CMPE209: Important Review Notes for Students

ISA: Samira C. Oliva Madrigal

Modular Arithmetic

1. How it works
    - It is an arithmetic number system where elements wrap
    - Extremely important, will probably see it through all revolutions in crypto.

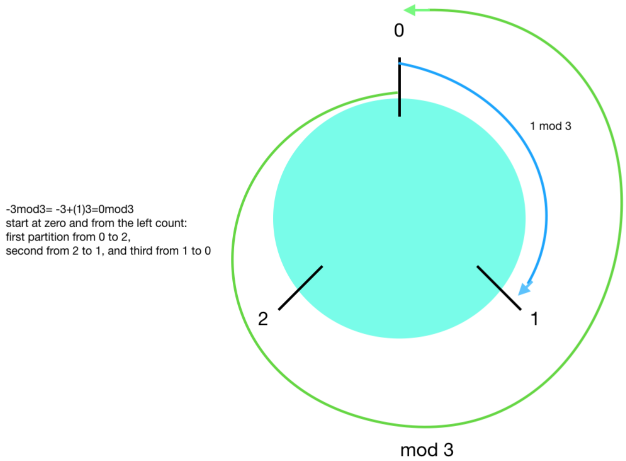

Figure 1: Modular arithmetic example

2. Examples in $\mathbb{Z}$ with the different cases
    - $a \pmod p = r$ such that a/p = q*p + r where q = quotient and r = remainder
    - $-a \pmod{-p} = -(a \pmod p)$
    - $-a \pmod p = -a + kp$ where k is a multiple of the modulous p; add multiples of p
    - $a \pmod{-p} = -(-a \pmod p)$; same as above but with minus sign.
    - $a \pmod p = a$ if $a \geq 0$ and $a < p$
3. Modular arithmetic properties [3]
    - $[a \pmod p + b \pmod p] \pmod p = (a + b) \pmod p$
    - $[a \pmod p - b \pmod p] \pmod p = (a - b) \pmod p$
    - $[a \pmod p \times b \pmod p] \pmod p) = (a \times b) \pmod p$
    - Read chapter 2 in your text.
    - **modular exponentiation**
      methods to solve: modular exponentiation (aka repeated multiplication) [small base, prime modulous], Fermat's Little Theorem [prime modulous] (example under Fermat section), or Euler Totient Function (example under Euler section) reduce exponent mod $\phi(n)$]

- **example of method 1**: $3^{557} \pmod{925}$ has two routes:
  a) repeated division b) or use trick
  route a)
  step 1: express exponent as a sum of powers of 2, so: $557 = 2^9 + 2^5 + 2^3 + 2^2 + 2^0$
  step 2: calculate base to all powers of 2 mod (p) up to and including highest power above as follows:
  step 2: $3^{2^0} \pmod{p} = 3^1 = 3$
  step 2: $3^{2^1} \pmod{p} = 3^2 = 9$
  step 2: $3^{2^2} \pmod{p} = 3^4 = 81$ // same as $9^2 \pmod{p}$ or $[(9 \pmod{p}) \times (9 \pmod{p})] \pmod{p}$ which is $9x9 \pmod{p} = 81 \pmod{p}$
  step 2: $3^{2^3} \pmod{p} = 3^8 = 86$ // same as $81^2 \pmod{p}$
  step 2: $3^{2^4} \pmod{p} = 3^{16} = 921$ // same as $86^2 \pmod{p}$
  step 2: $3^{2^5} \pmod{p} = 3^{32} = 16$
  step 2: $3^{2^6} \pmod{p} = 3^{64} = [16 \times 16 \pmod{925}] = 256$
  step 2: $3^{2^7} \pmod{p} = 3^{128} = [256 \times 256 \pmod{925}] = 786$
  step 2: $3^{2^8} \pmod{p} = 3^{256} = [786 \times 786 \pmod{925}] = 821$
  step 2: $3^{2^9} \pmod{p} = 3^{512} = [821 \times 821 \pmod{925}] = 641$
  step 2: use pattern if cycle repeats to save on computations.
  note that each subsequent number will be the square of the previous
  In this case, the cycle is until we reach $3^{2^{14}}$ .... which is not useful
  but if our number required higher powers, then it would be useful as
  we would already know the answers for power after that.
  $641 \times 641 \pmod{925}$
  $181 \times 181 \pmod{925}$
  $386 \times 386 \pmod{925}$
  $71 \times 71 \pmod{925}$
  $416 \times 416 \pmod{925} = 81$
  step 3: select solutions from step 2 only for powers which are in step 1.
  step 3: $3^{557} = 3^{2^9+2^5+2^3+2^2+2^0} \pmod{925} = [641 \times 16 \times 86 \times 81 \times 3] \pmod{925}$
  step 3: $= [(641 \times 16) \pmod{925} \times (86 \times 81 \times 3) \pmod{925}] \pmod{925}$
  step 3: $= [226 \times 548] \pmod{925} = 823$ and voilá there is your answer
  route b): you look at an exponent like this: $3^{557} \pmod{925}$, and you try to do something similar but less systematic and often a lot of faster ...
  $3^{557} \pmod{925} \rightarrow [3^{250} \times 3^{250} \times 3^{57}] \pmod{925}$, when the exponent is not that large and can break into terms of at most 2^8, then no big deal especially if no calculator at hand, for example you get problem: $2^{24} \pmod{241}$

Fermat's Little Theorem

1. How it works
   - 1) if $a \in \mathbb{Z}^+$, p is prime, and $p \nmid a$, then $a^{p-1} \equiv 1 \pmod{p} \iff \gcd(a, p) = 1$
   - 2) if $a \in \mathbb{Z}^+$ and p is prime then $a^p \equiv a \pmod{p}$
2. Examples
   - Formula 1) example
     $3^{2000} mod(29)$
     $\rightarrow p - 1 = 29 - 1 = 28$
     $\rightarrow 2000 = (p-1)k + r = 28k + r = 28(71) + 12$
     $\rightarrow (3^{28}) \equiv 1 mod 29 \rightarrow$ and so must a multiple of it $(3^{28})^{71} \equiv 1 \pmod{29}$
     hence, $3^{2000} \pmod{29} \rightarrow [(3^{28})^{71} \pmod{29} \times (3^{12}) \pmod{29}] \pmod{29}$
     $\rightarrow [1 \times (3^{12}) mod(28)] \pmod{29} \rightarrow (3^{12}) \pmod{29}$ simpler to solve
     $\rightarrow (3^{12}) \pmod{29}$

$\rightarrow 3^3 = 27 \pmod{29}$
$\rightarrow [3^4 \pmod{29} \times 3^4 \pmod{29} \times 3^4 \pmod{29}] \pmod{29}$
$\rightarrow [81 \times 81 \times 81] \pmod{29}$
$\rightarrow [6561 \pmod{29} \times 81 \pmod{29}] \pmod{29}$
$\rightarrow [7 \times 23] \pmod{29}$
$\rightarrow 16 \pmod{29}$

Euler's Totient Function

1. How it works
   - $\phi(n)$:= the count of all numbers relatively prime to n and less than n.
   - now, if we have two primes p and q which are not equal, such that $n = pq$ then $\phi(n) := (p-1)(q-1)$
2. Examples
   - $\phi(1) := 1$
   - $\phi(21) := (2)(6) = 12$ since $n = 7 \times 3$, we broke it into it's prime factors.
3. Application to modular exponentiation
   - properties ∀ a, p such that gcd(a,p) = 1, $a^{\phi(p)} \equiv 1 mod(p)$ and $a^{\phi(p)+1} \equiv a mod(p)$
   - use when we want to reduce the exponent, for instance if given $233^{721} mod(p)$

Eucledian Algorithm

1. Gives us the GCD or greatest common divisor between two numbers, gcd(a,b) = c.

Algorithm 1 gcd(a,b)

```
1: if a < b:
2:     temp = a
3:     a = b
4:     b = temp
5: r = 1
6: while r > 0:
7:     r = a mod b
8:         if r > 0:
9:             a = b
10:            b = r
11: return b
```

2. when you do step 7 try to write out the equations like this: a mod (b) = q*b + r which is needed for EEA.
3. EXAMPLE: gcd(13,5) = ?
   a = b*q + r
   13 = 5*2 + 3
   5 = 3(1) + 2
   3 = 2(1) + 1
   2 = 1(2) + 0
   there gcd is 1, the last value of b when r >= 0.

Extended Eucledian Algorithm

1. How it works
   - EEA allows us to a find the solution to the Bezout's identiy,
   - $\gcd(a, b) = c = a(x) + p(y)$
   - we find c, x, and y
   - helpful to find the multiplicative inverse
2. The EEA algorithm
   - given $amod(p)$ find the MI(a).
     quick test without running EEA: is gcd(a, p) is not 1, then the MI(a) in mod p does not exist, else set up your equation like this:
     1 = a(x) + p(y)
     take smallest number as divisor, so if a > p, then:
     a = p(q) + r
     p = r(q_2) + r_2
     r = $r_2(q_3)$ + r_3
     $r_2 = r_3(q_4) + r_4$
     $r_3 = r_4(q_5) + r_5$
     and so on, until you get a remainder of 1, for example if our next line looked like this:
     $r_4 = r_5(q_6) + 1$
     then work from the bottom up, setting the last equation as r_4 - $r_5(q_6) = 1$
     and re-writing the subsequent equations similarly then starting with the last equation as the first, substitute the expression of equation above, for appropriate value in current equation.
3. Example: $7465 \pmod{2464} \rightarrow 1 = 7465(x) + 2464(y)$
   - step 1: eucledian algorithm
     7465 = 2464(3) + 73
     2464 = 73(33) + 55
     73 = 55(1) + 18
     55 = 18(3) + 1
   - step 2: extended eucledian algorithm
     55 - 18(3) = 1
     73 - 55(1) = 18
     2464 - 73(33) = 55
     7465 - 2464(3) = 73
   - step 3: substitutions
     substitute for 18:
     55 - **18**(3) = 1
     → 55 - **[73 - 55(1)]**(3) = 1
     → 55 - **[(3)73 + 55(3)]** = 1
     → -3(73) + 4(55) = 1
     substitute for 55:
     -3(73) + 4(**55**) = 1
     → -3(73) + 4(**2464 - 73(33)**) = 1
     → -3(73) + 4(2464) - 73(132) = 1
     → -135(73) + 4(2464) = 1
     substitute for 73:
     -135(**73**) + 4(2464) = 1 (mod 2064)
     → -135(**7465 - 2464(3)**) + 4(2464) = 1
     → -135(7465) + 409(2464) = 1 mod (2464) [any multiple of the modulous is zero]
     → -135(7465) = 1 mod (2464) [any multiple of the modulous is zero]
     the MI(7465) is then -135 or 2329 from -135 + 2464.

Set, Group, Abelian, Etc.

NOTE: do not confuse any symbols with actual addition, subtraction unless explicitly stated arithmetic addition or by context it is clear that we are talking about actual addition. Information taken from [2] and presented here in very compact review style.

1. *set* := set of objects.
   - We have infinite and finite sets.
   - Infinite ex: $\mathbb{Z}$, Finite ex: sequence $s_n :=< 1, 2, ..., n >$
   - cardinality of a set := number of objects in the set.
2. *group* := set of objects with a binary operator + (not necessariliy addition) with following 4 properties.
   - denote by $\{G, +\}$
   - closed under the operator
   - associativity holds
   - there exist identity element i s.t. $a + i = a$ (commonly denoted as 0 for above notation)
   - $\forall$ $a \in G$ $\exists$ a $b \in G$ s.t. $a + b = i$
   - NOTE: if operation is "addition", can think of its additive inverse; subtraction also allowed
   - ex: $a + b = 0$ so we say b is the additive inverse of a.
3. *abelian groups* := a group s.t. the operation is commutative (a+b = b+a)
   - ex: $\mathbb{Z}$ with operator = addition.
   - closed under the operator
   - associativity holds
   - there exist identity element i s.t. $a + i = a$
   - $\forall$ $a \in G$ $\exists$ a $b \in G$ s.t. $a + b = i$
4. *ring* := an abelian group with a second operation "x" with added properties on new operation
   - denote: $\{R, +, x\}$, + wrt which R is abelian, x need for R to be ring.
   - closed under the operator x
   - associativity holds wrt x
   - x is distributive over + , ex: a x (b + c) = a x b + a x c, etc.
   - ex: $\mathbb{Z}$ under arithmetic +, x.
5. *commutative ring* := ring if x operation is commutative i.e. (ab = ba $\forall a, b \in R$)
   - ex: set of all even integers (+,0,-) under arithmetic x and +.
   - ex: $\mathbb{Z}$ under arithmetic +, x.
   - ex: Z_n set of residues (requires gcd(a,n) =1 for a to have an MI).
6. *integral domain* := commutative ring with 2 additional properties:
   - $\exists$ identity element say "1" (symbolically) for x operation, s.t. $\forall a \in R$ 1a = a1 =a
   - if let "0" be the identity element for + operation, $\forall a, b \in R$ axb=0 $\iff$ a = 0 or b = 0.
   - ex: $\mathbb{Z}$ under arithmetic +, x.
   - ex: $\mathbb{R}$ under arithmetic +, x.

Finite Field Arithmetic

IMPORTANT

1. Finite Field := integral domain s.t. $\forall a \in F$ with $a \neq 0$ (the + identity element) $\exists b \in F$ s.t. ab = ba = "1" = identity element, we can denote the multiplicative inverse of a as: a^{-1} or MI(a).
   - denote as: $\{F, +, x\}$, more common $\mathbb{F}$
   - notation: a field of n-coordinate vectors or n-dimensional vector space $\mathbb{F}^n$
   - notation: a field of n by m matrices $\mathbb{F}^{nxm}$
   - order or cardinality or size of a field is the number of elements in it.
   - ex: Z_n for prime n.
2. Galois field = prime finite field
   - denote as $GF(n)$ where n = modulous (typically use p or n).
   - run EEA on Bezout's Identity to find MI(a) for an element in GF(n).
   - ex: Z_n for prime n.
3. $GF(2)$
   - addition and subtraction is XOR: $1 + 0 = 1 - 0 = 1 \oplus 0 = 1$.
   - multiplication is logical AND: 1 x 1 = 1 & 1 = 1, 1 x 0 = 0, and 0 x 0 = 0.
4. Polynomials with coefficients defined over $GF(2)$
   - addition and subtraction is XOR: $(x^2 + x) + (x + 1) = (x^2 + 1)$
   - normal mul with polynomials: $(x^2 + x^1) \times (x^1 + 1) = x^3 + x^2 + x^2 + x = x^3 + x$
   - normal poly division: $(x^4 + x^2 + 1)/x^2 \to x^4/x^2 = x^2(x^2) => x^4$
   - $(x^4 + x^2 + 1) - x^4 \to (x^2 + 1)$ now $(x^2/x^2) = 1(x^2) = x^2$
   - $(x^2 + 1) - x^2 = 1$
   - remainder is 1 and quotient is $x^2 + 1$.
5. Polynomial Arithmetic with coefficients defined over $GF(2)$
   - if we consider set of all possible polynomials over a field, it is a ring, actually a Commutative Ring
   - if we consider a finite set of polynomials with coefficients defined over a finite field we have a Finite Field
   - $GF(p^n)$ for p prime and n degree of irreducible polynomial defining the field $m(x)$, is a Finite Field
   - p is the characteristic of the field.
   - elements in $GF(p^n)$ are the same as elements in Z_p
   - order or cardinality of a field $q = p^n$
   - all elements are reduced via $m(x)$ and have degree at most $n - 1$
   - $m(x)$ is irreducible if cannot be expressed as product of two polynomials both of less degree and in F
   - $GF(2^1)$ or simply $GF(2)$ means coefficients are elements in {0,1}.
   - express a polynomial as codeword in $GF(2)$? simple:
$(MSb)x^5 + x^4 + x^3 + x^2 + x^1 + x^0(LSb) = 111111$
   - another:
$(MSb)x^5 + x^1 + x^0(LSb) = 100011$

- another:
  $x^(4) + x^1 = 010010$
- arithmetic: addition, subtraction are the same, bitwise XOR
- multiplication: direct multiplication and reduce mod $m(x)$ how? after multiply, (XOR result with $m(x)$ until get remainder with degree less than deg($m(x)$)), discard quotient.
- division: f(x)/g(x) => f(x)MI(g(x)); get MI same as above using EEA.
- some "tricks" for faster computation, especially for implementation, read your book and study the Kak notes.
- for pen and paper, do as above.
- EXAMPLES: consider $GF(2^3)$ and say my $m(x) = x^3 + x + 1$
- addition and subtraction are XOR: $(x^2 + x + 1) + (x + 1) = x^2$
- multiply out and reduce if result has degree >= $m(x)$
- $(x^2 + x + 1)x(x + 1) = (x^3 + x^2 + x^2 + x + x + 1) \to (x^3 + 1)$ and we need to reduce
- standard reduction is done by XORing from the most significant bit with $m(x)$ repeatedly, until we have an element in the field, of degree less than 3 in this case.
- $(x^3 + 1) \oplus x^3 + x + 1 = x = 010$
- division: $(x^2 + x + 1)/(x + 1) = (x^2 + x + 1) \times MI(x + 1)$.
- Find the MI using Bezout identity to set up equation and the EEA:
- Form is $a = b \times quotient + remainder$, then a = b and b = r until r = 1.
- $(x^3 + x + 1) = (x + 1)(x^2 + x) + 1$ //we hit remainder 1 on first equation :)
- $(x^3 + x + 1) - (x + 1)(x^2 + x) = 1 \pmod{m(x)}$
- $(x + 1)(x^2 + x) = 1 \pmod{m(x)}$, hence MI = $(x^2 + x)$.
- verify: $(x+1)(x^2+x) \to (x^3+x^2+x^2+x) \to (x^3+x) \pmod{m(x)} \equiv 1 \pmod{m(x)}$.

Chinese Remainder Theorem

Allows to compute modular exponentiation when the exponent is large and the modulous can be factored into primes and hence we have a system of equations as a set of linear congruences.

1. For simple but concise example on solving a system of such congruences see: [1] page 8-10.
   - Given a system of say 3 equations of this form:
     pronounce: э "eh", ю "yu", я "ya"
     $x \equiv a$ mod(э)
     $x \equiv b$ mod(ю)
     $x \equiv c$ mod(я)
   - What we do is this: if we take any of the above equations, we try to find the a multiple (α) of the other two moduli which is congruent to it. For example for the first equation we try to solve for a number x_1 such that $x_1 = \alpha$юя $\equiv a$ mod(э). Similarly we try to find two other numbers x_2 and x_3 for the other two equations such that $x_2 = \beta$эя $\equiv b$ mod(ю) and $x_3 = \gamma$эю $\equiv c$ mod(я)
   - our answer would then be x = $(x_1 + x_2 + x_3)$ mod(эюя)
2. A methodical way to the solution without guesswork may be as follows and is given in your book in the RSA chapter. For simplicity we consider splitting into two moduli.

- Consider the following example: Given b, e, and n, compute $s = b^e mod(n)$ for say some small base, a large exponent e, and large modulus n using CRT.
Step 1: factor n into it's prime factors p and q
Step 2: compute the following congruences:
$\mathrm{x_p} \equiv b^{emod(p-1)} \pmod{p}$
$\mathrm{x_q} \equiv b^{emod(q-1)} \pmod{q}$
Step 3: find a multiple of each modulus congruent to the other one:
$p\times$ю $\equiv \mathrm{x_q}$
$q\times$ я $\equiv \mathrm{x_p}$
These multiples are easily by solving the EEA from Bezout's identity once for either of the following equations:
$\mathrm{y_p} \equiv q \text{ x } q^{-1} \pmod{p}$
$\mathrm{y_q} \equiv p \text{ x } p^{-1} \pmod{q}$
Say we solve for $\mathrm{y_p} \equiv q \text{ x } q^{-1} \pmod{p}$
Our EEA equation is $gcd(q,p) = 1 = px + qy$
After solving we find our q^{-1} s.t.: $q \times q^{-1} \equiv 1 \pmod{p}$.
Now our multiple of q congruent to x_p in $\pmod{p}$ is simply:
я $= q^{-1} \times x_p$
we can verify the equation: since $q \times q^{-1} \equiv 1 \pmod{p} \rightarrow$
$q \times q^{-1} \times x_p \equiv 1 \times x_p \pmod{p}$
Now, if we go back to our equation before we cancel the modulus: and solve instead for the other parameter, we can find the MI of p.
Step 4: finally, compute the result as:
$s = (x_{\mathrm{p}}(q \times q^{-1}) + x_{\mathrm{q}}(p \times p^{-1})) \equiv mod(p \text{ x } q)$
$s = (q\times$я $+p\times$ю$)) \equiv mod(p \text{ x } q)$
EXAMPLE: b = 2, e= 23, p=233 and q=142.
2^{11} = 2048.. $x_p = 2^{23 \pmod{233-1}} \pmod{p} = 2^{23} \pmod{p} = 2^{11}2^{11}2 \pmod{233}$
$x_p = 142 \pmod{233}$
$x_q = 2^{23 \pmod{241-1}} \pmod{p} = 2^{23} \pmod{p} = 2^{11}2^{11}2 \pmod{241}$
$x_q = 121 \pmod{233}$
$y_p \equiv q \text{ x } q^{-1} \pmod{p}$
$y_q \equiv p \text{ x } p^{-1} \pmod{q}$
$\rightarrow gcd(233, 241) = 1 = qx + py$ using 233 as the modulus.
$241 = 233(1) + 8$
$233 = 8(29) + 1$ now rewrite:
$233 - 8(29) = 1$ (*)
$241 - 233(1) = 8$ now substitute for 8 in (*):
$233 - [241 - 233(1)](29) = 1$ (**) solve for the MI of 241:
-29(241) = 1 mod 233 and bring to positive -29+233 = 204. this is my q^{-1}
go back to (**) and solve for MI of 233 using 241 as the modulus:
233(30) = 1 mod 241, so 30 is my p^{-1}
my answer is now: $CT = PT^{23} \pmod{p \times q} \rightarrow$
$(q\times$ю$+p\times$я$) = [q(q^{-1} \times x_p) + p(p^{-1} \times x_q)] \pmod{p \times q}$
$[(241 \times 204)142) + (233 \times 30)121)]mod(233 \times 241) = 21811$.

3. Brute-force...
   - Once you compute $\mathrm{x_p}$ and $\mathrm{x_q}$, above,
   - for each equation, add the result to itself until you find a match occurs.

Discrete Log Problem

1. Very similar to computing logs in $\mathbb{R}_{>0}$

- modular domain

  Let the DLP be as follows: for p some prime number and a some primitive root of p whose powers 1 to $p-1$ are distinct and populate all nonzero elements in $mod(p)$. for any $j, i \in mod(p)$ that $j \equiv imod(p)$ and $i \in [0, p-1]$. For any j, we say that it's discrete log with respect to base $amod(p)$ is k such that $j \equiv a^k mod(p)$ for $k \in [0, p-1]$. Similarly, for groups, the DLP for any group G, is defined as:

  $$j = a^k \text{for } a, j \in G \tag{1}$$

  We want the smallest such k and this should be computationally infeasible if only j and a.

2. Examples, actual discrete log computation differs on scheme
   - Diffie-Hellman
   - ECC schemes (points on elliptic curves)
   - ElGamal

Asymmetric vs Symmetric Cryptography

1. Asymmetric or Public Key regards schemes which make use key pairs, meaning that each endpoint has a key pair $KP = \{PU, PK\}$ which consists of a public PU and a private key PK.
   - examples: RSA, ECC - based signature schemes
2. Symmetric schemes regards those for which share a private secret key
   - examples: ciphers, MACs

Overview of Course

1. We study the fundamental mathematics that form the basis of modern cryptosystems
2. We study the internal mechanism that form the basis of modern cryptosystems
3. We study specific cryptosystems
   - your text gives excellent presentation on these, please follow in detail.
   - Public-Key Cryptography: RSA, D-H, ECC (main core)
   - Symmetric Cryptography: block ciphers DES, AES (main core of the class besides the arithmetic such as CRT)
   - Cryptographic Hash Functions: SHA, SHA-3 (fundamental)
   - Applications: data encryption, user authentication, digital signatures, MACs, MICs (fundamental)
   - Infrastructures: Key Distribution Center (KDC) centralized/decentralized
4. We study specific attacks against these crytosystems
   - Passive, ex: eavesdropping, traffic analysis
   - Active, ex: intercepting traffic, man-in-the-middle
5. We see how these crytosystems are applied to protocols that secure communications in the Internet
   - HTTPS (HTTP over TLS), Kerberos - layer 5
   - TLS - between layer 4 and 5. It requires a TCP connection and is encapsulated inside L4, we say L5.
   - IPSec - encrypts and/or authenticates all traffic at layer 3

Advice

1. Study the textbook
2. Study textbook slides (for content not presented in the textbook)
3. Study outside lecture notes assigned
4. Pratice, pratice, pratice...
5. Very important: take care of your health!
6. Make time to workout and oxygenate your brain!
7. Make time for a social life!
8. Before you start hw, relax, concentrate, and burn through hw like a samurai!:)
9. Remember: If you do what you love, like they say, you'll never work a day in your life and you will enjoy what you do and have lots of fun!
10. Explore the world! there's a lot different areas, applications, and opportunities locally and abroad.

References

[1] Victor Adamchik. Modular Arithmetic.

[2] Avi Kak. Computer and Network Security.

[3] Stallings, William. *Cryptography and Network Security: Principles and Practice, 7th Edition.* Person, 2017.