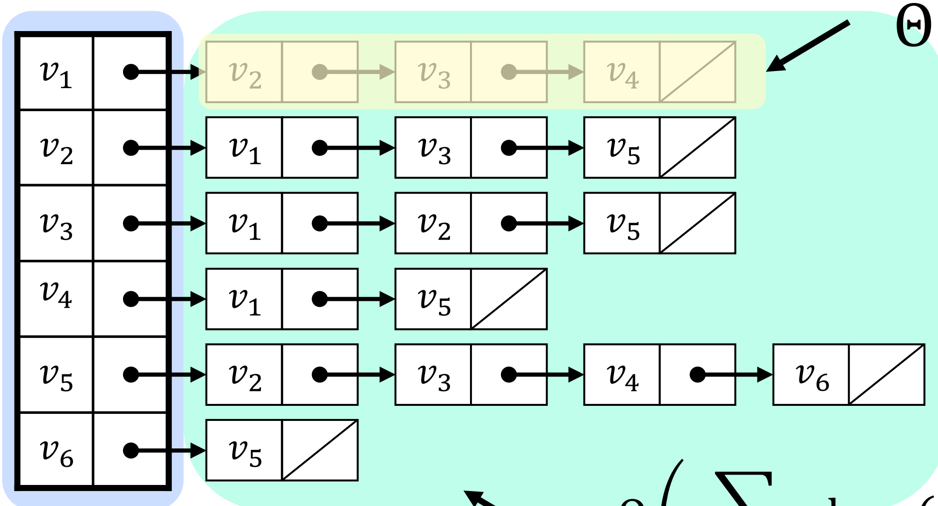
□ 証明 ：

1. 隣接行列$A(G)$ ： $n \times n$行列 $\therefore \Theta(n^2)$
2. 接続行列$B(G)$ ： $n \times m$行列 $\therefore \Theta(nm)$

グラフの表現

- **定理 (Theorem 2.2)** グラフG
  （点数$n = |V(G)|$, 辺数$m = |E(G)|$）
  3. 接続リストの大きさ
     - $\Theta(n+m)$

- 証明 ：



$\Theta(\deg_G(v))$

$\Theta(n)$

$$\Theta\left(\sum_{v \in V(G)} \deg_G(v)\right) = \Theta(m)$$

[1-1]定理1.3(次数総和と辺数)

連結グラフの点数と辺数

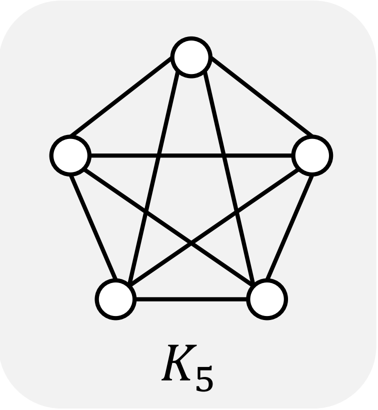
- 定理 (Theorem 2.3) 連結グラフG

（点数$n = |V(G)|$, 辺数$m = |E(G)|$）

$$m = \Omega(n) \text{ and } m = O(n^2)$$

□ 証明 :

- $m \geq n - 1$ ([1-2]系1.2(連結グラフの辺数))
  - $\lim_{n \to \infty} \frac{m}{n} \geq 1$
  - $m = \Omega(n)$
- $m \leq \frac{n(n-1)}{2}$ ([1-3]定理(完全グラフの点数))
  - $\lim_{n \to \infty} \frac{m}{n^2} \leq \frac{1}{2}$
  - $m = O(n^2)$

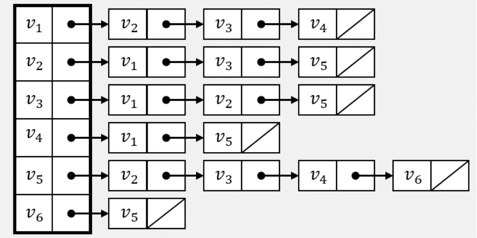


連結グラフの大きさ

- **定理（Theorem 2.4）** 連結グラフG
  （点数$n = |V(G)|$, 辺数$m = |E(G)|$）
  1. 隣接行列$A(G)$の大きさ
     - $\Theta(n^2)$
  2. 接続行列$B(G)$の大きさ
     - $\Omega(n^2)$ and $O(n^3)$
  3. 隣接リストの大きさ
     - $\Omega(n)$ and $O(n^2)$

$$A(G) = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

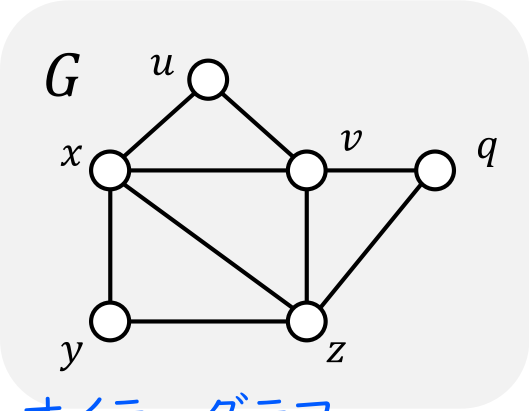
$$B(G) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



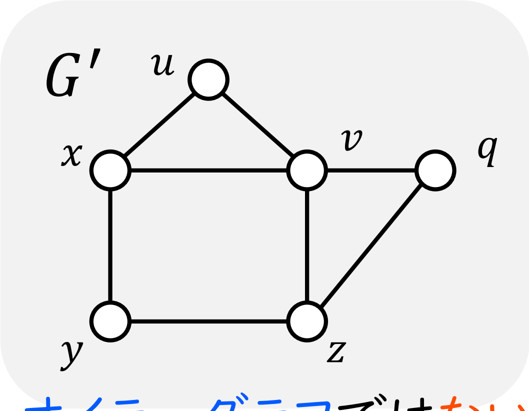
- **証明**：
  - ✓ $m = \Omega(n), m = O(n^2)$ （定理2.3）
  1. 隣接行列$A(G)$：$\Theta(n^2)$ （定理2.2）
  2. 接続行列$B(G)$：$\Theta(nm)$ （定理2.2）
  3. 隣接リスト：$\therefore \Theta(n+m)$ （定理2.2）

2-2（5）オイラーグラフ判定問題

- 連結オイラーグラフ判定問題
  - 入力：連結グラフ$G = (V, E)$
    - 入力サイズ：$\Theta(n+m), n = |V(G)|, m = |E(G)|$
  - 質問：Gはオイラーグラフか



オイラーグラフ



オイラーグラフではない

連結オイラーグラフ判定アルゴリズム

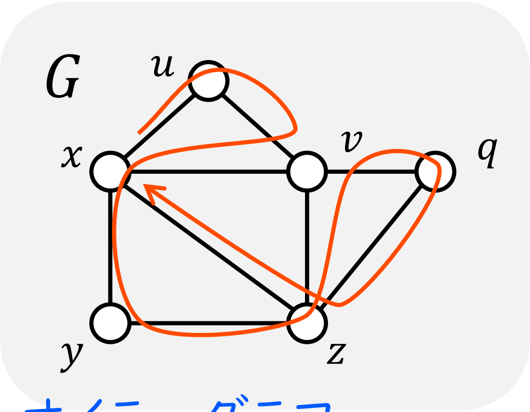
- Algorithm 2.0 (Euler Graph: Exhaustive Search)
  - 入力：連結グラフ$G = (V, E)$（$|V| = n, |E| = m$）
  - 出力："Yes" or "No"

  Step 1：Eのすべての順列を生成（$m!$個）

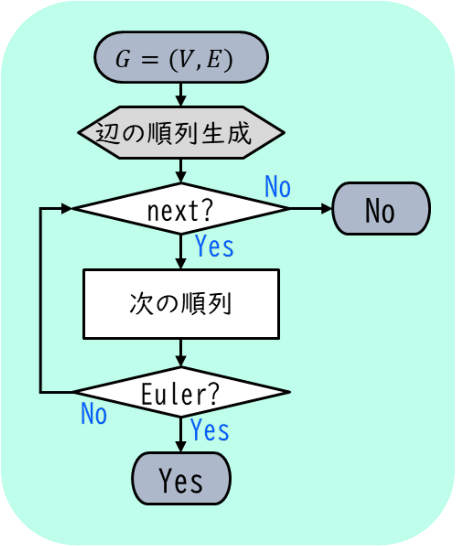
  Step 2:それぞれがオイラー閉トレイルに対応するかチェック
  - 一つでも対応していれば "Yes" と出力
  - すべて対応していなければ "No" と出力

➢ 正当性 ：trivial

➢ 計算複雑度 ：$\Omega(m!)$
- Inefficient !!



オイラーグラフ



連結オイラーグラフ判定アルゴリズム

- **Algorithm 2.1** (Euler Graph: Degree Check)
  - 入力： 連結グラフ$G = (V, E)$ ($|V| = n, |E| = m$)
  - 出力： "Yes" or "No"

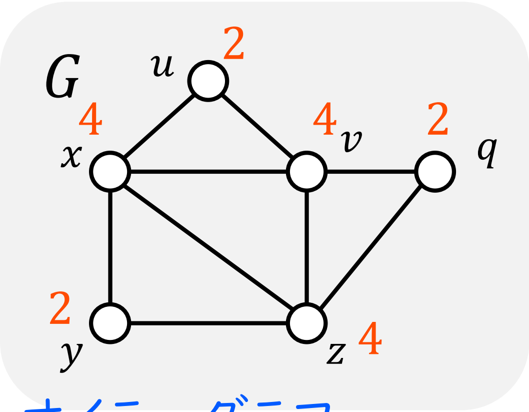
Step 0: Set $X = V$

Step 1: If $X = \emptyset$, then output "YES" and halt

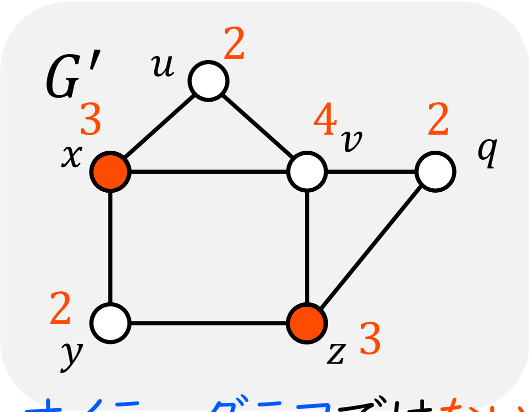
Step 2: Choose any $x \in X$ and set $X = X \setminus \{x\}$

Step 3: If $\deg_G(x)$ is odd then output "NO" and halt

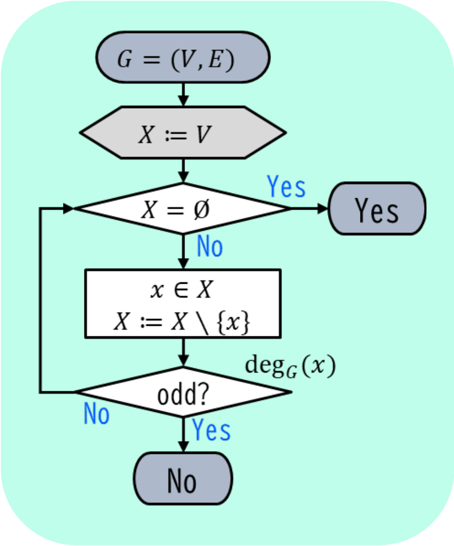
Step 4: return to Step 1



オイラーグラフ



オイラーグラフではない



連結オイラーグラフ判定アルゴリズム

- ■ 定理 (Theorem 2.5)
  - Alg.2.1は連結オイラーグラフ判定問題を線形時間で解く
- □ 証明
  - 正当性 ： [1-3]定理1.11(オイラーの定理)より
  - 時間計算量:
    - ■ $O(n+m)$:線形時間アルゴリズム(漸近的に最適)

■定理 (Theorem 2.5)
-Alg.2.1は連結オイラーグラフ判定問題を線形時間で解く
□証明
-正当性：[1-3]定理1.11(オイラーの定理)より
-時間計算量:
■$O(n+m)$:線形時間アルゴリズム(漸近的に最適)

■ Algorithm 2.1 (Euler Graph: Degree Check)

- 入力: 連結グラフ$G=(V,E)$ ($|V|=n, |E|=m$)
- 出力: “Yes” or “No”

Step 0: Set $X = V$ ← $O(n)$

Step 1: If $X=\emptyset$, then output “YES” and halt $O(1)$

Step 2: Choose any $x \in X$ and set $X = X \setminus \{x\}$ $O(1)$

Step 3: If $\deg_G(x)$ is odd then output “NO” and halt

Step 4: return to Step 1 ← $O(\deg_G(x))$

最大n回

$$\Theta\left(\sum_{v\in V(G)} \deg_G(v)\right) = \Theta(m)$$