Final Notes

L05: Modular Arithmetic

- $a|b \Leftrightarrow a$ divides $b \Leftrightarrow b/a$ is an integer
  - 反面: $a \nmid b$
  - a: factor/divisor of b
  - b: multiple of a

$(a \bmod mn) \bmod m = a \bmod m$

$\because \begin{cases} a = gmn + s & (0 \le s < mn) \\ a = j \cdot m + s' & (0 \le s' < m) \end{cases}$

$\Leftrightarrow s \bmod m = s'$

$\Leftrightarrow (a - gmn) \bmod m = s'$

$\Leftarrow a \bmod m = s' \checkmark$

※ $\begin{cases} (a+b) \bmod m = (a \bmod m + b \bmod m) \bmod m \\ (a \cdot b) \bmod m = (a \bmod m \cdot b \bmod m) \bmod m \end{cases}$

Arithmetic modulo (在 $\mathbb{Z}_m$ 上定义)

新运算: $\mathbb{Z}_m = \{0, 1, \cdots, m-1\}$

对于 $a, b \in \mathbb{Z}_m$

定义: $\begin{cases} a +_m b = (a+b) \bmod m \\ a \cdot_m b = (a \cdot b) \bmod m \end{cases}$

有交换、结合、分配律

Additive inverse & multiplicative inverse

① additive inverse (for a): $(-a \bmod m)$

有 $a +_m (-a \bmod m) = 0$

② multiplicative inverse (for a): $a \cdot_m b = 1$

for $\forall a \in \mathbb{Z}_m$, $m > 1$, $\gcd(a, m) = 1$

$\Rightarrow a$ 有 unique multiplicative inverse

if $\gcd(a, m) \neq 1$, a^{-1} 不存在

※找 a^{-1}: extended euclidean algorithm

用 bézout theorem 的办法找出

$1 = c_1 \cdot a + c_2 \cdot m$

$\Rightarrow 1 \equiv c_1 \cdot a + c_2 \cdot m \pmod m$

$\equiv c_1 \cdot a \pmod m$

$\therefore c_1 = a^{-1}$

Congruences (在 $\mathbb{Z}$ 上定义)

a is congruent to $b \Leftrightarrow a \bmod m = b \bmod m$

$\Leftrightarrow a \equiv b \pmod m$ (notation; m: modulus)

※ $a/c \neq b/d \pmod m$

L06: GCDs & Congruences

gcd: 最大公约数

Euclidean algorithm: 辗转相除法

Bézout's Theorem:

if $a, b \in \mathbb{Z}^+$, $\exists c_1, c_2 \in \mathbb{Z}$, s.t. $\gcd(a,b) = c_1 a + c_2 b$

找 c_1, c_2 的办法: (gcd 的 V 怪数都能找到 c_1, c_2)

例: 找 $\gcd(252, 198)$:

$252 = 198 \times 1 + 54$

$198 = 3 \cdot 54 + 36$

$54 = 1 \cdot 36 + 18$

$36 = 2 \times 18 + 0$

$\Rightarrow \gcd(252, 198) = 18$

$\Rightarrow 54 = 252 - 198 \times 1$

$36 = 198 - 3 \cdot 54$

$= 198 - 3 \cdot (252 - 198 \cdot 1)$

$= 4 \cdot 198 - 3 \cdot 252$

$= \cdots$

$= 4 \cdot 252 - 5 \cdot 198$

解同余方程

Linear congruence:

$ax \equiv b \pmod m$

① find a^{-1}

② $a^{-1} \cdot ax \equiv a^{-1} \cdot b \pmod m$

$\Rightarrow x \equiv a^{-1} \cdot b \pmod m$

if $\gcd(a, m) \neq 1$, 那么 $ax \equiv b \pmod m$ 可能有 $\begin{cases} 0 \text{ 解} \\ \text{or} \\ \text{多解} \end{cases}$ in $\mathbb{Z}_m$

中国剩余定理:

let $m_1, m_2, \cdots, m_n$ 两两互质且 $m_i > 1$, a_i 为任意数

$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \vdots \\ x \equiv a_n \pmod{m_n} \end{cases}$

那 $x \bmod (m_1 \cdots m_n)$ 有唯一确定值

Consider the 3 congruences from Sun-Tsu's problem:
$x \equiv 2 \pmod 3$, $x \equiv 3 \pmod 5$, $x \equiv 2 \pmod 7$.
Let $m = 3 \cdot 5 \cdot 7 = 105$, $M_1 = m/3 = 35$, $M_2 = m/5 = 21$, $M_3 = m/7 = 15$.
We see that

- 2 is an inverse of $M_1 \pmod 3$
- 1 is an inverse of $M_2 \pmod 5$
- 1 is an inverse of $M_3 \pmod 7$

Hence,
$x = a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3$
$= 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 = 233 \equiv 23 \pmod{105}$

L07: Cryptograph

Secret Key Cryptography

(1) Caesar Cipher (Shift Cypher)

secret key: k

encryption 加密: $f(p) = (p+k) \bmod 26$

$p \to f(p)$

decryption 解密: $f^{-1}(p) = (p-k) \bmod 26$

$f^{-1}(f(p)) \mapsto p$

(2) Affine Ciphers

secret key: (a, b), s.t. $\gcd(a, 26) = 1$ ($ax + b \equiv y \pmod{26}$ has a unique solution)

$f(p) = (ap+b) \bmod 26$

$f^{-1}(p)$: solve $ap + b \equiv c \pmod{26}$

(3) Block Ciphers

粗浅地理解为把 mod 26 变成 mod 一个大数

$f(p) = ap + b \pmod{2^{80}}$

$f^{-1}(y) = a^{-1}(y - b) \pmod{2^{80}}$

- $\frac{1}{2} \cdot 2^{80} \cdot 2^{80}$ different keys

(4) Advanced Encryption Standard

Public Key Cryptography & RSA

key: communicate physically / use DH protocol

↳ public key / private key

RSA Encryption ($n = p \cdot q$, which two large prime and hard to discover)

Alice $\xrightarrow{C}$ Bob

快速 $C = x^e \bmod n$ → 从这里算出 x 很困难

(p, q) private key

(n, e) is public key → $\gcd(e, (p-1)(q-1)) = 1$, 一般 randomly picked

RSA Decryption

① Find d. $d = e^{-1} \pmod{(p-1)(q-1)}$

② Find $c^d \bmod n = x$

proof: $c^d \equiv (x^e)^d \pmod n$

$\equiv x^{de} \pmod n$

(1) Step 1: show $\begin{cases} x^{de} \equiv x \pmod p \\ x^{de} \equiv x \pmod q \end{cases}$

(2) Step 2: show $x^{de} \equiv x \pmod{pq}$

Key exchange

Alice $\xrightarrow{k_1}$ $a^{k_1} \bmod p$ $\xrightarrow{k_2}$ Bob $(\text{算 } (a^{k_1})^{k_2} \bmod p)$

$(\text{算 } (a^{k_2})^{k_1} \bmod p)$ Alice $\xleftarrow{a^{k_2} \bmod p}$ Bob

shared key: $a^{k_1 k_2} \bmod p$

Ditte-Hellman key exchange

Fermat Little Theorem

$a^{p-1} \equiv 1 \pmod p$

Lemma: $a \bmod p, 2a \bmod p, \cdots, (p-1)a \bmod p$ 是 $\{1, 2, \cdots, p-1\}$

Step 1:

$\because d = e^{-1} \bmod (p-1)(q-1)$

$\therefore de = 1 + k(p-1)(q-1)$

$\Rightarrow c^d \equiv (x^e)^d$

$\equiv x^{de}$

$\equiv x^{1 + k(p-1)(q-1)} \pmod p$

$x^{k(p-1)(q-1)} \equiv (x^{p-1})^{k(q-1)} \pmod p$

$\equiv 1 \pmod p$

$\Rightarrow x^{1 + k(p-1)(q-1)} \equiv x \pmod p$

$\Rightarrow x^{de} \equiv x \pmod p$

同理, $x^{de} \equiv x \pmod q$

$\because \gcd(p, q) = 1$, by the Chinese Remainder Theorem

$\therefore x^{de} \equiv x \pmod{pq}$

$\Rightarrow c^d \equiv x \pmod{pq}$

L08: Algorithm

selection sort: 每次把后面 i 个最大的放在第 n-i 个位置上

max

大O表示法:

Big-Θ:

witness

$f(n)$ is $\Theta(g(n)) \Leftrightarrow \exists c_1, c_2, k > 0$, s.t. $c_1 g(n) \le f(n) \le c_2 g(n)$ $(x > k)$

※ $\log(n!) = \Theta(n \log n)$ (用放缩来证) → 但 $n! \neq \Theta(n^n)$ ($n! = O(n^n)$, $n! = \Omega((\frac{n}{2})^{\frac{n}{2}})$)

$\sum \frac{1}{i} = \Theta(\log n)$

$c_1^n >> n^{c_2}$

Big-Oh:

witness

$f(n)$ is $O(g) \Leftrightarrow \exists c_2, k > 0$, s.t. $f(n) \le c_2 g(n)$ $(x > k)$

Big-Omega:

witness

$f(n)$ is $\Omega(g) \Leftrightarrow \exists c_1, k > 0$, s.t. $f(n) \ge c_1 g(n)$ $(x > k)$

$f(n) = \Theta(g(n)) \Leftrightarrow f(n) = O(g(n))$ & $f(n) = \Omega(g(n))$

Comparison of algorithms:

计算 $\lim_{n \to \infty} \frac{T_1(n)}{T_2(n)}$

- $= 0$ 1 快
- $\to \infty$ 2 快
- $=$ Constant / 振荡 (cannot tell)

↳ 假如 $T_1(n)$, $T_2(n)$ 都是无穷时记为

也可以研究 $\lim_{n \to \infty} \frac{g_1(n)}{g_2(n)}$

($T_1(n) = \Theta(g_1(n))$ $T_2(n) = \Theta(g_2(n))$)

Insertion sort

sorted | 当前 | 未处理

每次把 key 插入到 sorted sequence 中合适的位置

Worse-case Analysis (default analysis)

To show worse-case $= O(n^2)$:

- show $O(n^2)$
- ※find a input which is $\Omega(n^2)$

L09: Mathematical induction

由 well-ordering property 证 mathematic induction

非空正整数集合在最小的元素

Proof by contradiction: Assume $\forall n P(n)$ is false.
Let $S = \{i \in \mathbb{Z}^+ \mid P(i) \text{ is false}\}$
By the well-ordering principle, S has a least element, say m.
$m \neq 1$ since $P(1)$ is true. So $m - 1 \in \mathbb{Z}^+$ and $P(m-1)$ is true.
$P(m)$ is false and $P(m-1)$ is true. This contradicts with $\forall k(P(k) \to P(k+1))$.

Strong Induction

$(P(1) \vee (P(1) \wedge P(2) \wedge \cdots \wedge P(k)) \to P(k+1))$

L10: Recursion

Recurrence: a recursively defined function over $\mathbb{N}$

Recursively defined sets: 有些理解没那么直观, 可从 base case - 步步推

※ Structural induction:

假如一个 set 是 recursively defined 的

证明它有某性质 P

- base case 有 P
- 这个结构由 base case construct
- construct 可保留 P

L15: probability

- experiment: 比如扔骰子
- sample space: 事件结果的集合
- event: sample space 的 subset
  - ↳ Complement of Events: $\bar{E}$
  - Union of Events: $E_1 \cup E_2$ ($P(E_1 \vee E_2) = P(E_1) + P(E_2) - P(E_1 \wedge E_2)$)
    - disjoint union: union 中每个事件相互无重合 ($P(E_1 \vee E_2) = P(E_1) + P(E_2)$)

sampling without replacement: 无放回抽样

sampling with replacement: 放回抽样

Probability distribution

pmf: probability mass distribution

pdf: probability density function

$\int_S = 1$

Uniform & non-uniform distribution

(sample space 里的 sample 被选到的概率一样)

Independence $\Leftrightarrow P(E \wedge F) = P(E) \cdot P(F)$

check this: 用定义 / 两个 event 完全无关

Pairwise & Mutual Independence

$E_1, \cdots, E_n$ pairwise independent
$\Leftrightarrow P(E_i \cap E_j) = P(E_i) \cdot P(E_j)$
for all $1 \le i < j \le n$

$E_1, \cdots, E_n$ mutual independent
$\Leftrightarrow P(E_1 \cap \cdots \cap E_n) = P(E_1) \cdots P(E_n)$
(且两两独立)

Mutual independent $\Rightarrow$ Pairwise independent

Conditional Probability & Bayes

$$P(E|F) = \frac{P(E \cap F)}{P(F)}$$

$$P(F|E) = \frac{P(E|F) \cdot P(F)}{P(E|F) \cdot P(F) + P(E|\bar{F}) \cdot P(\bar{F})} \quad \text{(Bayes)}$$

Random Variables

↳ 一个把 sample space 中的 element map 到 $\mathbb{R}$ 的函数
(这个 function 不是 random 的)

Bernoulli trials

结果 binary 且试验之间互相独立

Binomial distribution 二项分布

n次伯努利 experiment 的分布

$P(X=k) = C_n^k \cdot p^k \cdot (1-p)^{n-k}$
$E(X) = np$
$E(aX+b) = aE(X)+b$
$V(X) = np(1-p)$

Geometric distribution

X 是伯努利试验第一次成功时的次数

$P(X=k) = (1-p)^{k} \cdot p$
$E(X) = \frac{1}{p}$
$V(X) = \frac{1-p}{p^2}$

Expected Value

$E(X) = \sum p(x) \cdot X$

Indicator random variable

$X_i \in \{0, 1\}$，用来描述事件 E_i 是否发生

Independent random variables

$P(X=r_1 \ \& \ Y=r_2) = P(X=r_1) \cdot P(Y=r_2)$

$E(XY) = E(X) \cdot E(Y)$

Variance

$$V(X) = E((X-E(X))^2)$$
$$= \sum P(x) \cdot (X-E(X))^2$$
$$= E(X^2) - E(X)^2$$

$V(aX) = a^2 V(X)$

Bienaymé's Formula

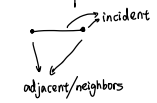
for independent random variables X, Y,

$V(X+Y) = V(X)+V(Y)$

if $X_1 \sim X_n$ (pairwise independent)

$\Rightarrow V(X_1+\cdots+X_n) = V(X_1)+\cdots+V(X_n)$

L18: Graph



simple graph: graph without loops/multiple edges

edge { directed (u, v) / undirected $\{u, v\}$

degree { undirected : 跟该 node incident 的边
directed : { In-degree 指向该顶点的 edge 的数量 $\deg^-(v)$
Out-degree 从该顶点出发的 edge 的数量 $\deg^+(v)$

$G_1 = (V, E)$, undirected, m edges $\Rightarrow \sum_{v \in V} \deg(v) = 2m$
$G_1 = (V, E)$, directed, m edges $\Rightarrow m = \sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v)$

Special graphs

① complete graph K_n
点之间两两都有连线

② cycle C_n $(n \ge 3)$
△ □ ⬠ ---

Bipartite graph ⊆ simple graph
↳能被完全分为两部分，边只能分别属于这两部分

complete Bipartite graph $K_{m,n}$
能被分 size 为 m, n 的两部分，使得两部份相互全连接
且每部分之间无连接

Subgraph

endpoint 为 edge 是 subset 即可

Adjacent list

形容 edge 用

adjacency matrices { undirected symmetric & diagonal entry = 0 (simple graph)
directed not necessarily symmetric

$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge} \\ 0 & \text{otherwise} \end{cases}$

↳ non-simple graph: a_{ij} 为 i, j 之间相连的边数

Incident matrices

n×m matrix, n = vertex 数, m = edge 数

$m_{ij} = \begin{cases} 1 & (v_i \text{ \& } e_j \text{ incident}) \\ 0 & (\text{otherwise}) \end{cases}$

Graph Isomorphism

对两个 simple graph $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$
isomorphic $\Leftrightarrow$ 存在一个 T (transformation), 使得
$T(v_1) = v_2$, $v_1 \in V_1$, $v_2 \in V_2$, 在
transform 之后 E_1 和 E_2 完全相同

determine if isomorphic { worse case: exponential
average case: linear

Path

$e_1, \cdots, e_n$ { directed: e_i connects x_{i-1} and x_i
undirected: $e_i = (x_{i-1}, x_i)$

for simple graph: path: $x_0, x_1, \cdots, x_n$ $(x_i \in V)$
if $u = v$, path is a circuit/cycle (始 = 终)
simple path/circuit: 每条 edge 经过一次

Connectivity

2 vertices u, v are connected $\Leftrightarrow$ there's a path from u to v

a/undirected graph is connected if 对 V 顶点对，都有通路

directed graph is { weakly connected if 忽略方向连通
strongly connected if 不忽略方向连通

Connected component
{ connected subgraph
不能被包含于更大的 connected subgraph 里

Strongly connected component { strongly connected subgraph
不能被包含于更大的 strongly connected subgraph 里

Euler circuit $\Leftrightarrow$ 每个点的度数都是偶数 & 图是 connected 的
起点 = 终点，每条边经过一次

Proof (cnt'd)

- The "if" direction: We will give an algorithm to find an Euler cycle when all degrees are even.
- First, consider the following simple algorithm:

```
u ← any vertex
while u has an edge not taken yet
    take that edge {u,v}
    u ← v
```

- Observation: This algorithm always finds a cycle, because all degrees are even
- Problem: This algorithm may not traverse all edges.

Proof (cnt'd)

- Idea: The subgraph consisting all edges not traversed must still have even degrees at all vertices.
- Then just repeat the algorithm on those edges, and "connect" the two cycles into one.

```
c ← empty cycle
while there are still edges not taken yet
    u ← any vertex already seen
    c' ← Find-Cycle(u)
    insert c' into c at u

Find-Cycle(u):
while u has an edge not taken yet
    take that edge {u,v}
    u ← v
```

Euler path $\Leftrightarrow$ 恰好两个点的 degree 是奇数，其他都是偶数 & 图 connected
起点，不用 = 终点，每边经过一次

证明：把两个奇点相连，则存在 euler circuit ∪

Hamilton Paths & Circuits
contains every vertex once