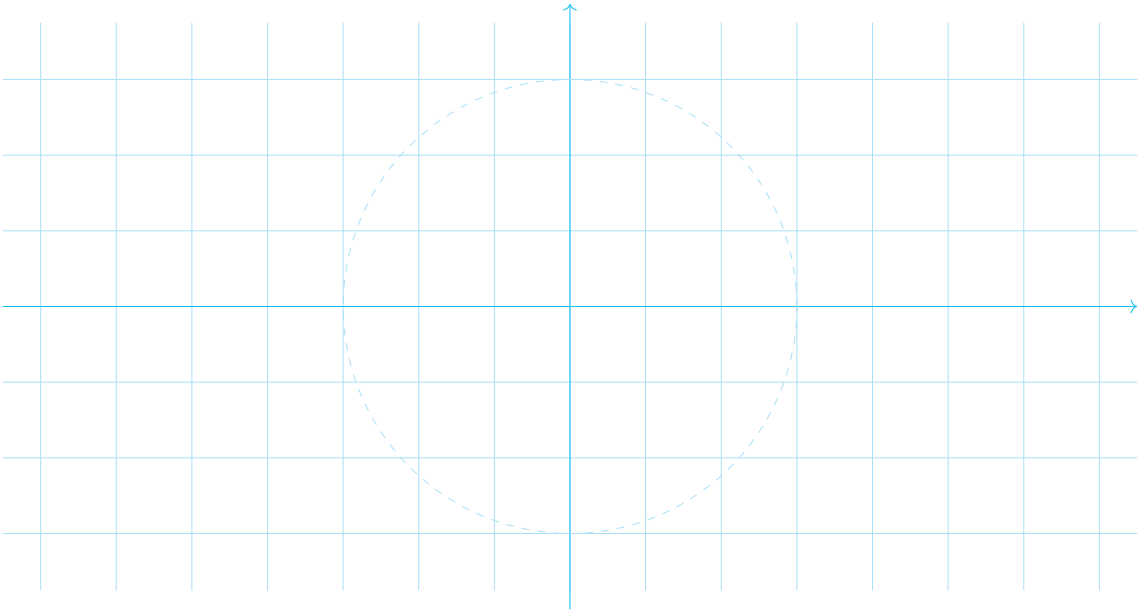
memo



離散数学

授業開始までしばらくお待ちください。

2024 離散数学
Discrete Mathematics
『代数系 I』

bit.ly/d-math

小林裕之

大阪工業大学 RD 学部システムデザイン工学科

OSAKA INSTITUTE OF TECHNOLOGY

9 of 14

a LaTeX + Beamer slideshow

授業の受講に関して

- 講義資料（スライド等）は **Google Drive** (`https://bit.ly/d-math`) に置く (紙の配布資料は行わない)。授業前には虫喰い状態のスライドのみを提供するが、授業後に `uncovered`フォルダに穴埋め版を置くので復習に活用されたい。
- ミニレポートは**Google Forms** (`https://forms.gle/hCyJBbFBMW9AisAt7`) に提出。
- 授業の録画はできるだけスライドと同じフォルダ内のフォルダに置くように努力する（が、必ず置きますとお約束はしません)。
- 授業中に計算間違い等を指摘してくれたらその都度 1 点。(内容に依るけど。)

成績評価について

- 出席そのものは評価せず。極論するとテストのみ出席で他は全欠席でも A 評価はあり得る。
- 基本的には**中間演習**と**期末試験**で評価。
- 毎回ミニレポートを課す。出す者は提出期間を厳守すること。
- 試験の不合格者は**毎回のミニレポート**と**出席**で少し救済する。
  (**しっかりした内容の**ミニレポートを概ね 9 割以上提出し、かつ大学の出欠管理システムで 8 割以上遅刻せず出席していた場合最大 10 点程度の救済。提出数や出席数が少ない場合は救済幅が縮小する。いずれかが 7 割を下回ったら一切救済しない。締め切り後の提出は認めない。)
- **授業中に**スライドの誤りを見つけて指摘してくれた者には、誤り一箇所につき先着一名様限り 100 点満点 1 点相当の加点を行う。(ただしごく軽微なものなど、内容によっては加点しない場合もあり。)

代数系

- 二項演算とは、　　から集合 B への写像を A **上の (二項) 演算**言う。
- 例えば『$\star$』という二項演算があったとすると、$\star$:
- 要素についてはこう書く:

例: $+$ は $\mathbb{N}$ 上の二項演算である。

二項演算とその表し方

本題に入るその前に…。

A の直積集合

- 二項演算とは、$A \times A$ から集合 B への写像を A **上の (二項) 演算**言う。
- 例えば『$\star$』という二項演算があったとすると、$\star$:
- 要素についてはこう書く:

例: $+$ は $\mathbb{N}$ 上の二項演算である。

A の直積集合

- 二項演算とは、$A \times A$ から集合 B への写像を A **上の (二項) 演算**言う。
- 例えば『$\star$』という二項演算があったとすると、$\star : A \times A \to B$
- 要素についてはこう書く:

例: $+$ は $\mathbb{N}$ 上の二項演算である。

二項演算とその表し方

本題に入るその前に…。

A の直積集合

- 二項演算とは、$A \times A$ から集合 B への写像を A **上の (二項) 演算**言う。
- 例えば『$\star$』という二項演算があったとすると、$\star : A \times A \to B$
- 要素についてはこう書く: $\langle a, b \rangle \mapsto c$

例: $+$ は $\mathbb{N}$ 上の二項演算である。

閉じた演算

A 上の二項演算 $\star$ が、$\forall a, b \in A$ に対して

であるとき、$\star$ は　　という。

例:

- $\times$ は $\mathbb{N}$ 上で閉じて
- $-$ は $\mathbb{N}$ 上で閉じて

閉じた演算

A 上の二項演算 $\star$ が、$\forall a, b \in A$ に対して

$$a \star b \in A$$

であるとき、$\star$ は　　という。

例:

- $\times$ は $\mathbb{N}$ 上で閉じて
- $-$ は $\mathbb{N}$ 上で閉じて

閉じた演算

A 上の二項演算 $\star$ が、$\forall a, b \in A$ に対して

$$a \star b \in A$$

であるとき、$\star$ は**閉じている**という。

例:

- $\times$ は $\mathbb{N}$ 上で閉じて
- $-$ は $\mathbb{N}$ 上で閉じて

閉じた演算

A 上の二項演算 $\star$ が、$\forall a, b \in A$ に対して

$$a \star b \in A$$

であるとき、$\star$ は**閉じている**という。

例:

- $\times$ は $\mathbb{N}$ 上で閉じている。
- $-$ は $\mathbb{N}$ 上で閉じて

閉じた演算

A 上の二項演算 $\star$ が、$\forall a, b \in A$ に対して

$$a \star b \in A$$

であるとき、$\star$ は**閉じている**という。

例:

- $\times$ は $\mathbb{N}$ 上で閉じている。
- $-$ は $\mathbb{N}$ 上で閉じていない。

剰余演算

前も軽くやったけどあらためて

ある整数を整数 p で割った余りを p **を法とした剰余**といい、$\bmod p$ を使って表す。
(例)

- $10 \mod 3 =$
- $121 \mod 11 =$

剰余の重要な性質:

$$(a \cdot b) \mod p = (a \mod p) \cdot (b \mod p) \mod p$$

mini Q. 証明せよ。(ヒント: $a = np + r_a, b = mp + r_b$(ただし n, m は整数) として考える)

剰余演算

前も軽くやったけどあらためて

ある整数を整数 p で割った余りを p **を法とした剰余**といい、$\bmod p$ を使って表す。
(例)

- $10 \mod 3 = 1$
- $121 \mod 11 =$

剰余の重要な性質:

$$(a \cdot b) \mod p = (a \mod p) \cdot (b \mod p) \mod p$$

mini Q. 証明せよ。(ヒント: $a = np + r_a, b = mp + r_b$(ただし n, m は整数) として考える)

剰余演算

前も軽くやったけどあらためて

ある整数を整数 p で割った余りを p **を法とした剰余**といい、$\bmod p$ を使って表す。
(例)

- $10 \mod 3 = 1$
- $121 \mod 11 = 0$

剰余の重要な性質:

$$(a \cdot b) \mod p = (a \mod p) \cdot (b \mod p) \mod p$$

mini Q. 証明せよ。(ヒント: $a = np + r_a, b = mp + r_b$(ただし n, m は整数) として考える)

剰余和と剰余積

通常の和と積の計算をして、最後に剰余を求めたものを p **を法とした剰余和**, p **を法とした剰余積** という。

記号は $\oplus_p$, $\odot_p$ を使う。

剰余和と剰余積

通常の和と積の計算をして、最後に剰余を求めたものを p **を法とした剰余和**, p **を法とした剰余積** という。

- $m \oplus_p n \stackrel{\triangle}{=} (m+n) \mod p$

記号は $\oplus_p$, $\odot_p$ を使う。

剰余和と剰余積

通常の和と積の計算をして、最後に剰余を求めたものを p **を法とした剰余和**, p **を法とした剰余積** という。

- $m \oplus_p n \stackrel{\triangle}{=} (m+n) \mod p$
- $m \odot_p n \stackrel{\triangle}{=} (m \times n) \mod p$

記号は $\oplus_p$, $\odot_p$ を使う。

範囲つき整数集合

単なる記号の定義のはなし

$\mathbb{Z}_n \triangleq \{0, 1, \cdots, n-1\}$ とする。

以上。

コラム: 閉じる。

- $\mathbb{Z}_N$ 上での演算を**閉じたい**ことはいっぱいある。
- 手っ取り早く**閉じる**には　　が便利。
- 例: 演算 $x \star v$ は、自機の位置 x と速度値 v を使って移動先を求めるのに使う演算である。この計算結果をスクリーン座標の範囲 (0〜79 とか) に**閉じる**ようにしたい。

```
WIDTH = 80                  # 移動範囲 (0-79)
def move(x, v):
  x = (x + v) % WIDTH       # +演算を {0, 1, ..., 79} に閉じた！
  return x
```

コラム: 閉じる。

- $\mathbb{Z}_N$ 上での演算を**閉じたい**ことはいっぱいある。
- 手っ取り早く**閉じる**には **剰余演算** (例: Python なら%)が便利。
- 例: 演算 $x \star v$ は、自機の位置 x と速度値 v を使って移動先を求めるのに使う演算である。この計算結果をスクリーン座標の範囲 (0〜79 とか) に**閉じる**ようにしたい。

```
WIDTH = 80                  # 移動範囲 (0-79)
def move(x, v):
  x = (x + v) % WIDTH       # +演算を {0, 1, ..., 79} に閉じた！
  return x
```

演算を表形式で書き下したものを**演算表**という。

$\{a, b, c\}$ 上の演算 $\star$ の演算表

$\star$	a	b	c
a	$a \star a$	$a \star b$	$a \star c$
b	$b \star a$	$b \star b$	$b \star c$
c	$c \star a$	$c \star b$	$c \star c$

4 進数一桁のかけ算の演算表

$\times$	0	1	2	3
0				
1				
2				
3				

演算表

かけ算九九の表もこれ。

演算を表形式で書き下したものを**演算表**という。

$\{a, b, c\}$ 上の演算 $\star$ の演算表

$\star$	a	b	c
a	$a \star a$	$a \star b$	$a \star c$
b	$b \star a$	$b \star b$	$b \star c$
c	$c \star a$	$c \star b$	$c \star c$

4 進数一桁のかけ算の演算表

$\times$	0	1	2	3
0	0	0	0	0
1				
2				
3				

演算表

かけ算九九の表もこれ。

演算を表形式で書き下したものを**演算表**という。

$\{a, b, c\}$ 上の演算 $\star$ の演算表

$\star$	a	b	c
a	$a \star a$	$a \star b$	$a \star c$
b	$b \star a$	$b \star b$	$b \star c$
c	$c \star a$	$c \star b$	$c \star c$

4 進数一桁のかけ算の演算表

$\times$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2				
3				

演算表

かけ算九九の表もこれ。

演算を表形式で書き下したものを**演算表**という。

$\{a, b, c\}$ 上の演算 $\star$ の演算表

$\star$	a	b	c
a	$a \star a$	$a \star b$	$a \star c$
b	$b \star a$	$b \star b$	$b \star c$
c	$c \star a$	$c \star b$	$c \star c$

4 進数一桁のかけ算の演算表

$\times$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	10	12
3	0	3	12	21

演算の演習

$\mathbb{Z}_4$ 上のふたつの演算 $\oplus_4$ と $\odot_4$ の演算表を作成せよ。

演算の演習

$\mathbb{Z}_4$ 上のふたつの演算 $\oplus_4$ と $\odot_4$ の演算表を作成せよ。

$\oplus_4$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

演算の演習

$\mathbb{Z}_4$ 上のふたつの演算 $\oplus_4$ と $\odot_4$ の演算表を作成せよ。

$\oplus_4$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$\odot_4$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

代数系とは?

CAS の A は Algebraic の A。

代数系 **algebraic system**

と、その上で をセットで
代数系という。

書き方:

(集合; 演算)

代数系とは?

CAS の A は Algebraic の A。

代数系 **algebraic system**

集合 ($\neq \emptyset$)と、その上で　　をセットで
代数系という。

書き方:

(集合; 演算)

代数系とは?

CAS の A は Algebraic の A。

代数系 **algebraic system**

集合 ($\neq \emptyset$)と、その上で**閉じた演算**をセットで**代数系**という。

書き方:

$$(\text{集合}; \text{演算})$$

代数系とは?

CAS の A は Algebraic の A。

代数系 **algebraic system**

集合 ($\neq \emptyset$)と、その上で**閉じた演算**をセットで**代数系**という。

書き方:

$$(\text{集合}; \text{演算})$$

うーん、なんだか抽象的すぎてよくわからん。

具体的な代数系の例

クイズ: この中にひとつだけ代数系でないものが混ざっています。どれだ？

- $(\mathbb{N};\ +)$
- $(\mathbb{Z};\ -)$
- $(\mathbb{R};\ +,\ -,\ \times)$
- $(\mathbb{Q};\ +,\ \times)$
- $(\{0,\ 1\};\ \times)$
- $(\mathbb{C};\ +,\ -,\ \times)$
- $(\mathbb{N};\ +,\ -,\ \times)$
- $(\{F,\ T\};\ \vee,\ \wedge)$

具体的な代数系の例

クイズ: この中にひとつだけ代数系でないものが混ざっています。どれだ？

- $(\mathbb{N};\ +)$
- $(\mathbb{Z};\ -)$
- $(\mathbb{R};\ +,\ -,\ \times)$
- $(\mathbb{Q};\ +,\ \times)$
- $(\{0,\ 1\};\ \times)$
- $(\mathbb{C};\ +,\ -,\ \times)$
- $(\mathbb{N};\ +,\ -,\ \times)$ 

  減算が**閉じていない**。
- $(\{F,\ T\};\ \vee,\ \wedge)$

作ってみよう! オレオレ代数系。

集合と、その上で閉じた演算さえ定義すれば何でもアリ。

1. **集合を考える。**　「あまり大きいと大変そうなので 3 要素くらいにしとこう…。」

2. **演算を考える。**　「いくつでもいいなら、じゃとりあえず 2 つ　で。」
   演算の具体的なルールは**演算表**で示せば良い。

3. **完成！**→

作ってみよう! オレオレ代数系。

集合と、その上で閉じた演算さえ定義すれば何でもアリ。

1. **集合を考える。**　「あまり大きいと大変そうなので 3 要素くらいにしとこう…。」

$$\{✈, ♥, ❀\}$$

2. **演算を考える。**　「いくつでもいいなら、じゃとりあえず 2 つ　で。」

   演算の具体的なルールは**演算表**で示せば良い。

3. **完成！**→

作ってみよう! オレオレ代数系。

集合と、その上で閉じた演算さえ定義すれば何でもアリ。

1. **集合を考える。**　「あまり大きいと大変そうなので 3 要素くらいにしとこう…。」

$$\{✈, ♥, ❀\}$$

2. **演算を考える。**　「いくつでもいいなら、じゃとりあえず 2 つ ⋆ と ✣ で。」

   演算の具体的なルールは**演算表**で示せば良い。

3. **完成！**→

作ってみよう! オレオレ代数系。

集合と、その上で閉じた演算さえ定義すれば何でもアリ。

1. **集合を考える。**　「あまり大きいと大変そうなので 3 要素くらいにしとこう…。」

$$\{✈, ♥, ❀\}$$

2. **演算を考える。**　「いくつでもいいなら、じゃとりあえず 2 つ ⋆ と ✣ で。」

演算の具体的なルールは**演算表**で示せば良い。

⋆	✈	♥	❀
✈	✈	♥	✈
♥	♥	❀	✈
❀	❀	✈	❀

3. **完成！**→

作ってみよう! オレオレ代数系。

集合と、その上で閉じた演算さえ定義すれば何でもアリ。

1. **集合を考える。**　「あまり大きいと大変そうなので 3 要素くらいにしとこう…。」

$$\{✈, ♥, ❀\}$$

2. **演算を考える。**　「いくつでもいいなら、じゃとりあえず 2 つ ⋆ と ✥ で。」

演算の具体的なルールは**演算表**で示せば良い。

⋆	✈	♥	❀
✈	✈	♥	✈
♥	♥	❀	✈
❀	❀	✈	❀

✥	✈	♥	❀
✈	✈	♥	❀
♥	❀	♥	❀
❀	♥	✈	✈

3. **完成！**→

作ってみよう! オレオレ代数系。

集合と、その上で閉じた演算さえ定義すれば何でもアリ。

1. 集合を考える。 「あまり大きいと大変そうなので 3 要素くらいにしとこう…。」

$$\{✈, ♥, ❀\}$$

2. 演算を考える。 「いくつでもいいなら、じゃとりあえず 2 つ $\star$ と ✣ で。」

演算の具体的なルールは**演算表**で示せば良い。

$\star$	✈	♥	❀
✈	✈	♥	✈
♥	♥	❀	✈
❀	❀	✈	❀

✣	✈	♥	❀
✈	✈	♥	❀
♥	❀	♥	❀
❀	♥	✈	✈

3. 完成！ $\to (\{✈, ♥, ❀\}; \star, ✣)$

演算の性質 (1)

演算の世界にもいろいろある礼儀作法。たくさん身につければより行儀良く振る舞う。まずは基本中の基本。

- $(a \star b) \star c = a \star (b \star c)$

  成り立たない演算の例: –

- $a \star b = b \star a$

  成り立たない演算の例: –

※結合律を満たす演算については今後 $(a \star b) \star c$ を $a \star b \star c$ と書く場合もある。

演算の性質 (1)

演算の世界にもいろいろある礼儀作法。たくさん身につければより行儀良く振る舞う。まずは基本中の基本。

- $(a \star b) \star c = a \star (b \star c)$

結合律

成り立たない演算の例: –

- $a \star b = b \star a$

成り立たない演算の例: –

※結合律を満たす演算については今後 $(a \star b) \star c$ を $a \star b \star c$ と書く場合もある。

演算の性質 (1)

演算の世界にもいろいろある礼儀作法。たくさん身につければより行儀良く振る舞う。まずは基本中の基本。

- $(a \star b) \star c = a \star (b \star c)$ **結合律**

  成り立たない演算の例: –

- $a \star b = b \star a$ **交換律**

  成り立たない演算の例: –

※結合律を満たす演算については今後 $(a \star b) \star c$ を $a \star b \star c$ と書く場合もある。

練習

問: 以下 $\mathbb{R}$ 上の演算 $\star$ について、結合律・交換律はそれぞれ成り立つかどうか？

1. $a \star b = a^b$
2. $a \star b = |a - b|$

問

演算 $\star$ が、$a \star b \stackrel{\triangle}{=} b$ であるなら $\star$ は結合律を満たす。証明せよ。

練習

問: 以下 $\mathbb{R}$ 上の演算 $\star$ について、結合律・交換律はそれぞれ成り立つかどうか？

1. $a \star b = a^b$ 両方成り立たない。

2. $a \star b = |a - b|$

問

演算 $\star$ が、$a \star b \stackrel{\triangle}{=} b$ であるなら $\star$ は結合律を満たす。証明せよ。

練習

問: 以下 $\mathbb{R}$ 上の演算 $\star$ について、結合律・交換律はそれぞれ成り立つかどうか？

1. $a \star b = a^b$ 両方成り立たない。

2. $a \star b = |a - b|$ 結合律は成り立たない。交換律は成り立つ。

問

演算 $\star$ が、$a \star b \stackrel{\triangle}{=} b$ であるなら $\star$ は結合律を満たす。証明せよ。

練習

問: 以下 $\mathbb{R}$ 上の演算 $\star$ について、結合律・交換律はそれぞれ成り立つかどうか？

1. $a \star b = a^b$ 両方成り立たない。

2. $a \star b = |a - b|$ 結合律は成り立たない。交換律は成り立つ。

問

演算 $\star$ が、$a \star b \overset{\triangle}{=} b$ であるなら $\star$ は結合律を満たす。証明せよ。

$(a \star b) \star c = b \star c = c, a \star (b \star c) = a \star c = c$ よって結合律を満たす。

剰余和・剰余積の性質

問: 剰余和について考えよ。

- **剰余和**は**結合律**$(a \oplus b) \oplus c = a \oplus (b \oplus c)$ を満たすか？
- **剰余和**は**交換律**$a \oplus b = b \oplus a$ を満たすか？

問: 剰余積について考えよ。

- **剰余積**は**結合律**$(a \odot b) \odot c = a \odot (b \odot c)$ を満たすか？
- **剰余積**は**交換律**$a \odot b = b \odot a$ を満たすか？

剰余和・剰余積の性質

問: 剰余和について考えよ。

- **剰余和**は**結合律** $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ を満たすか？

  自明ではないような気もするけれど **yes**

- **剰余和**は**交換律** $a \oplus b = b \oplus a$ を満たすか？

問: 剰余積について考えよ。

- **剰余積**は**結合律** $(a \odot b) \odot c = a \odot (b \odot c)$ を満たすか？

- **剰余積**は**交換律** $a \odot b = b \odot a$ を満たすか？

剰余和・剰余積の性質

問: 剰余和について考えよ。

- **剰余和**は**結合律**$(a \oplus b) \oplus c = a \oplus (b \oplus c)$ を満たすか？

  自明ではないような気もするけれど **yes**

- **剰余和**は**交換律**$a \oplus b = b \oplus a$ を満たすか？ **yes**

問: 剰余積について考えよ。

- **剰余積**は**結合律**$(a \odot b) \odot c = a \odot (b \odot c)$ を満たすか？

- **剰余積**は**交換律**$a \odot b = b \odot a$ を満たすか？

剰余和・剰余積の性質

問: 剰余和について考えよ。

- **剰余和**は**結合律**$(a \oplus b) \oplus c = a \oplus (b \oplus c)$ を満たすか？

  自明ではないような気もするけれど **yes**

- **剰余和**は**交換律**$a \oplus b = b \oplus a$ を満たすか？ **yes**

問: 剰余積について考えよ。

- **剰余積**は**結合律**$(a \odot b) \odot c = a \odot (b \odot c)$ を満たすか？

  自明ではないような気もするけれど **yes**

- **剰余積**は**交換律**$a \odot b = b \odot a$ を満たすか？

剰余和・剰余積の性質

問: 剰余和について考えよ。

- **剰余和**は**結合律**$(a \oplus b) \oplus c = a \oplus (b \oplus c)$ を満たすか？

  自明ではないような気もするけれど **yes**

- **剰余和**は**交換律**$a \oplus b = b \oplus a$ を満たすか？ **yes**

問: 剰余積について考えよ。

- **剰余積**は**結合律**$(a \odot b) \odot c = a \odot (b \odot c)$ を満たすか？

  自明ではないような気もするけれど **yes**

- **剰余積**は**交換律**$a \odot b = b \odot a$ を満たすか？ **yes**

唐突な問題

大きな数を扱うセキュリティ分野ではこの計算テクニックが必須

問

1. 3^{12} を 11 で割った余りはいくつか?
2. 2^{100} を 31 で割った余りはいくつか?
3. 150^{19} を 299 で割った余りはいくつか?

ヒント: 剰余積の結合則とほぼ同様に考えて $(a \cdot b \cdot c) \mod p = a \odot b \odot c$ が言える。

唐突な問題

大きな数を扱うセキュリティ分野ではこの計算テクニックが必須

問

1. 3^{12} を 11 で割った余りはいくつか?
2. 2^{100} を 31 で割った余りはいくつか?
3. 150^{19} を 299 で割った余りはいくつか?

ヒント: 剰余積の結合則とほぼ同様に考えて $(a \cdot b \cdot c) \mod p = a \odot b \odot c$ が言える。

1. $3^{12} = 3^{3\times 4} = 27^4$ より、

唐突な問題

大きな数を扱うセキュリティ分野ではこの計算テクニックが必須

問

1. 3^{12} を 11 で割った余りはいくつか?
2. 2^{100} を 31 で割った余りはいくつか?
3. 150^{19} を 299 で割った余りはいくつか?

ヒント: 剰余積の結合則とほぼ同様に考えて $(a \cdot b \cdot c) \mod p = a \odot b \odot c$ が言える。

1. $3^{12} = 3^{3\times 4} = 27^4$ より、 $27 \odot_{11} 27 \odot_{11} 27 \odot_{11} 27 = 5 \odot_{11} 5 \odot_{11} 5 \odot_{11} 5 = 3 \odot_{11} 3 = 9$

唐突な問題

大きな数を扱うセキュリティ分野ではこの計算テクニックが必須

問

1. 3^{12} を 11 で割った余りはいくつか?
2. 2^{100} を 31 で割った余りはいくつか?
3. 150^{19} を 299 で割った余りはいくつか?

ヒント: 剰余積の結合則とほぼ同様に考えて $(a \cdot b \cdot c) \mod p = a \odot b \odot c$ が言える。

1. $3^{12} = 3^{3\times 4} = 27^4$ より、 $27 \odot_{11} 27 \odot_{11} 27 \odot_{11} 27 = 5 \odot_{11} 5 \odot_{11} 5 \odot_{11} 5 = 3 \odot_{11} 3 = 9$
2. $2^{100} = 2^{5\times 20} = 32^{20}$ より 1

唐突な問題

大きな数を扱うセキュリティ分野ではこの計算テクニックが必須

問

1. 3^{12} を 11 で割った余りはいくつか?
2. 2^{100} を 31 で割った余りはいくつか?
3. 150^{19} を 299 で割った余りはいくつか?

ヒント: 剰余積の結合則とほぼ同様に考えて $(a \cdot b \cdot c) \mod p = a \odot b \odot c$ が言える。

1. $3^{12} = 3^{3\times 4} = 27^4$ より、 $27 \odot_{11} 27 \odot_{11} 27 \odot_{11} 27 = 5 \odot_{11} 5 \odot_{11} 5 \odot_{11} 5 = 3 \odot_{11} 3 = 9$
2. $2^{100} = 2^{5\times 20} = 32^{20}$ より 1
3. p. 21参照。電卓片手に自分でやってみよう。(一部 CAS 電卓などでは結合則云々抜きにそのまま mod 計算できてしまいますが……)

ミニレポート課題 (提出期間: 本日〜次回の授業の前日)

p. 19 の問題を解け。

解答を PC 文書や手書きで作成し、PDF にして Google Forms (**https://forms.gle/hCyJBbFBMW9AisAt7**) から提出せよ (要組織アカウントによるログイン)。ただし写真等の画像ファイルの場合は、解像度や露出・照明状態などを十分考慮し、きちんと読解可能なクオリティのものとすること。スマートフォンの場合はスキャナアプリの類の利用を必須とする。

コラム: 公開鍵暗号技術

一番使われている RSA 暗号とは?

現在最も広く用いらている**公開鍵暗号システム**である**RSA**では、**秘密鍵** Dと**公開鍵** Eおよび法とする**数** Nを用いる。
平文 P を暗号文 C に暗号化する計算は

$$C =$$

で、復号化する計算は

$$P =$$

である。どちらの計算も**剰余積の結合律**を使えば十分現実的に計算できることがわかる。
なお、平文 P、暗号文 C および法とする数 N をもとに**鍵** E **や** D **を逆算**することは、**離散対数問題**と言われていて**とても難しい**。
なお鍵は適当でいいはずはなくちゃんとした条件がある。例えば $(D, E, N) = (7, 19, 299)$ は条件を満たす。**平文『123』**を暗号化しよう。$123^7 \mod 299 = 150$ より、**暗号文は『150』**である。復号化は $150^{19} \mod 299 = \cdots$、練習問題だと思ってやってみよう。ちゃんと 123 に戻るかな?

コラム: 公開鍵暗号技術

一番使われている RSA 暗号とは?

現在最も広く用いらている**公開鍵暗号システム**である**RSA**では、**秘密鍵** D と**公開鍵** E および法とする**数** N を用いる。
平文 P を暗号文 C に暗号化する計算は

$$C = P^E \mod N$$

で、復号化する計算は

$$P =$$

である。どちらの計算も**剰余積の結合律**を使えば十分現実的に計算できることがわかる。
なお、平文 P、暗号文 C および法とする数 N をもとに**鍵** E **や** D **を逆算**することは、**離散対数問題**と言われていて**とても難しい**。
なお鍵は適当でいいはずはなくちゃんとした条件がある。例えば $(D, E, N) = (7, 19, 299)$ は条件を満たす。**平文『123』**を暗号化しよう。$123^7 \mod 299 = 150$ より、**暗号文は『150』**である。復号化は $150^{19} \mod 299 = \cdots$、練習問題だと思ってやってみよう。ちゃんと 123 に戻るかな?

コラム: 公開鍵暗号技術

一番使われている RSA 暗号とは?

現在最も広く用いらている**公開鍵暗号システム**である**RSA**では、**秘密鍵** D と**公開鍵** E および法とする**数** N を用いる。
平文 P を暗号文 C に暗号化する計算は

$$C = P^E \mod N$$

で、復号化する計算は

$$P = C^D \mod N$$

である。どちらの計算も**剰余積の結合律**を使えば十分現実的に計算できることがわかる。
なお、平文 P、暗号文 C および法とする数 N をもとに**鍵** E **や** D **を逆算**することは、**離散対数問題**と言われていて**とても難しい**。
なお鍵は適当でいいはずはなくちゃんとした条件がある。例えば $(D, E, N) = (7, 19, 299)$ は条件を満たす。**平文『123』**を暗号化しよう。$123^7 \mod 299 = 150$ より、**暗号文は『150』**である。復号化は $150^{19} \mod 299 = \cdots$、練習問題だと思ってやってみよう。ちゃんと 123 に戻るかな?

実はアレにはこれが入ってた！

実はアレにはこれが入ってた！

$$C = P^E \mod N$$
$$P = C^D \mod N$$

実はアレにはこれが入ってた！

$$C = P^E \mod N$$
$$P = C^D \mod N$$

実はアレにはこれが入ってた！

$$C = P^E \mod N$$
$$P = C^D \mod N$$