創造科学の世界B

１０．計算機科学最大の問題
（グラフ理論とアルゴリズム）

2008/6/13(金）

草苅 良至

100万ドルの問題を教えます。

問題に懸賞金が付いてます。
問題を解けば、
１００万ドルが手に入ります。
関連のサイト（英語）

http://www.claymath.org/millennium/

「P　vs　NP　問題」

Clay Mathematics Institute - Mozilla Firefox

ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

Clay Mathematics Institute

Dedicated to increasing and disseminating mathematical knowledge

HOME | ABOUT CMI | PROGRAMS | NEWS & EVENTS | AWARDS | SCHOLARS | PUBLICATIONS

P vs NP Problem

Suppose that you are organizing housing accommodations for a group of four hundred university students. Space is limited and only one hundred of the students will receive places in the dormitory. To complicate matters, the Dean has provided you with a list of pairs of incompatible students, and requested that no pair from this list appear in your final choice. This is an example of what computer scientists call an NP-problem, since it is easy to check if a given choice of one hundred students proposed by a coworker is satisfactory (i.e., no pair taken from your coworker's list also appears on the list from the Dean's office), however the task of generating such a list from scratch seems to be so hard as to be completely impractical. Indeed, the total number of ways of choosing one hundred students from the four hundred applicants is greater than the number of atoms in the known universe! Thus no future civilization could ever hope to build a supercomputer capable of solving the problem by brute force; that is, by checking every possible combination of 100 students. However, this apparent difficulty may only reflect the lack of ingenuity of your programmer. In fact, one of the outstanding problems in computer science is determining whether questions exist whose answer can be quickly checked, but which require an impossibly long time to solve by any direct procedure. Problems like the one listed above certainly seem to be of this kind, but so far no one has managed to prove that any of them really are so hard as they appear, i.e., that there really is no feasible way to generate an answer with the help of a computer. Stephen Cook and Leonid Levin formulated the P (i.e., easy to find) versus NP (i.e., easy to check) problem independently in 1971.

- The Millennium Problems
- Official Problem Description — Stephen Cook
- Lecture by Vijaya Ramachandran at University of Texas (video)
- Minesweeper

Return to top

目次

I: グラフ理論入門

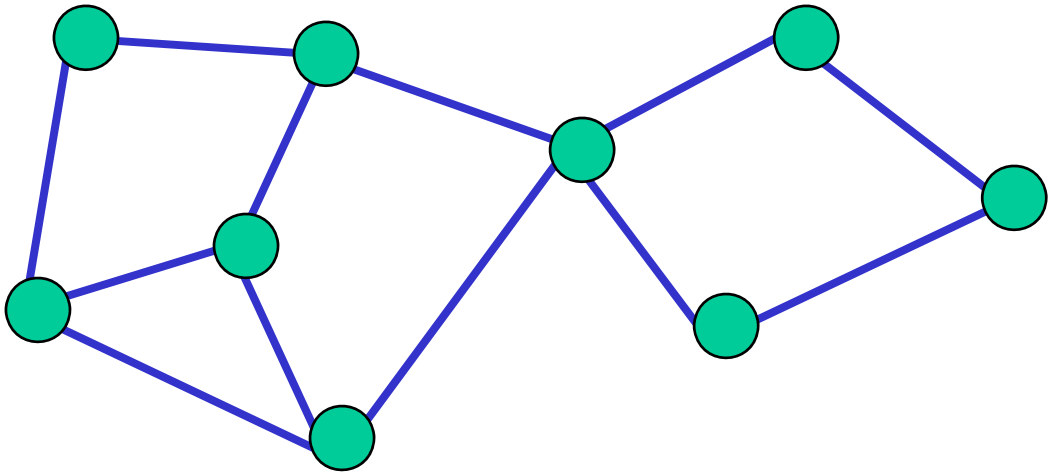
1. グラフとは？

いくつかの点集合と、
それらを結ぶいくつかの
辺から構成される図形

いくつかの点

2点間を結ぶ辺

グラフの例

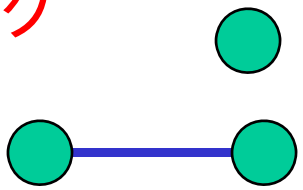


2. なんでグラフとか考えるの？

グラフ:いろいろなものを単純に抽象化して
考えることができる。

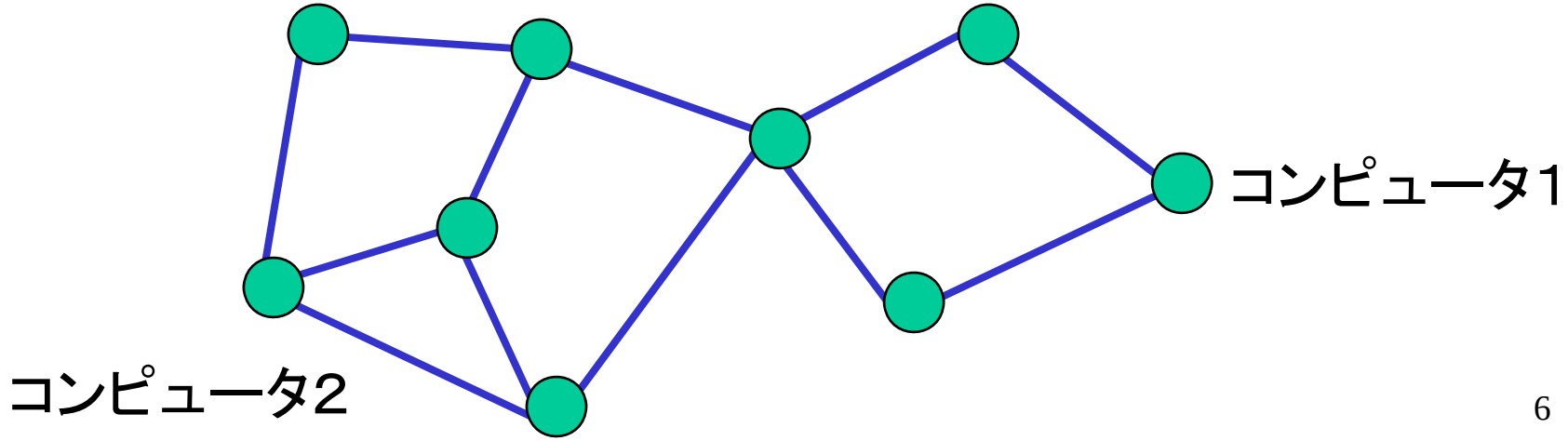
現実世界とグラフ

（1）ネットワーク

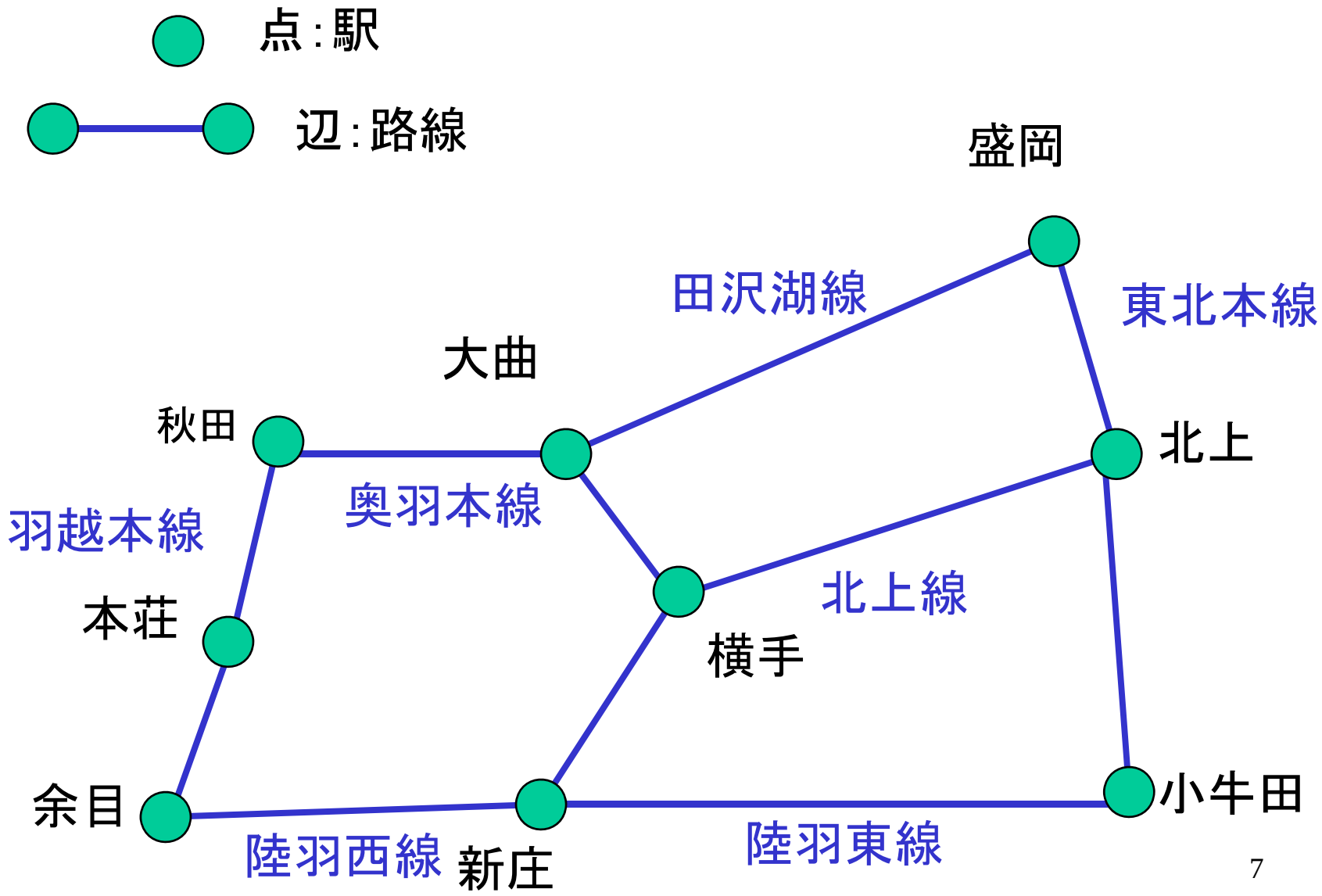


点：コンピュータ

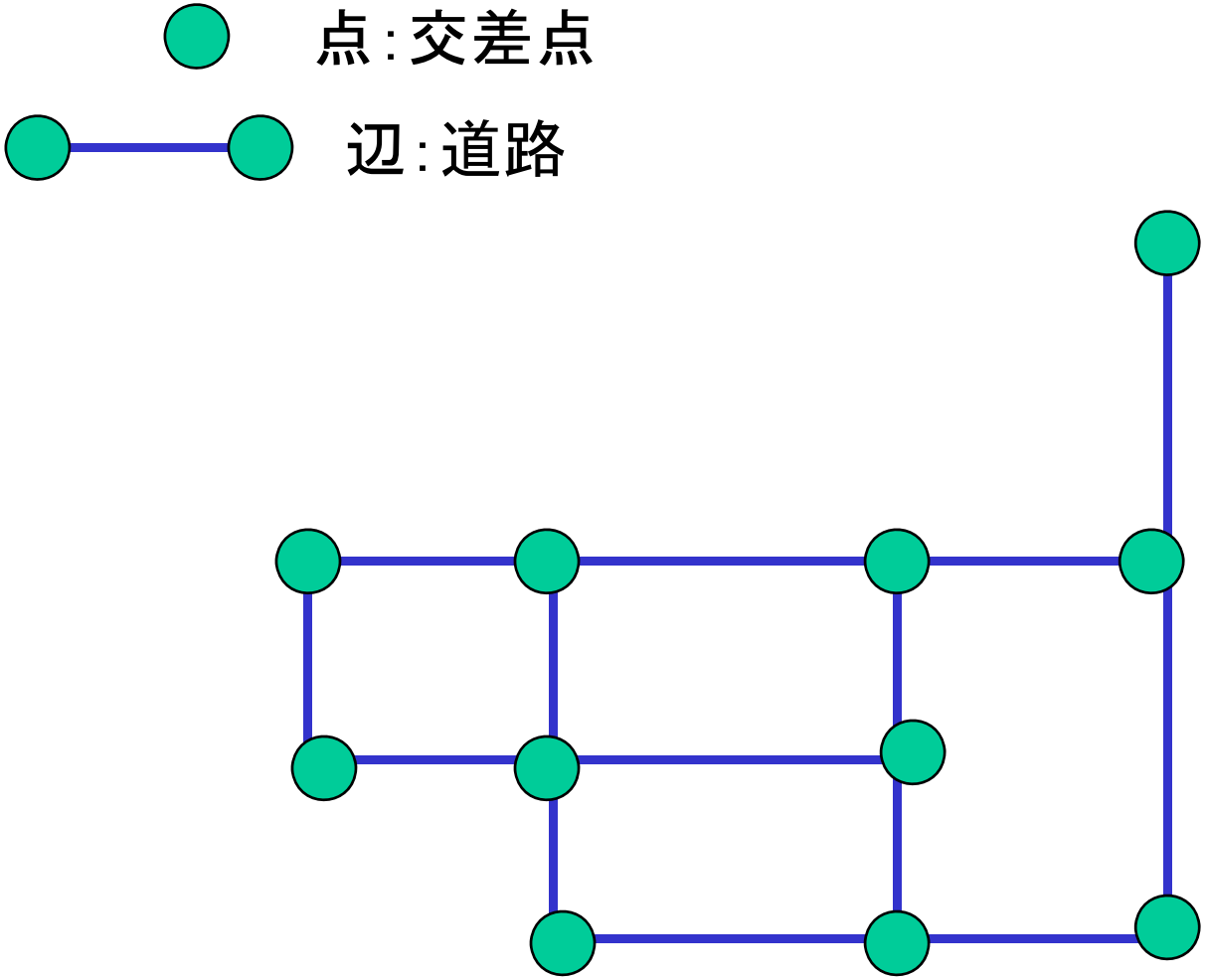
辺：配線



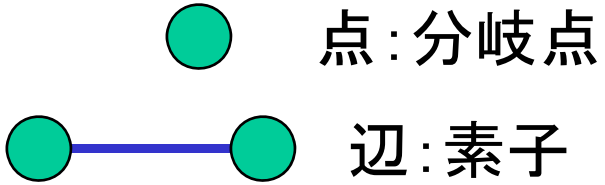
（2）路線図


点：駅
辺：路線
盛岡
田沢湖線
東北本線
大曲
秋田
北上
奥羽本線
羽越本線
北上線
本荘
横手
余目
小牛田
陸羽西線
新庄
陸羽東線

（3）道路地図

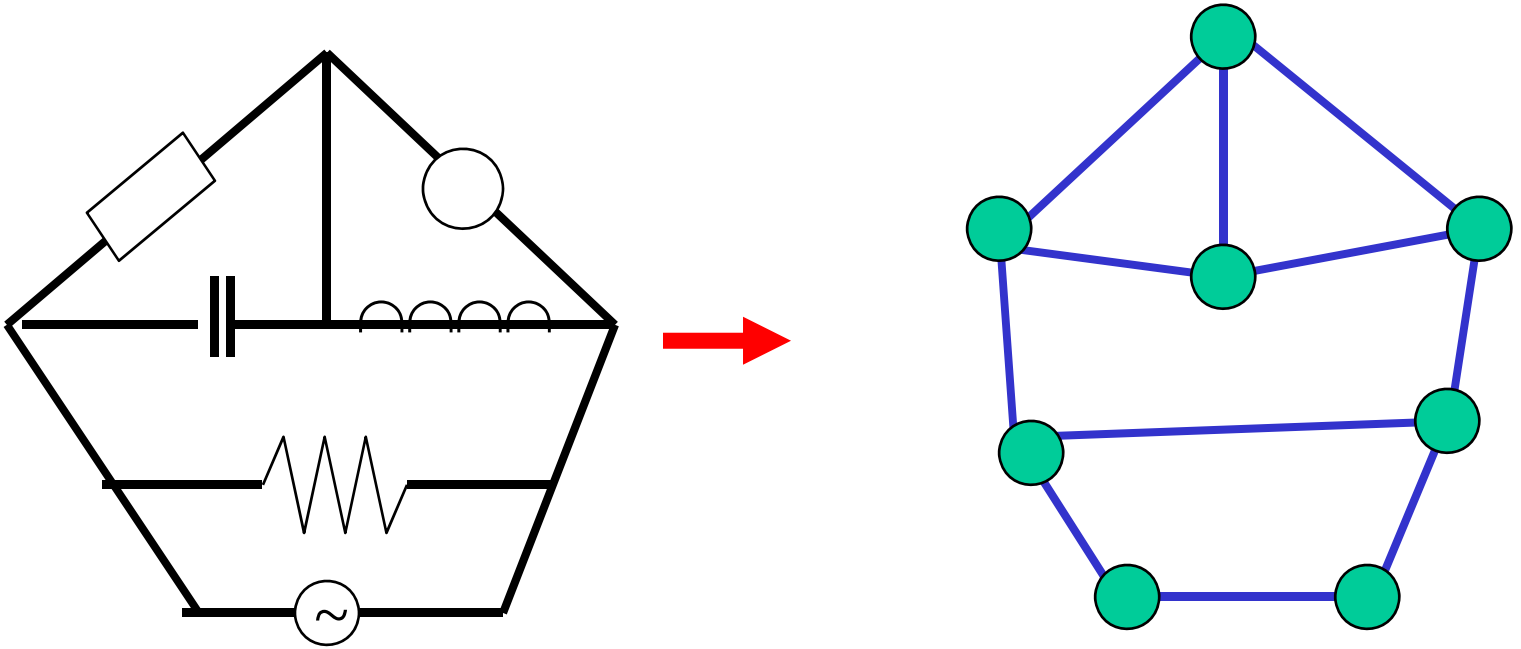


(4)回路図

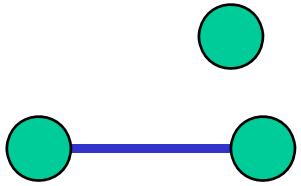


点:分岐点

辺:素子

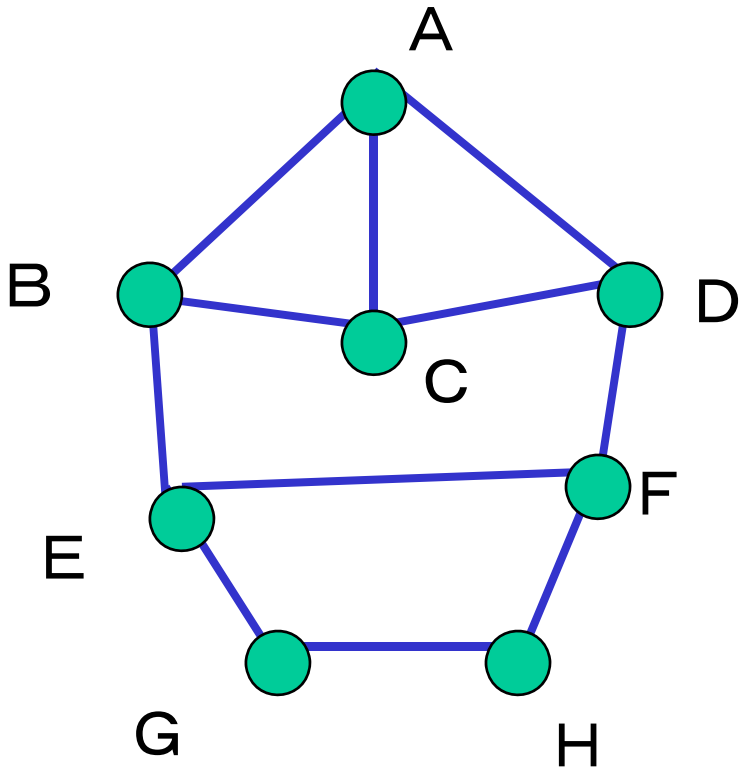


（5）友達関係図



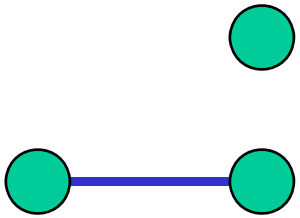
点：人間

辺：友達関係



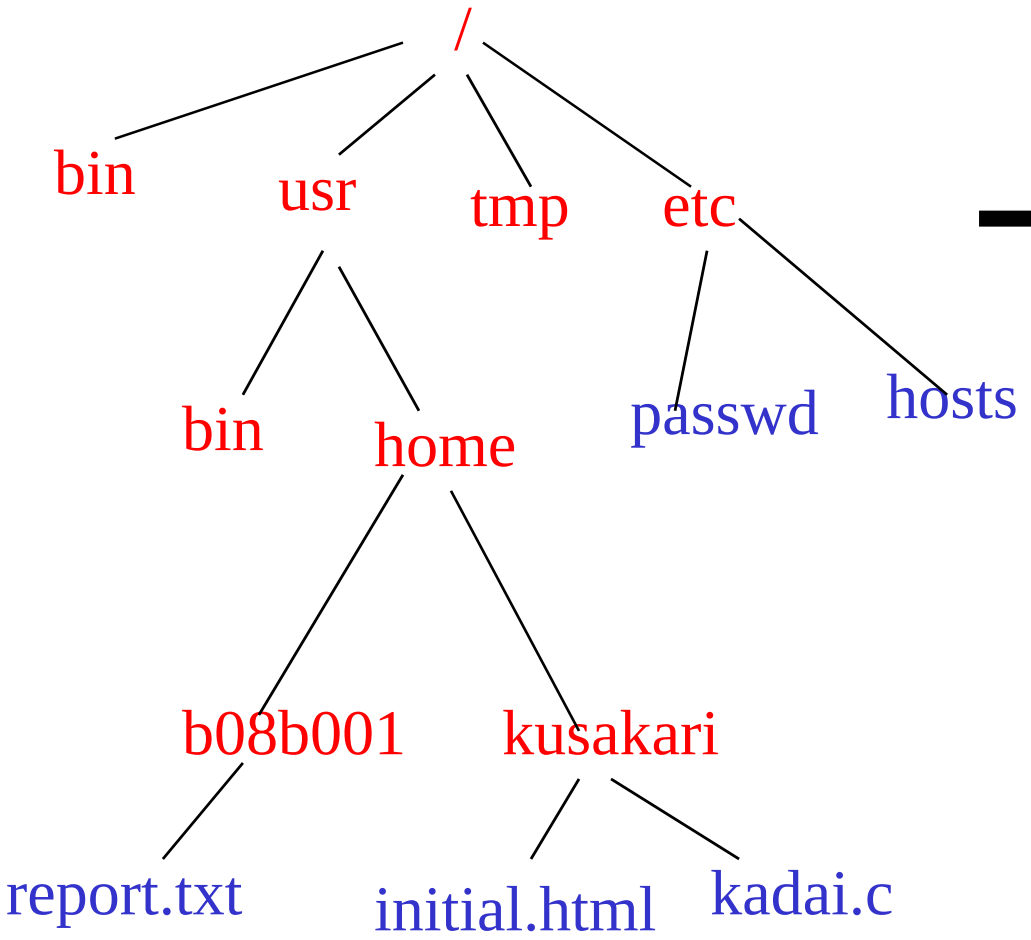
A君とB君は友達だけど、
A君とE君は友達でない。

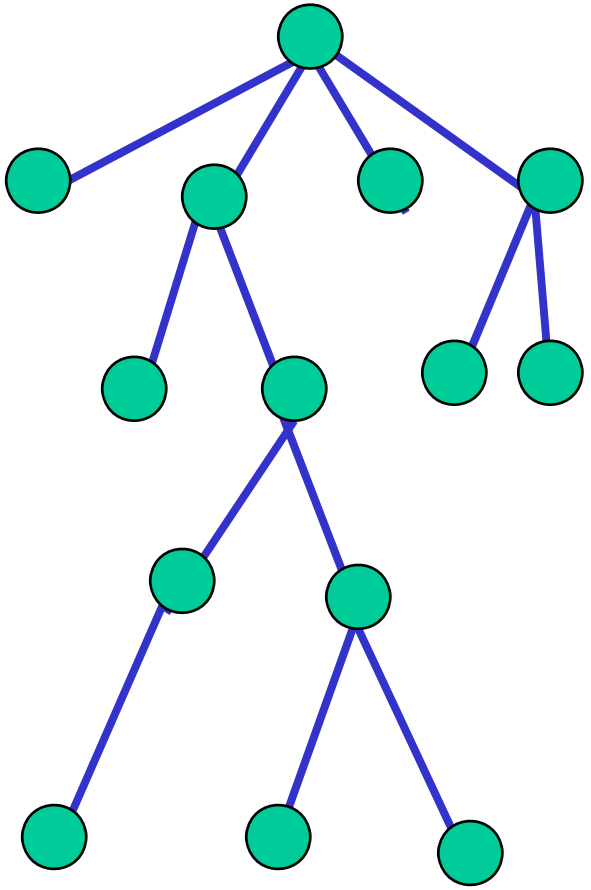
（6）UNIXのファイル構造



点：ディレクトリまたはファイル

辺：包含関係、位置関係





3. グラフの仲間

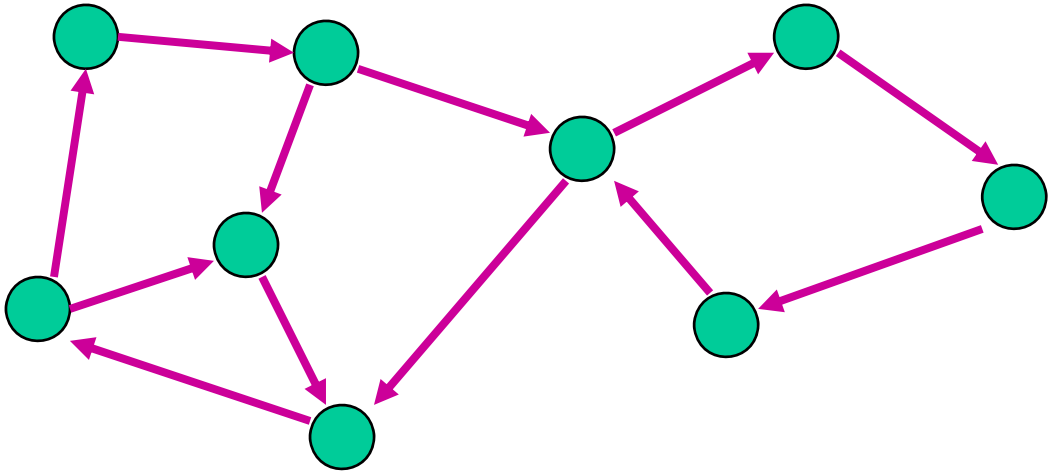
有向グラフ

いくつかの点

2つの点を結ぶ
いくつかの有向辺

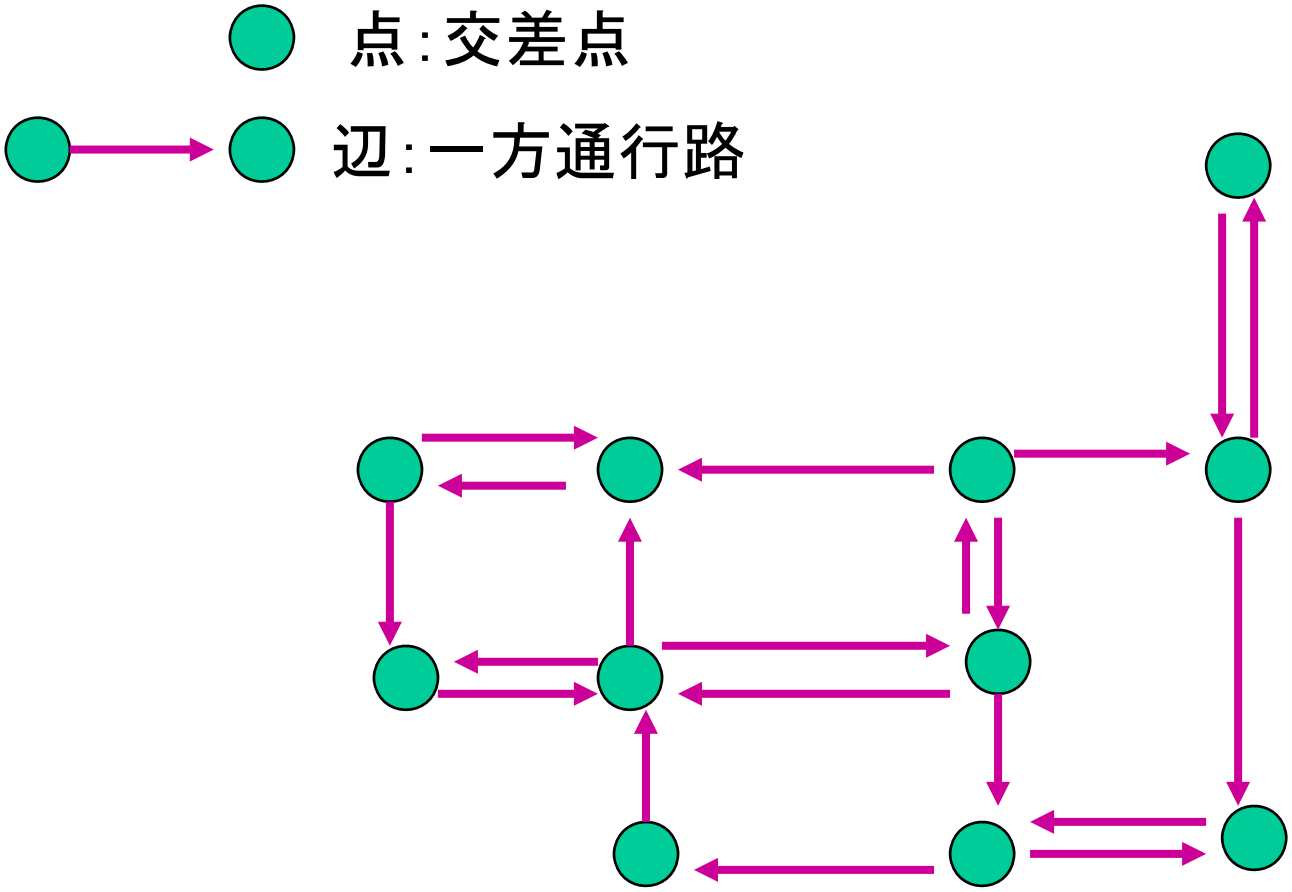
今までのグラフ
は無向グラフ

有向グラフの例

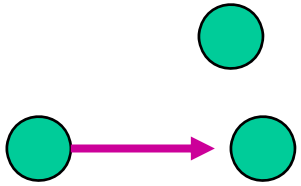


現実世界と有向グラフ

(1)一方通行のある道路地図

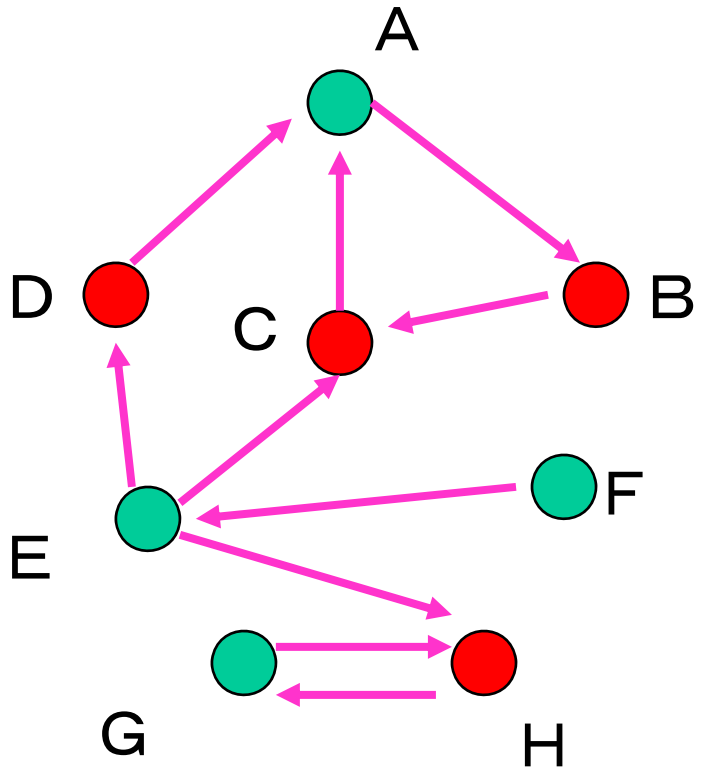


（2）恋愛感情の図



点：人間

辺：恋愛感情



A君はBさんを好きだけど、
BさんはCさんが好き。

グラフ理論で考えること、

- 平面上に書けるグラフってどんなのだろう？
  （グラフの平面性判定といいます。
  VLSI基板上の配線などに応用されます。）

- 隣接している国同士は違う色で塗るとしたら、
  何色必要なの？
  （彩色問題といいます。中でも4色問題が有名です。
  工程管理などに応用されます。）

- ネットワークで、
  途中のコンピュータが何台まで故障しても、
  相手のコンピュータと通信できるの？
  （グラフの連結度の問題といいます。
  ネットワーク設計、解析などに応用されます。）

II：アルゴリズム論入門

1．アルゴリズムとは？

アルゴリズム：問題をとくための、（有限回の機械的な）手順。

アルゴリズムの性質

あいまい性が無い。

有限のステップで終了する。

2. アルゴリズム例

例1：10進数から2進数への変換

入力 $D = (d_{n-1}d_{n-2}\cdots d_1 d_0)_{10}$

出力 $B = (b_{m-1}b_{m-1}\cdots b_1 b_0)_2$

人間が理解しやすい数

$(19)_{10} \Leftrightarrow (10011)_2$

$(1989)_{10} \Leftrightarrow (11111000101)_2$

$(2007)_{10} \Leftrightarrow (11111010111)_2$

コンピュータが理解しやすい数

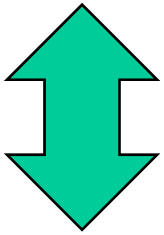
10進数と2進数

$$d_i \in \{0,1,2,3,4,5,6,7,8,9\}$$

$$D = (d_{n-1}d_{n-2}\cdots d_1 d_0)_{10}$$

$$= d_{n-1}\times 10^{n-1} + \cdots + d_1 \times 10^1 + d_0 \times 10^0$$

各桁の数字は、その位の値が何個あるのかを示している。



$$b_j \in \{0,1\}$$

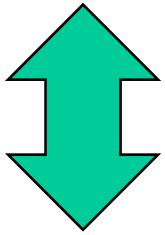
$$B = (b_{m-1}b_{m-2}\cdots b_1 b_0)_2$$

$$= b_{m-1}\times 2^{m-1} + \cdots + b_1 \times 2^1 + b_0 \times 2^0$$

各桁の数字は、その位の値が何個あるのかを示している。

$$(19)_{10} = 1 \times 10^1 + 9 \times 10^0$$

$$= 10 + 9$$



$$16 + 2 + 1$$

$$= 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$= (10011)_2$$

2進数変換アルゴリズム

[step1]: $D_0 := D, i = 0$ とする。

初期設定

[step2]: $D_i > 0$ の間以下を繰り返す;

(2-1) $b_i := D_i \bmod 2$

2で割った余り

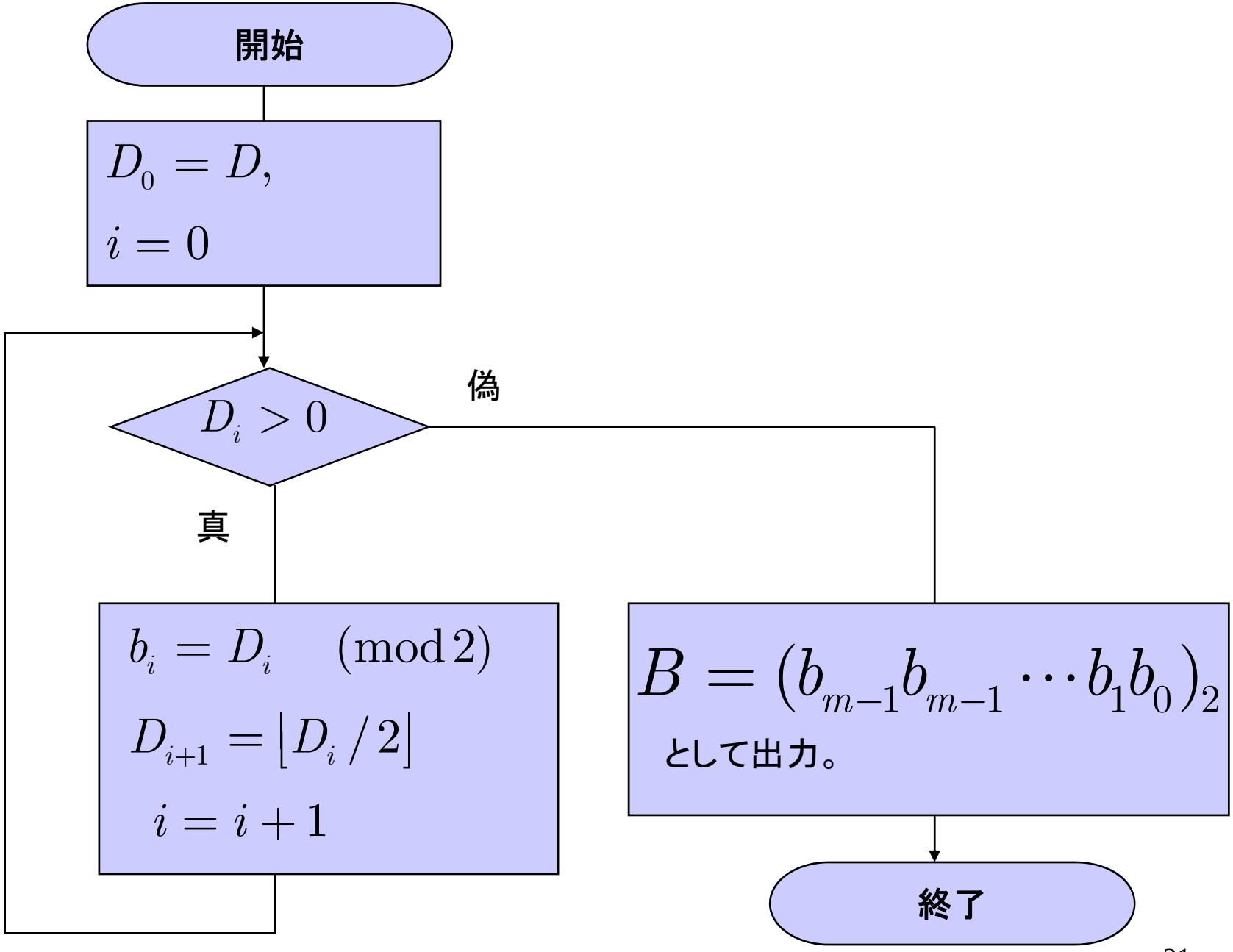
(2-2) $D_{i+1} := \lfloor D_i / 2 \rfloor$

2で割った商（切り捨て）

(2-3) $i := i + 1$

i を1増加させる。

[step3]: $B = (b_{m-1} \cdots b_1 b_0)_2$ を出力して終了する。


開始
$D_0 = D,$
$i = 0$
$D_i > 0$
偽
真
$b_i = D_i \pmod 2$
$D_{i+1} = \lfloor D_i / 2 \rfloor$
$i = i + 1$
$B = (b_{m-1} b_{m-1} \cdots b_1 b_0)_2$
として出力。
終了

実行例

19の2進数は？

19割2は、
商が9で余りが1.

$$b_0 := 19 \bmod 2 = 1$$

$$D_1 := \lfloor 19/2 \rfloor = 9$$

19/2 =9 ;1 下位ビット
9/2 = 4 ;1
4/2 = 2 ;0
2/2 = 1 ;0
1/2 = 0 ;1 上位ビット

$$(19)_{10} = (10011)_2$$

アルゴリズムの正当性
（なぜ、正しく2進数にしてくれるのか？）

$$D = D_0 = b_{m-1} \times 2^{m-1} + \cdots + b_1 \times 2^1 + b_0 \times 2^0$$

$$= \underbrace{\left(\underbrace{\left(\underbrace{\left(b_{m-1} \right)}_{D_{m-1}} \times 2 + b_{m-2} \right)}_{D_{m-2}} \times 2 + \cdots \right) \times 2 + b_1}_{D_1} \times 2 + b_0}_{D_0}$$

$$= D_1 \times 2 + b_0$$

除算における商
と余りの関係式

$$(19)_{10} = (9)_{10} \times 2 + 1$$

奇数なので1余る

$$= \big((4)_{10} \times 2 + 1\big) \times 2 + 1$$

$$= \Big(\big((2)_{10} \times 2 + 0\big) + 1\Big) \times 2 + 1$$

$$= \Big(\big(\big((1)_{10} \times 2 + 0\big) \times 2 + 0\big) \times 2 + 1\Big) \times 2 + 1$$

$$= \Big(\big(\big((1)_{2} \times 2 + 0\big) \times 2 + 0\big) \times 2 + 1\Big) \times 2 + 1$$

$$= \big((10)_{2} \times 2 + 0\big) \times 2 + 1\big) \times 2 + 1$$

$$= \big((100)_{2} \times 2 + 1\big) \times 2 + 1$$

$$= (1001)_{2} \times 2 + 1 = (10011)_{2}$$

例2：最大公約数を求めるアルゴリズム

ユークリッドの互除法

[step1]: $P = A; Q = B$ とする。

初期設定

[step2]: $R = P \bmod Q$

Qで割った余り

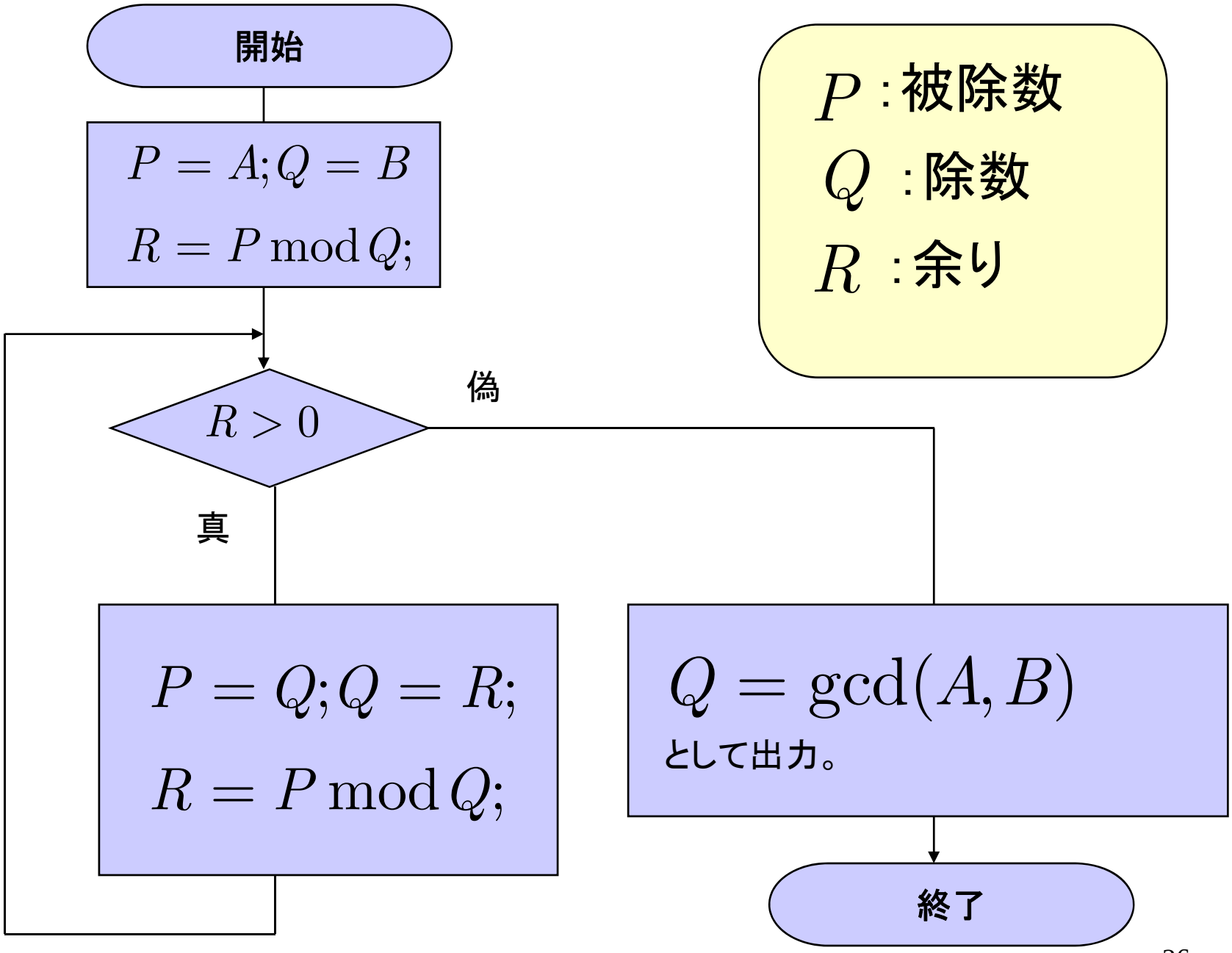
[step3]: $R > 0$ の間以下を繰り返す;

(3-1) $P = Q; Q = R$ とする。

(3-2) $R = P \bmod Q$ とする。

[step3]: $Q = \gcd(A, B)$ として出力する。

P：被除数　　Q：除数　　R：余り


開始
$P = A; Q = B$
$R = P \bmod Q;$
P : 被除数
Q : 除数
R : 余り
$R > 0$
偽
真
$P = Q; Q = R;$
$R = P \bmod Q;$
$Q = \gcd(A, B)$
として出力。
終了

実行例

$$A = 36, B = 21$$

$$P \qquad Q \qquad R$$

$$36 = \quad 21 \times 1 \quad + 15$$

$$21 = \quad 15 \times 1 \quad + 6$$

$$15 = \quad 6 \times 1 \quad + 3$$

$$6 = \quad 3 \times 2 \quad + 0$$

割り切れたときのQが最大公約数

$$\therefore \gcd(36, 21) = 3$$

3. アルゴリズムとプログラム

アルゴリズム

＋

データ構造

プログラミング

プログラム言語

プログラム

2進数変換アルゴリズム

＋

整数型データ

C言語風プログラム
intは整数のデータ型。

```
int * bin(int D){
        int B[100];
        Di=D;i=0;
        while(Di>0){
              B[i]=Di (mod) 2;
              Di=Di/2;
               i=i+1;
          };
          return B;
}
```

４．アルゴリズムの評価

主に、主要な繰り返し回数を考えます。

（１）の１０進数を２進数に変換するアルゴリズムでは、

変換される２進数のビット数回の繰り返しがおきます。

１０進数を n とすると、
プログラムにした場合 $\log_2 n$ に比例する時間で終了する。

（２）ユークリッドの互除法では、

小さい数 n の対数回 $\log_2 n$ ぐらい の繰り返しがおきます。
（なんでかはちょっと考えてみましょう。）
プログラムにした場合 $\log_2 n$ に比例する時間で終了する。

アルゴリズム論で考える事

- いろんな問題をコンピュータに解かせる場合に、
  どんな手順で処理をすすめたらいいのか考える。
  （アルゴリズムの設計といいます。）

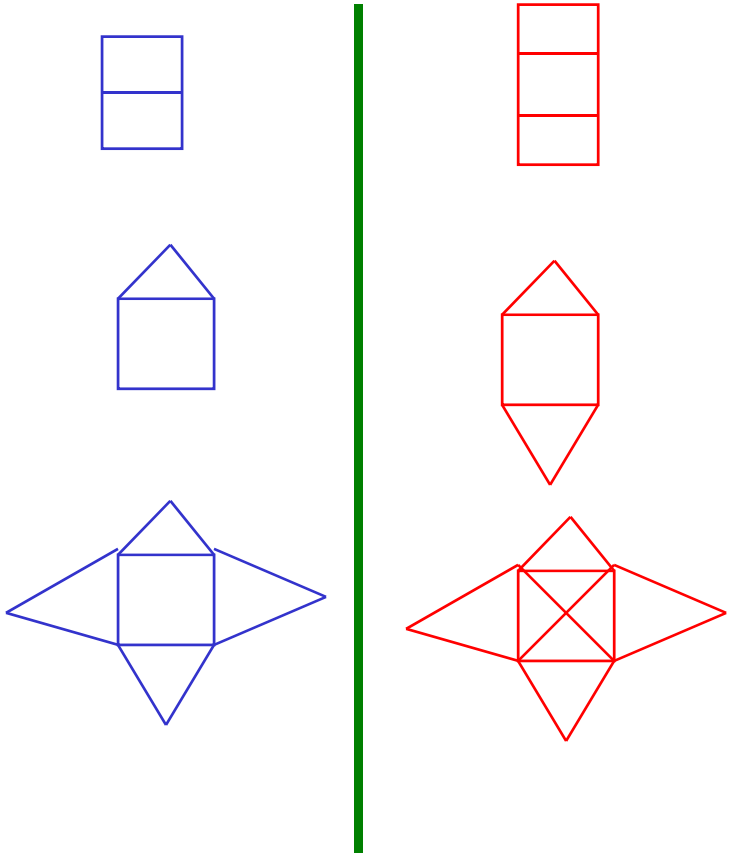
- アルゴリズムをプログラムに直した場合に、
  どのくらいの時間で終了するのかを考える。
  （アルゴリズムの解析といいます。）

- そもそも、計算機で解ける問題とは、
  どのような問題なのかを考える。
  つまり、アルゴリズムがあるのかどうかを考える。
  （計算量理論といいます。）

III:グラフ理論とアルゴリズム

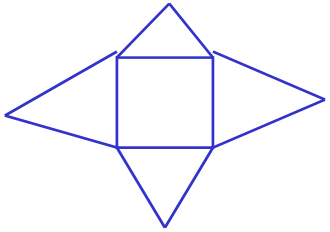
さて、グラフ上の問題を解くアルゴリズムを考えてみましょ。

1. オイラー回路問題



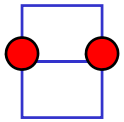
グラフが与えられたとき、そのグラフが一筆書きできるの？

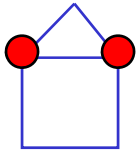
オイラー回路とは？



始点と終点が
等しい。

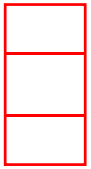
オイラー回路

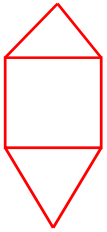


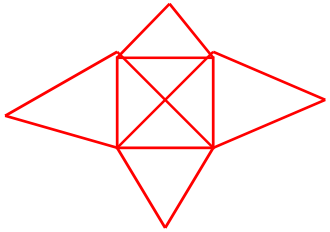


始点、終点
が異なる

一筆がきできる。

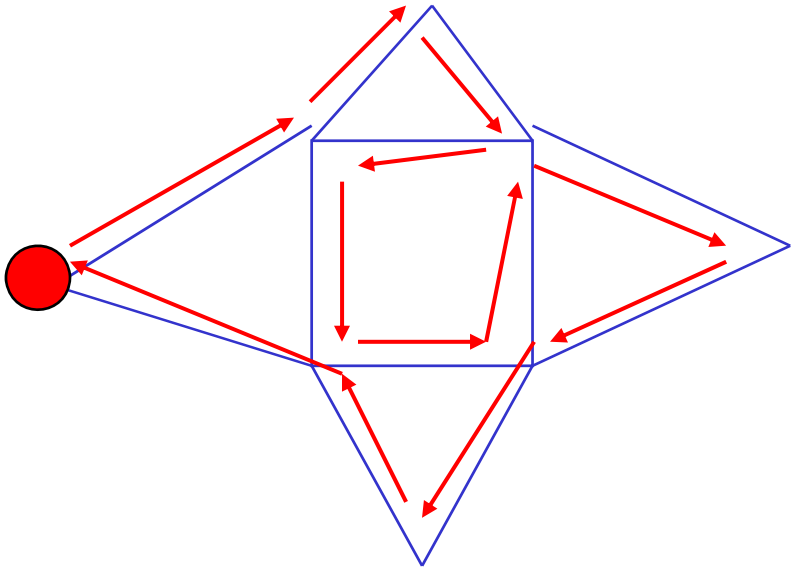






一筆がきできない。

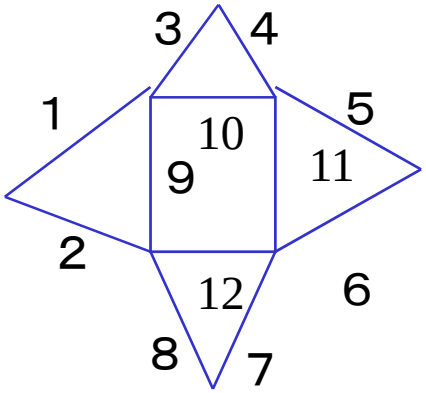
オイラー回路の例



オイラー回路問題

グラフGが与えられたとき、そのグラフにオイラー回路があるか判別せよ。

つまり、グラフGは、任意の点を始点として、
一筆がきできるか判別せよ。



m辺あるとする。

一つのアルゴリズム

アルゴリズム1 （辺順列法）

step1:全ての辺に番号をつける。

step2：1-mの数字を適当にならべる。

step3：ならべた数字に対応する辺の順序で、
一筆書きできるかしらべる。

step4:全てのならべ方を調べてなければ、
step2に戻る。

1 2 3 4 5 6 7 8 9 10 11 12

1 3 4 10 9 12 11 5 6 7 8 2

辺順列法の手間(計算量)

m個の数字の並べ方は、ｍ！であるので、
ｍ！に比例する手間がかかる。

ちなみに、

$$m! > 2^m \quad (m>=4 \text{ のとき})$$

アルゴリズム オイラー1は 指数時間アルゴリズムであるという。

100MIPSの計算機としましょ。
（1秒間に100万回の計算ができる。）

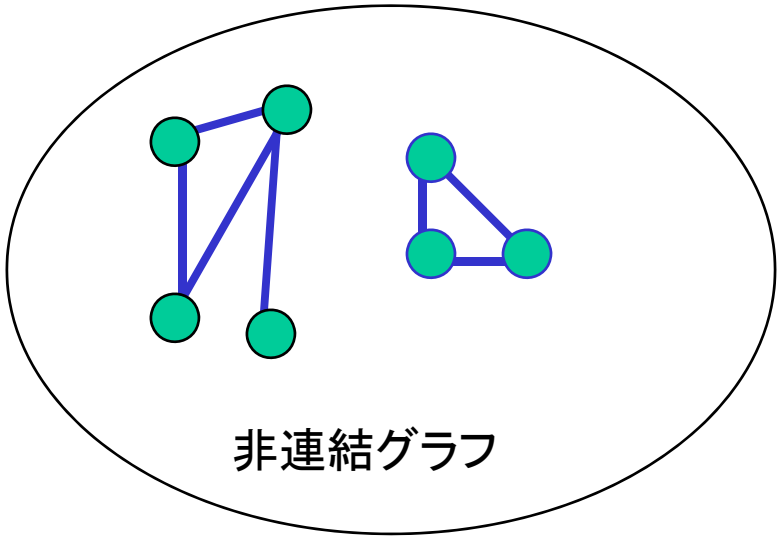
辺数mと計算時間

m	2^m	ｍ！
10	0.00005秒	0.04秒
20	0.01秒	771年
30	10秒	
40	3時間	
50	130日	

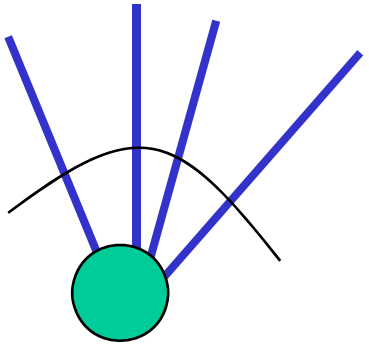
グラフ理論を使った高速化

定理1

グラフGにオイラー回路があるための、必要十分条件は、
（Gが連結で）Gのすべての 点の次数が偶数であること。



非連結グラフ



4次の点

次数：接続する辺の本数

アルゴリズム2（次数法）

step1:点の次数が偶数かどうか調べる。

step2:全ての点を調べてなければ、step1にもどる。

次数法の手間は辺数m　に比例した時間があれば十分である。

辺数mと計算時間

m	m	m^2	m^5	2^m	$m!$
10	0.0000001秒	0.000001秒	0.002秒	0.00005秒	0.04秒
20	0.0000002秒			0.01秒	771年
30	0.0000003秒			10秒	
40				3時間	
50				130日	
1000	0.00001秒		10秒		
1万	0.0001秒	1秒	3時間		

多項式時間アルゴリズム

指数時間アルゴリズム

ずいぶん賢くなった。めでたし、めでたし。

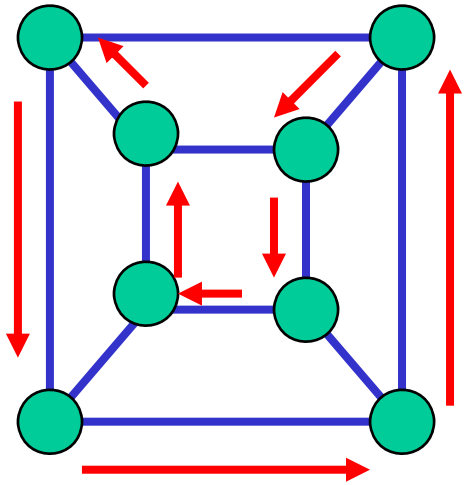
２．ハミルトン閉路問題

さて、オイラー回路と似た別の問題を考えよう。

同じ辺を1度しか通らずに全ての辺を1度は通る回路が、
オイラー回路である。

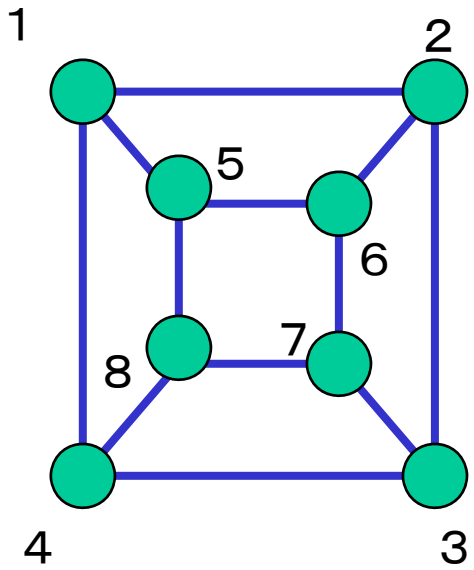
同じ点を1度しか通らずに全ての点を1度は通るような閉路を
ハミルトン閉路という。（オイラー回路は、同じ点を何度も通っていた。）

ハミルトン閉路の例



ハミルトン閉路問題

グラフGが与えられたとき、そのグラフにハミルトン閉路があるか判別せよ。



n点あるとする。

一つのアルゴリズム

アルゴリズム1 （点順列法）

step1:全ての点に番号をつける。

step2：1-nの数字を適当にならべる。

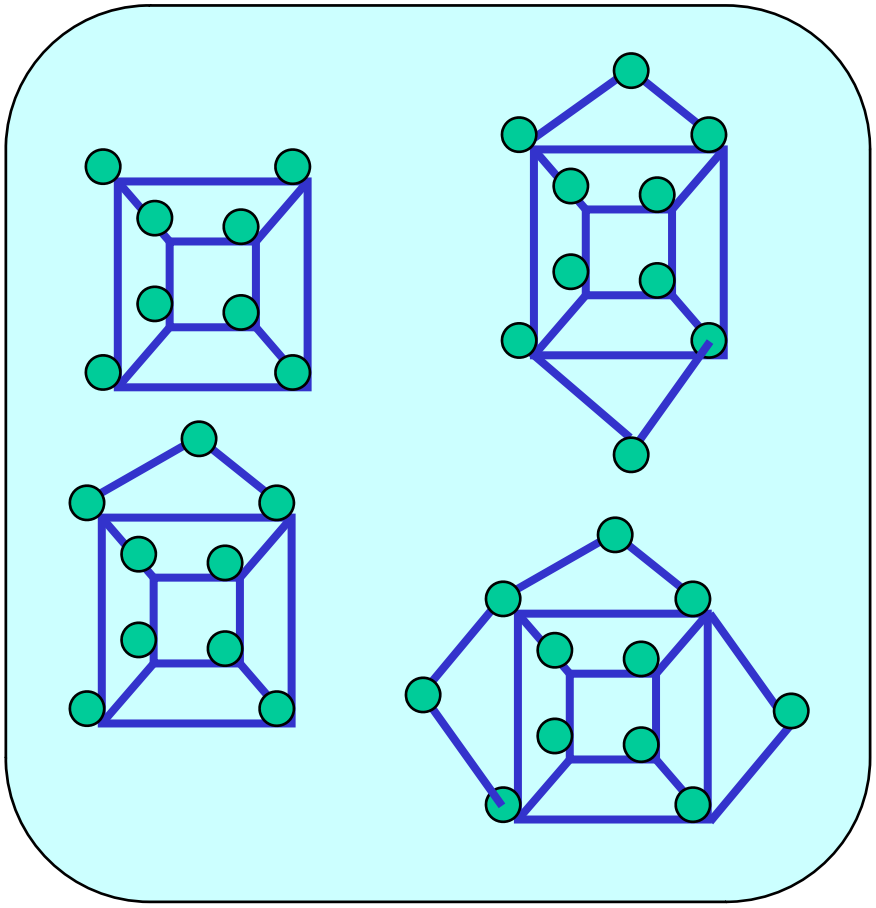
step3：ならべた数字に対応する点の順序で、ハミルトニアン閉路ができるか調べる。

step4:全てのならべ方を調べてなければ、step2に戻る。

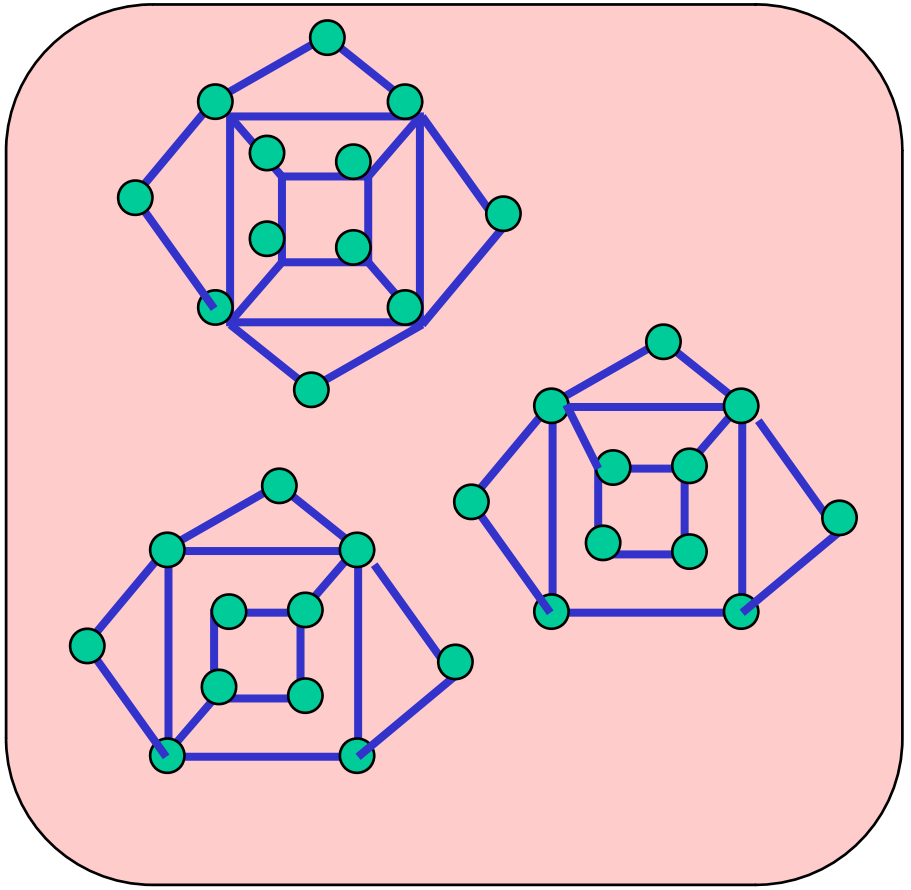
1 2 3 4 5 6 7 8 ✗

1 2 3 7 6 5 8 2 ○

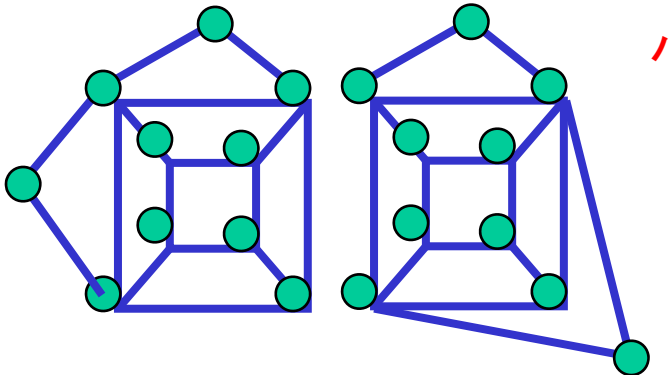
○



×



ハミルトン閉路あり



ハミルトン閉路なし

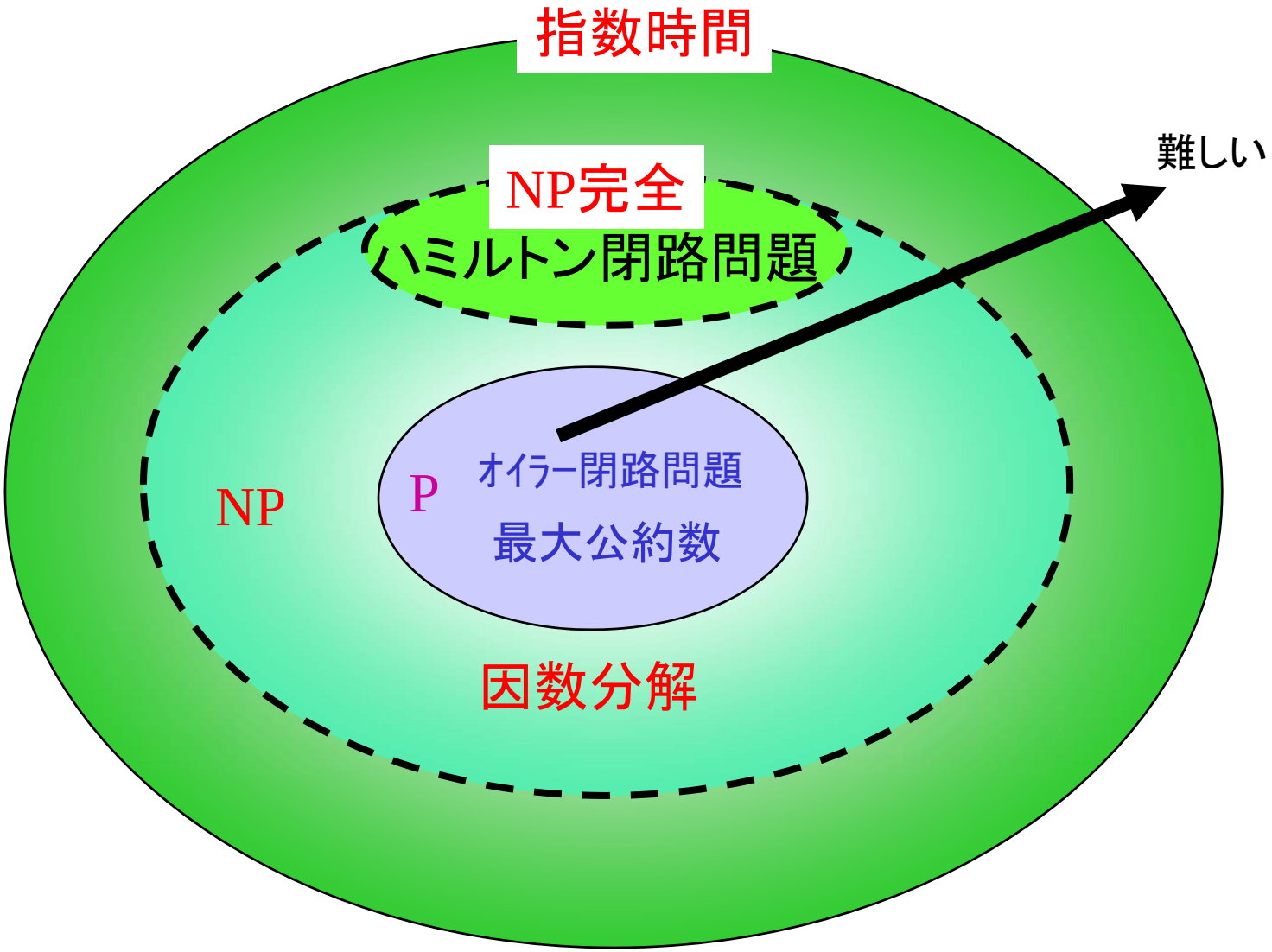
?

ハミルトン閉路問題を解くアルゴリズムの高速化は？

実は、この問題が現在のコンピュータサイエンスの
最大の未解決問題です。

だれも、ハミルトン閉路問題を解く
高速な（多項式時間の）アルゴリズムを作れていない。

IV: 問題の難しさの階層



クラスPの問題

クラスPに属する問題の特徴。

問題から答えを求めることが容易
（多項式時間で問題を解くアルゴリズムが存在）

例 :オイラー回路問題、２進数変換、最大公約数等

クラスNPの問題

クラスNPに属する問題の特徴。

問題と答えが両方与えられたら、
答えのチェックは容易に行える。
（多項式時間でチェックが行える。問題から答えを求めることは計算量が多くてもかまわない。）

厳密な定義は
省く。

例　、因数分解、ハミルトン閉路、オイラー回路

ハミルトン閉路も、チェックは容易。

他のNP問題

因数分解問題

1143816257578888676692357799761466120102182967212423625625618429357069352457338978305971235639587050589890751475992900268795435 41

チェック（掛け算）

3490529510847650949147849619903898133417764638493387843990820577

×

3276913299326670954996198819083446141317764296799294253979828853 3

クラスNP完全の問題

クラスNP完全に属する問題の特徴。

（1）問題と答えが両方与えられたら、答えのチェックは容易に行える。

（2）この問題を解く多項式時間アルゴリズムが発見されれば、クラスNPに属するすべての問題が多項式時間で溶ける。

厳密な定義は省く。

例　ハミルトン閉路、巡回セールスマン問題

「P　vs NP 問題」
（計算機科学最大の問題）

答えを多項式時間でチェックできる問題（NP問題）には、すべて答えを多項式時間で求めるアルゴリズムが存在する（Pに属する）のか？

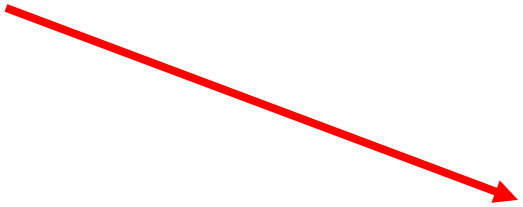
もし、ハミルトン閉路問題を解く多項式時間アルゴリズムを見つければ、「P　vs　NP　問題」を解いたことになる。

チューリング賞間違いなし

コンピュータサイエンスの
ノーベル賞のようなもの。

V:まとめ

- グラフ理論を紹介した。
- アルゴリズム論を紹介した。
- グラフ理論とアルゴリズムの関係を
  オイラー閉路問題を例にとって説明した。
- 「P vs NP 問題」という現在のコンピュータサイエンス
  (情報科学)最大の未解決問題を紹介した。



100万ドル(1億円)の懸賞が付いています。

参考文献

（1）中村亨、「数学21世紀の7大難問（数学の未来をのぞいてみよう）」、
（ブルーバックス）、2004、講談社、ISBN 4-06-257429-2

（2）一松信ほか、「数学7つの射解決問題（あなたも100万ドルにチャレンジしよう）」、
2002、森北出版、ISBN 4-627-01961-0