

# 楽しく計算するには ~計算を科学する~ らく

怠けて, 手を抜き, いいかげんに ×

無駄を省き, 効率的に, 高速に ○

東京大学大学院

情報理工学系研究科

数理情報学専攻

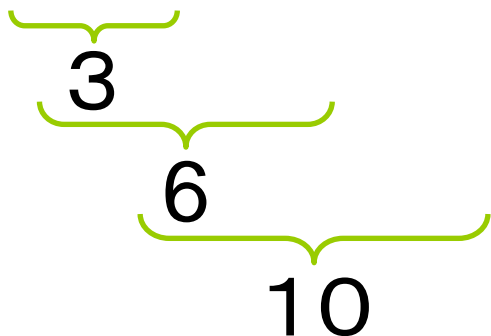
牧野 和久

# 効率性

例1.  $1 + 2 + 3 + 4 + \dots + 1000$

何回の演算(+, -, ×, ÷)で計算できる？

(A)  $1 + 2 + 3 + 4 + \dots + 1000$  単純な方法



1)  $1 + 2 = 3$

2)  $3 + 3 = 6$

3)  $6 + 4 = 10$

.....

999回

## 効率性

例1.  $1 + 2 + 3 + 4 + \dots + 1000$

何回の演算(+, -, ×, ÷)で計算できる？

(B)  $1 + 2 + 3 + 4 + \dots + 1000$  公式を用いる

$$= 1000 \times (1 + 1000) \div 2$$

3回

(A) 999回 >> (B) 3回

# 効率性

例2.  $a^n$ , ただし,  $a, n$  は正整数

何回の演算(+, -, ×, ÷)で計算できる?

(A)  $\overbrace{a \times a \times a \times a \times \cdots \times a}^{n\text{個}}$       単純な方法

$\underbrace{a \times a}_{a^2}$       1)  $a \times a = a^2$

$\underbrace{a^2 \times a}_{a^3}$       2)  $a^2 \times a = a^3$

$\underbrace{a^3 \times a}_{a^4}$       3)  $a^3 \times a = a^4$

.....

$n-1$ 回

## 効率性

例2.  $a^n$ , ただし,  $a, n$  は正整数

何回の演算(+, -, ×, ÷)で計算できる?

(B)  $n=2^k$  のとき (説明を簡単にするため)

$$1) a \times a = a^2 = 2^1$$

$$2) a^2 \times a^2 = a^4 = 2^2$$

$$3) a^4 \times a^4 = a^8 = 2^3$$

$$4) a^8 \times a^8 = a^{16} = 2^4$$

.....

$$k) a^{2^{k-1}} \times a^{2^{k-1}} = a^{2^k} \quad k\text{回}$$

## 効率性

例2.  $a^n$ , ただし,  $a, n$  は正整数

何回の演算(+, -, ×, ÷)で計算できる?

$n = 2^k$  のとき (説明を簡単にするため)

(A)  $2^k - 1$  回 >> (B)  $k$  回

1023

10

# 効率性

$n$ 桁のとき

例3. 何回, 回転されれば鍵は開けられる?

- ・ 番号が既知のとき

各桁: 高々5回

5回  $\times$  4桁 = 高々20回

$5n$ 回

- ・ 番号が未知のとき

全通り: 高々10000回

$10^n$ 回

# 効率性

例4. 素因数分解

323を素因数分解せよ.

答  $323 = 17 \times 19$

← 易しい

→ 難しい



# 例4. 素因数分解

RSA Laboratories RSA576  
1993年 1万ドル

188198812920607963838697239461650439807  
163563379417382700763356422988859715234  
665485319060606504743045317388011303396  
716199692321205734031879550656996221305  
168759307650257059 174桁

= 398075086424064937397125500550386491199  
064362342526708406385189575946388957261  
768583317 ✕

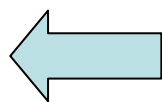
472772146107435302536223071973048224632  
914695302097116459852171130520711256363  
590397527

## 効率性

例4. 素因数分解 難しい???

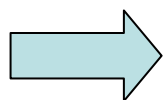
323を素因数分解せよ.

答  $323 = 17 \times 19$



易しい

一方向性関数



難しい?

(公開鍵暗号の基礎)

RSA暗号の基礎

Rivest, Shamir, Adleman 1978

(Turing Award 2003)

# コンピュータ(計算機)って何でもできる？

計算(機)??

計算可能??

1930年代頃 盛んに研究

計算モデル: 有限状態機械,  
プッシュダウン・オートマトン  
チューリング機械, RAM,  $\lambda$ -計算,  
Postシステム, 帰納的関数, . . .

問題Aが計算モデルMで計算可能:

有限ステップで問題Aを解く

計算モデルMでのアルゴリズムの存在

# 計算可能性

計算モデル: 有限状態機械

∩

プッシュダウン・オートマトン

∩

等価

チューリング機械, RAM,  $\lambda$ -計算,  
Postシステム, 帰納的関数, . . .

Church-Turingの提唱 **安定した概念**

A. Church, K. Gödel, S. C. Kleene, E. Post, A.A. Markov,  
A.M. Turing らによって様々な**妥当な**計算モデルとそれに  
基づく計算可能性が提案されたがすべて**等価**

コンピュータ(計算機)って何でもできる？

どんな問題も計算可能?? NO!

Postの対応問題

入力: 0-1列集合  $A = \{a_1, a_2, \dots, a_k\}$ ,

$B = \{b_1, b_2, \dots, b_k\}$

出力: 次の条件を満たす  $i_1, i_2, \dots, i_m$  有限個  
が存在すれば, YES, さもなければ, NO

$$a_{i_1} a_{i_2} \dots a_{i_m} = b_{i_1} b_{i_2} \dots b_{i_m}$$

## Postの対応問題

入力: 0-1列集合  $A = \{a_1, a_2, \dots, a_k\}$ ,

$$B = \{b_1, b_2, \dots, b_k\}$$

出力: 次の条件を満たす  $i_1, i_2, \dots, i_m$

が存在すれば, YES, さもないければ, NO

$$a_{i_1} a_{i_2} \dots a_{i_m} = b_{i_1} b_{i_2} \dots b_{i_m}$$

例.  $A = \{1, 10111, 10\}$ ,  $B = \{111, 10, 0\}$   
 $a_1 \quad a_2 \quad a_3 \quad b_1 \quad b_2 \quad b_3$

$$10111 \mid 1 \mid 1 \mid 10 = 10 \mid 111 \mid 111 \mid 0$$

$a_2 \quad a_1 a_1 \quad a_3 \quad b_2 \quad b_1 \quad b_1 \quad b_3$

YES

## Postの対応問題

入力: 0-1列集合  $A = \{a_1, a_2, \dots, a_k\}$ ,

$$B = \{b_1, b_2, \dots, b_k\}$$

出力: 次の条件を満たす  $i_1, i_2, \dots, i_m$  有限個  
が存在すれば, YES, さもないと, NO

$$a_{i_1} a_{i_2} \dots a_{i_m} = b_{i_1} b_{i_2} \dots b_{i_m}$$

定理: Postの対応問題は計算不可能.

$m=1$  のとき  $a_1 = b_1 ? a_2 = b_2 ? \dots a_k = b_k ?$

$m=2$  のとき  $a_1 a_2 = b_1 b_2 ? a_1 a_3 = b_1 b_3 ? \dots$

⋮

解がないとき, いつ止める? 止められない!!!

# 計算可能であればいい??

問題Aが計算可能:

有限ステップで問題Aを解くアルゴリズムの存在

時間量, 領域量

基準は入力サイズ

例. クラス40人の‘良い’座席表を作る.

入力サイズ  $n = 40$



入力 サイズ	計算時間				
	$n$	$n^2$	$n^3$	$2^n$	$n!$
10	$1 \times 10^{-9}$ 秒	$1 \times 10^{-8}$ 秒	$1 \times 10^{-7}$ 秒	$1 \times 10^{-7}$ 秒	$3.6 \times 10^{-4}$ 秒
20	$2 \times 10^{-9}$ 秒	$4 \times 10^{-8}$ 秒	$8 \times 10^{-7}$ 秒	$1 \times 10^{-4}$ 秒	7.5年
30	$3 \times 10^{-9}$ 秒	$9 \times 10^{-8}$ 秒	$2.7 \times 10^{-6}$ 秒	$1.1 \times 10^{-1}$ 秒	$8.4 \times 10^{14}$ 年
40	$4 \times 10^{-9}$ 秒	$1.6 \times 10^{-7}$ 秒	$6.4 \times 10^{-6}$ 秒	1.8分	$2.6 \times 10^{30}$ 年
50	$5 \times 10^{-9}$ 秒	$2.5 \times 10^{-7}$ 秒	$1.3 \times 10^{-5}$ 秒	31時間	$2.3 \times 10^{48}$ 年
100	$1 \times 10^{-8}$ 秒	$1 \times 10^{-6}$ 秒	$1 \times 10^{-4}$ 秒	$4.1 \times 10^{12}$ 年	$2.9 \times 10^{140}$ 年
1000	$1 \times 10^{-7}$ 秒	$1 \times 10^{-4}$ 秒	0.1秒	...	...

10BIPS(billion instructions per second)の計算機を用いた場合

# 計算可能であればいい??

問題Aが計算可能:

有限ステップで問題Aを解くアルゴリズムの存在

時間量, 領域量

基準は入力サイズ

例. クラス40人の‘良い’座席表を作る.

入力サイズ  $n = 40$

速い: 多項式時間で解ける

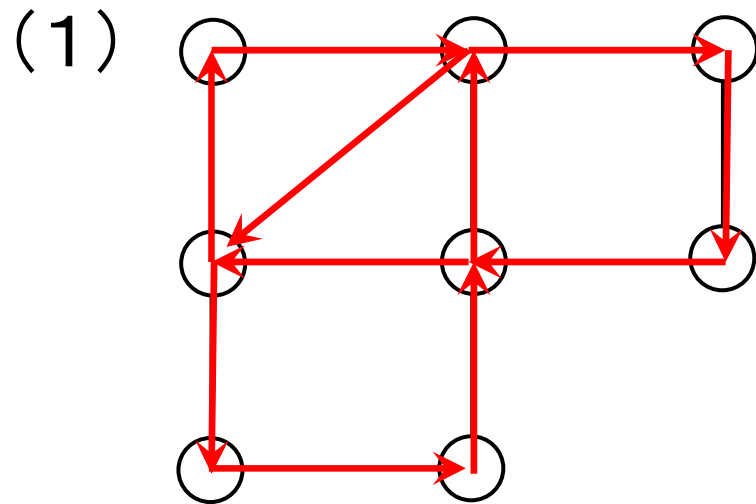
遅い: 指数時間で解ける

# P vs NP

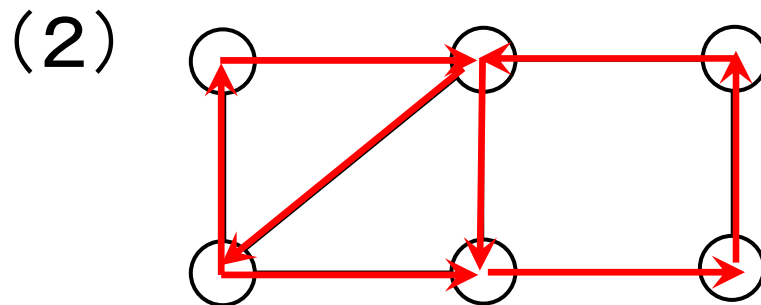
- Euler閉路問題 多項式時間で解ける
- Hamilton閉路問題 ???????

# Euler閉路問題

Euler閉路: すべての辺を丁度1回通る閉路



Euler閉路あり

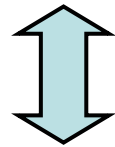


Euler閉路なし??

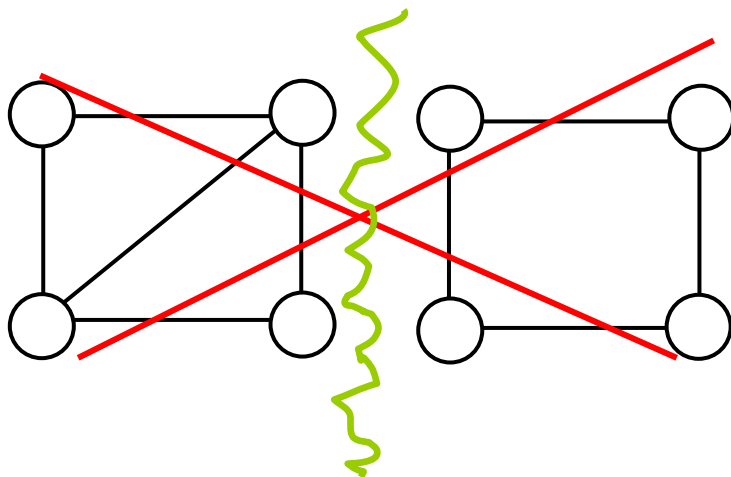
# Euler閉路問題

Euler閉路：すべての辺を丁度1回通る閉路

Euler閉路が存在



連結, かつ, すべての点の次数が偶数



連結でない

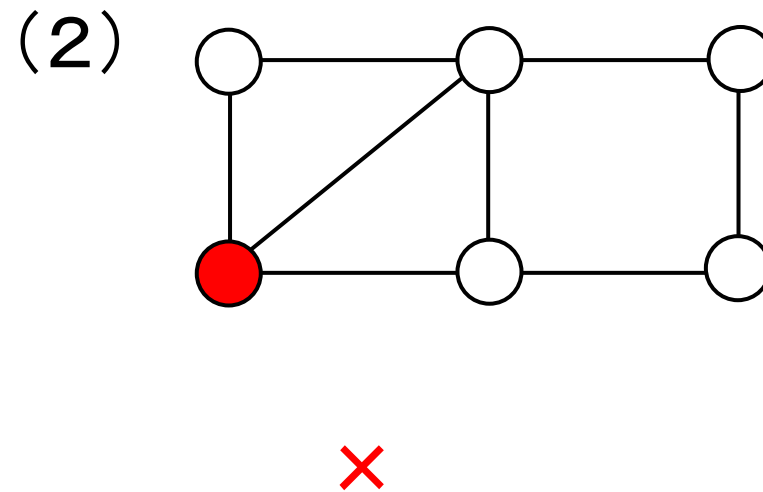
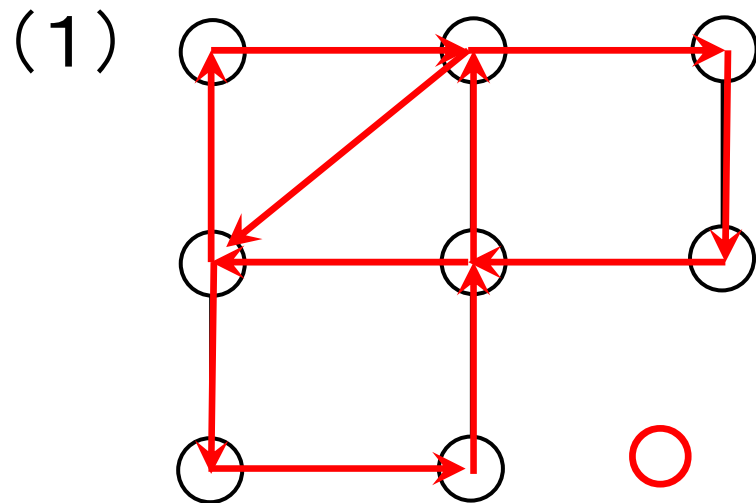
# Euler閉路問題

Euler閉路: すべての辺を丁度1回通る閉路

∃ Euler閉路が存在



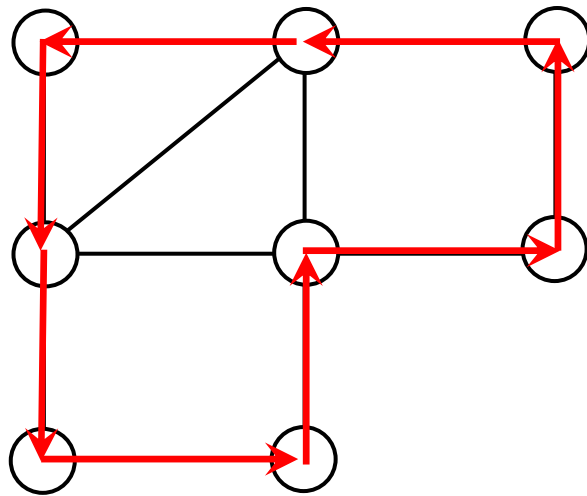
▽ 連結, かつ, すべての点の次数が偶数



# Hamilton閉路問題

Hamilton閉路: すべての点を丁度1回通る閉路

(1)



Hamilton閉路あり

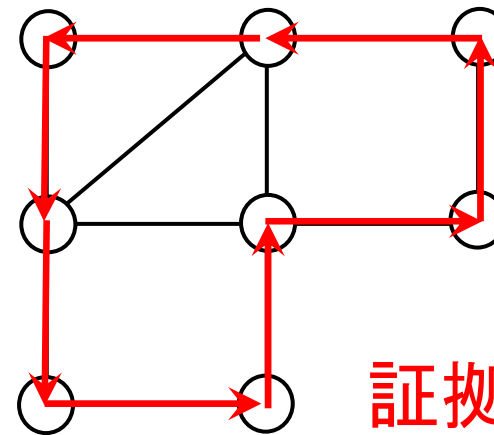




# Hamilton閉路問題

Hamilton閉路：すべての点を丁度1回通る閉路

(1) Hamilton閉路あり



証拠：閉路

(2) Hamilton閉路なし

証拠？？？ 全ての閉路を確かめる？？

# P vs NP

Euler閉路問題

多項式時間で解ける P

Hamilton閉路問題

??????

NP完全

## まとめ

- 効率的に解くことの重要性
- 計算可能性
- 多項式時間
- P vs NP (証拠)