Data Structure

Lecture#3: Mathematical Preliminaries

U Kang
Seoul National University

In This Lecture

- Set concepts and notation
- Logarithms
- Summations
- Recurrence Relations
- Recursion
- Induction Proofs

Set (1)

$\{1, 4\}$	A set composed of the members 1 and 4
$\{x \mid x \text{ is a positive integer}\}$	A set definition using a **set former** Example: the set of all positive integers
$x \in \mathbf{P}$	x is a member of set **P**
$x \notin \mathbf{P}$	x is not a member of set **P**
$\emptyset$	The null or empty set
$\lvert\mathbf{P}\rvert$	Cardinality: size of set **P** or number of members for set **P**
$\mathbf{P} \subseteq \mathbf{Q}$, $\mathbf{Q} \supseteq \mathbf{P}$	Set **P** is included in set **Q**, set **P** is a subset of set **Q**, set **Q** is a superset of set **P**
$\mathbf{P} \cup \mathbf{Q}$	Set Union: all elements appearing in **P** OR **Q**
$\mathbf{P} \cap \mathbf{Q}$	Set Intersection: all elements appearing in **P** AND **Q**
$\mathbf{P} - \mathbf{Q}$	Set difference: all elements of set **P** NOT in set **Q**

Figure 2.1 Set notation.

Set (2)

- Powerset 2^S of a set S : set of all possible subsets for S.
  - E.g., powerset of S = {a, b, c} is {∅, {a}, {b}, {c}, {a, b}, {a, c}, {b, c}, {a,b,c}}
- Let |S| = n. |powerset of S| = ?
- A set with duplicates is called multiset (or bag or multilist)

Relation (1)

- A relation R over a set S is a set of ordered pairs from S.
- Example: $<$ $>$ $=$
- Common Properties:
  - R is **reflexive** if aRa for all $a \in \mathbf{S}$.
  - R is **symmetric** if whenever aRb, then bRa, for all $a, b \in \mathbf{S}$.
  - R is **antisymmetric** if whenever aRb and bRa, then $a = b$, for all $a, b \in \mathbf{S}$.
  - R is **transitive** if whenever aRb and bRc, then aRc, for all $a, b, c \in \mathbf{S}$.

Relation (2)

	<	<=	=	“Friend of”
Reflexive				
Symmetric				
Antisymmetric				
Transitive				

- R is **reflexive** if aRa for all $a \in \mathbf{S}$.
- R is **symmetric** if whenever aRb, then bRa, for all $a, b \in \mathbf{S}$.
- R is **antisymmetric** if whenever aRb and bRa, then $a = b$, for all $a, b \in \mathbf{S}$.
- R is **transitive** if whenever aRb and bRc, then aRc, for all $a, b, c \in \mathbf{S}$.

Relation (3)

	<	<=	=	“Friend of”
Reflexive	X	O	O	X
Symmetric	X	X	O	O
Antisymmetric	O	O	O	X
Transitive	O	O	O	X

- R is **reflexive** if aRa for all $a \in \mathbf{S}$.
- R is **symmetric** if whenever aRb, then bRa, for all $a, b \in \mathbf{S}$.
- R is **antisymmetric** if whenever aRb and bRa, then $a = b$, for all $a, b \in \mathbf{S}$.
- R is **transitive** if whenever aRb and bRc, then aRc, for all $a, b, c \in \mathbf{S}$.

Relation (4)

- Equivalence relation: reflexive, symmetric, and transitive
  - E.g. 1) =
  - E.g. 2) = under modulo operation
    - Def) If n = qm + r, then n = r (mod m).
    - x = x (mod m)
    - If x = y (mod m), then y = x (mod m)
    - If x = y (mod m) and y = z (mod m), then x = z (mod m)
- An equivalence relation partitions a set into equivalence classes
  - 1 = 6 = 11 = … (mod 5)
  - 2 = 7 = 12 = … (mod 5)

Relation (5)

- Partial order: antisymmetric and transitive
  - E.g. 1) <, <=
  - E.g. 2) "heavier and taller" relation
- Total order: partial order where all pairs are comparable
  - E.g., <, <=

Notation for Numbers

- b: bit, B: byte
- KB: kilobyte (1000 byte ~ 2^{10} byte)
- MB: megabyte (10^6 byte ~ 2^{20} byte)
  - 1 million byte
- GB: gigabyte (10^9 byte ~ 2^{30} byte)
  - 1 billion byte
- TB: terabyte (10^{12} byte ~ 2^{40} byte)
  - 1 trillion byte
- PB: petabyte (10^{15} byte ~ 2^{50} byte)
  - 1 quadrillion byte
- 2^{32} byte ~ 4 GB

Notation - Excercises

- Assume you want to assign unique numerical ids to 50 million people. How many bits are required to express the ids?
- What is the amount of memory required to load 10,000 images at once?
- How much disk storage is required to save 20 images for each of 50 million people.
- Train yourself to answer each of the above questions in 5~10 seconds (w/o pencil and paper)

Miscellaneous Notation

- ms: millisecond
- Factorial n! = n * (n-1) * … * 1
- Permutation of a sequence S: S in some order
  - Random permutation: permutation in a random order

```
/** Randomly permute the values in array A */
static <E> void permute(E[] A) {
  for (int i = A.length; i > 0; i--) // for each i
    swap(A, i-1, DSutil.random(i));  //   swap A[i-1] with
}                                    //   a random element
```

Miscellaneous Notation

- Logic Notation
  - A => B : A implies B, or If A then B
  - A ⇔ B : A if and only if B (= A iff B)
  - A ∧ B
  - A ∨ B
  - ~A, $\bar{A}$: not A

- Flooring and Ceiling
  - Flooring(x): greatest integer <=x. E.g., $\lfloor 1.4 \rfloor = 1$
  - Ceiling(x): least integer >= x. E.g., $\lceil 1.4 \rceil = 2$

Logarithm

- Definition:

Logarithms have the following properties, for any positive values of m, n, and r, and any positive integers a and b.

1. $\log(nm) = \log n + \log m$.
2. $\log(n/m) = \log n - \log m$.
3. $\log(n^r) = r \log n$.
4. $\log_a n = \log_b n / \log_b a$.

=> No need to worry about the base of the algorithm

5. $2^{logn} = n$

Logarithm

- Application of Logarithm
  - What is the minimum # of bits to represent *n* distinct items?
  - Binary search – given an array with *n* sorted items, how quickly can we find an item of interest, in the worst case?
- Notation
  - log log n = log (log n)
  - $\log^2 n = (\log n)^2$
  - log*n : # of times to take $\log_2$ before the value <= 1
    - E.g. log* 1024 = 4 (1024 => 10 => 3.33 => 1.74 => 0.8)

Summation

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}.$$

$$\sum_{i=1}^{n} i^2 = \frac{2n^3+3n^2+n}{6} = \frac{n(2n+1)(n+1)}{6}.$$

$$\sum_{i=1}^{\log n} n = n \log n.$$

$$\sum_{i=0}^{\infty} a^i = \frac{1}{1-a} \text{ for } 0 < a < 1.$$

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1} \text{ for } a \neq 1.$$

$$\sum_{i=1}^{n} \frac{i}{2^i} = 2 - \frac{n+2}{2^n}.$$

Summation

- Harmonic Series
  - $\sum_{i=1}^{n} \frac{1}{i} \sim \log(n)$
  - Pf?

Questions?