Chapter 7: Advanced Counting Techniques

§7.1: Recurrence Relations

- A *recurrence relation* (R.R., or just *recurrence*) for a sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more previous elements $a_0, \ldots, a_{n-1}$ of the sequence, for all $n \geq n_0$.
  - A recursive definition, without the base cases.
- A particular sequence (described non-recursively) is said to *solve* the given recurrence relation if it is consistent with the definition of the recurrence.
  - A given recurrence relation may have many solutions.

Recurrence Relation Example

- Consider the recurrence relation

$$a_n = 2a_{n-1} - a_{n-2} \quad (n \geq 2).$$

- Which of the following are solutions?

$$a_n = 3n$$

$$a_n = 2^n$$

$$a_n = 5$$

Example Applications

- Recurrence relation for growth of a bank account with P% interest per given period:

$$M_n = M_{n-1} + (P/100)M_{n-1}$$

- Growth of a population in which each organism yields 1 new one every period starting 2 periods after its birth.

$$P_n = P_{n-1} + P_{n-2} \quad \text{(Fibonacci relation)}$$

Solving Compound Interest RR

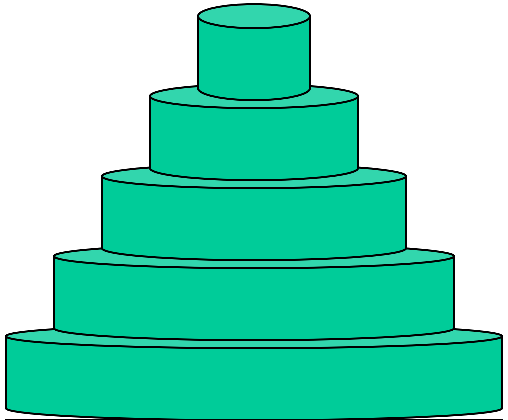
- $M_n = M_{n-1} + (P/100)M_{n-1}$

  $= (1 + P/100)\ M_{n-1}$

  $= r\ M_{n-1}$ (let $r = 1 + P/100$)

Tower of Hanoi Example

- Problem: Get all disks from peg 1 to peg 2.
  - Only move 1 disk at a time.
  - Never set a larger disk on a smaller one.



Peg #1	Peg #2	Peg #3

Hanoi Recurrence Relation

- Let H_n = # moves for a stack of n disks.
- Optimal strategy:
  - Move top $n-1$ disks to spare peg. (H_{n-1} moves)
  - Move bottom disk. (1 move)
  - Move top $n-1$ to bottom disk. (H_{n-1} moves)
- Note: $H_n = 2H_{n-1} + 1$

Solving Tower of Hanoi RR

$H_n = 2\,H_{n-1} + 1$

Finding Recurrence Relation

Ex: Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length 5?

Codeword Enumeration

Ex: Consider a string of decimal digits a valid codeword if it contains an even number of 0 digits. For example, 1230407869 is valid, whereas 120987045608 is not valid. Let a_n be the number of valid n-digit codewords. Find a recurrence relation for a_n .

Catalan Numbers

Ex: Find a recurrence relation for C_n , the number of ways to parenthesize the product of n+1 numbers, $x_0, x_1, \ldots, x_n$, to specify the order of multiplication. For example, $C_3 = 5$.

§7.2: Solving Recurrences

General Solution Schemas

- A *linear homogeneous recurrence of degree k with constant coefficients* ("*k*-LiHoReCoCo") is a recurrence of the form

  $$a_n = c_1 a_{n-1} + \ldots + c_k a_{n-k},$$

  where the c_i are all real, and $c_k \neq 0$.
- The solution is uniquely determined if k initial conditions $a_0 \ldots a_{k-1}$ are provided.

Solving LiHoReCoCos

- Basic idea: Look for solutions of the form $a_n = r^n$, where r is a constant.
- This requires the *characteristic equation*:

$$r^n = c_1 r^{n-1} + \ldots + c_k r^{n-k}, \textit{ i.e.},$$

$$r^k - c_1 r^{k-1} - \ldots - c_k = 0$$

- The solutions (*characteristic roots*) can yield an explicit formula for the sequence.

Solving 2-LiHoReCoCos

- Consider an arbitrary 2-LiHoReCoCo:

  $$a_n = c_1 a_{n-1} + c_2 a_{n-2}$$

- It has the characteristic equation (C.E.):

  $$r^2 - c_1 r - c_2 = 0$$

- **Thm. 1:** If this CE has 2 roots $r_1 \neq r_2$, then

  $$a_n = \alpha_1 r_1^{\,n} + \alpha_2 r_2^{\,n} \quad \text{for } n \geq 0$$

  for some constants α_1, α_2.

Example

- Solve the recurrence $a_n = a_{n-1} + 2a_{n-2}$ given the initial conditions $a_0 = 2$, $a_1 = 7$.
- Solution:

Example Continued…

- To find α_1 and α_2, solve the equations for the initial conditions a_0 and a_1:

  Simplifying, we have the pair of equations:

  which we can solve easily by substitution:

- Final answer:

Check: $\{a_{n\geq 0}\} = 2, 7, 11, 25, 47, 97 \ldots$

Example

- Find an explicit formula for the Fibonacci numbers.

The Case of Degenerate Roots

- Now, what if the C.E. $r^2 - c_1 r - c_2 = 0$ has only 1 root r_0?
- **Theorem 2:** Then,

  $a_n = \alpha_1 r_0^{\,n} + \alpha_2 n r_0^{\,n}$, for all $n \geq 0$,

  for some constants α_1, α_2.
- **Ex:** $\boldsymbol{a_n = 6a_{n-1} - 9a_{n-2},\ a_0 = 1,\ a_1 = 6}$

k-LiHoReCoCos

- Consider a k-LiHoReCoCo: $$a_n = \sum_{i=1}^{k} c_i a_{n-i}$$
- It's C.E. is: $$r^k - \sum_{i=1}^{k} c_i r^{k-i} = 0$$
- **Thm.3:** If this has k distinct roots r_i, then the solutions to the recurrence are of the form:

$$a_n = \sum_{i=1}^{k} \alpha_i r_i^n$$

for all $n \geq 0$, where the α_i are constants.

Example

- **Ex:** $a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$,

  $a_0 = 2, a_1 = 5, a_2 = 15$

Degenerate *k*-LiHoReCoCos

- Suppose there are t roots $r_1,\ldots,r_t$ with multiplicities $m_1,\ldots,m_t$. Then:

$$a_n = \sum_{i=1}^{t} \left(\sum_{j=0}^{m_i - 1} \alpha_{i,j} n^j \right) r_i^n$$

for all $n \geq 0$, where all the α are constants.

Example

- **Ex:** $a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3},$

  $a_0 = 1, a_1 = -2, a_2 = -1$

LiNoReCoCos

- Linear *nonhomogeneous* RRs with constant coefficients may (unlike LiHoReCoCos) contain some terms $F(n)$ that depend *only* on n (and *not* on any a_i's). General form:

$$\underbrace{a_n = c_1 a_{n-1} + \ldots + c_k a_{n-k}} + F(n)$$

The *associated homogeneous recurrence relation* (associated LiHoReCoCo).

Solutions of LiNoReCoCos

- A useful theorem about LiNoReCoCos:
  - If $a_n = p(n)$ is any *particular* solution to the LiNoReCoCo

$$a_n = \left(\sum_{i=1}^{k} c_i a_{n-i} \right) + F(n)$$

  - Then *all* its solutions are of the form:

$$a_n = p(n) + h(n),$$

where $a_n = h(n)$ is any solution to the associated homogeneous RR $a_n = \left(\sum_{i=1}^{k} c_i a_{n-i} \right)$

Example

- Find all solutions to $a_n = 3a_{n-1}+2n$. Which solution has $a_1 = 3$?
  - Notice this is a 1-LiNoReCoCo. Its associated 1-LiHoReCoCo is $a_n = 3a_{n-1}$, whose solutions are all of the form $a_n = \alpha 3^n$. Thus the solutions to the original problem are all of the form $a_n = p(n) + \alpha 3^n$. So, all we need to do is find one $p(n)$ that works.

Trial Solutions

- If the extra terms $F(n)$ are a degree-t polynomial in n, you should try a degree-t polynomial as the particular solution $p(n)$.
- This case: $F(n)$ is linear so try $a_n = cn + d$.

(for all n)

(collect terms)

So

So is a solution.

- Check: $a_{n\geq 1} = \{-5/2, -7/2, -9/2, \ldots\}$

Finding a Desired Solution

- From the previous, we know that all general solutions to our example are of the form:

  Solve this for α for the given case, $a_1 = 3$:

- The answer is

Example

Ex: $a_n = 5a_{n-1} - 6a_{n-2} + 7^n$

§7.3: Divide & Conquer R.R.s

Main points so far:

- Many types of problems are solvable by reducing a problem of size n into some number a of independent subproblems, each of size $\leq \lceil n/b \rceil$, where $a \geq 1$ and $b > 1$.
- The time complexity to solve such problems is given by a recurrence relation:
  - $T(n) = a \cdot T(\lceil n/b \rceil) + g(n)$

Divide+Conquer Examples

- **Binary search:** Break list into 1 sub-problem (smaller list) (so a=1) of size $\leq\lceil n/2 \rceil$ (so b=2).
  - So $T(n) = T(\lceil n/2 \rceil)+c$ ($g(n)=c$ constant)
- **Merge sort:** Break list of length n into 2 sublists (a=2), each of size $\leq\lceil n/2 \rceil$ (so b=2), then merge them, in $g(n) = \Theta(n)$ time.
  - So $T(n) = 2T(\lceil n/2 \rceil) + cn$ (roughly, for some c)

Divide+Conquer Examples

- **Finding the Maximum and Minimum:** Break list into 2 sub-problem (smaller list) (so a=2) of size $\leq\lceil n/2 \rceil$ (so b=2).
  - So $T(n) = 2T(\lceil n/2 \rceil)+2$ ($g(n)$=2 constant)

Fast Multiplication Example

- The ordinary grade-school algorithm takes $\Theta(n^2)$ steps to multiply two n-digit numbers.
  - This seems like too much work!
- So, let's find an asymptotically *faster* multiplication algorithm!
- To find the product cd of two $2n$-digit base-b numbers, $c=(c_{2n-1}c_{2n-2}\ldots c_0)_b$ and $d=(d_{2n-1}d_{2n-2}\ldots d_0)_b$,

First, we break c and d in half:

$c=b^nC_1+C_0$, $d=b^nD_1+D_0$, and then... (see next slide)

Derivation of Fast Multiplication

$$cd = (b^n C_1 + C_0)(b^n D_1 + D_0)$$

$$= b^{2n} C_1 D_1 + b^n (C_1 D_0 + C_0 D_1) + C_0 D_0$$ (Multiply out polynomials)

$$= b^{2n} C_1 D_1 + C_0 D_0 +$$

$$b^n (C_1 D_0 + C_0 D_1 + (C_1 D_1 - C_1 D_1) + (C_0 D_0 - C_0 D_0))$$

Zero

$$= (b^{2n} + b^n) C_1 D_1 + (b^n + 1) C_0 D_0 +$$

$$b^n (C_1 D_0 - C_1 D_1 - C_0 D_0 + C_0 D_1)$$

$$= (b^{2n} + b^n) C_1 D_1 + (b^n + 1) C_0 D_0 +$$

$$b^n (C_1 - C_0)(D_0 - D_1)$$ (Factor last polynomial)

Recurrence Rel. for Fast Mult.

Notice that the time complexity $T(n)$ of the fast multiplication algorithm obeys the recurrence:

- $T(2n)=3T(n)+\Theta(n)$

  i.e.,

- $T(n)=3T(n/2)+\Theta(n)$

  So $a=3$, $b=2$.

Time to do the needed adds & subtracts of n-digit and $2n$-digit numbers

The Master Theorem

Consider a function $f(n)$ that, for all $n=b^k$ for all $k\in\mathbf{Z}^+$, satisfies the recurrence relation:

$$f(n) = a\,f(n/b) + cn^d$$

with $a\geq 1$, integer $b>1$, real $c>0$, $d\geq 0$. Then:

$$f(n) \in \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log_b n) & \text{if } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

Examples

Consider a function $f(n)$ that, for all $n=2^k$ for all $k\in\mathbf{Z}^+$, satisfies the recurrence relation:

$f(n) = 5f(n/2) + 3$. Then: $f(n) =$

Complexity of Merge Sort:

$$M(n) = 2M(n/2) + n$$
$$\therefore M(n) =$$

Example

- Recall that complexity of fast multiply was:

  $T(n)=3T(n/2)+\Theta(n)$

- Thus, a=3, b=2, d=1. So $a > b^d$, so case 3 of the master theorem applies, so:

  $T(n) =$

  which is $\Theta(n^{1.58...})$, so the new algorithm is strictly faster than ordinary $\Theta(n^2)$ multiply!

Example

- **The Closest-Pair Problem:** a set of n points, $(x_1, y_1), \cdots, (x_n, y_n)$

$$d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

- How can this closest pair of points be found in an efficient way?

$$T(n)=2T(n/2)+7n$$

$$T(n) =$$

§7.4: Generating Functions

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

$$G(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k + \cdots = \sum_{k=0}^{\infty} a_k x^k$$

Examples

- What is the generating function of the sequence 1,1,1,1,1,1?

$$G(x) =$$

- What is the generating function of the sequence $\{a_k\},\ a_k = C(m,k)$?

$$G(x) =$$

Examples

- The function $f(x)=1/(1-x)$ is the generating function of the sequence 1,1,1,…for $|x|<1$.
- The function $f(x)=1/(1-ax)$ is the generating function of the sequence $1,a,a^2,\ldots$ for $|ax|<1$.

Theorem

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

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

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

Convolution of a_k and b_k

Example

What sequence has the generating function

$f(x)=1/(1-x)^2$?

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

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

Extended Binomial Coefficient

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

Note: u positive integer, $\binom{u}{k} = 0$ **if** $k > u$

Examples:

$$\binom{-2}{3} =$$

$$\binom{1/2}{3} =$$

Extended Binomial Theorem

$$(1+x)^u = \sum_{k=0}^{\infty} \binom{u}{k} x^k, \quad \text{where } |x| < 1$$

Can be proved using Maclaurin series.

Examples:

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

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

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 $2 \le e_1 \le 5,\ 3 \le e_2 \le 6,\ 4 \le e_3 \le 7.$

Sol: Find the coefficient of x^{17} ,

The answer is

Example

Solve the recurrence relation:

$$a_k = 3a_{k-1} \quad \text{for } k = 1, 2, 3, \cdots \text{ and } a_0 = 2.$$

Sol: Let $G(x)$ be the generating function of $\{a_k\}$, $G(x) = \sum_{k=0}^{\infty} a_k x^k$

Example(Cont'd)

$$G(x) =$$

$$\therefore a_k =$$