UNIT-III: Advanced Counting Techniques

Syllabus

Recurrence Relations, Solving Linear Recurrence Relations, Divide-and-Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion-Exclusion, Applications of Inclusion- Exclusion.

Self-study: Euclidean algorithm and prime factorization.

Faculty – Dr. Sridhar V

1 Recurrence Relation

A recurrence relation is an equation that defines a sequence based on a rule that gives the next term as a function of the previous term(s).

The simplest form of a recurrence relation is the case where the next term depends only on the immediately previous term. If we denote the n^{th} term in the sequence by x_n, such a recurrence relation is of the form $x_{n+1} = f(x_n)$ for some function f. One such example is $x_{n+1} = 2 - x_n/2$.

A recurrence relation can also be of higher order, where the term x_{n+1} could depend not only on the previous term x_n but also on earlier terms such as x_{n-1}, x_{n-2} etc. A second order recurrence relation depends just on x_n and x_{n-1} and is of the form $x_{n+1} = f(x_n, x_{n-1})$ for some function f with two inputs. For example, the recurrence relation $x_{n+1} = x_n + x_{n-1}$ can generate the Fibonacci numbers.

To generate sequence based on a recurrence relation, one must start with some initial values. For a first order recursion $x_{n+1} = f(x_n)$, one just needs to start with an initial value x_0 and can generate all remaining terms using the recurrence relation. For a second order recursion $x_{n+1} = f(x_n, x_{n-1})$, one needs to begin with two values x_0 and x_1. Higher order recurrence relations require correspondingly more initial values.

Note: A recurrence relation can be viewed as determining a discrete dynamical system.

2 Solving Linear Recurrence Relations

A wide variety of recurrence relations occur in models. Some of these recurrence relations can be solved using iteration or some other ad hoc technique. However, one important class of recurrence relations can be explicitly solved in a systematic way. These are recurrence relations that express the terms of a sequence as linear combinations of previous terms.

2.1 Linear Homogeneous Recurrence Relation with Constant Coefficients

A linear homogeneous recurrence relation of degree k with constant coefficients is a recurrence relation of the form $a_n = c_1a_{n-1} + c_2a_{n-2} + \ + c_ka_{n-k}$, where $c_1, c_2, ..., c_k$ are real numbers, and $c_k \neq 0$.

The recurrence relation in the definition is **linear** because the right-hand side is a sum of previous terms of the sequence each multiplied by a function of n. The recurrence relation is **homogeneous** because no terms occur that are not multiples of the a_js. The coefficients of the terms of the sequence are all **constants**, rather than functions that depend on n. The **degree** is k because an is expressed in terms of the previous k terms of the sequence.

A consequence of the second principle of mathematical induction is that a sequence satisfying the recurrence relation in the definition is uniquely determined by this recurrence relation and the k initial conditions $a_0 = C_0, a_1 = C_1, ..., a_{k-1} = C_{k-1}$.

Example: The recurrence relation $f_n = f_{n-1} + f_{n-2}$ is a linear homogeneous recurrence relation of degree two. The recurrence relation $a_n = a_{n-5}$ is a linear homogeneous recurrence relation of degree five.

Example: The recurrence relation $a_n = a_{n-1} + a_{n-2}^2$ is not linear. The recurrence relation $H_n = 2H_{n-1} + 1$ is not homogeneous. The recurrence relation $B_n = nB_{n-1}$ does not have constant coefficients.

Note: Linear homogeneous recurrence relations are studied for two reasons. First, they often occur in modeling of problems. Second, they can be systematically solved.

Example: What is the solution of the recurrence relation $a_n = a_{n-1} + 2a_{n-2}$ with $a_0 = 2$ and $a_1 = 7$?

Solution: The characteristic equation of the recurrence relation is $r^2 - r - 2 = 0$. Its roots are $r = 2$ and $r = -1$. Hence, the sequence a_n is a solution to the recurrence relation if and only if $a_n = \alpha_1 2^n + \alpha_2(-1)^n$, for some constants α_1 and α_2. From the initial conditions, it follows that $a_0 = 2 = \alpha_1 + \alpha_2$, $a_1 = 7 = \alpha_1 \cdot 2 + \alpha_2 \cdot (-1)$.

Solving these two equations shows that $\alpha_1 = 3$ and $\alpha_2 = -1$. Hence, the solution to the recurrence relation and initial conditions is the sequence a_n with $a_n = 3 \cdot 2^n - (-1)^n$.

Example: Find an explicit formula for the Fibonacci numbers.

Solution: Recall that the sequence of Fibonacci numbers satisfies the recurrence relation $f_n = f_{n-1} + f_{n-2}$ and also satisfies the initial conditions $f_0 = 0$ and $f_1 = 1$.

The roots of the characteristic equation $r^2 - r - 1 = 0$ are $r1 = (1 + \sqrt{5})/2$ and $r_2 = (1 - \sqrt{5})/2$.

Therefore, the Fibonacci numbers are given by

$$f_n = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)^n$$

for some constants α_1 and α_2. The initial conditions $f_0 = 0$ and $f_1 = 1$ can be used to find these constants. We have

$$f_0 = \alpha_1 + \alpha_2 = 0,$$

$$f_1 = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right) + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right) = 1.$$

The solution to these simultaneous equations for α_1 and α_2 is

$$\alpha_1 = \tfrac{1}{\sqrt{5}}\ ,\ \alpha_2 = -\tfrac{1}{\sqrt{5}}\ .$$

Consequently, the Fibonacci numbers are given by

$$f_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2}\right)^n.$$

Note: If there is one characteristic root of multiplicity two, then $a_n = nr_0^n$ is another solution of the recurrence relation when r_0 is a root of multiplicity two of the characteristic equation.

Example: What is the solution of the recurrence relation $a_n = 6a_{n-1} - 9a_{n-2}$ with initial conditions $a_0 = 1$ and $a_1 = 6$?

Solution: Do it yourself. Answer - $a_n = 3^n + n3^n$.

Example: Find the solution to the recurrence relation $a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$ with initial conditions $a_0 = 1$, $a_1 = -2$, and $a_2 = -1$.

Solution: Do it yourself. Answer - $a_n = (1 + 3n - 2n^2)(-1)^n$.

2.2 Linear Nonhomogeneous Recurrence Relation with Constant Coefficients

A recurrence relation of the form $a_n = c_1 a_{n-1} + c_2 a_{n-2} + + c_k a_{n-k} + F(n)$, where $c_1, c_2, ..., c_k$ are real numbers and $F(n)$ is a function not identically zero depending only on n, is called a linear nonhomogeneous recurrence relation with constant coefficients. The recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \; + c_k a_{n-k}$, is called the associated homogeneous recurrence relation. It plays an important role in the solution of the nonhomogeneous recurrence relation.

Example: The recurrence relation $a_n = 3a_{n-1} + 2n$ is an example of a linear nonhomogeneous recurrence relation with constant coefficients.

Example: Each of the recurrence relations $a_n = a_{n-1} + 2^n$, $a_n = a_{n-1} + a_{n-2} + n^2 + n + 1$, $a_n = 3a_{n-1} + n3^n$, and $a_n = a_{n-1} + a_{n-2} + a_{n-3} + n!$ is a linear nonhomogeneous recurrence relation with constant coefficients. The associated linear homogeneous recurrence relations are $a_n = a_{n-1}$, $a_n = a_{n-1} + a_{n-2}$, $a_n = 3a_{n-1}$, and $a_n = a_{n-1} + a_{n-2} + a_{n-3}$, respectively.

Note: The key fact about linear nonhomogeneous recurrence relations with constant coefficients is that every solution is the sum of a particular solution and a solution of the associated linear homogeneous recurrence relation.

Example: Find all solutions of the recurrence relation $a_n = 3a_{n-1} + 2n$. What is the solution with $a_1 = 3$?

Solution: The associated linear homogeneous equation is $a_n = 3a_{n-1}$. Its solutions are $a_n^{(h)} = \alpha\, 3^n$, where α is a constant.

We now find a particular solution. Because $F(n) = 2n$ is a polynomial in n of degree one, a reasonable trial solution is a linear function in n, say, $p_n = cn + d$, where c and d are constants.

To determine whether there are any solutions of this form, suppose that $p_n = cn + d$ is such a solution. Then the equation $a_n = 3a_{n-1} + 2n$ becomes $cn + d = 3(c(n-1) + d) + 2n$.

Simplifying and combining like terms gives $(2 + 2c)n + (2d - 3c) = 0$. It follows that $cn + d$ is a solution if and only if $2 + 2c = 0$ and $2d - 3c = 0$.

This shows that $cn + d$ is a solution if and only if $c = -1$ and $d = -3/2$. Consequently, $a_n^{(p)} = -n - 3/2$ is a particular solution.

Therefore, the final solutions can be written as $a_n = a_n^{(p)} + a_n^{(h)} = -n - \dfrac{3}{2} + \alpha \cdot 3^n$, where α is a constant.

To find the solution with $a_1 = 3$, let $n = 1$ in the formula we obtained for the general solution. We find that $3 = -1 - \dfrac{3}{2} + 3\alpha$, which implies that $\alpha = 11/6$. So, the solution we seek is $a_n = -n - \dfrac{3}{2} + \dfrac{11}{6} 3^n$.

Example: Find all solutions of the recurrence relation $a_n = 5a_{n-1} - 6a_{n-2} + 7^n$.

Solution: Do it yourself. Answer - $a_n = \alpha_1 \cdot 3^n + \alpha_2 \cdot 2^n + \frac{49}{20}7^n$.

3 Divide-and-Conquer Algorithms and Recurrence Relations

Many recursive algorithms take a problem with a given input and divide it into one or more smaller problems. This reduction is successively applied until the solutions of the smaller problems can be found quickly. These procedures follow an important algorithmic paradigm known as divide-and-conquer, and are called divide-and-conquer algorithms, because they divide a problem into one or more instances of the same problem of smaller size and they conquer the problem by using the solutions of the smaller problems to find a solution of the original problem, perhaps with some additional work.

Divide-and-Conquer Recurrence Relations: Suppose that a recursive algorithm divides a problem of size n into subproblems, where each subproblem is of size n/b (for simplicity, assume that n is a multiple of b; in reality, the smaller problems are often of size equal to the nearest integers either less than or equal to, or greater than or equal to, n/b). Also, suppose that a total of $g(n)$ extra operations are required in the conquer step of the algorithm to combine the solutions of the subproblems into a solution of the original problem. Then, if $f(n)$ represents the number of operations required to solve the problem of size n, it follows that f satisfies the recurrence relation $f(n) = af(n/b) + g(n)$.

This is called a divide-and-conquer recurrence relation.

Example - Binary Search The binary search algorithm reduces the search for an element in a search sequence of size n to the binary search for this element in a search sequence of size $n/2$, when n is even. (Hence, the problem of size n has been reduced to one problem of size $n/2$.) Two comparisons are needed to implement this reduction (one to determine which half of the list to use and the other to determine whether any terms of the list remain). Hence, if $f(n)$ is the number of comparisons required to search for an element in a search sequence of size n, then $f(n) = f(n/2) + 2$ when n is even.

Example - Finding the Maximum and Minimum of a Sequence Consider the following algorithm for locating the maximum and minimum elements of a sequence $a_1, a_2, ..., a_n$. If $n = 1$, then a_1 is the maximum and the minimum. If $n > 1$, split the sequence into two sequences, either where both have the same number of elements or where one of the sequences has one more element than the other. The problem is reduced to finding the maximum and minimum of each of the two smaller sequences. The solution to the original problem results from the comparison of the separate maxima and minima of the two smaller sequences to obtain the overall maximum and minimum. Let $f(n)$ be the total number of comparisons needed to find the maximum and minimum elements of the sequence with n elements. We have shown that a problem of size n can be reduced into two problems of size $n/2$, when n is even, using two comparisons, one to compare the maxima of the two sequences and the other to compare the minima of the two sequences. This gives the recurrence relation $f(n) = 2f(n/2) + 2$ when n is even.

Example - Merge Sort The merge sort algorithm splits a list to be sorted with n items, where n is even, into two lists with $n/2$ elements each, and uses fewer than n comparisons to merge the two sorted lists of $n/2$ items each into one sorted list. Consequently, the number of comparisons used by the merge sort to sort a list of n elements is less than $M(n)$, where the function $M(n)$ satisfies the divide-and-conquer recurrence relation $M(n) = 2M(n/2) + n$.

4 Generating Function

Generating functions are used to represent sequences efficiently by coding the terms of a sequence as coefficients of powers of a variable x in a formal power series. Generating functions can be used to solve many types of counting problems, such as the number of ways to select or distribute objects of different kinds, subject to a variety of constraints, and the number of ways to make change for a dollar using coins of different denominations. Generating functions can be used to solve recurrence relations by translating a recurrence relation for the terms of a sequence into an equation involving a generating function. This equation can then be solved to find a closed form for the generating function. From this closed form, the coefficients of the power series for the generating function can be found, solving the original recurrence relation. Generating functions can also be used to prove combinatorial identities by taking advantage of relatively simple relationships between functions that can be translated into identities involving the terms of sequences. Generating functions are a helpful tool for studying many properties of sequences besides those described in this section, such as their use for establishing asymptotic formulae for the terms of a sequence.

Definition: The generating function for the sequence $a_0, a_1, ..., a_k, ...$ of real numbers is the infinite series $G(x) = a_0 + a_1x + \ + a_kx^k + \cdots = \sum_{k=0}^{\infty} a_kx^k$.

Remark: The generating function for a_k given in above definition is sometimes called the ordinary generating function of a_k to distinguish it from other types of generating functions for this sequence.

Example: The generating functions for the sequences a_k with $a_k = 3$, $a_k = k + 1$, and $a_k = 2^k$ are $\sum_{k=0}^{\infty} 3x^k$, $\sum_{k=0}^{\infty}(k+1)x^k$, $\sum_{k=0}^{\infty} 2^kx^k$, respectively.

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 $1 + x + x^2 + x^3 + x^4 + x^5$.
We know that $\dfrac{(x^6-1)}{(x-1)} = 1 + x + x^2 + x^3 + x^4 + x^5$ when $x \neq 1$. Consequently, $G(x) = \dfrac{(x^6-1)}{(x-1)}$ is the generating function of the sequence $1, 1, 1, 1, 1, 1$. [Because the powers of x are only place holders for the terms of the sequence in a generating function, we do not need to worry that $G(1)$ is undefined.]

Example: The function $f(x) = \dfrac{1}{(1-ax)}$ is the generating function of the sequence $1, a, a^2, a^3, ...$ because $\dfrac{1}{(1-ax)} = 1 + ax + a^2x^2 + \cdots$ when $|ax| < 1$, or equivalently, for $|x| < 1/|a|$ for $a \neq 0$.

4.1 Counting Problems and Generating Functions

We know a few techniques to count the r-combinations from a set with n elements when repetition is allowed and additional constraints may exist. Such problems are equivalent to counting the solutions to equations of the form $e_1 + e_2 + \cdots + e_n = C$, where C is a constant and each e_i is a nonnegative integer that may be subject to a specified constraint.

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 with the indicated constraints 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 we obtain a term equal to x^{17} 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 the exponents e_1, e_2, and e_3 satisfy the equation $e_1 + e_2 + e_3 = 17$ and the given constraints.

It is not hard to see that the coefficient of x^{17} in this product is 3. Hence, there are three solutions.

4.2 Solution of Recurrence Relations Using Generating Functions

We can find the solution to a recurrence relation and its initial conditions by finding an explicit formula for the associated generating function.

Example: Solve the recurrence relation $a_k = 3a_{k-1}$ for $k = 1, 2, 3, ...$ and initial condition $a_0 = 2$.

Solution: Let $G(x)$ is generating function for the sequence a_k, *i.e.* $G(x) = \sum_{k=0}^{\infty} a_k x^k$.

First note that

$xG(x) = \sum_{k=0}^{\infty} a_k x^{k+1} = \sum_{k=1}^{\infty} a_{k-1} x^k$.

Using the recurrence relation, we see that

$G(x) - 3xG(x) = \sum_{k=0}^{\infty} a_k x^k - 3\sum_{k=1}^{\infty} a_{k-1} x^k$

$G(x) - 3xG(x) = a_0 + \sum_{k=1}^{\infty}(a_k - 3\,a_{k-1})x^k = 2,$

because $a_0 = 2$ and $a_k = 3a_{k-1}$.

Thus, $G(x) - 3xG(x) = (1 - 3x)G(x) = 2$.

Solving for $G(x)$ shows that $G(x) = 2/(1 - 3x)$.

We know that the identity $1/(1 - ax) = \sum_{k=0}^{\infty} a^k x^k$. Therefore, we can write

$G(x) = 2\sum_{k=0}^{\infty} 3^k x^k = \sum_{k=0}^{\infty} 2 \cdot 3^k x^k$.

Consequently, $a_k = 2 \cdot 3^k$.

4.3 Proving Identities via Generating Functions

Combinatorial identities can be established using combinatorial proofs and generating functions.

Example: Use generating functions to show that $\sum_{k=0}^{n} C(n,k)^2 = C(2n,n)$ whenever n is a positive integer.

Solution: First note that by the binomial theorem $C(2n,n)$ is the coefficient of x^n in $(1+x)^{2n}$.

However, we also have

$$(1+x)^{2n} = [(1+x)^n]^2 = [C(n,0) + C(n,1)x + C(n,2)x^2 + ... + C(n,n)x^n]^2.$$

The coefficient of x^n in this expression is

$$C(n,0)C(n,n) + C(n,1)C(n,n-1) + C(n,2)C(n,n-2) + \ + C(n,n)C(n,0).$$

This equals $\sum_{k=0}^{n} C(n,k)^2$, since $C(n,n-k) = C(n,k)$. It is because both $C(2n,n)$ and $\sum_{k=0}^{n} C(n,k)^2$ represent the coefficient of x^n in $(1+x)^{2n}$, they must be equal.

5 Inclusion-Exclusion

5.1 The Principle of Inclusion-Exclusion

The number of elements in the union of the two sets A and B is the sum of the numbers of elements in the sets minus the number of elements in their intersection.

Mathematically, $|A \cup B| = |A| + |B| - |A \cap B|$.

Example: Suppose that there are 1807 freshmen at your school. Of these, 453 are taking a course in computer science, 567 are taking a course in mathematics, and 299 are taking courses in both computer science and mathematics. How many are not taking a course either in computer science or in mathematics?

Solution: To find the number of freshmen who are not taking a course in either mathematics or computer science, subtract the number that are taking a course in either of these subjects from the total number of freshmen. Let A be the set of all freshmen taking a course in computer science, and let B be the set of all freshmen taking a course in mathematics. It follows that $|A| = 453$, $|B| = 567$, and $|A \cap B| = 299$. The number of freshmen taking a course in either computer science or mathematics is

$$|A \cup B| = |A| + |B| - |A \cap B| = 453 + 567 - 299 = 721.$$

Consequently, there are 1807 - 721 = 1086 freshmen who are not taking a course in computer science or mathematics.

Example: A total of 1232 students have taken a course in Spanish, 879 have taken a course in French, and 114 have taken a course in Russian. Further, 103 have taken courses in both Spanish and French, 23 have taken courses in both Spanish and Russian, and 14 have taken courses in both French and Russian. If 2092 students have taken at least one of Spanish, French, and Russian, how many students have taken a course in all three languages?

Solution: Do it yourself. Answer - 7.

Remark: **General Form of the Principle of Inclusion-Exclusion** - Let $A_1, A_2, ..., A_n$ be finite sets. Then

$$|A_1 \cup A_2 \cup ... \cup A_n| = \sum_{1 \leq i \leq n} |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| + \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| - ... + (-1)^{n+1} |A_1 \cap A_2 \cap ... \cap A_n|.$$

5.2 Applications of Inclusion-Exclusion

Many counting problems can be solved using the principle of inclusion-exclusion. For instance, we can use this principle to find the number of primes less than a positive integer. Many problems can be solved by counting the number of onto functions from one finite set to another. The inclusion-exclusion principle can be used to find the number of such functions.

5.2.1 The Number of Onto Functions

The principle of inclusionexclusion can also be used to determine the number of onto functions from a set with m elements to a set with n elements.

Result: Let m and n be positive integers with $m \geq n$. Then, there are

$$n^m - C(n, 1)(n-1)^m + C(n, 2)(n-2)^m - ... + (-1)^{n-1} C(n, n-1) \cdot 1^m$$

onto functions from a set with m elements to a set with n elements.

Example: How many onto functions are there from a set with six elements to a set with three elements?

Solution: The required number is $3^6 - C(3, 1)2^6 + C(3, 2)1^6 = 729 - 192 + 3 = 540$.

5.2.2 Derangements

A derangement is a permutation of objects that leaves no object in its original position. The principle of inclusion-exclusion can be used to count the permutations of n objects that leave no objects in their original positions.

Example: The permutation 21453 is a derangement of 12345 because no number is left in its original position. However, 21543 is not a derangement of 12345, because this permutation leaves 4 fixed.

Result: The number of derangements of a set with n elements is

$$D_n = n! \left[1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + ... + (-1)^n \frac{1}{n!} \right].$$