ICS 6D Notes

CHAPTER 8: INDUCTION AND RECURSION

* Sequence: special type of function where domain is set of consecutive integers
  + Denoted by {gk}
* gk--not g(k)—is a term of the sequence g with index k.
* Increasing sequence: ak < ak+1
* Non-decreasing sequence:  ak ≤ ak+1
* Geometric sequence: each term is multiplied by a fixed number called common ratio.
  + sk = a · rk, for k ≥ 0
* Arithmetic sequence: each term is added by a fixed number called common difference.
  + tn = t0 + dn, for n ≥ 0, where d is the common difference
* Recurrence relation: sequence defined by a function of previous terms
  + Geometric and arithmetic sequences are recurrence relations by nature
* Fibonacci sequence: fn = fn−1 + fn−2 for n≥2
* Summation (Σ): sum of terms in a sequence
  + Summation form: with the sigma and lower and upper limits denoted
  + Expanded form: each term written in expression [Ex: 13 + 23 + 33 + 43]
* Closed form for sum: mathematical expression for the value of the sum (w/o using summation notation)
  + Closed form arithmetic sequence sum:
  + Closed form geometric sequence sum:
* Mathematical Induction: a proof technique useful for proving statements about elements in a sequence
  + Two components:
    - Base case: establishes truth in theorem for the first value of sequence
    - Inductive step: establishes upon the truth for k that theorem holds for k+1
  + Theorem:
    - S(n) is true for all positive integers n, if:
      * 1) S(1) is true (the base case)
      * 2) For all k ∈ Z+, S(k) implies S(k+1) (the inductive step)
  + Inductive hypothesis: the presumption that S(k) is true in the statement “S(k) implies S(k+1)”
* Proving **IDENTITY**, **INEQUALITY, DIVISIBILITY** by induction.
  + Theorem: “For [range], [equation/inequality].”
  + Base Case: Prove theorem true for first value
  + Inductive Step: Prove theorem for k+1
* **RECURRENCE RELATION** by induction
  + Recurrence relation (RR): Ex) g0 = 1, g1 = g0 + 2n
  + Explicit formula (EF): Defines the recurrence relation without using a “previous term”
  + Theorem: RR = EF
  + Base case: RR(0) = EF(0)
  + Inductive step: RR(left) => RR(right), using EF(right) to get to RR(right)
* Inductive proof for: **CLOSED FORM SUM OF TERMS OF GEOMETRIC AND ARITHMETIC SEQUENCE, SET OPERATIONS**
* Principle of **strong** induction: proves k+1, by assuming k and any value less than k to be true
  + Base case: S(0) and S(1) are true
  + Inductive step: For every k ≥ 1, S(0) ^ S(1) ^ S(2) ^ … ^ S(k)) implies S(k+1)
* Well-ordering principle: states that any non-empty subset of the non-negative integers has a smallest element

[8.7]

* Recursive definition of a function: when the input value of the function is defined in terms of output values from the function’s smaller input values
* Components of a recursive definition of a *set*:
  + Basis: explicitly states that one or more specific elements are in the set
  + Recursive rule: shows how to construct larger elements in the set from elements already known to be in the set. (There is often more than one recursive rule).
  + Exclusion statement: states that an element is in the set only if it is stated in the basis or can be constructed by the recursive rule
* Empty string: denoted by lambda (λ):
* Infinite set: B\*, where B is a set

[8.8]

* Structural induction: type of induction used to prove theorems about recursively-defined sets that follow the structure of the recursive definition
* Recursive algorithm: an algorithm that calls itself

[8.11]

* Characteristic equation for a linear recurrence relation: can be used to solve for x, which is the base of the exponent in the solution.

[9.2]

* Ring: a closed mathematical system with n elements (**Zn**)
* Theorem: Equivalence mod n -- two integers have the same result mod n if and only if their difference is divisible by n
* Theorem: Congruence mod n -- x is congruent to y mod n if n|(x - y)
  + x  ≡  y (mod n).

[9.3 Prime Factorizations]

* The Fundamental Theorem of Arithmetic: Every positive integer other than 1 can be expressed uniquely as a product of prime numbers where the prime factors are written in non-decreasing order.
* The greatest common divisor (GCD) of non-zero integers x and y is the largest positive integer that is a factor of both x and y.
* The least common multiple (LCM) of non-zero integers x and y is the smallest positive integer that is an integer multiple of both x and y.
* Relatively prime: when two integers’ GCD is 1

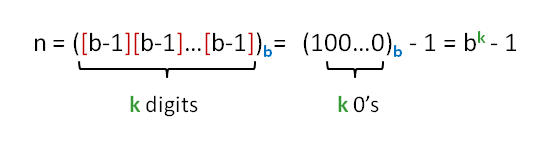
[9.4 Factoring and Primality Testing]

* Theorem: There is an infinite number of primes
* The Prime Number Theorem: The number of prime numbers in range 2 to x is: x/lnx
  + The *fraction* of prime numbers in range 3 to x is: 1/lnx
    - This is ALSO the probability/likelihood of randomly selecting a prime number

[9.5 GCD and Euclid’s algorithm]

* GCD Theorem/Euclid’s Algorithm: Let x and y be two positive integers. Then gcd(x, y) = gcd(x, y mod x)
* Extended Euclid’s Algorithm: expresses gcd(x,y) as linear combination of x and y
  + gcd(x, y) = sx + ty
* x hasr an inverse mod n if and only if x and n are relatively prime.

[9.6 Number Representation]

* Bit: digit in binary notation
* Base: premise of a unique number representation
* Decimal Expansion: expansion of a number by its base
  + EX: 23013 = (2 \* 3^3) + (3 \* 3^2) + (0 \* 3^1) + (1 + 3^0)
* Hexidecimal: numbers of base 16, so the digits are 0-9 and A-F
* Byte: 8 bits, or 2 hexidecimal numbers
* Converting decimal to any base b:
  + N divmod b: (n div b, n mod b)
  + Base expansion b: all the mod numbers from bottom up
* Largest number represented by k digits and base b:
  + 

[9.7 Fast Exponentiation]

* Fast integer exponentiation:
  + Binary expansion on the exponent
  + Multiply each base accordingly to each term produced from expanded exponent
  + Can mod each term for smaller numbers

[9.8 Intro to Cryptography]

* Cryptography: science of protecting and authenticating data and communication
* Sender sends encrypted message (turns plaintext to cyphertext) to the receiver who decrypts the cyphertext back to plaintext with the secret key.
* Private Key Cryptosystem: Alice and Bob must meet in advance to decide on secret key securely
* Simple Encryption Scheme Requirements: to verify that encryption/decryption is successful:
  + 1) encryption scheme is one-to-one (no two distinct plaintexts map to same ciphertext)
    - If m ≠ m' and m, m' ∈ ZN then (m + k) mod N ≠ (m' + k) mod N
  + 2) decryption scheme is the inverse of encryption scheme
    - If m ∈ ZN then (((m + k) mod N)-k) mod N = m

[9.9 The RSA Cryptosystem]

* Public key cryptography: relies on the difficulty of decrypting a message without the private decryption key, though the encryption key is public; only the receiver knows how to decrypt all the encrypted messages that anyone can encrypt with the public key
* RSA Cryptosystem: The security of the RSA cryptosystem rests on the assumption that it is *difficult to factor large numbers*.
* To prove the viability of the RSA Cryptosystem:
  + Chinese Remainder Theorem*:* to establish the correctness
    - Let p and q be prime numbers and pq = N. Suppose that m ∈ **Z**N and gcd(m, N)=1. Then m(p-1)(q-1) mod N = 1
  + Fermat’s Little Theorem: to establish the validity of the RSA cryptosystem
    - If m ∈ **Z**N and gcd(m, N) =1, then RSA encryption and decryption always yield m as the unique result.
* Decryption yields: cd mod N = me⋅d mod N

[10.1 Sum and Product Rules]

* Product rule: a way to count sequences by multiplying the cardinality of each set.
* Alphabet (Σ): set of characters
* Σn : set of all characters of n length from the set Σ
  + Ex: If Σ = {0, 1}, Σ6 is the set of all binary strings of length 6 [Ex: 100110]
* Sum rule: If sets are mutually disjoint (no combinations are made; only one selection made from all sets), then we add the cardinality of each set

[10.2 Bijection Rule]

* Bijection rule: Let S and T be two finite sets. If there is a bijection from S to T, then |S| = |T|.
* k-to-1 rule: requires a well-defined function from objects we can count to objects we would like to count
  + Let X and Y be finite sets. The function f:X→Y is a k-to-1 correspondence if for every y ∈ Y, there are exactly k different x ∈ X such that f(x) = y.
  + Suppose there is a k-to-1 correspondence from a finite set A to a finite set B. Then |B| = |A|/k.

[10.4 Counting Permutations]

* r-permutation: a sequence of r items with no repetitions, all taken from the same set
* P(n,r) = n! / (n-r)!
  + n = # of elements in set; r = denotes of r-permutation
* Permutation (without the parameter r): a sequence that contains each element of a finite set exactly once

[10.5 Counting subset]

* r-combination = r-subset
* # of r-subsets from S = (# of r-permutations from S) / (r!)

= P(n, r) / r!

= n! / r! (n-r)!

* + This is “n choose r” 🡪 C(n,r) 🡪 n! / r! (n-r)!

[12.1 Probability of an Event]

* Experiment: procedure that results in a number of possible outcomes
* Sample Space: the set of all possible outcomes
* Event: a subset of the sample space
  + the event E that the sum of the dice is exactly 8 is the following set: E={(2,6),(3,5)(4,4),(5,3),(6,2)}
* Countably infinite: there is a one-to-one correspondence between the elements of the set and integers
  + (Z x Z), set of binary strings of any length, etc.
* Uncountably infinite: describes a set that is not countably infinite
  + Set of real numbers
* Probability distribution (over the outcomes of an experiment with a countable sample space S): a function p from S to the set of real numbers in the interval from 0 to 1 with the property that
* Uniform distribution: The probability distribution in which every outcome has the same probability
  + Under uniform distribution, p(s) = 1/|S| for each s ∈ S, with S = sample space