Divide-and-Conquer Algorithms and Recurrence Relations

Section 8.3

Section Summary

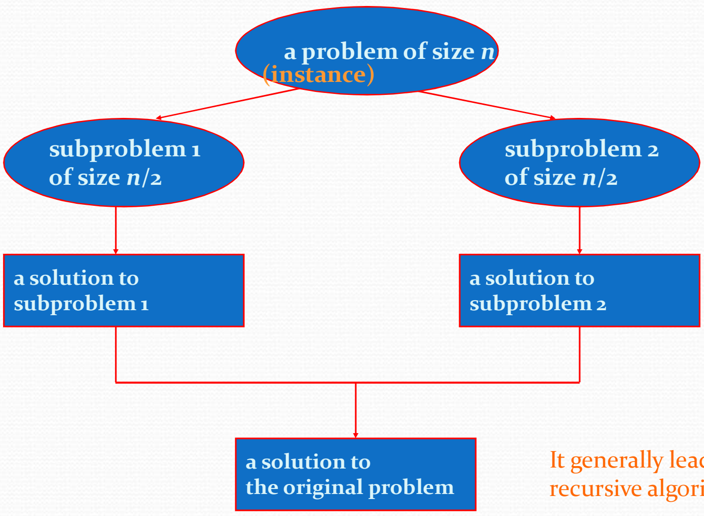
- Divide-and-Conquer Algorithms and Recurrence Relations
- Examples
  - Binary Search
  - Merge Sort
  - Fast Multiplication of Integers
- Master Theorem
- Closest Pair of Points (*not covered yet in these slides*)

Divide-and-Conquer

The most-well known algorithm design strategy:

1. Divide instance of problem into two or more smaller instances
2. Solve smaller instances recursively
3. Conquer the solution to original (larger) instance by combining these solutions

Divide-and-Conquer (cont.)



It generally leads to a recursive algorithm!

Divide-and-Conquer Algorithmic Paradigm

Definition: A *divide-and-conquer algorithm* works by first *dividing* a problem into one or more instances of the same problem of smaller size and then *conquering* the problem using the solutions of the smaller problems to find a solution of the original problem.

Examples:

- Binary search, covered in Chapters 3 and 5: It works by comparing the element to be located to the middle element. The original list is then split into two lists and the search continues recursively in the appropriate sublist.
- Merge sort, covered in Chapter 5: A list is split into two approximately equal sized sublists, each recursively sorted by merge sort. Sorting is done by successively merging pairs of lists.

Divide-and-Conquer Recurrence Relations

- Suppose that a recursive algorithm divides a problem of size n into a (a is a positive integer) subproblems.
- Assume each subproblem is of size n/b. (b is a positive integer)
- Suppose $g(n)$ extra operations are needed in the conquer step.
- Then $f(n)$ represents the number of operations to solve a problem of size n satisisfies the following recurrence relation:

  $$f(n) = af(n/b) + g(n)$$
- This is called a *divide-and-conquer recurrence relation.*

Example: Binary Search

- Binary search reduces the search for an element in a sequence of size n to the search in a sequence of size $n/2$. Two comparisons are needed to implement this reduction;
  - one to decide whether to search the upper or lower half of the sequence and
  - the other to determine if the sequence has elements.
- Hence, if $f(n)$ is the number of comparisons required to search for an element in a sequence of size n, then

$$f(n) = f(n/2) + 2$$

when n is even.

Example: Merge Sort

- The merge sort algorithm splits a list of n (assuming n is even) items to be sorted into two lists with $n/2$ items. It uses fewer than n comparisons to merge the two sorted lists.
- Hence, the number of comparisons required to sort a sequence of size n, is no more than than $M(n)$ where

$$M(n) = 2M(n/2) + n.$$

Example: Fast Multiplication of Integers

- An algorithm for the fast multiplication of two $2n$-bit integers (assuming n is even) first splits each of the $2n$-bit integers into two blocks, each of n bits.
- Suppose that a and b are integers with binary expansions of length $2n$. Let
  $a = (a_{2n-1}a_{2n-2} \dots a_1a_0)_2$ and $b = (b_{2n-1}b_{2n-2} \dots b_1b_0)_2$.
- Let $a = 2^nA_1 + A_0$, $b = 2^nB_1 + B_0$, where
  $A_1 = (a_{2n-1} \dots a_{n+1}a_n)_2$, $A_0 = (a_{n-1} \dots a_1a_0)_2$,
  $B_1 = (b_{2n-1} \dots b_{n+1}b_n)_2$, $B_0 = (b_{n-1} \dots b_1b_0)_2$.
- The algorithm is based on the fact that ab can be rewritten as:
- $ab = 2^{2n} A_1B_1 + 2^n (A_1 B_0 + A_0 B_1) + A_0B_0$.
  $ab = (2^{2n} + 2^n)A_1B_1 + 2^n (A_1 - A_0)(B_0 - B_1) + (2^n + 1)A_0B_0$.
- This identity shows that the multiplication of two $2n$-bit integers can be carried out using three multiplications of n-bit integers, together with additions, subtractions, and shifts.
- Hence, if $f(n)$ is the total number of operations needed to multiply two n-bit integers, then

  $f(2n) = 3f(n) + Cn$

  where Cn represents the total number of bit operations; the additions, subtractions and shifts that are a constant multiple of n-bit operations.

Estimating the Size of Divide-and-Conquer Functions

Theorem 1: Let f be an increasing function that satisfies the recurrence relation

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

whenever n is divisible by b, where $a \geq 1$, b is an integer greater than 1, and c is a positive real number. Then

$$f(n) \text{ is } \begin{cases} O(n^{\log_b a}) & \text{if } \; a > 1 \\ O(\log n) & \text{if } \; a = 1. \end{cases}$$

Furthermore, when $n = b^k$ and $a \neq 1$, where k is a positive integer,

$$f(n) = C_1 n^{\log_b a} + C_2$$

where $C_1 = f(1) + c/(a-1)$ and $C_2 = -c/(a-1)$.

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

Theorem 1:

If $a = 1$, $f(n) \in \Theta(\log n)$

If $a > 1$, $f(n) \in \Theta(n^{\log_b a})$

$f(n) = af(n/b) + C = a^2 f(n/b^2) + aC + C$

$= a^3 f(n/b^3) + a^2 C + aC + C$

$= \ldots\ldots$

$= a^k f(n/b^k) + a^{k-1} C + a^{k-2} C + \ldots\ldots + C$

Let $n = b^k$, $n/b^k = 1$,

$= a^k f(1) + \sum_{j=0}^{k-1} a^j c$

$f(n) = af(n/b) + C$

If a = 1, then f(n) = f(1) + Ck

Since $n= b^k$, $k = \log_b n$, hence

$f(n) = f(1) + c \log_b n$

When a>1 and **$n= b^k$**,

$f(n)= a^k f(1) +$ $c(a^k - 1)/(a-1)$

$= a^k[f(1) + c/(a-1)] - c/(a-1)$

$= C_1 n^{\log_b a} + C_2$

$a^k = a^{\log_b n} = n^{\log_b a}$

Complexity of Binary Search

Binary Search Example: Give a big-O estimate for the number of comparisons used by a binary search.

Solution: Since the number of comparisons used by binary search is $f(n) = f(n/2) + 2$ where n is even, by Theorem 1, it follows that $f(n)$ is $O(\log n)$.

Estimating the Size of Divide-and-conquer Functions (*continued*)

Theorem 2. Master Theorem: Let f be an increasing function that satisfies the recurrence relation

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

whenever $n = b^k$, where $a \geq 1$, b is an integer greater than 1, k is a positive integer greater than 1, and c and d are real numbers with c positive and d nonnegative. Then

$$f(n) \text{ is } \begin{cases} O(n^d) & \text{if } a < b^d, \\ O(n^d \log n) & \text{if } a = b^d, \\ O(n^{\log_b a}) & \text{if } a > b^d. \end{cases}$$

Complexity of Merge Sort

Merge Sort Example: Give a big-O estimate for the number of comparisons used by merge sort.

Solution: Since the number of comparisons used by merge sort to sort a list of n elements is less than $M(n)$ where $M(n) = 2M(n/2) + n$, by the master theorem $M(n)$ is $O(n \log n)$.

Complexity of Fast Integer Multiplication Algorithm

Integer Multiplication Example: Give a big-O estimate for the number of bit operations used needed to multiply two n-bit integers using the fast multiplication algorithm.

Solution: We have shown that $f(n) = 3f(n/2) + Cn,$ when n is even, where $f(n)$ is the number of bit operations needed to multiply two n-bit integers. Hence by the master theorem with $a = 3$, $b = 2$, $c = C$, and $d = 1$ (so that we have the case where $a > b^d$), it follows that $f(n)$ is $O(n^{\log 3})$.

Note that $\log 3 \approx 1.6$. Therefore the fast multiplication algorithm is a substantial improvement over the conventional algorithm that uses $O(n^2)$ bit operations.

Homework

Sec. 8.3 22

Generating Functions

Section 8.4

Section Summary

- Generating Functions
- Useful Generating Functions
- Counting Problems and Generating Functions
- Solving Recurrence Relations Using Generating Functions
- Proving Identities Using Generating Functions

Generating Functions

Definition: The *generating function for the sequence* $a_0, a_1, \ldots, a_k, \ldots$ of real numbers is the infinite series

$$G(x) = a_0 + a_1 x + \cdots + a_k x^k + \cdots = \sum_{k=0}^{\infty} a_k x^k.$$

A generating function is a clothesline on which we hang up a sequence of numbers for display.

— Herbert Wilf, *Generating functionology* (1994)

- *Questions about the convergence of these series are ignored.*
- *The fact that a function has a unique power series around x = 0 will also be important*

Generating Functions

Examples:

- The sequence $\{a_k\}$ with $a_k = 3$ has the generating function $\sum_{k=0}^{\infty} 3x^k = \frac{3}{1-x}\ for|x| < 1$
- The sequence $\{a_k\}$ with $a_k = k + 1$ has the generating function has the generating function $\sum_{k=0}^{\infty}(k+1)x^k.$
- The sequence $\{a_k\}$ with $a_k = 2^k$ has the generating function has the generating function $\sum_{k=0}^{\infty} 2^k x^k.$

Generating Functions for Finite Sequences

- Generating functions for finite sequences of real numbers can be defined by extending a finite sequence $a_0, a_1, \dots, a_n$ into an infinite sequence by setting $a_{n+1} = 0$, $a_{n+2} = 0$, and so on.
- The generating function $G(x)$ of this infinite sequence $\{a_n\}$ is a polynomial of degree n because no terms of the form $a_j x^j$ with $j > n$ occur, that is,

$$G(x) = a_0 + a_1 x + \cdots + a_n x^n.$$

Generating Functions for Finite Sequences (continued)

Example: What is the generating function for the sequence 1,1,1,1,1,1?

Solution: The generating function of 1,1,1,1,1,1 is

$1 + x + x^2 + x^3 + x^4 + x^5$.

By Theorem 1 of Section 2.4, we have

$(x^6 - 1)/(x - 1) = 1 + x + x^2 + x^3 + x^4 + x^5$

when $x \neq 1$.

Consequently $G(x) = (x^6 - 1)/(x - 1)$ is the generating function of the sequence.

Generating Functions for Finite Sequences (continued)

Example:

Let $a_k = C(m,k), k = 0,1,2,\cdots,m$ **. The generating function for this sequence is**

$G(x) = C(m,0) + C(m,1)x + C(m,2)x^2 + \cdots + C(m,m)x^m = (1+x)^m$

Counting Problems and Generating Functions

Generating functions can be used to solve a wide variety of counting problems, such as

✓ **Count the number of combinations from a set when repetition is allowed and additional constraints exist.**

✓ **Count the number of permutations**

25

Counting Problems and Generating Functions

Example: Find the number of solutions of

$e_1 + e_2 + e_3 = 17,$

where e_1, e_2, and e_3 are nonnegative integers with $2 \le e_1 \le 5$, $3 \le e_2 \le 6$, and $4 \le e_3 \le 7$.

Solution: The number of solutions is the coefficient of x^{17} in the expansion of

$(x^2 + x^3 + x^4 + x^5)\ (x^3 + x^4 + x^5 + x^6)\ (x^4 + x^5 + x^6 + x^7).$

This follows because a term equal to is obtained in the product by picking a term in the first sum x^{e_1}, a term in the second sum x^{e_2}, and a term in the third sum x^{e_3}, where $e_1 + e_2 + e_3 = 17$.

There are three solutions since the coefficient of x^{17} in the product is 3.

〖Example 〗Determine the number of ways to insert tokens worth \$1,\$2 and \$5 into a vending machine to pay for an item that costs *r* dollars in both the cases when the order in which the tokens are inserted does not matter and when the order does matter.

Solution:

(1) The order in which the tokens are inserted does not matter

$$G(x) = (1 + x + x^2 + x^3 + \cdots)(1 + x^2 + x^4 + x^6 + \cdots)(1 + x^5 + x^{10} + x^{15} + \cdots)$$

The coefficient of x^r in the expansion of $G(x)$ is the solution of this problem.

(2) The order in which the tokens are inserted does matter

- **The number of ways to insert exactly n tokens to produce a total of r\$ is the coefficient of x^r in** $(x + x^2 + x^5)^n$
- **Since any number of tokens may be inserted, the number of ways to produce r\$ using \$1,\$2 and \$5 tokens, is the coefficient of x^r in**

$$1 + (x + x^2 + x^5) + (x + x^2 + x^5)^2 + \ldots = \frac{1}{1 - (x + x^2 + x^5)}$$

Counting Problems and Generating Functions (*continued*)

Example: Use generating functions to find the number of k-combinations of a set with n elements, i.e., $C(n,k)$.

Solution: Each of the n elements in the set contributes the term $(1 + x)$ to the generating function

$$f(x) = \sum_{k=0}^{n} a^k x^k.$$

Hence $f(x) = (1 + x)^n$ where $f(x)$ is the generating function for $\{a^k\}$, where a^k represents the number of k-combinations of a set with n elements.

By the binomial theorem, we have

$$f(x) = \sum_{k=0}^{n} \binom{n}{k} x^k,$$

where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Hence,

$$C(n,k) = \frac{n!}{k!(n-k)!}.$$

Useful Facts About Power Series

【Theorem 1】Let $f(x) = \sum_{k=0}^{\infty}$

(1) $f(x) + g(x) = \sum_{k=0}^{\infty} (a_k + b_k) x^k$

(2) $\alpha \cdot f(x) = \sum_{k=0}^{\infty} \alpha \cdot a_k x^k$ $\quad \alpha$

(3) $x \cdot f'(x) = \sum_{k=0}^{\infty} k \cdot a_k x^k$

(4) $f(\alpha x) = \sum_{k=0}^{\infty} \alpha^k \cdot a_k x^k$

(5) $f(x) g(x) = \sum_{k=0}^{\infty} \left(\sum_{j=0}^{k} a_j b_{k-j}\right) x^k$

Proof:

$$\sum_{k=0}^{\infty} k a_k x^k = \sum_{k=0}^{\infty} a_k \cdot x \cdot k x^{k-1}$$

$$= x \sum_{k=0}^{\infty} a_k (x^k)'$$

$$= x\left(\sum_{k=0}^{\infty} a_k x^k\right)'$$

$$= x f'(x)$$

30

Useful Facts About Power Series

Let $f(x)=\sum_{k=0}^{\infty} a_k x$

(1) $f(x)+g(x)=$

(2) $\alpha\cdot f(x)=\sum_{k=0}^{\infty}$

(3) $x\cdot f'(x)=$

(4) $f(\alpha x)=\sum_{k=0}^{\infty}$

(5) $f(x)g(x)=\sum_{k}$

Proof:

$$\sum_{k=0}^{\infty}(\sum_{j=0}^{k}a_j b_{k-j})x^k$$

$$=a_0b_0+(a_0b_1+a_1b_0)x+(a_0b_2+a_1b_1+a_2b_0)x^2$$

$$+\ldots+(\sum_{j=1}^{k}a_j b_{k-j})x^k+\ldots$$

$$=(a_0+a_1x+a_2x^2+\ldots)(b_0+b_1x+b_2x^2+\ldots)$$

$$=f(x)\cdot g(x)$$

31

Using the above properties, the generating functions of some sequences can be obtained easily.

〖Example 〗 What is the generating function for the sequence 0,1,2,3,4,5,…?

Solution:

$$b_k=k$$

$$G(x)=\sum_{k=0}^{\infty}kx^k$$

$$=x(\frac{1}{1-x})'$$

$$=\frac{x}{(1-x)^2}$$

32

【Example 】 **Suppose that the generating function of the sequence: $a_0, a_1, a_2, \cdots, a_n, \cdots$ is $G(x)$. What is the generating function for the sequence $b_k = \sum_{i=0}^{k} a_i$?**

Solution:

$a_k \leftrightarrow G(x), \quad b_k \leftrightarrow$

$c_k = 1$

$b_k = \sum_{i=0}^{k} a_i$

$= \sum_{i=0}^{k} a_i \times c_{k-i}$

$F(x) = G(x) \cdot \frac{1}{1-x}$

For example:

$1,1,1,\cdots \longleftrightarrow \frac{1}{1-x}$

$\Downarrow$

$1,2,3,4,\cdots,k+1,\cdots \longleftrightarrow \frac{1}{(1-x)^2}$

33

【Example 】 **Let $f(x) = \frac{1}{1-4x^2}$. Find the coefficient $a_0, a_1, a_2, \cdots, a_n, \cdots$ in the expansion $f(x) = \sum_{k=0}^{\infty} a_k x^k$.**

Solution:

$$f(x) = \frac{1}{1-4x^2} = \frac{1}{(1-2x)(1+2x)} = \frac{1}{2}\left(\frac{1}{1-2x} + \frac{1}{1+2x}\right)$$

$$2^k \qquad (-2)^k$$

$$a_k = \frac{1}{2}(2^k + (-2)^k) = \begin{cases} 2^k & k \text{ is even} \\ 0 & k \text{ is odd} \end{cases}$$

34

※ The extended binomial coefficient

Recall:

$$\binom{m}{k} = C(m,k) = \frac{m!}{k!(m-k)!}$$

Where *m*, *k* are nonnegative integers, *k*≤*m*

【Definition 2】 Let *u* be a real number and *k* a nonnegative integer. Then the *extended binomial coefficient* is defined by

$$\binom{u}{k} = \begin{cases} u(u-1)\cdots(u-k+1)/k! & \text{if } k>0 \\ 1 & \text{if } k=0 \end{cases}$$

35

〖Example 〗 (1) $\binom{1/2}{3} = ?$ (2) $\binom{-n}{k} = ?$

Solution:

$$(1)\binom{1/2}{3} = \frac{(1/2)(1/2-1)(1/2-2)}{3!} = 1/16$$

$$(2)\binom{-n}{k} = \frac{(-n)(-n-1)\ldots(-n-k+1)}{k!}$$

$$= \frac{(-1)^k n(n+1)\ldots(n+k-1)}{k!}$$

$$= (-1)^k C(n+k-1,k)$$

36

❈ The extended Binomial Theorem

【Theorem 2】Let x be a real number with $|x| < 1$ and let u be a real number. Then

$$(1+x)^u = \sum_{k=0}^{\infty} \binom{u}{k} x^k$$

37

〖Example 〗Find the generating functions for

$$(1+x)^{-n} \, and (1-x)^{-n}$$

where *n* is a positive integer, using the extended Binomial Theorem.

Solution:

By the extended Binomial Theorem , it follows that

$$(1+x)^{-n} = \sum_{k=0}^{\infty} \binom{-n}{k} x^k = \sum_{k=0}^{\infty} (-1)^k C(n+k-1,k) x^k$$

$$(1-x)^{-n} = \sum_{k=0}^{\infty} \binom{-n}{k} (-x)^k = \sum_{k=0}^{\infty} (-1)^k C(n+k-1,k)(-1)^k x^k = \sum_{k=0}^{\infty} C(n+k-1,k) x^k$$

38

Useful Generating Functions

TABLE 1 Useful Generating Functions.

$G(x)$	a_k
$(1+x)^n = \sum_{k=0}^{n} C(n,k)x^k = 1 + C(n,1)x + C(n,2)x^2 + \cdots + x^n$	$C(n,k)$
$(1+ax)^n = \sum_{k=0}^{n} C(n,k)a^k x^k = 1 + C(n,1)ax + C(n,2)a^2x^2 + \cdots + a^n x^n$	$C(n,k)a^k$
$(1+x^r)^n = \sum_{k=0}^{n} C(n,k)x^{rk} = 1 + C(n,1)x^r + C(n,2)x^{2r} + \cdots + x^{rn}$	$C(n,k/r)$ if $r \mid k$; 0 otherwise
$\frac{1-x^{n+1}}{1-x} = \sum_{k=0}^{n} x^k = 1 + x + x^2 + \cdots + x^n$	1 if $k \le n$; 0 otherwise
$\frac{1}{1-x} = \sum_{k=0}^{\infty} x^k = 1 + x + x^2 + \cdots$	1
$\frac{1}{1-ax} = \sum_{k=0}^{\infty} a^k x^k = 1 + ax + a^2x^2 + \cdots$	a^k
$\frac{1}{1-x^r} = \sum_{k=0}^{\infty} x^{rk} = 1 + x^r + x^{2r} + \cdots$	1 if $r \mid k$; 0 otherwise
$\frac{1}{(1-x)^2} = \sum_{k=0}^{\infty} (k+1)x^k = 1 + 2x + 3x^2 + \cdots$	$k+1$
$\frac{1}{(1-x)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)x^k = 1 + C(n,1)x + C(n+1,2)x^2 + \cdots$	$C(n+k-1,k) = C(n+k-1,n-1)$
$\frac{1}{(1+x)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)(-1)^k x^k = 1 - C(n,1)x + C(n+1,2)x^2 - \cdots$	$(-1)^k C(n+k-1,k) = (-1)^k C(n+k-1,n-1)$
$\frac{1}{(1-ax)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)a^k x^k = 1 + C(n,1)ax + C(n+1,2)a^2x^2 + \cdots$	$C(n+k-1,k)a^k = C(n+k-1,n-1)a^k$
$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$	$1/k!$
$\ln(1+x) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} x^k = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots$	$(-1)^{k+1}/k$

Note: The series for the last two generating functions can be found in most calculus books when power series are discussed.

Sequence	**Generating function**
(1) $C(n,k)$	$\sum_{k=0}^{\infty} C(n,k)x^k = (1+x)^n$
(2) $C(n,k)a^k$	$(1+ax)^n$
(3) 1,1,…,1	$1+x+x^2+\cdots+x^n = \frac{1-x^{n+1}}{1-x}$
(4) $1,1,1,\cdots$	$\frac{1}{1-x}$
(5) a^k	$\frac{1}{1-ax}$
(6) $k+1$	$\frac{1}{(1-x)^2}$

40

Sequence	Generating function
(7) $C(n+k-1,k)$	$(1-x)^{-n}$
(8) $(-1)^k C(n+k-1,k)$	$(1+x)^{-n}$
(9) $C(n+k-1,k)a^k$	$(1-ax)^{-n}$
(10) $\frac{1}{k!}$	e^x
(11) $\frac{(-1)^{k+1}}{k}$	$\ln(1+x)$

41

Counting Problems and Generating Functions (*continued*)

Example: Use generating functions to find the number of k-combinations of a set with n elements, i.e., $C(n,k)$.

Solution: Each of the n elements in the set contributes the term $(1 + x)$ to the generating function

$$f(x) = \sum_{k=0}^{n} a^k x^k.$$

Hence $f(x) = (1 + x)^n$ where $f(x)$ is the generating function for $\{a^k\}$, where a^k represents the number of k-combinations of a set with n elements.

By the binomial theorem, we have

$$f(x) = \sum_{k=0}^{n} \binom{n}{k} x^k,$$

where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Hence,

$$C(n,k) = \frac{n!}{k!(n-k)!}.$$

〖Example 〗Use generating functions to find the number of r-combinations from a

$$(1-x)^{-n} = (1+(-x))^{-n} = \sum_{r=0}^{\infty} \binom{-n}{r} (-x)^r .$$

$$\binom{-n}{r} (-1)^r = (-1)^r C(n+r-1, r) \cdot (-1)^r$$
$$= C(n+r-1, r).$$

Solution:

Since there are n e
times, one times a

$G(x) = (1+$

the number of r-con
repetitions allowed, is the coefficient a_r of x^r in the expansion of $G(x)$. Since

$$\frac{1}{(1-x)^n} = \sum_{k=0}^{\infty} C(n+k-1, k) x^k$$

Then the coefficient a_r equals $C\ (n+r-1,r)$

43

5. Using Generating Functions to Solve Recurrence Relations

The Methods of Solving Recurrence Relations

- **Iterative approach**
- **Use a systematic way to solve an important class of recurrence relations**
- **Generating functions**

Method:

(1) Use the recurrence relation to find the generating function of this sequence;

(2) $G(x) \leftrightarrow a_n$

44

〖Example 〗 Use generating functions to solve the recurrence relation $a_n = 2a_{n-1} + 3a_{n-2} + 4^n + 6$ with initial conditions $a_0 = 20, a_1 = 60$.

Solution:

$$a_n = 2a_{n-1} + 3a_{n-2} + 4^n + 6 \qquad \times x^n$$

$$a_n x^n = 2a_{n-1}x^n + 3a_{n-2}x^n + 4^n x^n + 6x^n$$

$$\sum_{n=2}^{\infty} a_n x^n = 2\sum_{n=2}^{\infty} a_{n-1}x^n + 3\sum_{n=2}^{\infty} a_{n-2}x^n + \sum_{n=2}^{\infty} 4^n x^n + 6\sum_{n=2}^{\infty} x^n$$

$$G(x) - a_0 - a_1 x \qquad 2x\sum_{n=1}^{\infty} a_n x^n \qquad 3x^2\sum_{n=0}^{\infty} a_n x^n \qquad \frac{1}{1-4x} - 1 - 4x \qquad 6\left(\frac{1}{1-x} - 1 - x\right)$$

$$2x(G(x) - a_0) \qquad 3x^2 G(x)$$

45

$$(1-2x-3x^2)G(x) = \frac{20 - 80x + 2x^2 + 40x^3}{(1-4x)(1-x)}$$

$$G(x) = \frac{20 - 80x + 2x^2 + 40x^3}{(1-4x)(1-x)(1+x)(1-3x)}$$

$$= \frac{16/5}{1-4x} + \frac{-3/2}{1-x} + \frac{31/20}{1+x} + \frac{67/4}{1-3x}$$

$$\frac{16}{5} \times 4^n \quad -\frac{2}{3} \times 1^n \quad \frac{31}{20} \times (-1)^n \quad \frac{67}{4} \times 3^n$$

$$a_n = \frac{16}{5} \times 4^n - \frac{2}{3} + \frac{31}{20} \times (-1)^n + \frac{67}{4} \times 3^n$$

46

The method of proving combinatorial identities:

- **Use combinatorial proofs**
- **Use generating functions**

〖Example 〗 Use generating functions to prove Pascal's identity $C(n,r) = C(n-1,r) + C(n-1,r-1)$ when *n* and *r* are positive integers with $r < n$.

Proof:

$$G(x)=(1+x)^n=\sum_{r=0}^{n}C(n,r)x^r$$

$$(1+x)^n=(1+x)(1+x)^{n-1}=(1+x)^{n-1}+x(1+x)^{n-1}$$

$$\sum_{r=0}^{n}C(n,r)x^r=\sum_{r=0}^{n-1}C(n-1,r)x^r+\sum_{r=0}^{n-1}C(n-1,r)x^{r+1}$$

$$1+\sum_{r=1}^{n-1}C(n,r)x^r+x^n$$

$$=\sum_{r=0}^{n-1}C(n-1,r)x^r+\sum_{r=1}^{n}C(n-1,r-1)x^r$$

$$=1+\sum_{r=1}^{n-1}C(n-1,r)x^r+\sum_{r=1}^{n-1}C(n-1,r-1)x^r+x^n$$

$$\sum_{r=1}^{n-1}\underline{C(n,r)}x^r=\sum_{r=1}^{n-1}[\underline{C(n-1,r)+C(n-1,r-1)}]x^r$$

49

Homework

第8版 Sec. 8.4 6(d,f), 10 (c, d, e), 16, 24(a), 32, 36, 45