Fast Multiplication of Integers

Example 4, p. 528 (p. 475, 6th ed), shows a method of multiplying 2n bit integers which is more efficient than the standard method shown in Example 10, p. 252 (p.225 6th ed). This is interesting, but somewhat technical and is left to the curiosity of the ambitious reader.

Fast Matrix Multiplication

Example 5, p. 529 (p.476, 6th ed) describes a method for matrix multiplication of a 2n x 2n matrix which works by manipulating the main n x n blocks to improve efficiency.

If f(n) is the number of multiplications and additions used, it turns out that

$f(n) = 7f(n/2) + 15n^2/4$

We apply this by embedding our matrix in a matrix of size $n = 2^k$ and repeatedly dividing by 2, and estimating the form of our recurrence relation for f.

$f(n) = 7f(n/2) + 15n^2/4$ has the form

$f(n) = af(n/b) + g(n)$ [now apply this formula again]

$= a^2f(n/b^2) + ag(n/b) + g(n)$

$= a^3f(n/b^3) + a^2g(n/b^2) + ag(n/b) + g(n)$

$= \ldots$

$= a^kf(n/b^k) + \Sigma_{j=0}^{k-1} a^jg(n/b^j)$ [now use $n=b^k$]

$= a^kf(1) + \Sigma_{j=0}^{k-1} a^jg(n/b^j)$

$= a^kf(1) + \Sigma_{j=0}^{k-1} a^jg(b^{k-j})$

It turns out that this type of formula can be used to estimate the big-O growth of f.

Divide-and-Conquer Recurrences

Usually, we do not try to solve such divide-and conquer recurrences, but we use them to derive a **big-O estimate** for the complexity of an algorithm.

Divide & Conquer, first theorem

Theorem 1 (see p 530, p 477, 6th ed). If f is an increasing function satisfying:

$f(n) = af(n/b) + c$, $b|n$, $b>1$ an integer and $c>0$ real then

if $a > 1$ then f(n) is O(n^log$_b$a)

if $a = 1$ then f(n) is O(log n).

If $n = b^k$, $k > 0$ an integer then

f(n) = C$_1$n^log$_b$a + C$_2$. (^ means power)

$C_1 = f(1) + c/(a-1)$, $C_2 = -c/(a-1)$

Proof: to iterate, assume $n = b^k$.(else bound n above by b^k)

$f(n) = af(n/b) + c$.

$= a(af(n/b^2) + c) + c = a^2f(n/b^2) + ac + c$

$= a^3f(n/b^3) + a^2c + ac + c = \ldots$

$= a^kf(n/b^k) + c\Sigma_{i=0}^{k-1}a^i = a^kf(1) + c\Sigma_{i=0}^{k-1}a^i$,(geometric series)

Case 1: a = 1: $f(n) = f(1) + ck = f(1) + c\log_b n$

Case 2: a > 1: $f(n) = a^kf(1) + c(a^k - 1)/(a - 1)$

$= a^k[f(1) + c/(a-1)] - c/(a-1)$ (used geometric series formula)

= un^log$_b$a + v, since a^k = a^log$_b$n = n^log$_b$a

Note: $\log(u^v) = v\log(u)$, so

$\log(a^{\log(n)}) = \log(n)\log(a) = \log(n^{\log(a)})$.

So $a^{\log(n)} = n^{\log(a)}$. Here log means $\log_b$

Divide-and-Conquer

Example 7 p. 531 (p 478, 6th ed) using Theorem 1:

For binary search, we have the number of operations $f(n) = f(n/2) + 2$, so $a = 1$, $c = 2$.

Consequently, by theorem 1,
$f(n)$ is $O(\log n)$.

The binary search algorithm has logarithmic time complexity.

Divide-and-Conquer

Theorem 2: (p 532. p 479 6th ed. No proof here)
Let f be an increasing function that satisfies the recurrence relation

$f(n) = af(n/b) + cn^d$

whenever $n = b^k$, where k is a positive integer, a, c, and d are real numbers with $a \geq 1$, and b is an integer greater than 1. Then f(n) is
$O(n^d)$, if $a < b^d$,
$O(n^d \log n)$ if $a = b^d$,
$O(n^{\log_b a})$ if $a > b^d$

Divide-and-Conquer

Example 7 p. 531 using Theorem2:

For binary search, we have the number of operations $f(n) = f(n/2) + 2$, so $a = 1$, $b = 2$, and $d = 0$
($d = 0$ because here, g(n) does not depend on n).

Consequently, $a = b^d$, and therefore,
$f(n)$ is $O(n^d \log n) = O(\log n)$.

The binary search algorithm has logarithmic time complexity.

Divide-and-Conquer

Example 11, p532 (p479 6th ed).

Fast matrix multiplication of n x n matrices uses $f(n) = 7f(n/2) + 15n^2/4$ additions and multiplications, so by Th. 2, since $7 > 2^2$, ($a > b^d$) $f(n)$ is $O(n^{\log 7})$, and $\log_2 7 \sim 2.8$. The standard method of multiplying matrices is $O(n^3)$.

Let's look at …

Relations

(Chapter 9)

Relations

If we want to describe a relationship between elements of two sets A and B, we can use **ordered pairs** with their first element taken from A and their second element taken from B.

Since this is a relation between **two sets**, it is called a **binary relation**.

Definition: Let A and B be sets. A binary relation from A to B is a subset of $A\times B$.

In other words, for a binary relation R we have $R \subseteq A\times B$. We use the notation aRb to denote that $(a, b)\in R$ and $a\underline{R}b$ to denote that $(a, b)\notin R$.

Relations

When (a, b) belongs to R, a is said to be **related** to b by R.

Example: Let P be a set of people, C be a set of cars, and D be the relation describing which person drives which car(s).

P = {Carl, Suzanne, Peter, Carla},

C = {Mercedes, BMW, tricycle}

D = {(Carl, Mercedes), (Suzanne, Mercedes), (Suzanne, BMW), (Peter, tricycle)}

This means that Carl drives a Mercedes, Suzanne drives a Mercedes and a BMW, Peter drives a tricycle, and Carla does not drive any of these vehicles.

Functions as Relations

You might remember that a **function** f from a set A to a set B assigns a unique element of B to each element of A.

The **graph** of f is the set of ordered pairs (a, b) such that b = f(a).

Since the graph of f is a subset of $A \times B$, it is a **relation** from A to B.

Moreover, for each element **a** of A, there is exactly one ordered pair in the graph that has **a** as its first element.

Functions as Relations

Conversely, if R is a relation from A to B such that every element in A is the first element of exactly one ordered pair of R, then a function can be defined with R as its graph.

This is done by assigning to an element $a \in A$ the unique element $b \in B$ such that $(a, b) \in R$.

Relations on a Set

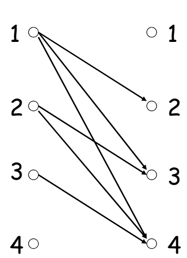
Definition: A relation on the set A is a relation from A to A.

In other words, a relation on the set A is a subset of $A \times A$.

Example: Let A = {1, 2, 3, 4}. Which ordered pairs are in the relation R = {(a, b) | a < b} ?

Relations on a Set

Solution: R = {(1, 2),(1, 3),(1, 4),(2, 3),(2, 4),(3, 4)}



R	1	2	3	4
1		X	X	X
2			X	X
3				X
4				

Relations on a Set

How many different relations can we define on a set A with n elements?

A relation on a set A is a subset of $A \times A$.
How many elements are in $A \times A$?

There are n^2 elements in $A \times A$, so how many subsets (= relations on A) does $A \times A$ have?

The number of subsets that we can form out of a set with m elements is 2^m. Therefore, 2^{n^2} subsets can be formed out of $A \times A$.

Answer: We can define 2^{n^2} different relations on A.

Properties of Relations

We will now look at some useful ways to classify relations.

Definition: A relation R on a set A is called **reflexive** if $(a, a) \in R$ for every element $a \in A$.

Are the following relations on {1, 2, 3, 4} reflexive?

R = {(1, 1), (1, 2), (2, 3), (3, 3), (4, 4)}	No.
R = {(1, 1), (2, 2), (2, 3), (3, 3), (4, 4)}	Yes.
R = {(1, 1), (2, 2), (3, 3)}	No.

Definition: A relation on a set A is called **irreflexive** if $(a, a) \notin R$ for every element $a \in A$.

Properties of Relations

Definitions:

A relation R on a set A is called **symmetric** if $(b, a) \in R$ whenever $(a, b) \in R$ for all $a, b \in A$.

A relation R on a set A is called **antisymmetric** if $a = b$ whenever $(a, b) \in R$ and $(b, a) \in R$.

A relation R on a set A is called **asymmetric** if $(a, b) \in R$ implies that $(b, a) \notin R$ for all $a, b \in A$.

Properties of Relations

Are the following relations on {1, 2, 3, 4} symmetric, antisymmetric, or asymmetric?

R = {(1, 1), (1, 2), (2, 1), (3, 3), (4, 4)}	symmetric
R = {(1, 1)}	sym. and antisym.
R = {(1, 3), (3, 2), (2, 1)}	antisym. and asym.
R = {(4, 4), (3, 3), (1, 4)}	antisym.

Properties of Relations

Definition: A relation R on a set A is called **transitive** if whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$ for $a, b, c \in A$.

Are the following relations on {1, 2, 3, 4} transitive?

R = {(1, 1), (1, 2), (2, 2), (2, 1), (3, 3)}	Yes.
R = {(1, 3), (3, 2), (2, 1)}	No.
R = {(2, 4), (4, 3), (2, 3), (4, 1)}	No.

Counting Relations

Example: How many different reflexive relations can be defined on a set A containing n elements?

Solution: Relations on R are subsets of $A \times A$, which contains n^2 elements.

Therefore, different relations on A can be generated by choosing different subsets out of these n^2 elements, so there are 2^{n^2} relations.

A **reflexive** relation, however, **must** contain the n elements (a, a) for every $a \in A$.

Consequently, we can only choose among $n^2 - n = n(n - 1)$ elements to generate reflexive relations, so there are $2^{n(n-1)}$ of them.

Combining Relations

Relations are sets, and therefore, we can apply the usual **set operations** to them.

If we have two relations R_1 and R_2, and both of them are from a set A to a set B, then we can combine them to $R_1 \cup R_2$, $R_1 \cap R_2$, or $R_1 - R_2$.

In each case, the result will be **another relation from A to B**.

Combining Relations

… and there is another important way to combine relations.

Definition: Let R be a relation from a set A to a set B and S a relation from B to a set C. The **composite** of R and S is the relation consisting of ordered pairs (a, c), where $a \in A$, $c \in C$, and for which there exists an element $b \in B$ such that $(a, b) \in R$ and $(b, c) \in S$. We denote the composite of R and S by **$S \circ R$**.

In other words, if relation R contains a pair (a, b) and relation S contains a pair (b, c), then $S \circ R$ contains a pair (a, c).

Combining Relations

Example: Let D and S be relations on A = {1, 2, 3, 4}.

D = {(a, b) | b = 5 - a} "b equals (5 – a)"

S = {(a, b) | a < b} "a is smaller than b"

D = {(1, 4), (2, 3), (3, 2), (4, 1)}

S = {(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)}

$S \circ D$ = { (2, 4), (3, 3), (3, 4), (4, 2), (4, 3), (4, 4)}

D maps an element a to the element (5 – a), and afterwards S maps (5 – a) to all elements larger than (5 – a), resulting in **$S \circ D$ = {(a,b) | b > 5 – a}** or **$S \circ D$ = {(a,b) | a + b > 5}.**

Combining Relations

We already know that **functions** are just **special cases** of **relations** (namely those that map each element in the domain onto exactly one element in the codomain).

If we formally convert two functions into relations, that is, write them down as sets of ordered pairs, the composite of these relations will be exactly the same as the composite of the functions (as defined earlier).

Combining Relations

Definition: Let R be a relation on the set A. The powers R^n, n = 1, 2, 3, …, are defined inductively by

$R^1 = R$

$R^{n+1} = R^n \circ R$

In other words:

$R^n = R \circ R \circ \dots \circ R$ (n times the letter R)

Combining Relations

Theorem: The relation R on a set A is transitive if and only if $R^n \subseteq R$ for all positive integers n.

Remember the definition of transitivity:

Definition: A relation R on a set A is called transitive if whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$ for $a, b, c \in A$.

The composite of R with itself contains exactly these pairs (a, c).

Therefore, for a transitive relation R, $R \circ R$ does not contain any pairs that are not in R, so $R \circ R \subseteq R$.

Since $R \circ R$ does not introduce any pairs that are not already in R, it must also be true that $(R \circ R) \circ R \subseteq R$, and so on, so that $R^n \subseteq R$.

Combining Relations

Another Example: Let X and Y be relations on A = {1, 2, 3, …}.

X = {(a, b) | b = a + 1} "b equals a plus 1"

Y = {(a, b) | b = 3a} "b equals 3 times a"

X = {(1, 2), (2, 3), (3, 4), (4, 5), …}

Y = {(1, 3), (2, 6), (3, 9), (4, 12), …}

$X \circ Y$ = { (1, 4), (2, 7), (3, 10), (4, 13), …}

Y maps an element a to the element 3a, and afterwards X maps 3a to 3a + 1.

$X \circ Y$ = {(a,b) | b = 3a + 1}

n-ary Relations

In order to study an interesting application of relations, namely **databases**, we first need to generalize the concept of binary relations to **n-ary relations**.

Definition: Let $A_1, A_2, \ldots, A_n$ be sets. An **n-ary relation** on these sets is a subset of $A_1 \times A_2 \times \ldots \times A_n$.
The sets $A_1, A_2, \ldots, A_n$ are called the **domains** of the relation, and n is called its **degree**.

n-ary Relations

Example:
Let $R = \{(a, b, c) \mid a = 2b \wedge b = 2c \text{ with } a, b, c \in \mathbf{Z}\}$
What is the degree of R?
The degree of R is 3, since its elements are triples.
What are its domains?
Its domains are all equal to the set of integers.
Is (2, 4, 8) in R?
No.
Is (4, 2, 1) in R?
Yes.