2017/10/10 Tue.

問題解決技法入門

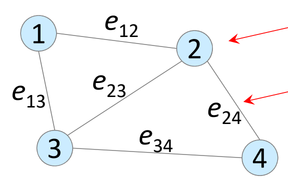
# 2. Graph Theory
# 1. グラフの基礎

堀田 敬介

## Graph

厳密には
$G=(f, V, E)$
$f: E \rightarrow V \times V$

- グラフ Graph　$G=(V,E)$
  - 点と枝，およびその接続関係



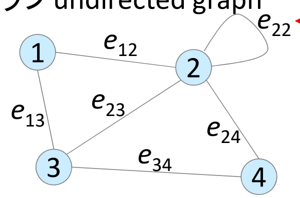

点，頂点 vertex, node

枝，辺，弧 edge, arc

  - 点集合 $V = \{1, 2, \ldots, n\}$
  - 枝集合 $E = \{e_{12}, e_{13}, e_{23}, \ldots\}$
  - 点1と点2は隣接している（A vertex 1 is adjacent to 2.）
  - 枝$e_{12}$は点1に接続している（An edge $e_{12}$ is incident to 1.）

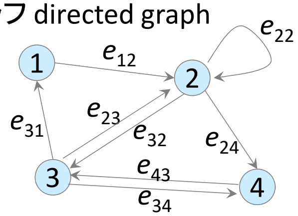
## Graph

- グラフ　$G=(V,E)$
  - 無向グラフ undirected graph



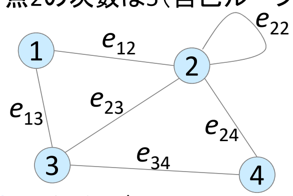

自己ループselfloop

  - 有向グラフ directed graph

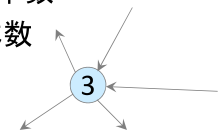



## Graph

- グラフ　$G=(V,E)$
  - 次数 degree …　点に接続している枝の本数
    - Ex) 点1の次数は2
    - Ex) 点2の次数は5（自己ループは2回カウント）

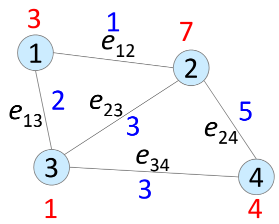



  - 入次数…有向グラフで入ってくる枝の本数
  - 出次数…有向グラフで出ていく枝の本数
    - Ex) 点3の次数5，入次数2，出次数3



## Graph

- グラフ $G$=($V$,$E$) のコスト
  - コスト cost
    - ラベル label
    - ポテンシャル potential
    - 重み weight
    - 流量 flow
    - 容量 capacity
    - 距離 distance
    - etc.

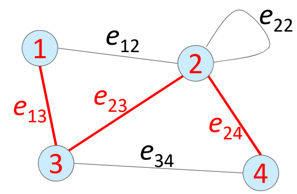



※点や枝に付随する数値(図の赤や青の数値)は,コストcostとよばれる.コストには,上記にあげたような様々な意味を持たせて利用する
※点や枝にこれらの数値が付随するグラフを特に「ネットワーク」とよんだ時代もあったが,別の意味で使われることが多い言葉なので,「コスト付きのグラフ」や単に「グラフ」でよいだろう

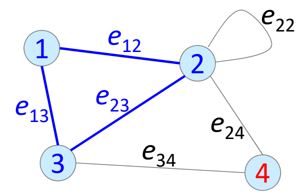
## Graph

もっと細かい定義…
- ✓ 初等的な路 elementary path
- ✓ 単純な路 simple path
- ✓ etc.

- グラフ $G$=($V$,$E$) の路と閉路
  - 路 path
    - Ex) 1, $e_{13}$, 3, $e_{23}$, 2, $e_{24}$ , 4
    - Ex) 1, 3, 2, 4
    - Ex) $e_{13}$, $e_{23}$, $e_{24}$

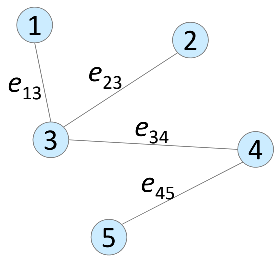



  - 閉路 cycle
    - Ex) 1, $e_{13}$, 3, $e_{23}$, 2, $e_{12}$ , 1
    - Ex) 1, 3, 2, 1
    - Ex) $e_{13}$, $e_{23}$, $e_{12}$

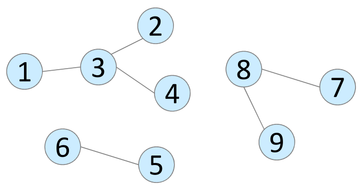



## Graph

- 様々なグラフ(1)
  - 木 tree
    - 連結で閉路を含まない無向グラフ




  - 森 forest
    - 閉路を含まない無向グラフ

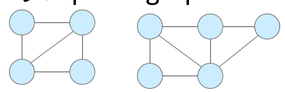


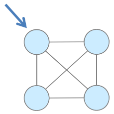

  - 連結成分 connected component

## Graph

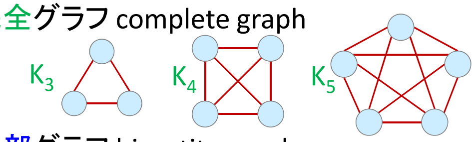
- 様々なグラフ(2)
  - 平面グラフ plane graph※

※平面グラフplane graph と同型なグラフを平面的グラフplanar graphといいます

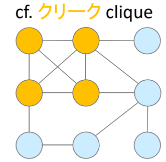




  - 完全グラフ complete graph




cf. クリーク clique



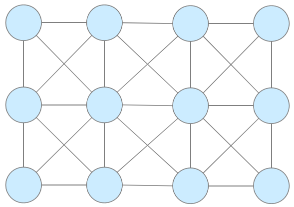
  - 二部グラフ bipartite graph


- ✓ 最大クリーク
- ✓ 友達の集合
- ✓ 点彩色・辺彩色

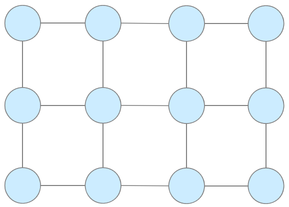
# 練習

- 問:これは何? 木? 平面? 完全? 二部?

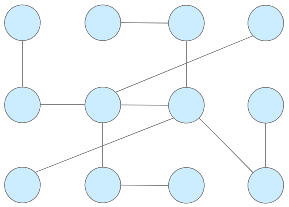
(1)



(2)



(3)

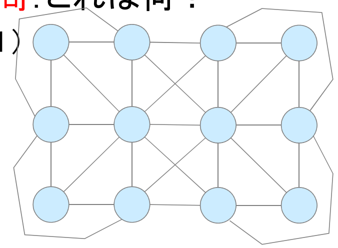


| | (1) | (2) | (3) |
|---|---|---|---|
| 木 | ? | ? | ? |
| 平面 | ? | ? | ? |
| 完全 | ? | ? | ? |
| 二部 | ? | ? | ? |

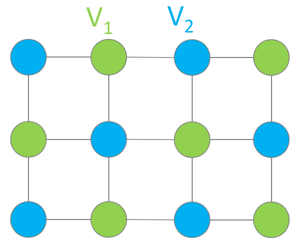
# 練習(解答)

- 問:これは何?

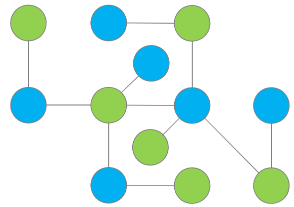
(1)



(2)




(3)

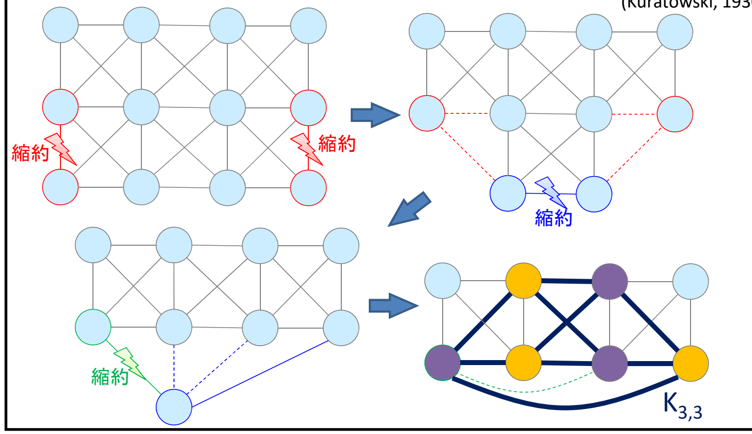


| | (1) | (2) | (3) |
|---|---|---|---|
| 木 | × | × | ○ |
| 平面 | × | ○ | ○ |
| 完全 | × | × | × |
| 二部 | × | ○ | ○ |

# 補足:平面グラフ(平面的グラフ)

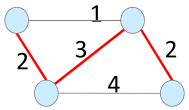
定理:グラフが平面に描画できるための必要十分条件は, $K_5$, $K_{3,3}$のどちらも位相的マイナーとしてもたないこと

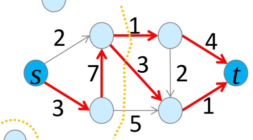
(Kuratowski, 1930)

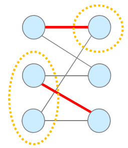



# 参考: Graph を使って何をする?

- 全域木 spanning tree
  - 最小全域木 minimum spanning tree
- フロー flow, カット cut
  - 最大流 maximum flow
  - 最小カット minimum cut
  - 最小費用流 minimum cost flow
- マッチ match, 被覆 cover
  - 最大マッチング maximum matching
  - 最小被覆 minimum covering



詳細は, 専門科目「ネットワークモデル分析」で学ぼう

- ✓ 補助ネットワーク auxiliary network, 残余ネットワーク residual network
- ✓ 増加道 augmenting path
- ✓ 最大フロー・最小カット定理
- ✓ 劣モジュラ関数 submodular function
- ✓ 最大マッチング・最小被覆定理
- ✓ ダルメジ-メンデルゾン分解 DM decomposition
- ✓ マトロイド matroid
- ✓ 幅優先探索 BFS, Breadth-First Search
- ✓ 深さ有線探索 DFS, Depth-First Search
- ✓ 連結性: $k$点連結, $k$枝連結
- ✓ 強連結 strongly connected, 半順序 partial order
- ✓ 安定集合 stable set, 独立集合 independent set

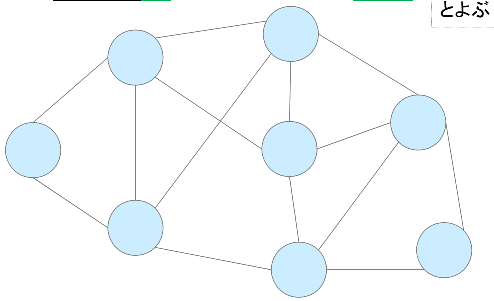
## 2種類の閉路

- オイラー閉路 Eulerian cycle
  - グラフの全ての枝を1度だけ通る閉路
- ハミルトン閉路 Hamiltonian cycle
  - グラフの全ての点を1度だけ通る閉路

注1)どの枝(点)から始めても構わない

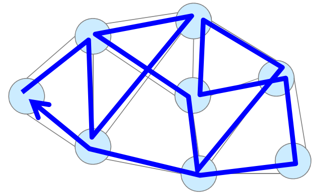
注2)スタート枝(点)とゴール枝(点)が異なる場合は，それぞれ，
- オイラー路(path)
- ハミルトン路(path)

とよぶ

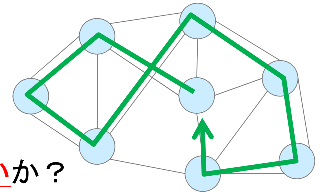


## 2種類の閉路（解答例）

- オイラー閉路 Eulerian cycle
  - グラフの全ての枝を1度だけ通る閉路



- ハミルトン閉路 Hamiltonian cycle
  - グラフの全ての点を1度だけ通る閉路



- 問：どっちの問題がより難しいか？

※ P = Polynomial　　× NP≠Not Polynomial
※ NP = Non-deterministic Polynomial

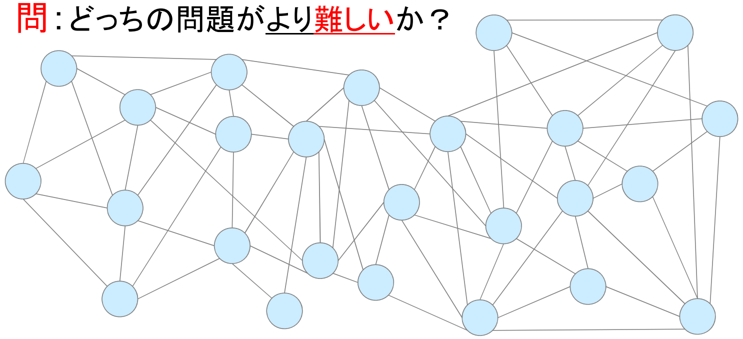
※与えられたグラフの
オイラー閉路を求める問題は，クラスPに属す（多項式時間で解ける polynomial-time solvable）
ハミルトン閉路を求める問題は，NP完全問題 NP complete problem

※NP完全問題とは，クラスNPに属し，かつ，NPの全ての問題から多項式時間帰着可能な問題
polynomial-time reducible

※「P≠NP予想」 未解決（7つの ミレニアム懸賞問題 millennium prize problems の1つ）

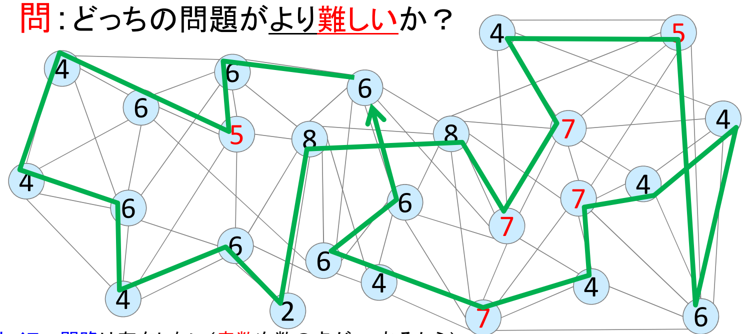
## 2種類の閉路

- オイラー閉路 Eulerian cycle
  - グラフの全ての枝を1度だけ通る閉路
- ハミルトン閉路 Hamiltonian cycle
  - グラフの全ての点を1度だけ通る閉路
- 問：どっちの問題がより難しいか？



## 2種類の閉路（解答例）

- オイラー閉路 Eulerian cycle
  - グラフの全ての枝を1度だけ通る閉路
- ハミルトン閉路 Hamiltonian cycle
  - グラフの全ての点を1度だけ通る閉路
- 問：どっちの問題がより難しいか？



オイラー閉路は存在しない（奇数次数の点が6つあるから）

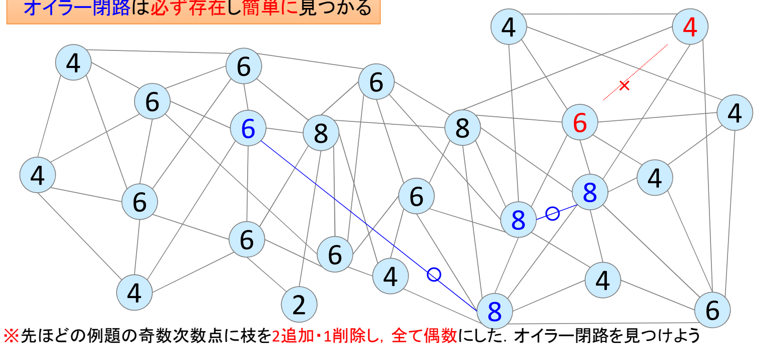
## 2種類の閉路（解説）

オイラー閉路は（存在する場合）
本当に簡単に見つけられる？

- オイラー閉路 Eulerian cycle
  - グラフの全ての枝を1度だけ通る閉路

奇数次数の点があれば(あるときは必ず偶数個)
オイラー閉路は存在しない
次数が全て偶数なら
オイラー閉路は必ず存在し簡単に見つかる

- ハミルトン閉路 Hamiltonian cycle
  - グラフの全ての点を1度だけ通る閉路

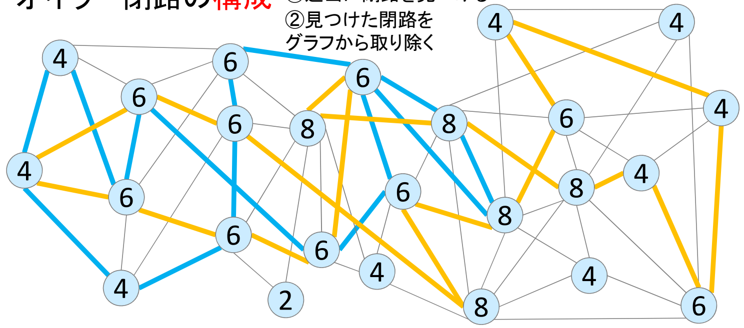


※先ほどの例題の奇数次数点に枝を2追加・1削除し，全て偶数にした．オイラー閉路を見つけよう

## 2種類の閉路（解説）

- オイラー閉路 Eulerian cycle
  - グラフの全ての枝を1度だけ通る閉路
- ハミルトン閉路 Hamiltonian cycle
  - グラフの全ての点を1度だけ通る閉路
- オイラー閉路の構成

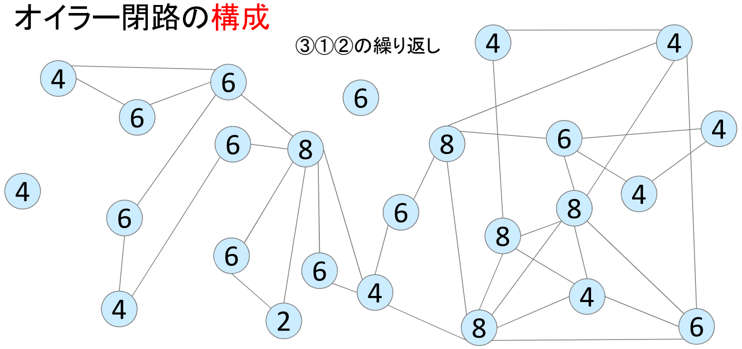
①適当に閉路を見つける
②見つけた閉路を
グラフから取り除く



## 2種類の閉路（解説）

- オイラー閉路 Eulerian cycle
  - グラフの全ての枝を1度だけ通る閉路
- ハミルトン閉路 Hamiltonian cycle
  - グラフの全ての点を1度だけ通る閉路
- オイラー閉路の構成

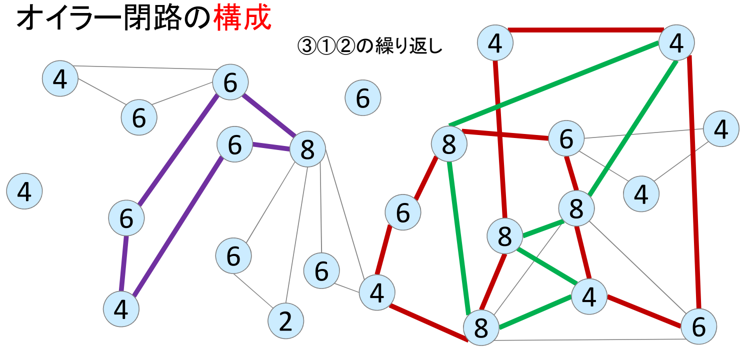
③①②の繰り返し



## 2種類の閉路（解説）

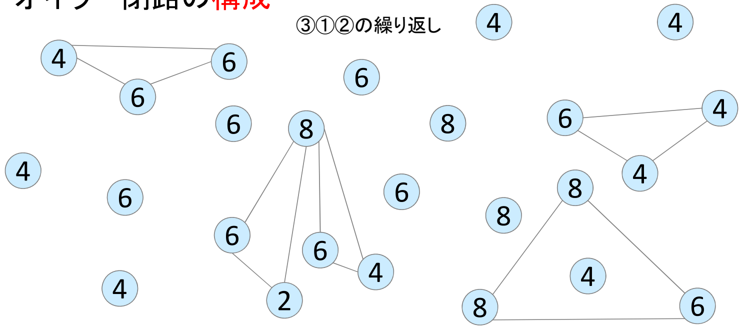
- オイラー閉路 Eulerian cycle
  - グラフの全ての枝を1度だけ通る閉路
- ハミルトン閉路 Hamiltonian cycle
  - グラフの全ての点を1度だけ通る閉路
- オイラー閉路の構成

③①②の繰り返し



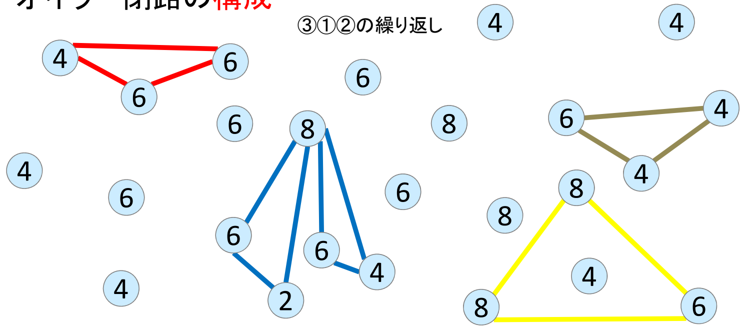
## 2種類の閉路（解説）

- オイラー閉路 Eulerian cycle
  - グラフの全ての枝を1度だけ通る閉路
- ハミルトン閉路 Hamiltonian cycle
  - グラフの全ての点を1度だけ通る閉路
- オイラー閉路の構成



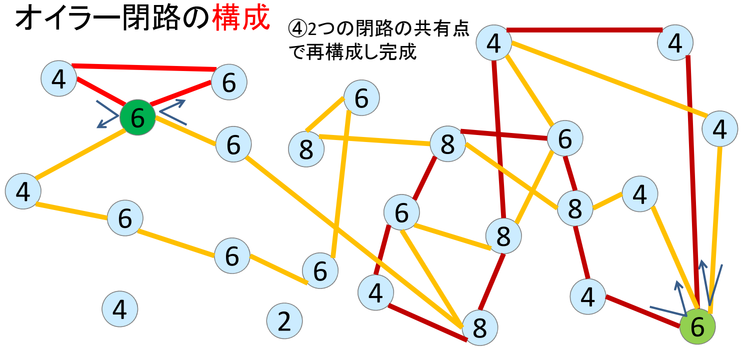

## 2種類の閉路（解説）

- オイラー閉路 Eulerian cycle
  - グラフの全ての枝を1度だけ通る閉路
- ハミルトン閉路 Hamiltonian cycle
  - グラフの全ての点を1度だけ通る閉路
- オイラー閉路の構成




## 2種類の閉路（解説）

- オイラー閉路 Eulerian cycle
  - グラフの全ての枝を1度だけ通る閉路
- ハミルトン閉路 Hamiltonian cycle
  - グラフの全ての点を1度だけ通る閉路
- オイラー閉路の構成




## 四色定理

【四色定理】
平面グラフは4彩色可能
（高々4色で塗り分けられる）

- 与えられたグラフは平面
- 隣接する地域を別の色で塗る
- 隣接とは辺で接していることであり，点で接する場合は除く

〔練習〕
横浜市（18区）を4色で塗り分けよう

区の境界線を辺とし，
辺が交わる箇所を点とする
グラフG=(V,E)を考え，4彩色せよ


地図出展： テクノコ白地図イラスト（http://technocco.jp/）

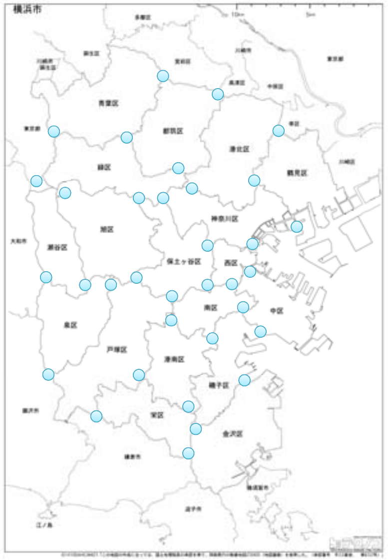
# 四色定理

【四色定理】
平面グラフは4彩色可能
（高々4色で塗り分けられる）

- 与えられたグラフは平面
- 隣接する地域を別の色で塗る
- 隣接とは辺で接していることであり，点で接する場合は除く

〔練習〕
横浜市（18区）を4色で塗り分けよう

区の境界線を辺とし，
辺が交わる箇所を点とする
グラフG=(V,E)を考え，4彩色せよ



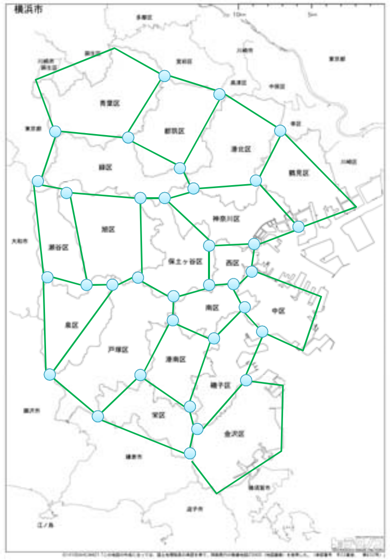
# 四色定理

【四色定理】
平面グラフは4彩色可能
（高々4色で塗り分けられる）

- 与えられたグラフは平面
- 隣接する地域を別の色で塗る
- 隣接とは辺で接していることであり，点で接する場合は除く

〔練習〕
横浜市（18区）を4色で塗り分けよう

区の境界線を辺とし，
辺が交わる箇所を点とする
グラフG=(V,E)を考え，4彩色せよ



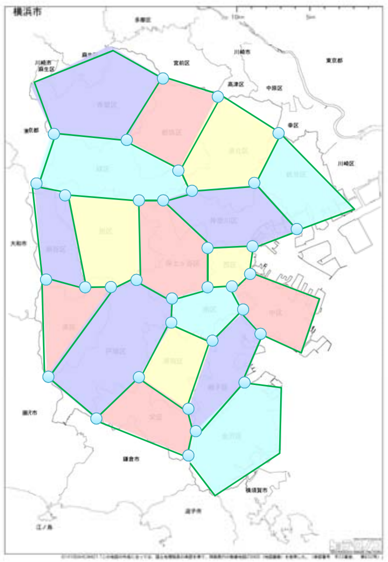
# 四色定理

【四色定理】
平面グラフは4彩色可能
（高々4色で塗り分けられる）

- 与えられたグラフは平面
- 隣接する地域を別の色で塗る
- 隣接とは辺で接していることであり，点で接する場合は除く

〔練習〕
横浜市（18区）を4色で塗り分けよう

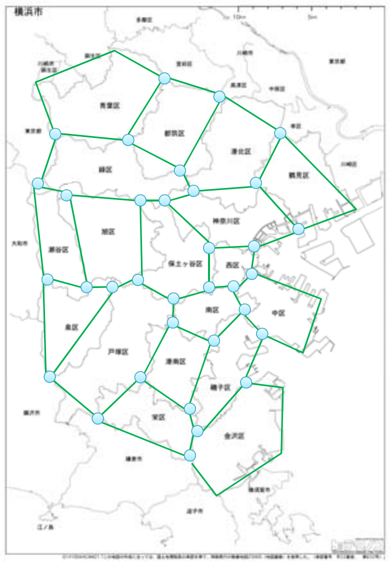
区の境界線を辺とし，
辺が交わる箇所を点とする
グラフG=(V,E)を考え，4彩色せよ



# ハミルトン閉路

【ハミルトン閉路】
全ての点を通る閉路

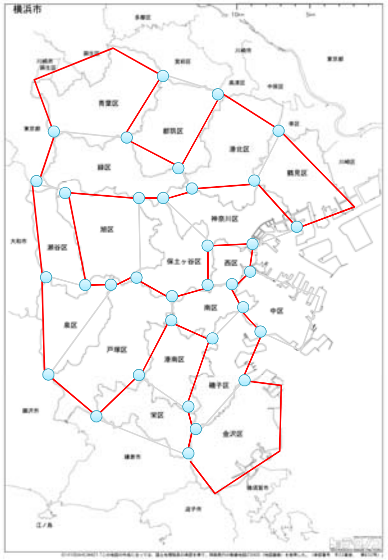
〔練習〕
横浜市（18区）のグラフについて，
ハミルトン閉路が存在するなら，
それを求めよ



# ハミルトン閉路

【ハミルトン閉路】
全ての点を通る閉路

〔練習〕
横浜市（18区）のグラフについて，
ハミルトン閉路が存在するなら，
それを求めよ

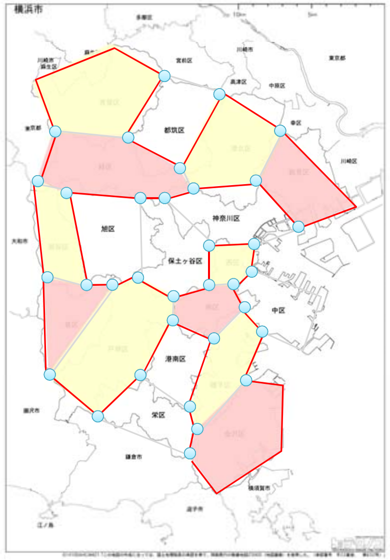


# 四色定理 と ハミルトン閉路

【四色定理】
平面グラフは4彩色可能
（高々4色で塗り分けられる）

【ハミルトン閉路】
全ての点を通る閉路

平面グラフでハミルトン閉路が存在すれば，閉路の内側と外側が出来る．内側を2色交互に，外側を2色交互に塗れば4彩色ができる

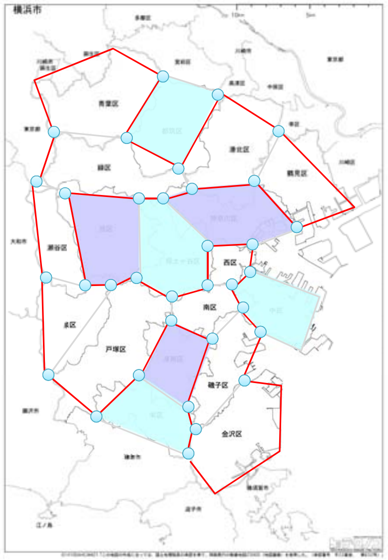


# 四色定理 と ハミルトン閉路

【四色定理】
平面グラフは4彩色可能
（高々4色で塗り分けられる）

【ハミルトン閉路】
全ての点を通る閉路

平面グラフでハミルトン閉路が存在すれば，閉路の内側と外側が出来る．内側を2色交互に，外側を2色交互に塗れば4彩色ができる

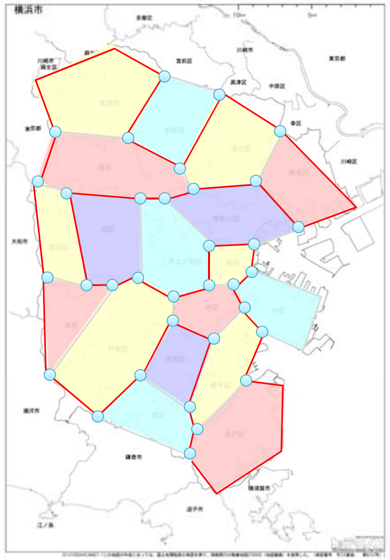


# 四色定理 と ハミルトン閉路

【四色定理】
平面グラフは4彩色可能
（高々4色で塗り分けられる）

【ハミルトン閉路】
全ての点を通る閉路

平面グラフでハミルトン閉路が存在すれば，閉路の内側と外側が出来る．内側を2色交互に，外側を2色交互に塗れば4彩色ができる



# 参考文献


- D.Jungnickel, ``*Graph, Networks and Algorithms*'', Springer (2002)
- J.Gross & J.Yellen, ``*Graph Theory and Its Applications*'', CRC Press (1999)
- 伊里正夫・藤重悟・大山達雄「グラフ・ネットワーク・マトロイド」産業図書 (1986)
- 藤重悟「グラフ・ネットワーク・組合せ論」共立出版 (2002)
- 茨木俊秀・永持仁・石井利昌「グラフ理論」朝倉書店 (2010)
- 地図出展：テクノコ白地図イラスト（http://technocco.jp/）2015.5.4


もっと知りたい人は
関連する授業をとろう！

- ✓「ネットワークモデル分析」
- ✓「最適化モデル分析」
- ✓ etc.

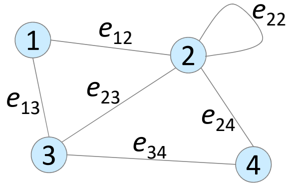
---

補足：コンピュータ上等で処理するため

# Graph

注）どんなグラフも表現出来るわけではない
- ✓ 多重辺
- ✓ 自己ループ
- ✓ etc.

- グラフ　$G=(V,E)$ の行列表現




有向グラフの場合はどうなるか考えてみよう
- ➢ 枝が出る→−1
- ➢ 枝が入る→+1

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 0 & 1 \\ 4 & 0 & 1 & 1 & 0 \end{array}$$

**隣接**行列
adjacency matrix

$$\begin{array}{c|cccccc} & e_{12} & e_{13} & e_{22} & e_{23} & e_{24} & e_{34} \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 1 & 1 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 & 1 \\ 4 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

**接続**行列
incidence matrix

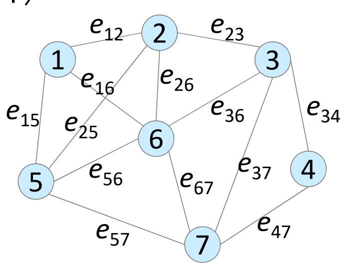
---

補足

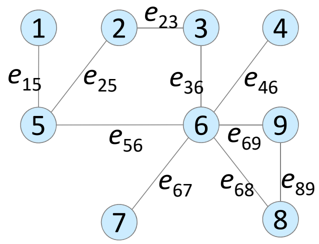
# 練習1

- **問**：次のグラフG=(V,E)の点集合V と枝集合E を示せ．
  また，グラフを接続行列と隣接行列で表せ．
  さらに，各点の次数を求めよ

（1）




（2）

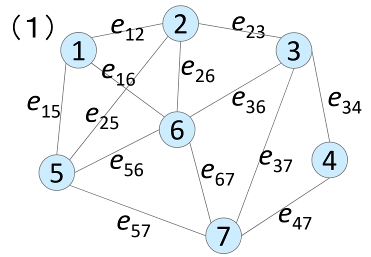



---

補足

# 練習1（解答）

（1）




隣接行列

$$\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 4 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 5 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 6 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 7 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array}$$

接続行列

点集合V = {1,2,3,4,5,6,7}
枝集合E = {$e_{12}$, $e_{15}$, $e_{16}$, $e_{23}$, $e_{25}$, $e_{26}$, $e_{34}$, $e_{36}$, $e_{37}$, $e_{47}$, $e_{56}$, $e_{57}$, $e_{67}$}

各点の次数 degree
点1(3), 点2(4), 点3(4), 点4(2),
点5(4), 点6(5), 点7(4)

$$\begin{array}{c|ccccccccccccc} & e_{12} & e_{15} & e_{16} & e_{23} & e_{25} & e_{26} & e_{34} & e_{36} & e_{37} & e_{47} & e_{56} & e_{57} & e_{67} \\ \hline 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 5 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 6 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}$$

補足

# 練習2

- 問：隣接行列・接続行列で表されたグラフG=(V,E)をそれぞれ図示せよ

（1）隣接行列

$$\begin{array}{c|c} & \begin{matrix}1 & 2 & 3 & 4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{pmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 1 & 1\\ 1 & 1 & 0 & 1\\ 1 & 1 & 1 & 0\end{pmatrix}\end{array}$$

（2）接続行列

$$\begin{array}{c|c} & \begin{matrix}e_{12} & e_{13} & e_{14} & e_{23} & e_{24} & e_{34}\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{pmatrix}1 & 1 & 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 1 & 1 & 0\\ 0 & 1 & 0 & 1 & 0 & 1\\ 0 & 0 & 1 & 0 & 1 & 1\end{pmatrix}\end{array}$$

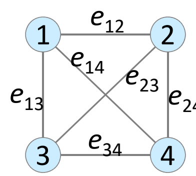
補足

# 練習2（解答）

- 問：隣接行列・接続行列で表されたグラフG=(V,E)をそれぞれ図示せよ

（1）隣接行列

$$\begin{array}{c|c} & \begin{matrix}1 & 2 & 3 & 4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{pmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 1 & 1\\ 1 & 1 & 0 & 1\\ 1 & 1 & 1 & 0\end{pmatrix}\end{array}$$




（2）接続行列

$$\begin{array}{c|c} & \begin{matrix}e_{12} & e_{13} & e_{14} & e_{23} & e_{24} & e_{34}\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{pmatrix}1 & 1 & 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 1 & 1 & 0\\ 0 & 1 & 0 & 1 & 0 & 1\\ 0 & 0 & 1 & 0 & 1 & 1\end{pmatrix}\end{array}$$