授業アンケート実施「教員独自の設問」

(1) 情報理工学科の他の科目との連携は
適切だったと思いますか
(5：独立し過ぎ ← 適切 → 重複し過ぎ：1)

(2) 情報理工学科の学科専門科目として
適切な内容だったと思いますか
(5：専門的過ぎ ← 適切 → 概論的過ぎ：1)

(3) 情報理工学科の教職課程（教科「数学」）の
コンピュータ区分の科目として
適切な内容だったと思いますか
(5：専門的過ぎ ← 適切 → 概論的過ぎ：1)

期末試験のお知らせ

7月24日（月）11:00 ～ 12:00
（60分試験）

2-403教室（四谷キャンパス）

- 最終回(7/17)の講義内容まで
- 学生証必携

レポート提出について

- 期日：8月2日（水）20時頃まで
- 内容：
  配布プリントのレポート問題の例のような内容
  及び授業に関連する内容で、
  授業内容の理解または発展的な取組みを
  アピールできるようなもの
- 提出方法：
  ⋆ 市谷本館1階106室前のメイルポスト
  ⋆ 電子メイル

本講義最後の話題は、

計算量

について

問題の難しさを如何に計るか？

重要な難しさのクラス

多項式時間 P $\cdots \exists k : O(n^k)$

- **"事実上計算可能" な難しさ**
- **計算モデルの変更に関して頑健（複数テープ TM などに変更しても不変）**

「しらみつぶし」が入ると大体 $O(2^n)$ **程度以上**

（指数時間 EXP $\cdots \exists k : O(2^{n^k})$**）**

"事実上計算不可能"

例：素数判定・素因数分解

素数判定は、試行除算だと指数時間だが
実は多項式時間で解けるのだった !!

(Agrawal-Kayal-Saxena “PRIMES is in P”)

このような効率の良い素数判定は、
具体的に素因数を見付けている訳ではない

素因数分解は P であるかどうか未解決
（多項式時間アルゴリズムが知られていない）

現状で知られているのは、
“準指数時間” $L_N[u, v]$ $(0 < u < 1)$
のアルゴリズム
（現時点で最高速なのは $u = 1/3$）

計算困難な問題の数理技術としての利用

素因数分解の困難さを利用した暗号方式
$\cdots$ RSA 暗号 (Rivest-Shamir-Adleman)

鍵となる整数 n の素因数分解を
知っていれば解読できるが、
知らないと解読できない

$\longrightarrow$ 詳しくは暗号理論などの授業で

例：巨大な指数の冪の計算

RSA 暗号では、次の計算が必要になる：

$$C \equiv M^e \pmod{N}$$

ここで、 C, M, e, N **はどれも数百桁程度**

(500〜1000 **bit)**

単純に M **を** e **回掛けるのでは、**

数百桁回の乗算（と mod N **の計算）が必要**

$\longrightarrow$ **事実上不可能（指数時間）**

計算を実行するには高速化の工夫が必要

例：巨大な指数の冪の計算

RSA 暗号では、次の計算が必要になる：

$$C \equiv M^e \pmod N$$

ここで、 C, M, e, N **はどれも数百桁程度**
(500〜1000 **bit)**

単純に M **を** e **回掛けるのでは、**
数百桁回の乗算（と $\bmod N$ **の計算）が必要**
$\longrightarrow$ **事実上不可能（指数時間）**

計算を実行するには高速化の工夫が必要

例：巨大な指数の冪の計算

RSA 暗号では、次の計算が必要になる：

$$C \equiv M^e \pmod{N}$$

ここで、 C, M, e, N **はどれも数百桁程度**

(500〜1000 **bit)**

単純に M **を** e **回掛けるのでは、**
数百桁回の乗算（と $\mathrm{mod}\, N$ **の計算）が必要**
$\longrightarrow$ **事実上不可能（指数時間）**

計算を実行するには高速化の工夫が必要

例：巨大な指数の冪の計算（M^e の高速計算）

$$e = e_0 + e_1 \cdot 2 + e_2 \cdot 2^2 + \cdots + e_{n-1} \cdot 2^{n-1} + e_n \cdot 2^n$$

：e の二進法表示（二進 n 桁、$e_i = 0, 1$）

M^{2^k} **は、次の漸化式で、k 回の乗算で計算できる**

- $M^{2^0} = M$
- $M^{2^{k+1}} = (M^{2^k})^2$

M^e **は、**

$$M^e = \prod_{k=0}^{n} (M^{2^k})^{e_k} = \prod_{k\,:\,e_k=1} M^{2^k}$$

により、$O(n)$ 回の乗算で計算できる（多項式時間）

例：巨大な指数の冪の計算（M^e の高速計算）

$$e = e_0 + e_1 \cdot 2 + e_2 \cdot 2^2 + \cdots + e_{n-1} \cdot 2^{n-1} + e_n \cdot 2^n$$

：e の二進法表示（二進 n 桁、$e_i = 0, 1$）

M^{2^k} は、次の漸化式で、k 回の乗算で計算できる

- $M^{2^0} = M$
- $M^{2^{k+1}} = (M^{2^k})^2$

M^e は、

$$M^e = \prod_{k=0}^{n} (M^{2^k})^{e_k} = \prod_{k\,:\,e_k=1} M^{2^k}$$

により、$O(n)$ 回の乗算で計算できる（多項式時間）

例：巨大な指数の冪の計算（M^e の高速計算）

$$e = e_0 + e_1 \cdot 2 + e_2 \cdot 2^2 + \cdots + e_{n-1} \cdot 2^{n-1} + e_n \cdot 2^n$$

：e の二進法表示（二進 n 桁、$e_i = 0, 1$）

M^{2^k} **は、次の漸化式で、k 回の乗算で計算できる**

- $M^{2^0} = M$
- $M^{2^{k+1}} = (M^{2^k})^2$

M^e **は、**

$$M^e = \prod_{k=0}^{n} (M^{2^k})^{e_k} = \prod_{k\,:\,e_k=1} M^{2^k}$$

により、$O(n)$ 回の乗算で計算できる（多項式時間）

例：並べ替え (sorting)

多くの数値データを大きさの順に並べ替える操作

n 個のデータの比較は $\dfrac{n(n-1)}{2}$ 通り

$\longrightarrow$ 全ての組合せを比較しても $O(n^2)$ で済む筈

- **具体的なアルゴリズムは？**
- **もっと早くなる？($o(n^2)$ になる？)**

例：並べ替え (sorting)

多くの数値データを大きさの順に並べ替える操作

n **個のデータの比較は** $\dfrac{n(n-1)}{2}$ **通り**

$\longrightarrow$ **全ての組合せを比較しても** $O(n^2)$ **で済む筈**

- **具体的なアルゴリズムは？**
- **もっと早くなる？($o(n^2)$ になる？)**

例：並べ替え (sorting)

多くの数値データを大きさの順に並べ替える操作

n 個のデータの比較は $\dfrac{n(n-1)}{2}$ 通り

$\longrightarrow$ 全ての組合せを比較しても $O(n^2)$ で済む筈

- 具体的なアルゴリズムは？
- もっと早くなる？($o(n^2)$ になる？)

並べ替えの例：バブルソート

- 端から順に、隣と比べて逆順なら入れ換える（末尾が決まる）
- （末尾を除いて）これを繰り返す

比較回数：$\dfrac{n(n-1)}{2}$ 回 $\longrightarrow$ 計算量 $O(n^2)$

並べ替えの例：バブルソート

- 端から順に、隣と比べて逆順なら入れ換える（末尾が決まる）

- （末尾を除いて）これを繰り返す

比較回数： $\dfrac{n(n-1)}{2}$ **回** $\longrightarrow$ **計算量** $O(n^2)$

並べ替えの例：マージソート

- 半分に分ける
- それぞれをソートする（分割統治）
- 両者を併せる

「それぞれをソート」の部分は再帰を用いる

計算量は？

並べ替えの例：マージソート

- 半分に分ける
- それぞれをソートする（分割統治）
- 両者を併せる

「それぞれをソート」の部分は再帰を用いる

計算量は？

並べ替えの例：マージソート

- 半分に分ける
- それぞれをソートする（分割統治）
- 両者を併せる

「それぞれをソート」の部分は再帰を用いる

計算量は？

並べ替えの例：マージソート

- 半分に分ける
- それぞれをソートする
- 両者を併せる $\longrightarrow$ 計算量は $O(n)$

計算量を $f(n)$ とすると、

$$f(n) = 2f\left(\frac{n}{2}\right) + O(n)$$

$$\longrightarrow f(n) = O(n \log n)$$

並べ替えの例：マージソート

- **半分に分ける**
- **それぞれをソートする**
- **両者を併せる** $\longrightarrow$ **計算量は** $O(n)$

計算量を $f(n)$ **とすると、**

$$f(n) = 2f\left(\frac{n}{2}\right) + O(n)$$

$$\longrightarrow f(n) = O(n \log n)$$

並べ替えの例：マージソート

- **半分に分ける**
- **それぞれをソートする**
- **両者を併せる** $\longrightarrow$ **計算量は** $O(n)$

計算量を $f(n)$ **とすると、**

$$f(n) = 2f\left(\frac{n}{2}\right) + O(n)$$

$$\longrightarrow f(n) = O(n \log n)$$

最悪計算量と平均計算量

計算量の理論では、入力データに対して

「どんな場合でも（最悪でも）これだけで出来る」

というのが計算量の定義（最悪計算量）だが、

実際に計算するには、ランダムなデータに対して

「平均的にはこれだけで出来る」

というのも重要である（平均計算量）

並べ替えの例：クイックソート

- **基準値 (pivot) を選ぶ**
- **それより大きい値と小さい値とに分ける**
- **それぞれをソートする（分割統治）**

計算量は

- **最悪では $O(n^2)$ にしかならない**
- **しかし平均では $O(n \log n)$ で、多くの場合、実際にはその中でもかなり速い**

並べ替えの例：クイックソート

- 基準値 (pivot) を選ぶ
- それより大きい値と小さい値とに分ける
- それぞれをソートする（分割統治）

計算量は

- 最悪では $O(n^2)$ にしかならない
- しかし平均では $O(n \log n)$ で、
  多くの場合、実際にはその中でもかなり速い

並べ替えの例：挿入ソート

実際に扱うデータはランダムとは限らない

ソート済みデータに変更があった場合など、
殆どソートされているデータに対して速い方法

- それまでのデータをソートしておく
- 次のデータを適切な場所に挿入する

最悪計算量は $O(n^2)$ だが、場合によっては使える

並べ替えの例：挿入ソート

実際に扱うデータはランダムとは限らない

ソート済みデータに変更があった場合など、
殆どソートされているデータに対して速い方法

- **それまでのデータをソートしておく**
- **次のデータを適切な場所に挿入する**

最悪計算量は $O(n^2)$ だが、場合によっては使える

並べ替えの例：挿入ソート

実際に扱うデータはランダムとは限らない

**ソート済みデータに変更があった場合など、
殆どソートされているデータに対して速い方法**

- **それまでのデータをソートしておく**
- **次のデータを適切な場所に挿入する**

最悪計算量は $O(n^2)$ だが、場合によっては使える

さて、

いよいよ、

非決定性計算量と ”P vs NP” 問題へ

非決定性計算モデルでの計算量

計算量にも “非決定性” の概念がある

あてずっぽうを許して、
うまくいけばどの位で解けるか
＝ 答を知って、その検証にどの位かかるか

非決定性多項式時間 (NP)：
非決定性の計算モデルで多項式時間で解ける

例：素因数分解は NP
$\cdots$ 素因数を知っていれば割算するだけ

非決定性計算モデルでの計算量

計算量にも “非決定性” の概念がある

あてずっぽうを許して、
うまくいけばどの位で解けるか
＝ 答を知って、その検証にどの位かかるか

非決定性多項式時間 (NP)：
非決定性の計算モデルで多項式時間で解ける

例：素因数分解は NP
… 素因数を知っていれば割算するだけ

非決定性計算モデルでの計算量

計算量にも “非決定性” の概念がある

あてずっぽうを許して、
うまくいけばどの位で解けるか
＝ 答を知って、その検証にどの位かかるか

非決定性多項式時間 (NP)：
非決定性の計算モデルで多項式時間で解ける

例：素因数分解は NP
$\cdots$ 素因数を知っていれば割算するだけ

非決定性計算モデルでの計算量

$$\mathsf{P} \subset \mathsf{NP} \subset \mathsf{EXP}$$

未解決問題 (P vs NP Problem)

$$\mathsf{P} = \mathsf{NP}$$

であるか否か？

“The Millennium Problems”
の 7 つの問題のうちの 1 つ
（賞金 $1M）

非決定性計算モデルでの計算量

$$\mathrm{P} \subset \mathrm{NP} \subset \mathrm{EXP}$$

未解決問題 (P vs NP Problem)

$$\mathrm{P} = \mathrm{NP}$$

であるか否か？

"The Millennium Problems"

の 7 つの問題のうちの 1 つ

(賞金 $1M)

非決定性計算モデルでの計算量

$$\mathbf{P} \subset \mathbf{NP} \subset \mathbf{EXP}$$

未解決問題 (P vs NP Problem)

$$\mathbf{P} = \mathbf{NP}$$

であるか否か？

"The Millennium Problems"
の 7 つの問題のうちの 1 つ
(賞金 $1M)

参考：The Millennium Problems

2000 年に Clay 数学研究所 (CMI) により
賞金 $1M が懸けられた 7 つの問題

- **Birch and Swinnerton-Dyer 予想**
- **Hodge 予想**
- **Navier-Stokes 方程式の解の存在と微分可能性**
- **P vs NP 予想**
- **Poincarè 予想（Perelman により解決 (2003)）**
- **Riemann 予想**
- **Yang-Mills 方程式と質量ギャップ問題**

多項式時間帰着可能性

問題 B が問題 A に多項式時間帰着可能
$\overset{\leftarrow}{\iff}$ $\exists f : \Sigma^* \longrightarrow \Sigma^*$：

- f：**多項式時間で計算可能**
  （多項式時間で計算する TM が存在）
- $w \in B \iff f(w) \in A$

- 問題 B が問題 A に多項式時間帰着可能のとき、
  $A \in P \Longrightarrow B \in P$

多項式時間帰着可能性

問題 B が問題 A に多項式時間帰着可能
$\overset{\leftarrow}{\iff} \exists f : \Sigma^* \longrightarrow \Sigma^*$：

- f：**多項式時間で計算可能**
  （多項式時間で計算する TM が存在）
- $w \in B \iff f(w) \in A$

- **問題 B が問題 A に多項式時間帰着可能のとき、**
  $\mathbf{A} \in \mathbf{P} \Longrightarrow \mathbf{B} \in \mathbf{P}$

NP 完全・NP 困難

- 問題 A が NP 困難 (NP-hard)
  $\overset{\leftarrow}{\Longleftrightarrow}$ 全ての NP 問題 B が
  問題 A に多項式時間帰着可能

- 問題 A が NP 完全 (NP-complete)
  $\overset{\leftarrow}{\Longleftrightarrow}$ 問題 A が NP かつ NP 困難

或る NP 完全な問題 A が P $\Longleftrightarrow$ P $=$ NP

NP 完全・NP 困難

- 問題 A が NP 困難 (NP-hard)
  $\stackrel{\leftarrow}{\Longleftrightarrow}$ 全ての NP 問題 B が
  問題 A に多項式時間帰着可能

- 問題 A が NP 完全 (NP-complete)
  $\stackrel{\leftarrow}{\Longleftrightarrow}$ 問題 A が NP かつ NP 困難

或る NP 完全な問題 A が P $\Longleftrightarrow$ P $=$ NP

充足可能性問題 (SAT)

NOT, OR, AND **からなる論理式**

$$f(A_1, \ldots, A_n)$$ **に対し、**

$f(a_1, \ldots, a_n) = 1$ **となる変数** A_i **の真理値の組**

$$(a_1, \ldots, a_n) \in \{0, 1\}^n$$

が存在するか？

定理 (Cook-Levin)

SAT は NP 完全

充足可能性問題 (SAT)

NOT, OR, AND からなる論理式 $f(A_1, \ldots, A_n)$ に対し、$f(a_1, \ldots, a_n) = 1$ となる変数 A_i の真理値の組 $(a_1, \ldots, a_n) \in \{0, 1\}^n$ が存在するか？

定理 (Cook-Levin)

SAT は NP 完全

論理積標準形

NOT, OR, AND **からなる全ての論理式は次の形で表せる：**

$$f(A_1, \ldots, A_n) = (X_{11} \vee \cdots \vee X_{1t_1}) \wedge \cdots \wedge (X_{s1} \vee \cdots \vee X_{st_s})$$

(各 X_{ij} **は** A_k **または** $\neg A_k$**)**

$\cdots$ **論理積標準形・連言標準形**
(conjunctive normal form, CNF)

特に、$\forall i : t_i = 3$ **となる論理式** $\cdots$ 3-**CNF**

3-充足可能性問題 (3-SAT)

3-**CNF** $f(A_1, \ldots, A_n)$ **に対し、**
$f(a_1, \ldots, a_n) = 1$ **となる変数** A_i **の真理値の組**

$$(a_1, \ldots, a_n) \in \{0, 1\}^n$$

が存在するか？

定理

- **SAT は** 3-**SAT に多項式時間帰着可能**
- **従って、**3-**SAT も NP 完全**

3-充足可能性問題 (3-SAT)

3-**CNF** $f(A_1, \ldots, A_n)$ **に対し、**
$f(a_1, \ldots, a_n) = 1$ **となる変数** A_i **の真理値の組**

$$(a_1, \ldots, a_n) \in \{0, 1\}^n$$

が存在するか？

定理

- **SAT は** 3-**SAT に多項式時間帰着可能**
- **従って、**3-**SAT も NP 完全**

他にも沢山の NP 完全問題が知られている

例えば、次のパズルの解の存在判定は全て NP 完全

- 数独
- カックロ
- ののぐらむ
- スリザーリンク
- ナンバーリンク
- ぬりかべ
- 美術館
- ましゅ
- 天体ショー
- フィルオミノ　　　　　　など

おしまい