- 作业讲解
  - TJ第16章1、3、12、17、18、24、32、34、35、36、39、~~40~~

TJ第16章练习12

- 要有过程
- 要首先证明运算的封闭性
- 对于integral domains，别忘了证明

for every $a, b \in R$ such that $ab = 0$, either $a = 0$ or $b = 0$.

TJ第16章练习17

- $0a$
  $=(1+(-1))a$ //ring with identity
  $=1a+(-1)a=a+(-1)a$ //分配律
  即$(-1)a$是a的加法逆元，即$(-1)a=-a$

TJ第16章练习18

(Proposition 16.1)

- 先证$\forall x \in R$, 0x=0
  - 因为：0x+xx=(0+x)x=xx → 0x=0
- 因此
  - $\underline{(-a)(-b)+(-a)b}=(-a)(-b+b)=0=(-a+a)b=\underline{(-a)b+ab}$
    $(-a)(-b)+(-a)b+ab=(-a)b+ab+ab$
    $(-a)(-b)+(-a+a)b=(-a+a)b+ab$
    $(-a)(-b)+0=0+ab$
    $(-a)(-b)=ab$

TJ第16章练习39

- $x^2=x$
  → $x^2-x=0$
  → $x^2-1x=0$
  → $x^2+(-1)x=0$ // Proposition 16.1
  → $(x-1)x=0$
  → $x-1=0$或$x=0$ // integral domain
  → $x=1$或$x=0$

- 教材讨论
  - TC第31章第1、2、3、4、5、6节

问题1：GCD和(Extended-)Euclid

- Define $\mathrm{lcm}(a_1, a_2, \ldots, a_n)$ to be the ***least common multiple*** of the n integers $a_1, a_2, \ldots, a_n$, that is, the smallest nonnegative integer that is a multiple of each a_i. Show how to compute $\mathrm{lcm}(a_1, a_2, \ldots, a_n)$ efficiently using the (two-argument) gcd operation as a subroutine.

问题1：GCD和(Extended-)Euclid (续)

- 什么是 $(\mathbb{Z}_n, +_n)$和 $(\mathbb{Z}_n^*, \cdot_n)$？它们为什么是有限交换群？How to compute multiplicative inverses in $(\mathbb{Z}_n^*, \cdot_n)$？

问题1：GCD和(Extended-)Euclid (续)

- 什么是 $(\mathbb{Z}_n, +_n)$ 和 $(\mathbb{Z}_n^*, \cdot_n)$？它们为什么是有限交换群？
How to compute multiplicative inverses in $(\mathbb{Z}_n^*, \cdot_n)$？
  - 1 $=_n$ ax+ny
  利用Extended-Euclid求gcd(a,n)=1，得到的x即a^{-1}

问题2：Z_n和Z_n*

- 什么是<a>？
  什么时候<a>=Z_n？
  如果<a>≠Z_n，那么<a>有什么特征？

问题2：Z_n和Z_n* (续)

- 你理解Euler’s phi function了吗？

  $$\phi(n) = n \prod_{p\,:\,p \text{ is prime and } p \mid n} \left(1 - \frac{1}{p}\right)$$

  它和 $(\mathbb{Z}_n^*, \cdot_n)$ 有什么关系？
  当n是质数时，你能算出ϕ(n)吗？
- 我们上次课讲解习题时，提到过一个公式：
  当m和n互质时，ϕ(mn)=ϕ(m)ϕ(n)
  你能证明吗？

问题2：Z_n和Z_n* (续)

Theorem 31.27 (Chinese remainder theorem)
Let $n = n_1 n_2 \cdots n_k$, where the n_i are pairwise relatively prime. Consider the correspondence

$$a \leftrightarrow (a_1, a_2, \ldots, a_k) , \tag{31.27}$$

where $a \in \mathbb{Z}_n$, $a_i \in \mathbb{Z}_{n_i}$, and

$$a_i = a \bmod n_i$$

for $i = 1, 2, \ldots, k$. Then, mapping (31.27) is a one-to-one correspondence (bijection) between $\mathbb{Z}_n$ and the Cartesian product $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_k}$. Operations performed on the elements of $\mathbb{Z}_n$ can be equivalently performed on the corresponding k-tuples by performing the operations independently in each coordinate position in the appropriate system. That is, if

$$\begin{aligned} a &\leftrightarrow (a_1, a_2, \ldots, a_k) , \\ b &\leftrightarrow (b_1, b_2, \ldots, b_k) , \end{aligned}$$

then

$$(a + b) \bmod n \leftrightarrow ((a_1 + b_1) \bmod n_1, \ldots, (a_k + b_k) \bmod n_k) , \tag{31.28}$$

$$(a - b) \bmod n \leftrightarrow ((a_1 - b_1) \bmod n_1, \ldots, (a_k - b_k) \bmod n_k) , \tag{31.29}$$

$$(ab) \bmod n \leftrightarrow (a_1 b_1 \bmod n_1, \ldots, a_k b_k \bmod n_k) . \tag{31.30}$$

问题2：Z_n和Z_n* (续)

- Draw the group operation tables for the groups $(\mathbb{Z}_4, +_4)$ and $(\mathbb{Z}_5^*, \cdot_5)$. Show that these groups are isomorphic by exhibiting a one-to-one correspondence α between their elements such that $a + b \equiv c \pmod 4$ if and only if $\alpha(a) \cdot \alpha(b) \equiv \alpha(c) \pmod 5$.

问题2：Z_n和Z_n* (续)

- Draw the group operation tables for the groups $(\mathbb{Z}_4, +_4)$ and $(\mathbb{Z}_5^*, \cdot_5)$. Show that these groups are isomorphic by exhibiting a one-to-one correspondence α between their elements such that $a + b \equiv c \pmod 4$ if and only if $\alpha(a) \cdot \alpha(b) \equiv \alpha(c)$ (mod 5).

- 任意Z_i，是不是一定能找到Z_j*与之同构？
- 任意Z_j*，是不是一定能找到Z_i与之同构？

问题3：powers of an element

- 在这个算法中，c的作用是什么？
  你能简要解释这个算法的正确性证明吗？

Just prior to each iteration of the **for** loop of lines 4–9,

1. The value of c is the same as the prefix $\langle b_k, b_{k-1}, \ldots, b_{i+1} \rangle$ of the binary representation of b, and
2. $d = a^c \bmod n$.

- 你会分析这个算法的运行时间吗？

```
MODULAR-EXPONENTIATION(a, b, n)
1   c = 0
2   d = 1
3   let ⟨b_k, b_{k-1}, ..., b_0⟩ be the binary representation of b
4   for i = k downto 0
5       c = 2c
6       d = (d · d) mod n
7       if b_i == 1
8           c = c + 1
9           d = (d · a) mod n
10  return d
```