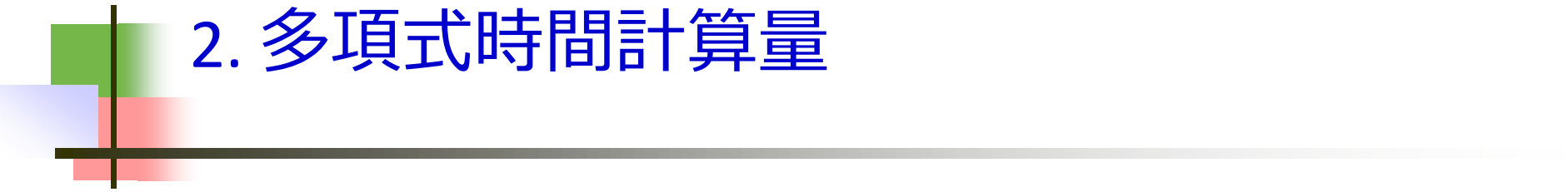
May 19, 2025



2. 多項式時間計算量

来嶋 秀治

滋賀大学 データサイエンス学部

計算機科学のMCMC法

Fully Polynomial-time Randomized Approximation Scheme (FPRAS)

“Valiant計画” (since 1979)

#P困難な$A \in \mathbb{R}$に対して，近似解$Z \in \mathbb{R}$を精度

$$\Pr[(1-\epsilon)A \leq Z \leq (1+\epsilon)A] \geq 1-\delta$$

で多項式時間$(\mathrm{poly}(\mathrm{input}, \epsilon^{-1}, \log \delta^{-1}))$で求められるか？

多項式時間(polynomial time)とは何が多項式なのか？

⇨ 「演算」の回数

「計算量」(決定性 Turing machine)

通常(形容詞のない場合)は
worst case time complexityを指す

◘ 計算量 = 演算回数(有限回に限る)

- **読み込み**，**書き出し**，**代入**，**比較**: O(桁数)
- **四則演算**(+, −,×,/) O(桁数)～ O(桁数^2)
- **論理演算**(¬,∧,∨, ⇒) $O(1)$/1演算
- 繰り返し(for, while)は繰り返し中の演算の総数
- 「パラメータ」
  - 固定(定数)なら$O(1)$
  - 可変なら四則演算に準ずる
- 「関数(ライブラリ)」
  - 計算量が解明されていれば，それに準拠
  - オラクル計算量では，通常の演算数とは別に数える

「計算」を掘り下げる

若干(?) 嘘を含むので注意

Turing machine

- できること
  - 有理数の四則演算
  - 論理演算（⇒,∧,∨, ¬, ⊤, ⊥; 例:$1 \vee 0 = 1, 1 \wedge 0 = \bot$）
  - 受理、非受理の判定（If文）
  - 大きな数が数えられる（=メモリがある: While文、For文）
  - 配列
- できないこと
  - 実数（無理数一般）の計算（二次方程式の解の公式とか）
  - 「難しい」関数の計算: sin, cos, tan, exp, log, etc.
  - 微分、積分（一般に）
  - 二階述語論理（停止性問題）

➢ 何をどこまで計算可能か都度確認。

Pythonではmathをimportしてexpはxは計算できる!?

$$\exp(3) \simeq 2.718 * 2.718 * 2.718$$

$$\exp(1.5) \simeq \sqrt{2.718 * 2.718 * 2.718} \quad \text{(開平根)}$$

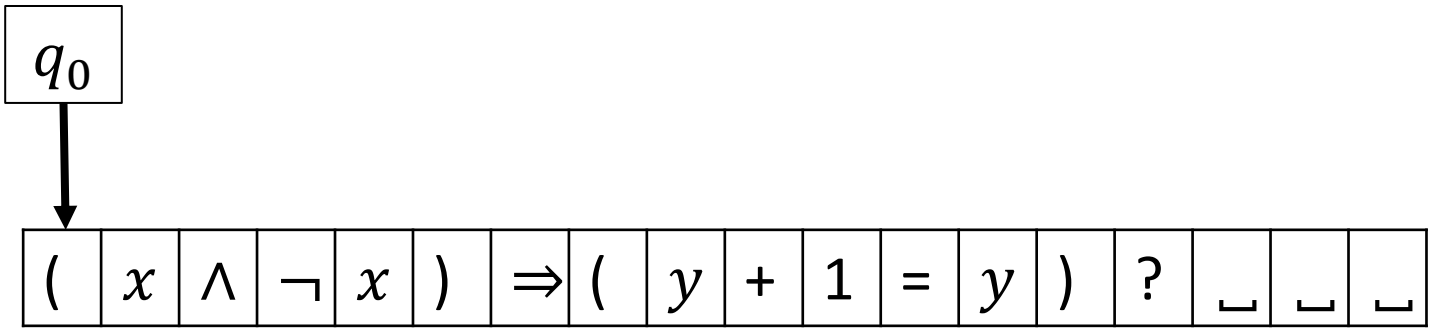
$$\exp(1.7) \simeq \sqrt[10]{\left(\left((2.718^2)^2\right)^2\right)^2} * 2.718$$

$$\exp(x) = 1 + \frac{1}{2}x^2 + \frac{1}{3!}x^3 + \cdots$$

「計算」を掘り下げる

- **計算 = accept-rejectの決定**
- 入力 = 文字列
- なぜ「多項式」時間？
  - 等価性、帰着

計算機科学における計算: Turing machine



Turing machine

- headerとtapeで構成された機械
  - 決められた規則に従って動作する
- headerは有限個の演算規則集 (有限オートマトン)
- tapeはメモリ
  - 文字を読んだり書いたり(書き換えたり) できる．長さに上限がない
- 文字は有限種類
- 最終的に，**accept(正しい) or reject(間違い)を判断して停止**
  - 判断できない計算問題が存在する (Turingの停止問題)

計算機科学の歴史（NP以前）

1874, Cantor, $\infty^n \neq 2^\infty$, 対角線論法

1900, Hilbert, 算術の無矛盾性 (23問題)

1931, Gödel, 不完全性定理

1936, Turing, Turing機械 + 停止問題

1936, Church, ラムダ計算

1945, von Neumann, ノイマン型コンピュータ(ENIAC)

1956, Kleene, regular language = finite automaton判定

1956, Chomsky, Chomsky階層

1971, Cook, NP完全の提案

Turing machineの先祖は論理学

Hilbert計画(1900)
数学が正しいことを証明せよ

- 形式証明: 論理演算規則による変形

  (例) 命題: $\forall x, \forall y, [\mathrm{even}(x) \wedge \neg\mathrm{even}(y) \rightarrow \mathrm{even}(x+y)]$(偶数+奇数=偶数)

  証明:

  $\mathrm{even}(x) \Rightarrow \exists a(x = 2a)$
  $\neg\mathrm{even}(y) \Rightarrow \exists b(y = 2b+1)$
  $(x = 2a) \wedge (y = 2b+1) \Rightarrow (x+y = 2(a+b)+1)$
  $\exists c(x+y = 2c+1) \Rightarrow \neg\mathrm{even}(x+y)$
  $\neg\mathrm{even}(x+y) \wedge \mathrm{even}(x+y) = \perp$

- 証明できればaccept (=定理)，矛盾が出ればreject

Gödel(1931):
証明とは数字(Gödel数)の計算である．計算できない命題が存在する．
計算機科学(Turing(1936), Kleene(1956), Chomsky(1956), ...?):
計算とは文字列のaccept-rejectの決定である．決定できない文字列が存在する．

「計算」を掘り下げる

- 計算 = accept-rejectの決定
- **入力 = 文字列**
- なぜ「多項式」時間？
  - 等価性、帰着

言語の認識

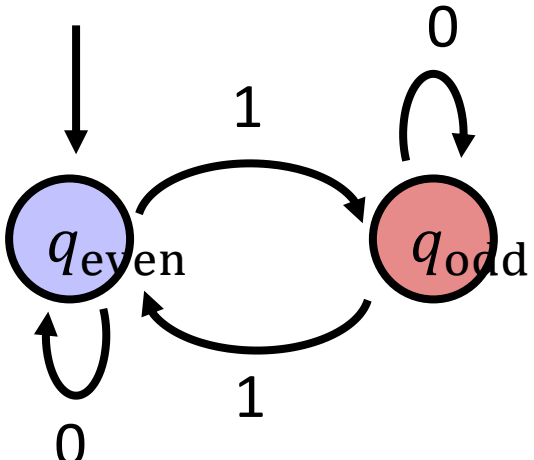
- Σ: 文字の集合(alphabet)
- Σ^k: 長さkの文字列の集合．$\Sigma^* = \cup_{k=0}^{\infty} \Sigma^k$
- $\mathcal{L} \subseteq \Sigma^*$: 言語(language)
- $w \in \mathcal{L}$: 語(word)
- $w \in \mathcal{L}$をacceptし，$w \notin \mathcal{L}$をrejectできることをrecognitionという．

定理 (Kleene 1956)
正規言語であることの必要十分条件は**有限オートマトン**でrecognizeできる言語

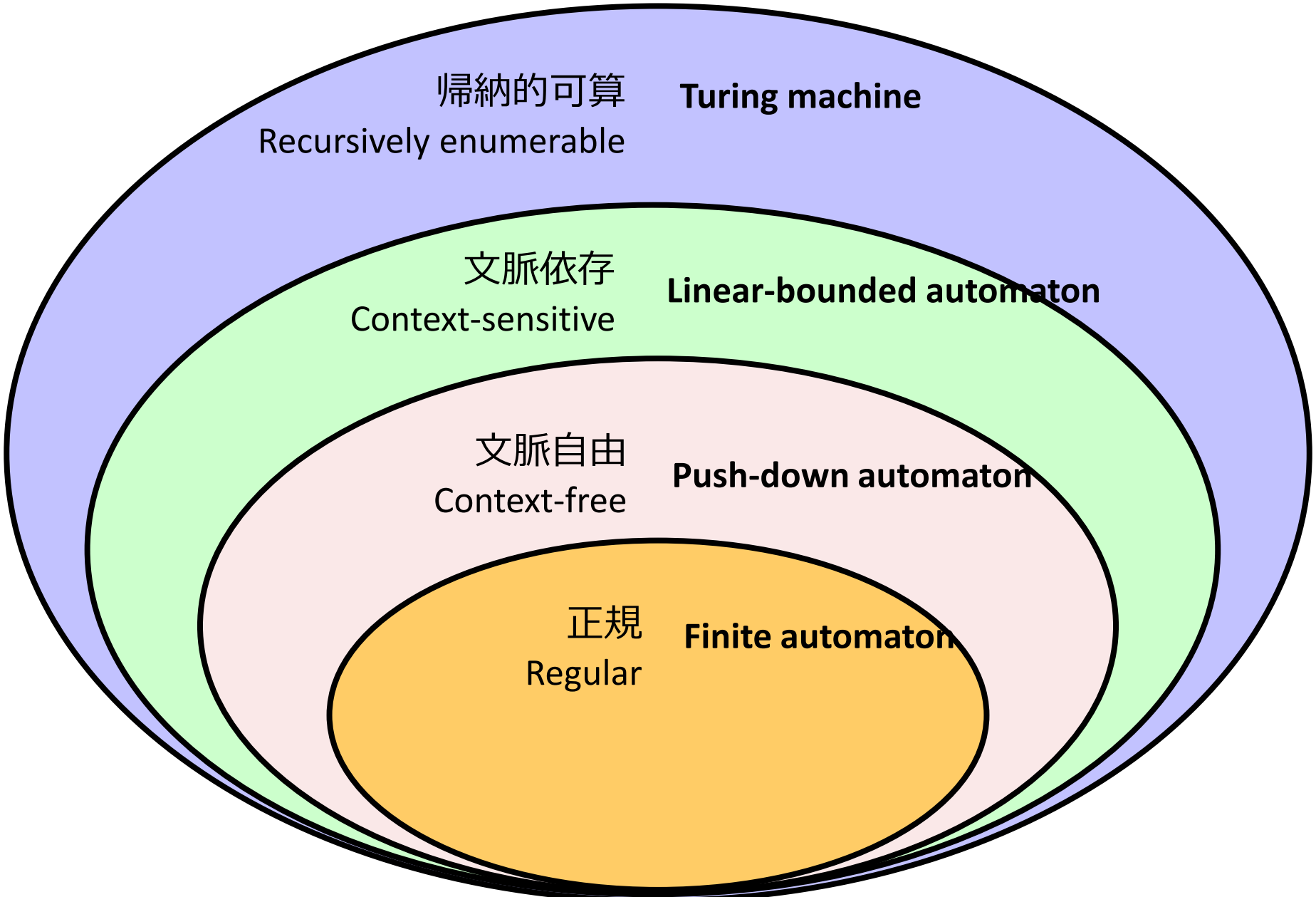
$\Sigma = \{0,1\}$とする．

有限オートマトンは
大きな数を数えられない

- 正規言語の例
  - $\{w | w$の最初と最後は同じ文字$\} = 0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$
  - $\{w | w$の長さは3の倍数$\} = (\Sigma\Sigma\Sigma)^*$
  - $\{w | w$は1を偶数個もつ$\}$
- 正規言語でない例
  - $\{0^n 1^n | n \geq 0\}$
  - $\{w | w$は0と1を同数もつ$\}$



Chomsky hierarchy



決定問題(decision problem)

例1. グラフ3彩色

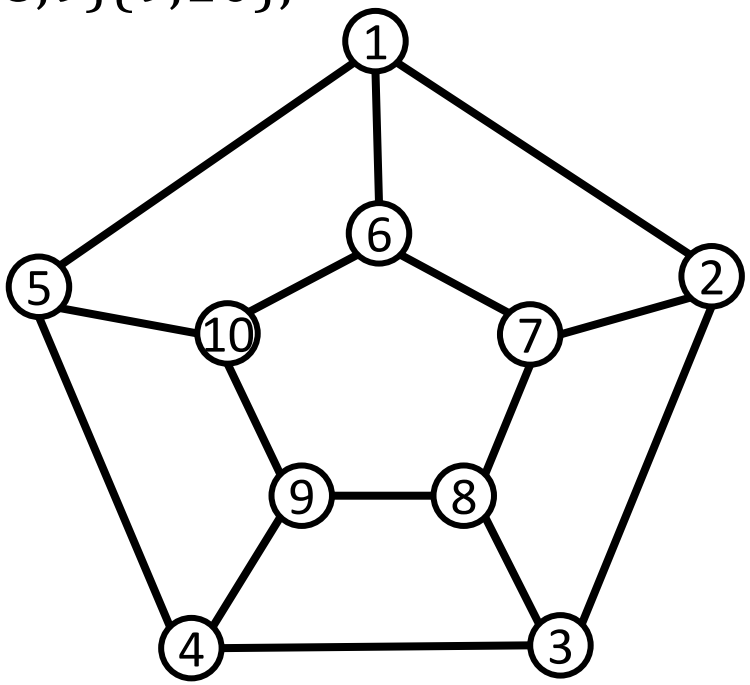
入力: $G = (V, E)$

問題: Gは3色で塗り分けられるか？

$\mathcal{L} \subseteq \Sigma^* = \{0, \ldots, 9, \{, \}, , , :, ;\}$: グラフ3彩色言語

$w = 10$: $\{1,2\}\{1,5\}\{1,6\}\{2,3\}\{2,7\}\{3,4\}\{3,8\}\{4,5\}\{4,9\}\{5,10\}$
$\{6,7\}\{6,10\}\{7,8\}\{8,9\}\{9,10\}$;

$w \in \mathcal{L}$



決定問題(decision problem)

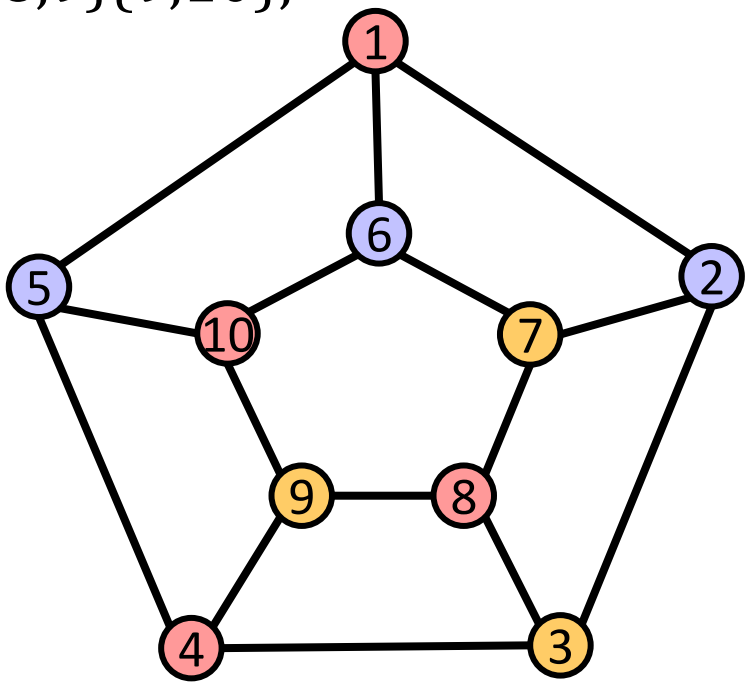
例1. グラフ3彩色

入力: $G = (V, E)$

問題: Gは3色で塗り分けられるか？

$\mathcal{L} \subseteq \Sigma^* = \{0, \ldots, 9, \{, \}, , , :, ;\}$: グラフ3彩色言語

$w = 10$: $\{1,2\}\{1,5\}\{1,6\}\{2,3\}\{2,7\}\{3,4\}\{3,8\}\{4,5\}\{4,9\}\{5,10\}$
$\{6,7\}\{6,10\}\{7,8\}\{8,9\}\{9,10\}$;
$w \in \mathcal{L}$



決定問題(decision problem)

例1. グラフ3彩色

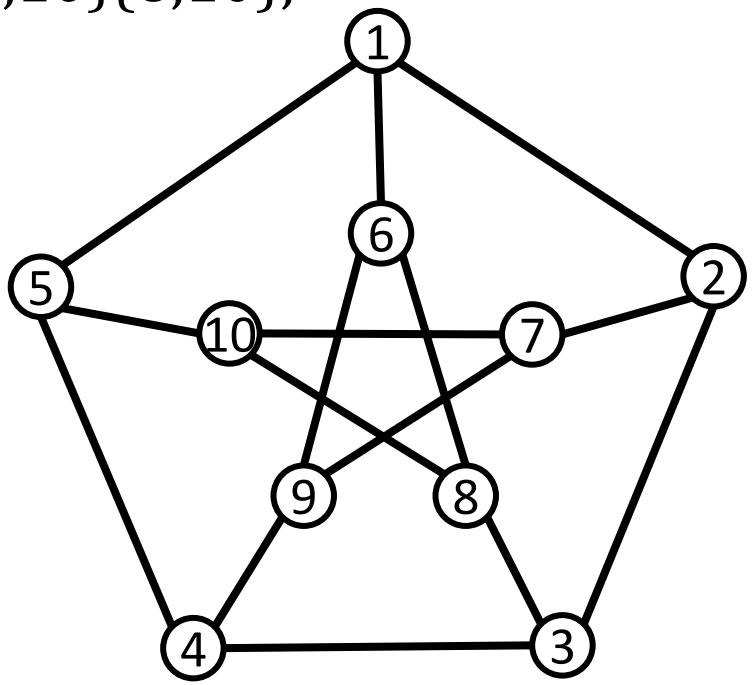
入力: $G = (V, E)$

問題: Gは3色で塗り分けられるか？

$\mathcal{L} \subseteq \Sigma^* = \{0, \dots, 9, \{, \}, , , :, ;\}$: グラフ3彩色言語

$w = 10$: $\{1,2\}\{1,5\}\{1,6\}\{2,3\}\{2,7\}\{3,4\}\{3,8\}\{4,5\}\{4,9\}\{5,10\}$
$\{6,8\}\{6,9\}\{7,9\}\{7,10\}\{8,10\}$;

$w \notin \mathcal{L}$



例2. Partition (cf. 0-1ナップサック問題)

入力: 整数 $z_1, z_2, \ldots, z_n; k$.

問題: 合計がkとなる組合せは存在するか？

例: $2,3,5,7,11,13,17,19; 38$.

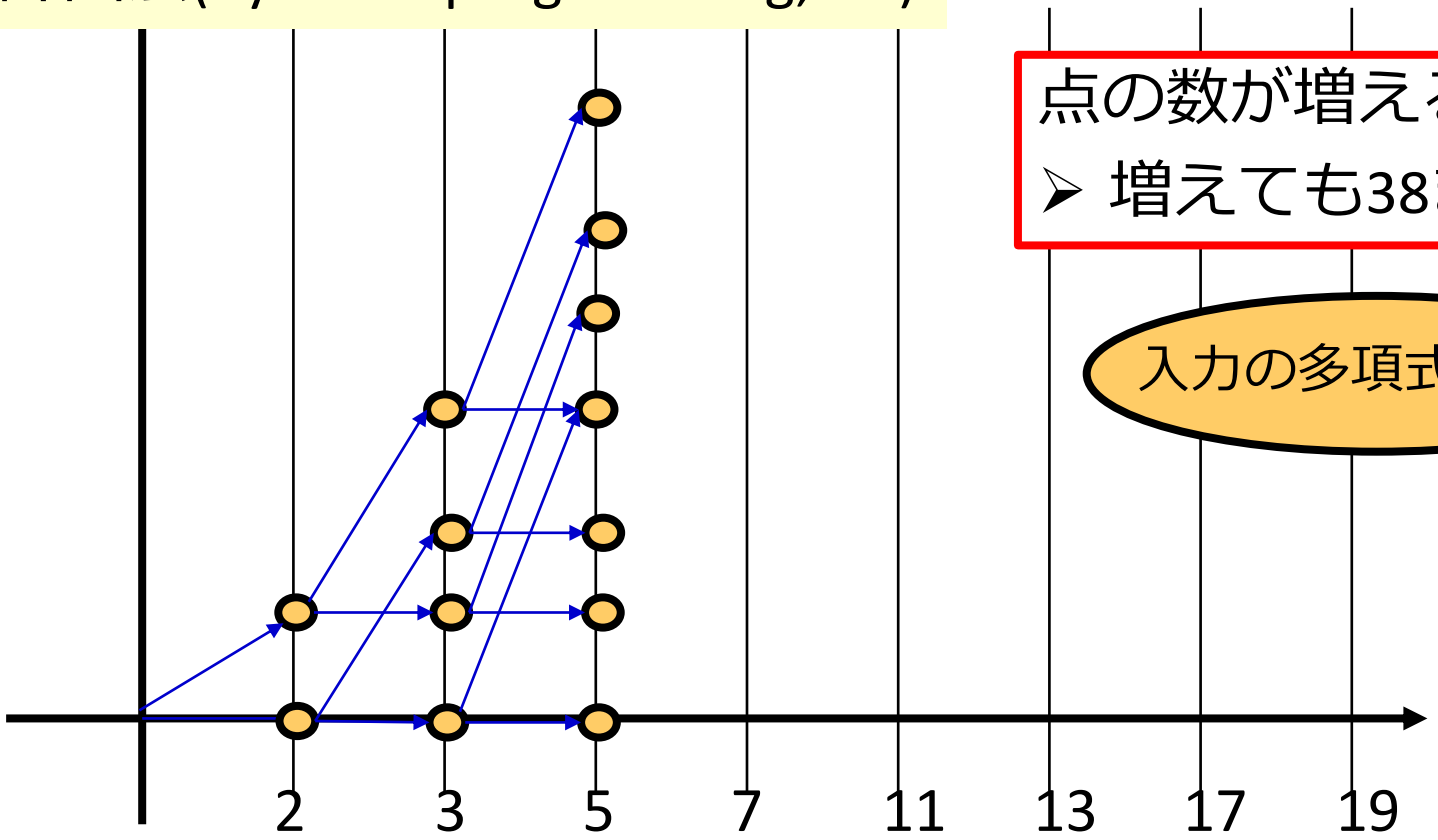
例2. Partition (cf. 0-1ナップサック問題)

入力: 整数 $z_1, z_2, \ldots, z_n; k$

問題: 合計がkとなる組合せは存在するか？

例: 2,3, 5, 7, 11, 13, 17, 19; 38.

動的計画法(dynamic programming; DP)



点の数が増える？

- 増えても38まで.

入力の多項式時間？

例2. Partition (cf. 0-1ナップサック問題)

入力: 整数 $z_1, z_2, \ldots, z_n; k$

問題: 合計がkとなる組合せは存在するか？

例: $271,314, 141,200, 223, 264, 316, 331; 1008.$

動的計画法(dynamic programming; DP)

点の数が増える？

- 増えても1008まで.

入力の多項式時間？

271+200+223+314

例2. Partition (cf. 0-1ナップサック問題)

入力: 整数 $z_1, z_2, \ldots, z_n; k$

問題: 合計がkとなる組合せは存在するか？

例,

$$2749384726478576374 7,$$
$$1104853494039284584 3,$$
$$2434920934039405932 1,$$
$$1433059409504953849 7,$$
$$1913470429434904434 0,$$
$$1941734436813491001 9,$$
$$2223343495808052092 4,$$
$$1700000000000000000;$$
$$7407649162334907932 9.$$

点の数が増える？

- 増えても7407649162334907932 9まで.

入力の多項式時間？

入力サイズは桁数で数える.

数え上げ問題 [Valiant 79]

決定問題の特に計算量の議論は多分に組合せ論的である.

- たとえばNP完全問題の難しさは，膨大な組合せの中から条件に合う解を見つける難しさに起因する.

組合せ論の興味の中心は，ずばり，組合せの個数である.

決定問題の解の個数は数えられるのであろうか？

$w \in \mathcal{L}$に対し，

$$\mathcal{C}(w) \subseteq \Sigma^*$$

をwの“解”の集合とする.

計算量理論の枠組みでは

Certificateとかwitness

と呼ばれる.

Q. $w \in \mathcal{L}$が与えられた時，$|\mathcal{C}(w)|$を求めよ.

数え上げ問題 (counting problem)

例1. グラフ3彩色

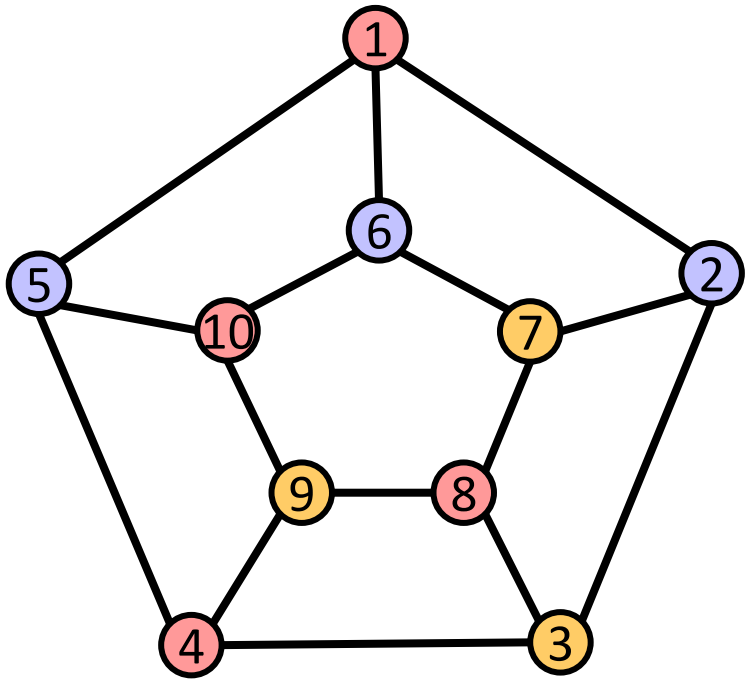
入力: $G = (V, E)$

問題: Gは3色で塗り分けられるか？

$\mathcal{C}(w) \subseteq \{r, b, y\}^{10}$

$rbyrbbyryrc \in \mathcal{C}(w)$

計算量理論の枠組みでは
Certificateとかwitness
と呼ばれる.



数え上げ問題 (counting problem)

例1. グラフ3彩色

入力: $G = (V, E)$

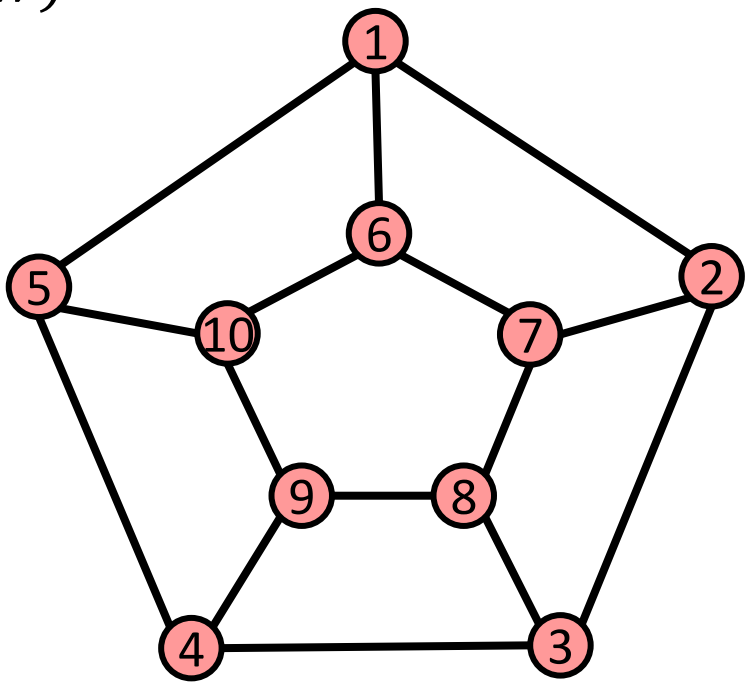
問題: Gは3色で塗り分けられるか？

$\mathcal{C}(w) \subseteq \{r, b, y\}^{10}$

$rbyrbbyryrc \in \mathcal{C}(w)$

$rrrrrrrrrr \notin \mathcal{C}(w)$

計算量理論の枠組みでは
Certificateとかwitness
と呼ばれる.



計算機科学のMCMC法

計算量クラス #P [Valiant '79]

$\mathcal{L}$がNPに属するとき，$w \in \mathcal{L}$に対し，

$\mathcal{C}(w) \subseteq \Sigma^*$を$w$のcertificateの集合とする．

$|\mathcal{C}(w)|$を求める問題のクラスを#Pと呼ぶ．

1998年Gödel prize

Cf. 定理 [戸田1991]

$P^{\#P} = PH$

"Valiant計画" (since 1979)

#P問題$|\mathcal{C}(w)| \in \mathbb{R}$に対して，近似解$Z \in \mathbb{R}$を精度

$$\Pr[(1-\epsilon)|\mathcal{C}(w)| \leq Z \leq (1+\epsilon)|\mathcal{C}(w)|] \geq 1-\delta$$

で多項式時間$(\mathrm{poly}(\mathrm{input}, \epsilon^{-1}, \log \delta^{-1}))$で求められるか？

Fully Polynomial-time Randomized Approximation Scheme (**FPRAS**)

$A \in \mathbb{R}$に対して，近似解$Z \in \mathbb{R}$を精度

$$\Pr[(1-\epsilon)A \leq Z \leq (1+\epsilon)A] \geq 1-\delta$$

で多項式時間$(\mathrm{poly}(\mathrm{input}, \epsilon^{-1}, \log \delta^{-1}))$で求めるアルゴリズム．

「計算」を掘り下げる

- 計算 = accept-rejectの決定
- 入力 = 文字列
- **なぜ「多項式」時間？**
  - 等価性、帰着

P ≠ NP予想

「言語$\mathcal{L}$が**非**決定性Turing機械で多項式時間で判定可能
⇏ 言語$\mathcal{L}$は決定性Turing機械で多項式時間で判定可能」

(⇐は定義から導かれる)

Cf. Savitchの定理
「言語$\mathcal{L}$が**非**決定性Turing機械で多項式領域で判定可能
⇒ 言語$\mathcal{L}$は決定性Turing機械で多項式領域で判定可能」

(⇐は定義から導かれる)

そもそも、Turing machineにはたくさん種類がある

- 有限オートマトン
- プッシュダウンオートマトン
- 2スタックプッシュダウンオートマトン
- 決定性Turing machine
  - 1テープ1ヘッド
  - 多テープマシン
  - 多ヘッドマシン
  - 万能Turing machine
- 非決定性Turing machine
- オラクルTuring machine
- 確率的Turing machine

- 1回の演算にかかる時間が異なる.
  - 例: n桁+m桁
    - 1テープ1ヘッドだと$O((n+m)^2)$
    - 多ヘッドだと$O(n+m)$
- 「多項式」のくくりにすると
  どの決定性TMでも等価.

Cf. **Cantorの定理** $\infty^n \neq 2^\infty$

等価性

Reduction (還元, 帰着)：simulateできるか？

- 計算可能性の等価性
  - Turing完全
    - 例: 2-stack push-down automaton
- 多項式**領域**計算量の等価性
  - PSPACE完全
  - NL完全
  - Cf. Savitchの補題：$\mathrm{NSPACE}\big(f(n)\big) \subseteq \mathrm{SPACE}(f(n)^2)$
- 多項式**時間**計算量の等価性
  - NP完全
  - P完全

本講義の計算機モデル

非決定性Turing machineとは異なる
(cf. $RP \neq NP$予想)

◘ 確率的Turing machine

- 決定性Turing machine + random **bit** (0 or 1)生成
  - n bits生成は$O(n)$時間
  - [0,1] 実数乱数は生成できない

けっこう不便．たとえば，

命題
確率的Turing Machineでは「確率1/3」を有限時間停止では実現不可能．

命題
確率的Turing Machineでは「確率1/3」を$O(1)$期待時間で実現不可能．

このくらいの精度で議論を進める．

まとめ

Turing machine

- できること
  - 有理数の四則演算
  - 論理演算（⇒,∧,∨, ¬, ⊤, ⊥; 例:$1 \vee 0 = 1, 1 \wedge 0 = \bot$）
  - 受理、非受理の判定（If文）
  - 大きな数が数えられる（=メモリがある: While文、For文）
  - 配列
- できないこと
  - 実数（無理数一般）の計算（二次方程式の解の公式とか）
  - 「難しい」関数の計算: sin, cos, tan, exp, log, etc.
  - 微分、積分（一般に）
  - 二階述語論理（停止性問題）

➢ 何をどこまで計算可能か都度確認。

四則演算の計算量

- 整数xはおよそ$\log x$桁
- p桁とq桁の和は$\max\{p, q\} + 1$桁以下
  - 計算時間もそれくらい
- p桁−q桁は符号込みで$\max\{p, q\} + 2$桁以下
  - 計算時間もそれくらい
- p桁とq桁の積は$n + m$桁以下
  - 計算時間もまぁそれくらい
- 有理数 $\frac{x}{y}$ は$n + m$桁以下
  - 整数のペア(x, y)として表される（分数）

「計算量」(決定性 Turing machine)

通常(形容詞のない場合)は
worst case time complexityを指す

◘ 計算量 = 演算回数(有限回に限る)

- **読み込み**，**書き出し**，**代入**，**比較**: O(桁数)
- **四則演算**(+, −,×,/) O(桁数)～ O(桁数^2)
- **論理演算**(¬,∧,∨, ⇒) $O(1)$/1演算
- 繰り返し(for, while)は繰り返し中の演算の総数
- 「パラメータ」
  - 固定(定数)なら$O(1)$
  - 可変なら四則演算に準ずる
- 「関数(ライブラリ)」
  - 計算量が解明されていれば，それに準拠
  - オラクル計算量では，通常の演算数とは別に数える

計算機科学のMCMC法

計算量クラス #P [Valiant '79]

$\mathcal{L}$がNPに属するとき，$w \in \mathcal{L}$に対し，

$\mathcal{C}(w) \subseteq \Sigma^*$を$w$のcertificateの集合とする．

$|\mathcal{C}(w)|$を求める問題のクラスを#Pと呼ぶ．

1998年Gödel prize

Cf. 定理 [戸田1991]

$P^{\#P} = PH$

"Valiant計画" (since 1979)

#P問題$|\mathcal{C}(w)| \in \mathbb{R}$に対して，近似解$Z \in \mathbb{R}$を精度

$$\Pr[(1-\epsilon)|\mathcal{C}(w)| \leq Z \leq (1+\epsilon)|\mathcal{C}(w)|] \geq 1-\delta$$

で多項式時間$(\mathrm{poly}(\mathrm{input}, \epsilon^{-1}, \log \delta^{-1}))$で求められるか？

Fully Polynomial-time Randomized Approximation Scheme (**FPRAS**)

$A \in \mathbb{R}$に対して，近似解$Z \in \mathbb{R}$を精度

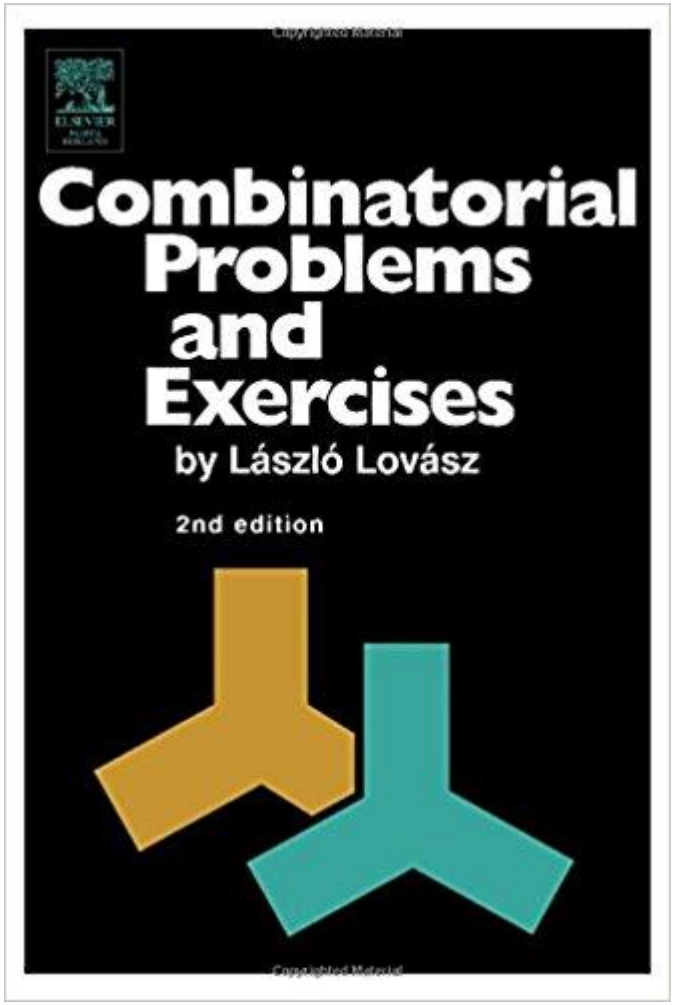
$$\Pr[(1-\epsilon)A \leq Z \leq (1+\epsilon)A] \geq 1-\delta$$

で多項式時間$(\mathrm{poly}(\mathrm{input}, \epsilon^{-1}, \log \delta^{-1}))$で求めるアルゴリズム．

次の時間の導入

組合せの数

Counting is a foundation of Combinatorics



Combinatorial Problems and Exercise [Lovasz 1979]

§1. Basic Enumeration

1. In a shop there are k kinds of postcards. We want to send postcards to n friends.

(i) How many different ways can this be done?

➢

(ii) What happens if we want to send them different cards?

➢

(iii) What happens if we want to send two different cards to each of them (but different persons may get the same card)?

➢

Combinatorial Problems and Exercise [Lovasz 1979]

§1. Basic Enumeration

1. In a shop there are k kinds of postcards. We want to send postcards to n friends.

(i) How many different ways can this be done?

- k^n

(ii) What happens if we want to send them different cards?

-

(iii) What happens if we want to send two different cards to each of them (but different persons may get the same card)?

-

Combinatorial Problems and Exercise [Lovasz 1979]

§1. Basic Enumeration

1. In a shop there are k kinds of postcards. We want to send postcards to n friends.

(i) How many different ways can this be done?

- k^n

(ii) What happens if we want to send them different cards?

- $\frac{k!}{(k-n)!}$ (which is 0 if $n > k$)

(iii) What happens if we want to send two different cards to each of them (but different persons may get the same card)?

-

Combinatorial Problems and Exercise [Lovasz 1979]

§1. Basic Enumeration

1. In a shop there are k kinds of postcards. We want to send postcards to n friends.

(i) How many different ways can this be done?

- k^n

(ii) What happens if we want to send them different cards?

- $\frac{k!}{(k-n)!}$ (which is 0 if $n > k$)

(iii) What happens if we want to send two different cards to each of them (but different persons may get the same card)?

- $\binom{k}{2}^n$

Combinatorial Problems and Exercise [Lovasz 1979]

§1. Basic Enumeration

2. We have k distinct post cards and want to send them all to our n friends (a friend can get any number of post cards, including 0).

(i) How many ways can this be done?

➢

(ii) What happens if we want to send at least one card to each friend?

➢

Combinatorial Problems and Exercise [Lovasz 1979]

§1. Basic Enumeration

2. We have k distinct post cards and want to send them all to our n friends (a friend can get any number of post cards, including 0).

(i) How many ways can this be done?

- n^k

(ii) What happens if we want to send at least one card to each friend?

-

Combinatorial Problems and Exercise [Lovasz 1979]

§1. Basic Enumeration

2. We have k distinct post cards and want to send them all to our n friends (a friend can get any number of post cards, including 0).

(i) How many ways can this be done?

- n^k

(ii) What happens if we want to send at least one card to each friend?

- $n! \cdot \left\{ {k \atop n} \right\}$

Stirling number of the second kind $\left\{ {k \atop n} \right\}$ counts the number of ways to partition a set of k elements into n nonempty subsets.

$$\left\{ {k \atop n} \right\} = \left\{ {k-1 \atop n-1} \right\} + n \left\{ {k-1 \atop n} \right\}$$

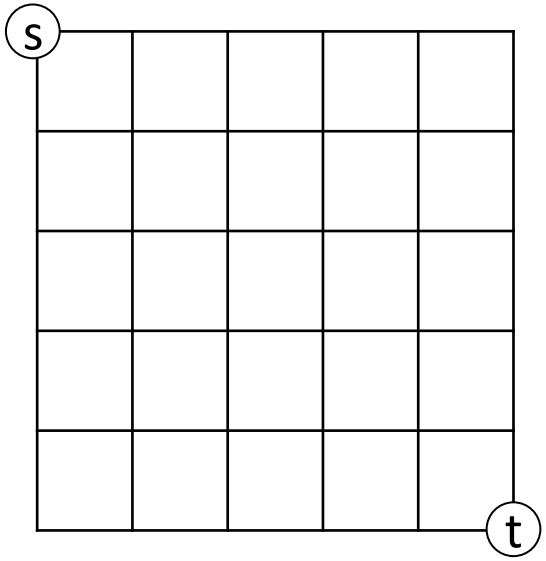
holds. [Wikipedia "Stirling number"]

Another interesting counting problem in [Lovasz 1979]

§1. Basic Enumeration

32. How many shortest paths from s to t in the $n \times n$ grid?

➢



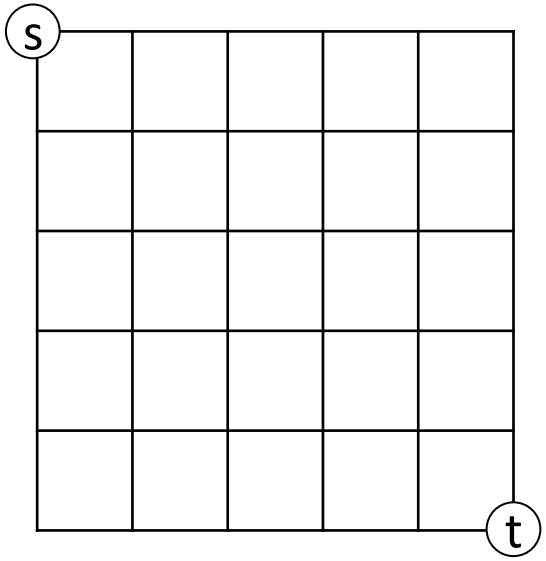
5×5 grid

Another interesting counting problem in [Lovasz 1979]

§1. Basic Enumeration

32. How many shortest paths from s to t in the $n \times n$ grid?

- $\binom{2n}{n}$



5×5 grid

Another interesting counting problem in [Lovasz 1979]

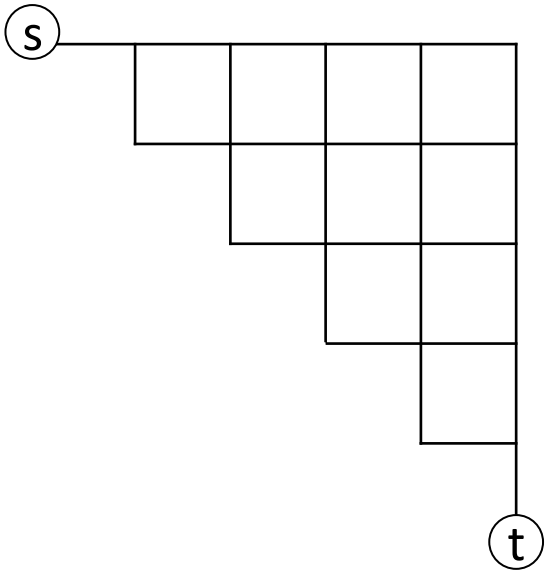
§1. Basic Enumeration

32. How many shortest paths from s to t in the $n \times n$ grid?

- $\binom{2n}{n}$

33. How many shortest paths from s to t in the $n \times n$ grid upper than diagonal?

-



5×5 grid

Another interesting counting problem in [Lovasz 1979]

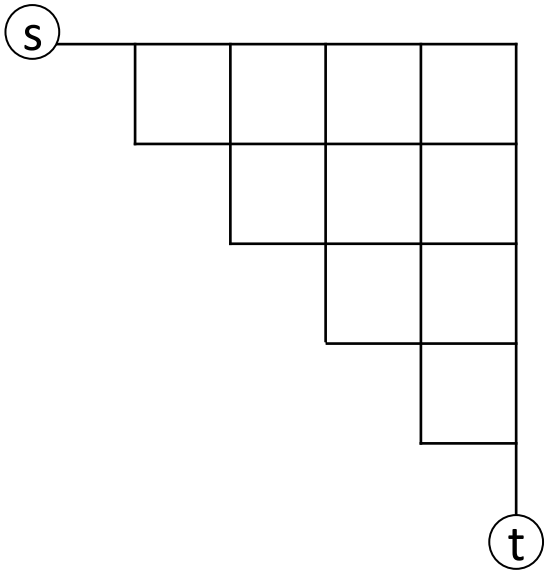
§1. Basic Enumeration

32. How many shortest paths from s to t in the $n \times n$ grid?

- $\binom{2n}{n}$

33. How many shortest paths from s to t in the $n \times n$ grid upper than diagonal?

- $\binom{2n}{n} - \binom{2n}{n-1} = \frac{2n!}{n!(n+1)!}$ (Catalan number)



5×5 grid

Counting is a foundation of Combinatorics

#permutations of n elements is $n!$

#k combinations of n elements is $\binom{n}{k} = \frac{n!}{(n-k)!k!}$

#Dyck path = #binary trees = #proper parentheses = …

is known as Catalan number = $\binom{2n}{n} - \binom{2n}{n-1} = \frac{2n!}{n!(n+1)!}$

spanning trees $\Rightarrow$ Matrix tree theorem

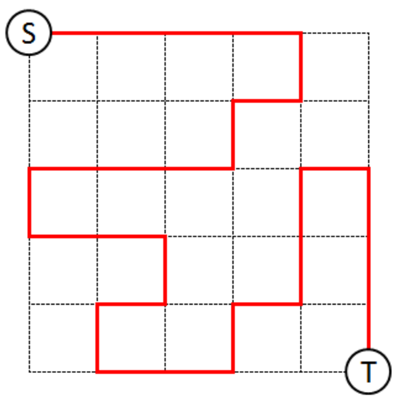
Many known formula for counting combinatorial objects
(while some of them do not have an "explicit form"
…such as Stirling number)

… And, many more combinatorial objects

for which efficient way to count is not known.

おねえさん問題

$n \times n$格子グラフ上で，左上(s)から右下(t)を結ぶ，
s-t単純経路の総数を求めよ．



画像引用 ： 日本科学未来館,『フカシギの数え方』 おねえさんといっしょ！みんなで数えてみよう！, YouTube, 2012/9/10公開,
https://www.youtube.com/watch?v=Q4gTV4r0zRs

おねえさん問題 youtube　search

0	1
1	2
2	12
3	184
4	8512
5	1262816
6	575780564
7	789360053252
8	3266598486981642
9	41044208702632496804
10	1568758030464750013214100
11	182413291514248049241470885236
12	64528039343270018963357185158482118
13	69450664761521361664274701548907358996488
14	227449714676812739631826459327989863387613323440
15	2266745568862672746374567396713098934866324885408319028
16	68745445609149931587631563132489232824587945968099457285419306
17	6344814611237963971310297540795524400449443986866480693646369387855336
18	1782112840842065129893384946652325275167838065704767655931452474605826692782532
19	15233449717048799930807428103192296908994542553232945557760298667373550605928775692 55844
20	3962892199823037560207299517133362502106339705739463771515237113377010682364035706704472064940398
21	31374751050137102720420538137382214513103312193698723653061351991346433379389385793965576992246021316463868
22	755970286667345339661519123315222619353103732072409481167391410479517925792743631234987038883317634987271171404439792
23	55435429355237477009914318489061437930690379970964331332556958646484008407334885544566386924020875711242060085408513482933945720
24	123717122312070647583387448626735708323730419890129435396787270804849516955159304856413945507921530371918580282125122809266003045813867910 94
25	84029748578811334710070837454368091272960542937753835498247426239370284978982152569291785770839709601216256025060273165497184021064940499783756042 47408
26	1736993158627927293117544042123649890037222958828814060466370372091034241327613476278921819349800610708229622314338049134829002672193112962770873889085390810 8906396

おわり