1 Recurrence Relations

Suppose $a_0, a_1, a_2, \ldots$ is a sequence. A *recurrence relation* for the n-th term a_n is a formula (*i.e.*, function) giving a_n in terms of some or all previous terms (*i.e.*, $a_0, a_1, \ldots, a_{n-1}$). To completely describe the sequence, the first few values are needed, where "few" depends on the recurrence. These are called the *initial conditions*.

If you are given a recurrence relation and initial conditions, then you can write down as many terms of the sequence as you please: just keep applying the recurrence. For example, $f_0 = f_1 = 1,\ f_n = f_{n-1} + f_{n-2}, n \geq 2$, defines the *Fibonacci Sequence* $1, 1, 2, 3, 5, 8, 13, \ldots$ where each subsequent term is the sum of the preceding two terms. On the other hand, if you are given a sequence, you may or may not be able to determine a recurrence relation with inital conditions which describes it. For example, $0, 1, 2, 3, 0, 2, 4, 6, 0, 4, 8, 12, \ldots$ satisfies $a_n = 2a_{n-4},\ a_0 = 0, a_1 = 1, a_2 = 2, a_3 = 3$, but I can't think of a recurrence relation and initial conditions that describes the sequence $\{p_n\}$ of prime numbers.

Deriving recurrence relations involves different methods and skills than solving them. These two topics are treated separately in the next 2 subsections. Another method of solving recurrences involves *generating functions*, which will be discussed later.

1.1 Deriving Recurrence Relations

It is typical to want to derive a recurrence relation with initial conditions (abbreviated to RRwIC from now on) for the number of objects satisfying certain conditions. The main technique involves giving counting argument that gives the number of objects of "size" n in terms of the number of objects of smaller size. This typically involves an analysis of several cases.

Suggestion. When attempting to derive a RRwIC, start by working out the first few cases directly. You'll need these for the initial conditions anyway, and doing this might help you see how to proceed. If you do enough cases, then you can use them later to check your recurrence.

Example 1 Fibonacci numbers. *Assume you start with one pair of newborn rabbits (one of each gender), and in each subsequent month each pair* *of rabbits which are more than 1 month old gives birth to a new pair of rabbits, one of each gender. Determine a RRwIC for f_n, the number of pairs of rabbits present at the end of n months.*

The statement tells us that $f_0 = 1$. Also, $f_1 = 1$ because the original pair of rabbits is not yet old enough to breed. At the end of two months, we have our pair from before, plus one new pair. At the end of 3 months, we have the f_2 pairs from before, and f_1 of them are old enough to breed, so we have $f_3 = f_2 + f_1 = 3$ pairs. Consider what happens at the end of n months. We still have the f_{n-1} pairs from the month before. The number of pairs old enough to breed is the number alive two months ago, or f_{n-2}, so we get f_{n-2} new pairs. Thus, $f_n = f_{n-1} + f_{n-2},\ n \geq 2$, and $f_0 = f_1 = 1$. Using the RRwIC yields the sequence 1, 1, 2, 3, 5, 8, ... which agrees with our initial counting.

Counting sequences. Suppose you want to derive a recurrence for a_n, the number of sequences with a certain property. A strategy that is often successful is breaking the counting argument into cases based on the first (or last) entry in the sequence.

Example 2 *Derive a RR with IC for u_n, the number of sequences (strings) of upper case letters that do not contain ZZ.*

By counting, $u_0 = 1, u_1 = 26, u_2 = 26^2 - 1 = 675, u_3 = 25(26^2 - 1) + 1 \cdot 25 \cdot 26 = 17525$. The method used to derive u_3 suggests breaking the counting into cases depending on the first letter in the sequence. Let's call a string *valid* if it does not contain ZZ. Consider a valid string of length n. There are two cases depending on whether the first letter is Z.

Case 1. *The first letter is not Z (25 choices).*
Then, the remaining $n-1$ letters can be any valid string of length $n-1$. Since there u_{n-1} of these, by the Rule of Product there are $25u_{n-1}$ valid string in which the first letter is not Z.

Case 2. *The first letter is Z (1 choice).*
Since the string is valid, the second letter is not Z (25 choices), and then the remaining $n-2$ letters can be any valid string of length $n-2$. Since there u_{n-2} of these, by the Rule of Product there are $25u_{n-2}$ valid string in which the first letter is Z.

Therefore, by the Rule of Sum, $u_n = 25u_{n-1} + 25u_{n-2},\ n \geq 2$. As a check on the work, computing with the recurrence, and $u_0 = 1$ and $u_1 = 25$ gives

$u_2 = 25 \cdot 26 + 25 \cdot 1 = 675$ and $u_3 = 25 \cdot 675 + 25 \cdot 26 = 17525$. Since these agree with what was obtained in step 1, there is some evidence that the RR with IC is correct.

Similar methods often work when you are considering the number of ways to accomplish a sequence of steps, where at each step one of a few things happens. For example, a RRwIC for the number of ways to climb n stairs where on each stride you climb 1, 2, or 3 stairs is $a_0 = a_1 = 1,\ a_2 = 2$ and for $n \geq 3, \quad a_n = a_{n-1} + a_{n-2} + a_{n-3}$.

Sometimes one or more of the cases do not involve the number of objects of a smaller size.

Example 3 *Derive a RRwIC for b_n, the number of bit strings of length n that contain* 00.

By counting, $b_0 = b_1 = 0,\ b_2 = 1$, and $b_3 = 3$. Call a bit string *good* if it contains 00. Consider a good bit string of length n. There are two cases to consider, depending on the first *bit* (element of the sequence).

Case 1. *The first element is 1 (1 choice).*
Then, the remaining $n-1$ bits can be any good bit string of length $n-1$, so there are b_{n-1} of these.

Case 2. *The first element is 0 (1 choice).*
If the next bit is 1, then the remaining $n-2$ bits can be any good bit string of length $n-2$, so there are b_{n-2} good bit strings that begin 01. On the other hand, if the second bit is 0, then we have 00 already, so the remaining $n-2$ bits can be *any* bit string of length $n-2$, and there are 2^{n-2} of these.

Thus, by the Rule of Sum, $b_n = b_{n-1} + b_{n-2} + 2^{n-2}, \quad n \geq 2$. Plugging in the initial conditions and computing yields $b_2 = 1$ and $b_3 = 3$, which agrees with what was obtained directly.

When the recurrence relation is for the number of sequences that do, or don't, contain a subsequence where all characters are the same, the methods above usually suffice. When the characters in the forbidden subsequence of interest are not all the same, the counting can be more complicated.

Example 4 *Derive a RRwIC for d_n, the number of sequences of the 26 upper case letters that do not contain DOG.*

By counting, $d_0 = 1$, $d_1 = 26$, $d_2 = 26^2$, and $d_3 = 26^3 - 1$. Call a sequence *dogless* if it does not contain DOG. There are two cases to consider, depending on the first letter.

Case 1. *The first letter is not D (25 choices).*
Then, the remaining $n-1$ letters can be any dogless sequence of length $n-1$, and there d_{n-1} of these. Thus, there are $25d_{n-1}$ dogless sequences in which the first character is not D.

Case 2. *The first letter is D (1 choice).*
The remaining $n-1$ letters must be a dogless sequence of length $n-1$ (d_{n-1} choices), but care needs to be taken because DOG is formed when this sequence starts with OG. We don't want to count these, and so we subtract the number of sequnces that contain DOG as the first three letters (i.e. formed in this way – d_{n-3} choices). Hence, the number of dogless sequences in this case is $d_{n-1} - d_{n-3}$.

By the Rule of Sum, $d_n = 25d_{n-1} + (d_{n-1} - d_{n-3}) = 26d_{n-1} - d_{n-3}$.

Another method involves remembering that if a_n counts the number of objects you want, then the number that you don't want equals the total number of object minus a_n.

Example 5 *Derive a RRwIC for z_n, the number of sequences of A's, B's and C's that contain an odd number of C's.*

By counting, $z_0 = 0, z_1 = 1, z_2 = 4$ and $z_3 = 13$. Let's call a sequence *odd* if it contains an odd number of C's, and *even* otherwise. Consider an odd sequence of length n. There are two cases to consider, depending on the first character in the sequence.

Case 1. *The first character is A or B (2 choices).*
Then, the remaining $n-1$ characters form an odd sequence of length $n-1$, so there are $2z_{n-1}$ of these.

Case 2. *The first character is C (1 choice).*
Then the remaining $n-1$ characters form an even sequence of length $n-1$, and the number of these is $3^{n-1} - z_{n-1}$.

Thus, by the Rule of Sum, $z_n = 2z_{n-1} + 3^{n-1} - z_{n-1} = z_{n-1} + 3^{n-1}$, $n \geq 1$. Plugging in the initial conditions and computing yields $z_1 = 1, z_2 = 4$, and $z_3 = 13$, which agrees with our inital counting.

Counting subsets. If you are trying to derive a RRwIC that involves subsets of $\{x_1, x_2, \ldots, x_n\}$ with certain properties, try breaking into cases depending on what happens to x_1.

Example 6 *Derive a RR with IC for p_n, the number of ways to partition an n-set into subsets. (Remember that the subsets in a partition are non-empty and pairwise disjoint.)*

By counting, $p_1 = 1,\ p_2 = 2,\ p_3 = 7$. Let's agree that $p_0 = 1$; we'll see why this is helpful in a moment. Consider a partition of the n-set $\{x_1, x_2, \ldots, x_n\}$. Then x_1 belongs to one of the subsets in the partition, call it S and let $k = |S|$. Then $1 \leq k \leq n$. Deleting S leaves a partition of the $(n-k)$-set formed by the remaining elements. Since there are $\binom{n-1}{k-1}$ choices for a k-subset containing x_1, and for each of these there are p_{n-k} partitions of the remaining elements, by the Rule of Product the number of partitions of $\{x_1, x_2, \ldots, x_n\}$ in which x_1 belongs to a subset of size k is $\binom{n-1}{k-1}p_{n-k}$. (Note: it is this line that would lead to wanting to define $p_0 = p_{n-n} = 1$.) Thus, by the Rule of Sum, $p_n = \sum_{k=1}^{n} \binom{n-1}{k-1}p_{n-k},\ n \geq 1$. Computing using this recurrence leads to $p_2 = \binom{1}{0}p_1 + \binom{1}{1}p_0 = 2$, and $p_3 = \binom{2}{0}p_2 + \binom{2}{1}p_1 + \binom{2}{2}p_0 = 7$, which agrees with what we obtained initially.

Summary. A good way to start is to work out the first few cases directly. For the general case, try to organise the counting into cases depening on what happens on the first move, or first step, or to the first object. Or, replace first by n-th.

2 Solving Recurrences

To *solve* a recurrence relation means to find a function defined on the collection of indices (i.e. subscripts, usually the natural numbers) that satisfies the recurrence. There are usually many such functions. If initial conditions are given, we will want to chose the one function that gives the correct initial values.

Example 7 *For example, if c is any constant, any function of the form $c2^n$ is a solution to the recurrence relation $a_n = 2a_{n-1}$. To see this, plug the corresponding value into both sides and verify that they are equal. This is the*

Induction part of a proof by mathematical induction: If $a_{n-1} = c2^{n-1}$, then $a_n = c2^n$. If the inital condition $a_0 = 5$ is specified, then the only choice for c that gives the correct initial value is $c = 5$. This is the Basis part of a proof by mathematical induction that $a_n = 5 \cdot 2^n$. It doesn't matter in which order the basis and induction are established, what matters is that both have been demonstrated to be true. Hence, the solution is $a_n = 5 \cdot 2^n$.

We will discuss four methods for solving recurrences: (1) Guess and Check, (2) Iteration, (3) Characteristic Equations, and (4) Generating Functions. The first three are discussed in this section, and the fourth in the section on generating functions.

Guess and Check. The method is to guess a solution and then prove by induction that your guess is correct. Obtaining the "right" guess is a matter of astute observation or dumb luck, though experience can help.

Example 8 *Consider the recurrence $h_n = 2h_{n-1} + 1,\ n \geq 1$ with initial condition $h_0 = 0$. Computing the first few values gives $h_0 = 1, h_1 = 1, h_2 = 3, h_3 = 7, h_4 = 15$ and $h_5 = 31$. It seems reasonable to guess that $h_n = 2^n - 1$ for all $n \geq 0$. It is easy to prove by induction that this is correct. If $n = 0$ then $h_0 = 0 = 2^0 - 1$, so that statement (that $h_n = 2^n - 1$) is true for $n = 1$. Assume that $h_k = 2^k - 1$ for some $k \geq 0$. Consider $h_{k+1} = 2h_k + 1 = 2(2^k - 1) + 1$ (by the induction hypothesis) $= 2^{k+1} - 1$, as desired. It now follows by induction that $h_n = 2^n - 1$ for all $n \geq 0$.*

Iteration. This is also known as **repeated substitution**. It is *most useful when the recurrence involves only one previous term*, and is what you should probably try fist in such cases. The method is to repeatedly apply the recurrence and reduce it to a summation that you can hopefully exaluate. Usually is is best not to collect terms in the summation as that can obscure what is going on.

It is useful to remember that if $x \neq 1$, then $1 + x + x^2 + \cdots + x^t = \frac{x^{t+1}-1}{x-1}$, and if $x = 1$ the sum is $t + 1$.

Example 9 *Solve the recurrence $t_n = 3t_{n-1} + 7,\ n \geq 1$ with initial condition $t_0 = 5$.*

We have

$$\begin{aligned}
t_n &= 3t_{n-1} + 7 \\
&= 3(3t_{n-2} + 7) + 7 = 3^2 t_{n-2} + 3 \cdot 7 + 7 \\
&= 3^2(3t_{n-3} + 7) + 3 \cdot 7 + 7 = 3^3 t_{n-3} + 3^2 7 + 3 \cdot 7 + 7 \\
&\vdots \\
&= 3^n t_0 + 3^{n-1} 7 + 3^{n-2} 7 + \cdots + 7 \\
&= 5 \cdot 3^n + 7 \tfrac{3^n - 1}{3-1} \\
&= 5 \cdot 3^n + (7/2)(3^n - 1) = \tfrac{17}{2} 3^n + \tfrac{7}{2}
\end{aligned}$$

In arguments like this we frequently make use of an elipsis (i.e. '...'), which should be read as "*and continuing in this way, we eventually arrive at*". Almost every time an elipsis appears, we are actually using mathematical induction but declining to write out the details. This is usually ok when the pattern is "obvious", though some people might insist on a formal argument (I won't).

Exactly the same argument as above will give the solution to any recurrence of the form $t_n = at_{n-1} + b$, where a and b are constants and some initial condition is also specified. In other words, there is a theorem to be found and proved if you choose. It will also work if $n - 1$ is replaced by $n - k$, but you'll need initial values for t_0 through t_{k-1} and get k solutions depending on the remainder of n on division by k.

Some recurrences can be solved exactly by iteration only when n is of a certain form (which depends on the recurrence). In such cases this information can sometimes be used to obtain good upper and lower bounds on the solution for arbitrary n.

Example 10 *Try to solve the recurrence* $t_n = t_{\lfloor \frac{n}{2} \rfloor} + t_{\lceil \frac{n}{2} \rceil} + n$, *with initial conditions* $t_0 = t_1 = 1$.

For arbitrary n it is almost impossible to see a pattern when iteration is applied. The form of the recurrence suggests it would be good to try to solve it for values of n where the floor and ceiling never come into play. Iterating the recurrence requires dividing the subscript by 2 at each step, and we want to choose n so that this always results in an integer. This requires that n must be a power of 2. When n is even (and all powers of 2 are even except

$2^0 = 1$), the recurrence becomes $t_n = 2t_{\frac{n}{2}} + n$. If $n = 2^k$, then iteration gives

$$\begin{aligned} t_{2^k} &= 2t_{2^{k-1}} + 2^k \\ &= 2(2t_{2^{k-2}} + 2^{k-1}) + 2^k \\ &= 2^2 t_{2^{k-2}} + 2^k + 2^k \\ &\vdots \\ &= 2^k t_1 + k2^k \\ &= 2^k + k2^k. \end{aligned}$$

If n is not a power of 2 we can make use of the above analysis to go a bit further and get good bounds on t_n, *provided one additional assumption is satisfied. For what follows to make sense, it must be assumed that t_n is a non-decreasing function of n.* That is, we assume that if $m \leq n$, then $t_m \leq t_n$. Now, choose k to be the smallest integer such that $n < 2^k$. That k is the smallest implies $2^{k-1} < n$, so $2^{k-1} < n < 2^k$. The assumption about t_n implies that $t_{2^{k-1}} \leq t_n \leq t_{2^k}$. Thus, from the analysis above, $2^{k-1} + (k-1)2^{k-1} \leq t_n \leq 2^k + k2^k$. But, if $2^{k-1} < n < 2^k$, then (since log is an increasing function), $k - 1 < \log_2(n) < k$. That is $k - 1 = \lfloor \log_2(n) \rfloor$ and $k = \lceil \log_2(n) \rceil$. Also, since $n < 2^k$, we have $n/2 < 2^{k-1}$, and since $2^{k-1} < n$ we have $2^k < 2n$. Plugging all of these into the inequality for t_n we get our bounds: $n/2 + \lfloor \log_2(n) \rfloor (n/2) \leq t_n \leq 2n + \lceil \log_2(n) \rceil (2n)$, or $(1/2)(n + \lfloor \log_2(n) \rfloor n) \leq t_n \leq 2(n + \lceil \log_2(n) \rceil n)$. That is, t_n is $\Theta(n \log(n))$.

Similar considerations as above apply to other recurrences involving floors and ceilings.

Example 11 *Try to solve the recurrence $t_n = 5t_{\lfloor n^{1/3} \rfloor} + 1$.*

Because of the special form of n we will use so that the floor function does not come into play when we iterate, it will turn out that we will want to know t_2 as our initial condition (it could readily be computed if it is not given). Let's assume $t_2 = 4$ is given.

We want to choose n of an appropriate form so that every time a (subsequent) cube root is taken, the result is an integer and the floor function never comes into play. If n is of the form 2 to some power, then taking the cube root amounts to dividing the power by 3, so we want the exponent itself to be of the form 3^t, and thus will want to look at values of n of the form

$n = 2^{(3^t)}$. In this case, iterating yields

$$\begin{aligned}
t_{2^{(3^t)}} &= 5t_{2^{(3^{t-1})}} + 1 \\
&= 5(5t_{2^{(3^{t-2})}} + 1) + 1 = 5^2 t_{2^{(3^{t-2})}} + 5 + 1 \\
&\vdots \\
&= 5^t t_{2^{(3^0)}} + 5^{t-1} + 5^{t-2} + \cdots + 5 + 1 \\
&= 5^t t_2 + \tfrac{5^t - 1}{4}. \\
&= 5^t 4 + \tfrac{5^t - 1}{4}
\end{aligned}$$

If n is not of the special form we used above to solve the recurrence exactly, then we can obtain reasonable bounds on t_n under the assumption that it is an increasing function of n. Choose the smallest t such that $n < 2^{(3^t)}$. That t is smallest means $2^{(3^{t-1})} < n$, so $2^{(3^{t-1})} < n < 2^{(3^t)}$. Taking logs throughout this inequality gives $3^{t-1} < \log_2(n) < 3^t$, and doing so again gives $(t-1)\log_2(3) < \log_2(\log_2(n)) < t\log_2(3)$, or $t-1 < c\log_2(\log_2(n)) < t$, where $c = 1/\log_2(3)$. Since t_n is assumed to an non-decreasing function, $t_{2^{(3^{t-1})}} \le n \le t_{2^{(3^t)}}$, so from the above we get $5^{t-1}4 + \frac{5^{t-1}-1}{4} \le t_n \le 5^t 4 + \frac{5^t-1}{4}$. Since $t-1 < c\log_2(\log_2(n)) < t$ we have $t-1 = \lfloor c\log_2(\log_2(n))\rfloor$ and $t = \lceil c\log_2(\log_2(n))\rceil$. Plugging these in tells us that t_n is "about" (*i.e.*, a constant multiple of) $5^{c\log_2(\log_2(n))}$.

There is more that can be done. Consider $5^{c\log_2(\log_2(n))}$. Since $5 = 2^{\log_2(5)}$ we can use the rules of exponents to our advantage:

$$5^{c\log_2(\log_2(n))} = (2^{\log_2(5)})^{c\log_2(\log_2(n))} = 2^{\log_2(5)c\log_2(\log_2(n))} = \log_2(n)^{c_1},$$

where $c_1 = c\log_2(5) = \log_2(5)/\log_2(3)$. Thus, t_n is "roughly" $\log_2(n)^{c_1}$, that is, t_n is $\Theta(log(n)^{c_1})$.

Similar means can be used for recurrences where the coefficients are not constant.

Example 12 *Try to solve the recurrence* $t_n = n \cdot t_{\lfloor n/3 \rfloor},\ t_1 = 1$.

Here, similarly to the above examples, we have a hope of obtaining an exact solution by iteration only when n is a power of 3. In that case

$$\begin{aligned}
t_{3^k} &= 3^k t_{3^{k-1}} \\
&= 3^k(3^{k-1} t_{3^{k-2}}) \\
&\vdots \\
&= 3^k 3^{k-1} \cdots 3^1 t_{3^0} \\
&= 3^{k+(k-1)+\cdots+1} 1 \\
&= 3^{\frac{k(k+1)}{2}}.
\end{aligned}$$

If n is not of this special form, a futher analysis can be carried out as above.

Solving Linear Recurrence Relations with Constant Coefficients. These are recurrence relations of the form

$$a_n = \alpha_1 a_{n-1} + \alpha_2 a_{n-2} + \cdots + \alpha_k a_{n-k} + h(n),$$

where $\alpha_1, \alpha_2, \ldots, \alpha_n$ are constants. The name arises because the formula giving a_n is a linear function of (some of) the previous terms. If $h(n)$ is zero, the recurrence is called *homogeneous*, otherwise it is *non-homogeneous*.

Solving non-homogeneous recurrence relations, when possible, requires solving an associated homogeneous recurrence as part of the process, so we will discuss **solving linear homogeneous recurrence relations with constant coefficients** (LHRRWCC's) first.

Theorems that tell us how to solve LHRWCC's, and hints at how you can prove them are given below. These lead to a general procedure for solving this type of recurrence. Consider the LHRWCC's $a_n = \alpha_1 a_{n-1} + \alpha_2 a_{n-2} + \cdots + \alpha_k a_{n-k}$ or, equivalently, $a_n - \alpha_1 a_{n-1} - \alpha_2 a_{n-2} - \cdots - \alpha_k a_{n-k} = 0$.

Theorem 1 *If b is a non-zero complex number, then $a_n = b^n$ satisfies the recurrence $a_n - \alpha_1 a_{n-1} - \alpha_2 a_{n-2} - \cdots - \alpha_k a_{n-k} = 0$ if and only if b is a root of the polynomial $x^k - \alpha_1 x^{k-1} - \alpha_2 x^{k-2} - \cdots - \alpha_{k-1} x - \alpha_k = 0$.*

The equation $x^k - \alpha_1 x^{k-1} - \alpha_2 x^{k-2} - \cdots - \alpha_{k-1} x - \alpha_k = 0$ is called the *characteristic equation* of the recurrence $a_n = \alpha_1 a_{n-1} + \alpha_2 a_{n-2} + \cdots + \alpha_k a_{n-k}$. By the Fundamental Theorem of Algebra, the characteristic equation has k roots, counting multiplicities. Some of the roots may be complex.

It is important to notice that Theorem 1 is an "if and only if" theorem. In particular, it tells us that any root of the characteristic equation gives a solution to the recurrence. Thus, solutions exist.

To prove one direction of **??**, substitute $a_n = b^n$ into the recurrence and take out the common factor of b^{n-k} to obtain $b^{n-k}(b^k - \alpha_1 b^{k-1} - \alpha_2 b^{k-2} - \cdots - \alpha_{k-1} b - \alpha_k) = 0$. Since b is not zero, $b^k - \alpha_1 b^{k-1} - \alpha_2 b^{k-2} - \cdots - \alpha_{k-1} b - \alpha_k$ must be zero. To prove the other direction, reverse the argument.

Theorem 2 *If b is a root of the characteristic equation of multiplicity $t \geq 1$, then each of $b^n, nb^n, n^2 b^n, \ldots, n^{t-1} b^n$ satisfies the recurrence.*

Taken together with Theorem 1, the above result implies that if the characteristic equation involves a polynomial of degree k, then there are k different expressions that satisfy the recurrence. Also, *notice that if* $t = 1$, *then the only expression in the list is* b^n.

The proof of Theorem 2 is not hard, and it also not easy to communicate the main idea briefly.

Sketch of the proof of Theorem 2. Let

$$p(x) = x^k - \alpha_1 x^{k-1} - \alpha_2 x^{k-2} - \cdots - \alpha_{k-1}x - \alpha_k.$$

If b is a root of multiplicity $t \geq 1$, then $p(x) = (x-b)^t q(x)$, where $q(x)$ is a polynomial of degree $k-t$. Suppose r is between 0 and $t-1$, and plug $n^r b^n$ into the recurrence. After factoring, one obtains

$$\begin{aligned} b^{n-k}(n^r b^k - \alpha_1(n-1)^r b^{k-1} - \alpha_2(n-2)^r b^{k-2} - \\ \cdots - \alpha_{k-1}(n-k+1)^r b - \alpha_k(n-k)^r) = 0. \end{aligned}$$

But this expression is what you get when let $x = b$ after performing the following operation r times starting with $x^{n-k}p(x)$: differentiate then multiply by x. Since $p(x) = (x-b)^t q(x)$, this is the same as performing the same sequence of operations on $(x-b)^t x^{n-k} q(x)$. The result of doing this is a sum of many terms, each of which involves $(x-b)$. Thus, when you let $x = b$ the result is zero, which is what was needed. (Note that if you performed the operation t or more times, then some term would not involve $(x-b)$, so the result is not guaranteed to be zero.)

Theorem 3 *If* $h_1(n), h_2(n), \ldots, h_k(n)$ *all satisfy the recurrence relation*

$$a_n = \alpha_1 a_{n-1} + \alpha_2 a_{n-2} + \cdots + \alpha_k a_{n-k}$$

then, for any choice of constants $c_1, c_2, \ldots, c_k$, *so does* $h(n) = c_1 h_1(n) + c_2 h_2(n) + \cdots + c_k h_k(n)$.

The function $h(n)$ in Theorem 3 is called the *general solution* to the recurrence relation. To prove Theorem 3, just plug $h(n)$ into the recurrence relation and do algebra.

If we are given a set of k initial values for the recurrence, then we want to choose the constants in Theorem 3 so that the expression produces the correct initial values, and therefore produces (and gives a formula for the n-th term of) the correct sequence. That is, we want to find a *particular solution* that gives the correct initial values. To do this, use the k initial values to generate a system of k equations in k unknowns, the unknowns being the constants (from the general solution) that we want to determine. It is possible for the system of equations to have no solution, but it turns out to always have a solution in the cases where the roots of the characteristic equation are distinct, or when the initial values given are consecutive (like $a_0, a_1, \dots a_{k-1}$). We won't explore why here.

General Procedure for Solving LHRWCC's

1. Determine the characteristic equation.
2. Find the roots of this equation and their multiplicities.
3. Write down the general solution.
4. Use the initial conditions to get a system of k equations in k unknowns, then solve it to obtain the solution you want. (Where k is the degree of the characteristic equation.)

Notes.

1. If the "largest" index term in the recurrence is a_n, and the "smallest" index term is a_{n-k}, the characteristic equation will be a polynomial of degree k. There will be a non-zero term involving x^{k-t} whenever a_{n-t} is involved in the recurrence.
2. There should be a total of k roots of the characteristic equation, counting multiplicities.
3. Any complex roots come in complex conjugate pairs. Thus, if the degree is odd there is at least one real root.
4. If the characteristic equation has *only* integer roots, and the coefficient of x^n is 1, then these are among the positive divisors of the constant term and their negatives.

5. Since each root of multiplicity t gives us t solutions to the recurrence relation, the general solution is a sum of k terms.

Example 13 *Solve the recurrence relation* $a_n = a_{n-1} + 8a_{n-2} - 12a_{n-3},\ n \geq 3$*, subject to the initial conditions* $a_0 = 0,\ a_1 = 1,\ a_2 = 3$.

To determine the characteristic equation, first bring all terms over to the LHS. Thus, the recurrence is $a_n - a_{n-1} - 8a_{n-2} + 12a_{n-3} = 0$, so the characteristic equation is $x^3 - x^2 - 8x + 12 = 0$. Testing the positive divisors of 12 and their negatives as possible roots reveals that 2 and -3 are roots, and after factoring the LHS of the characteristic equation we get $(x-2)^2(x+3) = 0$. Thus 2 is a root of multiplicity 2, and 3 is a root of multiplicity 1. Using Theorems 2 and 3, the general solution is then $h(n) = c_1 2^n + c_2 n 2^n + c_3(-3)^n$. Finally, we need to use the initial conditions to determine the constants.

$$\begin{array}{rcccccccc} a_0 = 0 & \Rightarrow & c_1 2^0 & + & c_2 0(2^0) & + & c_3(-3)^0 & = & 0 \\ a_1 = 1 & \Rightarrow & c_1 2^1 & + & c_2 1(2^1) & + & c_3(-3)^1 & = & 1 \\ a_2 = 2 & \Rightarrow & c_1 2^2 & + & c_2 2(2^2) & + & c_3(-3)^2 & = & 2 \end{array}$$

This system of 3 equations in 3 unknowns has the solution $c_1 = -2/25, c_2 = 3/10, c_3 = 2/25$, so the solution to the recurrence relation that also satisfies the initial conditions is $a_n = (-2/25)2^n + (3/10)n2^n + (2/25)(-3)^n$.

Example 14 *Solve* $a_n = 2a_{n-1} - 4a_{n-2}, +8a_{n-3}\ n \geq 3$*, given the initial conditions* $a_0 = 1, a_1 = 1, a_2 = 1$.

Here the characteristic equation is $x^3 - 2x^2 + 4x - 8 = 0$. By trial we find that 2 is a root, and the equation is $(x-2)(x^2+4) = 0$. Using the quadratic formula on the second factor tells us that the other roots are $2i$ and $-2i$. Since all of the roots have multiplicity 1, the general solution is $h(n) = c_1 2^n + c_2(2i)^n + c_3(-2i)^n$. Plugging in the initial conditions gives:

$$\begin{array}{rcccccl} a_0 = 1 & \Rightarrow & c_1 2^0 & + & c_2(2i)^0 & + & c_3(-2i)^0 = 1 \\ a_1 = 1 & \Rightarrow & c_1 2^1 & + & c_2(2i)^1 & + & c_3(-2i)^1 = 1 \\ a_2 = 1 & \Rightarrow & c_1 2^2 & + & c_2(2i)^2 & + & c_3(-2i)^2 = 1 \end{array}$$

I think that the solution is $c_1 = \frac{5}{8},\ c_2 = -\frac{3}{16} - \frac{1}{16}i,\ c_3 = \frac{9}{16} + \frac{1}{16}i$, but you should check for yourself.

Solving Non-homogeneous Linear Recurrence Relations with Constant Coefficients.

Recall that these are recurrence relations of the form $a_n = \alpha_1 a_{n-1} + \alpha_2 a_{n-2} + \cdots + \alpha_k a_{n-k} + h(n)$, where $\alpha_1, \alpha_2, \ldots, \alpha_n$ are constants, and $h(n)$ is not identically zero. We'll refer to $h(n)$ as the *non-homogeneous term*. We will need to use solutions to the recurrence relation obtained by replacing $h(n)$ by zero, which we'll call the *associated non-homogeneous recurrence relation.*

Theorems that give us a method for solving non-homogeneous recurrences are listed below, and followed by discussion of how to use the method.

Theorem 4 *If $f(n)$ is a solution to the associated homogeneous recurrence, and $g(n)$ is a solution to the non-homogeneous recurrence, then $f(n) + g(n)$ is also a solution to the non-homogeneous recurrence.*

To prove this theorem, plug $f(n)+g(n)$ into the non-homogeneous recurrence and do algebra.

Theorem 4 suggests that *the general solution to a non-homogeneous recurrence relation should be the sum of the general solution to the associated homogeneous recurrence and any particular solution to the non-homogeneous recurence*. This expression will involve k unknowns (from the general solution to the associated homogeneous recurrence) which can be determined by plugging in the k initial conditions in turn and solving the resulting system of k equations in k unknowns as before.

Theorem 5 *Suppose the non-homogeneous term is of the form $p(n)s^n$, where $p(n)$ is a polynomial of degree r and s is a constant. If s is not a root of the characteristic equation, then there is a particular solution of the form $q(n)s^n$, where q is a polynomial of degree at most r. If s is a root of the characteristic equation of multiplicity t, then there is a particular solution of the form $n^t q(n)s^n$, where q is a polynomial of degree at most r.*

Theorem 5 says that in some situations there is a particular solution of a certain form. To find it, use the **method of undetermined coefficients**. If the number s in is not a root of the characteristic equation, then plug $a_n = (\beta_r n^r + \beta_{r-1} n^{r-1} + \cdots + \beta_1 n + \beta_0)s^n$ into the recurrence. If s is a root of multiplicity t of the characteristic equation, then plug $a_n = n^t(\beta_r n^r +$

$\beta_{r-1}n^{r-1} + \cdots + \beta_1 n + \beta_0)s^n$ into the recurrence. (Remember to change all n's to $(n-1)$'s when you are replacing a_{n-1} and to $(n-2)$'s when you are replacing a_{n-2}, etc..) In either case, do some algebra. Both sides of the equation will be polynomials in n, and two polynomials are equal if and only if the coefficients of like powers of n are equal. This results in a system of $r+1$ linear equations in $r+1$ unknowns (the β's) to solve. A demonstration that the system of equations you get is always solvable completes the proof.

Note that non-homogeneous terms like n or 6^n are of the right form to apply Theorem 5: there is a 1 that's not written down. The term n is the same as $n1^n$, and 6^n is the same as $1 \cdot 6^n$. The two terms with the (implied) ones written in are clearly of the right form. In the last case, remember that a constant (in this case 1) is a polynomial of degree 0.

If you look in some combinatorics texts, for example Tucker, or Grimaldi, you will find tables of other situations in which the form of a particular solution is known. We will deal only with the cases covered by the above fact, and the next one.

Theorem 6 *If the non-homogeneous term is of the form $p_1(n)s_1^n + p_2(n)s_2^n + \cdots + p_m(n)s_m^n$, then there is a particular solution of the form $f_1(n) + f_2(n) + \cdots + f_m(n)$ where, for each i, $f_i(n)$ is a particular solution to the non-homogeneous recurence $a_n = \alpha_1 a_{n-1} + \alpha_2 a_{n-2} + \cdots + \alpha_k a_{n-k} + p_i(n)s_i^n$.*

Theorem 6 allows you to find a particular solution to a recurrence with a complicated non-homogeneous term by finding particular solutions to a bunch of simpler non-homogeneous recurrences, and then adding these together. If the non-homogeneous term is a sum of m terms as in the theorem, then there will be m non-homogeneous recurrences to solve, all with the same associated homogeneous recurrence. Thus, you will end up finding the general solution to the associated homogeneous recurrence once, using Theorem 6 and the method of undetermined coefficiants m times, adding all of these together, and then generating and solving a system of k equations in k unknowns. In total, you'll solve $m+1$ linear systems.

Procedure for solving non-homogeneous recurrences.

1. Write down the associated homogeneous recurrence and find its general solution.
2. Find a particular solution the non-homogeneous recurrence. This may involve solving several simpler non-homogeneous recurrences (using this same procedure).
3. Add all of the above solutions together to obtain the general solution to the non-homogeneous recurrence.
4. Use the initial conditions to get a system of k equations in k unknowns, then solve it to obtain the solution you want.

Note. You must solve the associated homogeneous recurrence first because you need to know the roots of the characteristic equation and their multiplicities *before* you can find the particular solution you want in the next step.

Example 15 *Solve* $a_n = 4a_{n-1} - 4a_{n-2} + n2^n + 3^n + 4,\ n \geq 2,\ a_0 = 0, a_1 = 1.$

The associated homogeneous recurrence is $a_n = 4a_{n-1} - 4a_{n-2}$. Its characteristic equation is $x^2 = 4x - 4$, or $(x-2)^2 = 0$. Thus, 2 is the only root and it has multiplicity 2. The general solution to the associated homogeneous recurrence is $c_1 2^n + c_2 n 2^n$.

To find a particular solution to the non-homogeneous recurrence, we add together particular solutions to the three "simpler" non-homogeneous recurrences:

- $a_n = 4a_{n-1} - 4a_{n-2} + n2^n$,
- $a_n = 4a_{n-1} - 4a_{n-2} + 3^n$, and
- $a_n = 4a_{n-1} - 4a_{n-2} + 4$.

Let's find a particular solution to $a_n = 4a_{n-1} - 4a_{n-2} + 4$ first. The non-homogeneous term is $4 = 4 \cdot 1^n$, and since 1 is not a root of the characteristic equation, Theorem 5 says there is a particular solution of the form $c1^n$. To determine c, plug $c1^n = c$ into the *non-homogeneous* recurrence and get

$c = 4c - 4c + 4 = 4$. Thus, a particular solution to the recurrence under consideration is $a_n = 4$.

Now let's do the same for $a_n = 4a_{n-1} - 4a_{n-2} + 3^n$. The non-homogeneous term is $3^n = 1 \cdot 3^n$. Since 3 is not a root of the characteristic equation, Theorem 5 says there is a particular solution of the form $c3^n$. To determine c, plug $c3^n$ into the non-homogeneous recurrence and get $c3^n = 4c3^{n-1} - 4c3^{n-2} + 3^n$. After dividing by the common factor of 3^{n-2}, we have $9c = 12c - 4c + 9$, or $c = 9$. Thus, a particular solution to the recurrence under consideration is $a_n = 9 \cdot 3^n$.

Let's find a particular solution to $a_n = 4a_{n-1} - 4a_{n-2} + n2^n$. The non-homogeneous term is $n2^n$. Since 2 is a root of the characteristic equation with multiplicity 2, NHLRR 2 says there is a particular solution of the form $n^2(un + v)2^n$. To determine u and v, plug this expression into the non-homogeneous recurrence and get

$$n^2(un+v)2^n = 4(n-1)^2(u(n-1)+v)2^{n-1} - 4(n-2)^2(u(n-2)+v)2^{n-2} + n2^n.$$

After dividing both sides by $2n - 2$ and doing a bit of algebra this becomes

$$\begin{aligned}
4un^3 + 4vn^2 &= 8u(n-1)^3 + 8v(n-1)^2 \\
&\quad -4u(n-2)^3 - 4v(n-2)^2 + 4n \\
&= 8u(n^3 - 3n^2 + 3n - 1) + 8v(n^2 - 2n + 1) \\
&\quad -4u(n^3 - 6n^2 + 2n - 8) - 4v(n^2 - 4n + 4) + 4n \\
&= (8u - 4u)n^3 + (-24u + 8v + 24u - 4v)n^2 \\
&\quad +(24u - 16v - 48u + 16v + 4)n \\
&\quad +(-8u + 8v + 32u - 16v)
\end{aligned}$$

Equating coefficients of like powers of n on the LHS and RHS gives:

$$\begin{aligned}
4u &= 4u, \\
4v &= (-24u + 8v + 24u - 4v) \\
0 &= (24u - 16v - 48u + 16v + 4) = -24u + 4 \\
0 &= (-8u + 8v + 32u - 16v) = -24u - 8v
\end{aligned}$$

The first two equations tell us nothing. The third equation implies $u = 1/6$. Substituting this value into the last equation and solving gives $v = 1/2$. Thus, a particular solution to the given recurrence is $n^2(\frac{1}{6}n + \frac{1}{2})2^n$.

By Theorem 6, combining the three particular solutions just obtained gives a particular solution to $a_n = 4a_{n-1} - 4a_{n-2} + n2^n + 3^n + 4$. It is $4 + 9 \cdot 3^n + n^2(\frac{1}{6}n + \frac{1}{2})2^n$.

The general solution is therefore $c_1 2^n + c_2 n 2^n + 4 + 9 \cdot 3^n + n^2(\frac{1}{6}n + \frac{1}{2})2^n$. To determine the constants, use the initial conditions.

$$
\begin{aligned}
a_0 = 0 \quad &\Rightarrow \quad c_1 + 4 + 9 = 0 \\
a_1 = 1 \quad &\Rightarrow \quad 2c_1 + 2c_2 + 4 + 27 + (1/6 + 1/2)2 = 1
\end{aligned}
$$

The first equation says $c_1 = -13$. Plugging this into the second equation gives $c_2 = -8/3$. Thus, the solution to the recurrence is

$$a_n = (-13)2^n + (-8/3)n2^n + 4 + 9 \cdot 3^n + n^2(\frac{1}{6}n + \frac{1}{2})2^n.$$