Lecture Notes

Last lecture we saw how to solve Divide and Conquer recurrences. These come up all the time in the analysis of recursive algorithms such as Merge Sort.

In this lecture, we'll consider another family of recurrences, called *linear recurrences*, that also frequently arise in computer science and other disciplines. We'll first see how to solve a specific linear recurrence and then generalize our method to work for all linear recurrences.

1 Graduate Student Job Prospects

In a certain new academic field (say computer science), there are only so many faculty positions available in all the universities of the world. Initially, there were not enough qualified candidates, and many positions were unfilled. But, over time, new graduates are filling the positions, making job prospects for later computer science students ever more bleak. Worse, the increasing number of professors are able to train an increasing number of graduate students, causing positions to fill ever more rapidly. Eventually, the universities will be saturated; new computer science graduates will have no chance at an academic career. Our problem is to determine when the universities will stop hiring new computer science faculty and, in particular, to answer the question, "Are the 6.042 TAs doomed?" Here are the details of the problem.

- There are a total of N faculty positions available worldwide. This number never changes due to budgetary constraints.
- Congress has passed a law forbidding forced retirement, and no one will retire voluntarily. (This is true and a problem!) In our analysis, therefore, once a faculty position is filled, it never opens up again.
- Each year, every professor trains exactly 1 student who will go on to become a professor the following year. The only exception is that first year professors do not train students; they are too busy publishing, teaching, getting grants, and serving on committees.
- In year 0, there are no computer science professors in the world. In year 1, the first professor is hired.

1.1 Finding a Recurrence

Ideally, we could find a formula for the number of professors in the world in a given year. Then we could determine the year in which all N faculty positions are filled. Let $f(n)$ be the number of professors during year n. To develop some intuition about the problem, we can compute values of this function for small n by hand.

$f(0) = 0$	No CS professors; the dark ages.
$f(1) = 1$	1 new professor; too busy to train a student.
$f(2) = 1$	1 old professor; now training a student.
$f(3) = 2$	1 new prof, 1 old prof; new prof too busy, old prof training.
$f(4) = 3$	1 new prof, 2 old profs; new prof too busy, both old profs training
$f(5) = 5$	2 new profs, 3 old profs
$f(6) = 8$	3 new profs, 5 old profs

In general, the number of professors in year n is equal to the number of professors last year plus the number of new hires. The number of professors last year is $f(n-1)$. The number of new hires is equal to the number of professors two years ago, $f(n-2)$, since each of these professors trained a student last year. These observations give the following recurrence equation for the number of professors:

$$f(n) = f(n-1) + f(n-2) \qquad (n \geq 2)$$

This is the familiar Fibonacci recurrence. Looking back, the values of $f(n)$ that we computed by hand are indeed the first few Fibonacci numbers. Fibonacci numbers arise in all sorts of applications. For example, we have already seen that the running time of Euler's GCD algorithm for inputs of size at most $f(n)$ is bounded by n. So how big is $f(n)$ anyway? Of course, we could compute as many Fibonacci numbers as we like using the recurrence, but it would be much nicer to find a closed form.

1.2 Solving the Recurrence

Solving the Fibonacci recurrence is easy because the recurrence is linear. (Well, "easy" in the sense that you can learn the technique in one lecture; discovering it actually took six centuries.)

Definition A recurrence is *linear* if it is of the form

$$\begin{aligned} f(n) &= a_1 f(n-1) + a_2 f(n-2) + \ldots + a_d f(n-d) \\ &= \sum_{i=1}^{d} a_i f(n-i) \end{aligned}$$

where $a_1, a_2, \ldots a_d$ are constants. The *order* of the recurrence is d.

For example, the Fibonacci recurrence is order 2 and has $a_1 = a_2 = 1$.

For now, let's try to solve just the Fibonacci recurrence; we'll see how to solve general linear recurrences later in the lecture. Our rule of thumb is that Guess and Verify is the first method to apply to a recurrence. It turns out that for a linear recurrence, an exponential solution is a good guess. However, since we know nothing beyond this, our initial Guess and Verify attempt will really only be a "dry run"; that is, we will not make an exact guess and will not verify it with a complete proof. Rather, the goal of this first attempt is only to clarify the form of the solution.

Guess $f(n) = c\alpha^n$

Here c and α are parameters introduced to improve our odds of having a correct guess; in the verification step, we can pick values that make the proof work. To further improve our odds, for now let's neglect the boundary conditions, $f(0) = 0$ and $f(1) = 1$.

Verification. *(dry run)* Plugging our guess into the recurrence $f(n) = f(n-1) + f(n-2)$ gives:

$$\begin{aligned} c\alpha^n &= c\alpha^{n-1} + c\alpha^{n-2} \\ \alpha^2 &= \alpha + 1 \\ \alpha^2 - \alpha - 1 &= 0 \\ \alpha &= \frac{1 \pm \sqrt{5}}{2} \end{aligned}$$

In the first step, both sides of the equation are divided by $c\alpha^{n-2}$. The second step only involves rearranging terms. In the final step, we can compute α by solving the quadratic equation.

This calculation suggests that c can be anything, but that α must be $(1 \pm \sqrt{5})/2$. Evidently, there are two solutions to the recurrence:

$$f(n) = c\left(\frac{1+\sqrt{5}}{2}\right)^n \qquad \text{or} \qquad f(n) = c\left(\frac{1-\sqrt{5}}{2}\right)^n$$

■

There are actually more solutions than suggested by our "dry run" of Guess and Verify; in fact, any linear combination of the above two solutions is also a solution. The following theorem states that this is true in general for linear recurrences.

Theorem 1.1 If $f(n) = \alpha_1^n$ and $f(n) = \alpha_2^n$ are two solutions to a linear recurrence (without boundary conditions), then $f(n) = c_1 a_1^n + c_2 a_2^n$ is also a solution for all constants c_1 and c_2.

Proof. Every linear recurrence has the following form:

$$f(n) = \sum_{i=1}^{d} a_i f(n-i)$$

We must prove that $f(n) = c_1\alpha_1^n + c_2\alpha_2^n$ is a solution; that is, we must prove:

$$c_1\alpha_1^n + c_2\alpha_2^n = \sum_{i=1}^{d} a_i(c_1\alpha_1^{n-i} + c_2\alpha_2^{n-i})$$

Substituting the two solutions $f(n) = \alpha_1^n$ and $f(n) = \alpha_2^n$ into the recurrence gives the following two equations:

$$\alpha_1^n = \sum_{i=1}^{d} a_i\alpha_1^{n-i}$$
$$\alpha_2^n = \sum_{i=1}^{d} a_i\alpha_2^{n-i}$$

Multiplying the first equation by c_1, multiplying the second equation by c_2, and adding the two gives the desired result. ■

The theorem implies that for all c_1 and c_2, the following is a solution to the Fibonacci recurrence without boundary conditions:

$$f(n) = c_1\left(\frac{1+\sqrt{5}}{2}\right)^n + c_2\left(\frac{1-\sqrt{5}}{2}\right)^n$$

All that remains is to find a solution that satisfies the boundary conditions $f(0) = 0$ and $f(1) = 1$. The first boundary condition requires that:

$$\begin{aligned} & f(0) = c_1\left(\frac{1+\sqrt{5}}{2}\right)^0 + c_2\left(\frac{1-\sqrt{5}}{2}\right)^0 = 0 \\ \Rightarrow \quad & c_1 + c_2 = 0 \end{aligned}$$

Similarly, the second boundary condition requires that:

$$\begin{aligned} & f(1) = c_1\left(\frac{1+\sqrt{5}}{2}\right)^1 + c_2\left(\frac{1-\sqrt{5}}{2}\right)^1 = 1 \\ \Rightarrow \quad & c_1\left(\frac{1+\sqrt{5}}{2}\right) + c_2\left(\frac{1-\sqrt{5}}{2}\right) = 1 \end{aligned}$$

We now have two linear equations in two unknowns! The system of equations is not degenerate, so there is a unique solution: $c_1 = 1/\sqrt{5}$ and $c_2 = -1/\sqrt{5}$. We now have a complete solution to the Fibonacci recurrence *with* boundary conditions:

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

This looks completely wrong! All Fibonacci numbers are integers, but this expression is full of square roots of five! Amazingly, however, the square roots always cancel out. This expression really does give the Fibonacci numbers if we plug in $n = 0, 1, 2, \ldots$. If one were required to guess the solution to the Fibonacci recurrence, then it is easy to imagine taking at least six centuries to come up with this!

1.3 Job Prospects

Let's return to the original question: how long until all N faculty positions are taken?

To answer this question, we must find the smallest n such that $f(n) \geq N$. This will imply that in year n, there are as many professors as university positions. Therefore, graduates after year n will have to find other employment, e.g. shuffling golden disks in an obscure monastic community for the next 10^{19} years.

Because $f(n)$ has such a complicated form, it is hard to compute the right value of n exactly. However, we can make a good approximation. Notice that $f(n)$ is the sum of two terms. The key observation is that the second term tends to zero as n grows large, since $|\frac{1-\sqrt{5}}{2}| = 0.618\ldots < 1$. This means that we can rewrite $f(n)$ with asymptotic notation as follows:

$$f(n) = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n + o(1)$$

Using this approximation for $f(n)$, we can determine the approximate year in which all faculty positions will be taken as follows:

$$\begin{aligned} & f(n) \geq N \\ \Leftrightarrow \quad & \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n + o(1) \geq N \\ \Leftrightarrow \quad & \frac{\log(\sqrt{5}\ N + o(1))}{\log(\frac{1+\sqrt{5}}{2})} \end{aligned}$$

This implies that all the faculty jobs will be filled in $\Theta(\log N)$ years. This makes sense, since the number of professors is increasing exponentially. For example, if there are $N = 10000$ jobs, then all will be taken in $n = 20.8$ years. Now, as to the fate of the TAs, there are fewer than ten thousand faculty jobs in computer science and hiring started about 20 years ago...

The solution to the Fibonacci recurrence has an interesting corollary: for large n, $\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$ is very close to an integer— namely, the n-th Fibonacci number. This expression is the first term in the closed form for the n-th Fibonacci number. Since the other term of the closed form goes to zero exponentially fast and Fibonacci numbers are always integers, the first term must be very close to an integer! This is one of many strange facts about the "Golden Ratio", $\frac{1+\sqrt{5}}{2} = 1.61803\ldots$.

2 General Linear Recurrences

The method we used to solve the Fibonacci recurrence can be used to solve any linear recurrence! Recall that every linear recurrence has the following form:

$$f(n) = \sum_{i=1}^{d} a_i f(n-i)$$

Substituting the guess $f(n) = \alpha^n$ into the recurrence gives:

$$\begin{aligned}
\alpha^n &= a_1\alpha^{n-1} + a_2\alpha^{n-2} + \ldots + a_d\alpha^{n-d} \\
\alpha^d &= a_1\alpha^{d-1} + a_2\alpha^{d-2} + \ldots + a_{d-1}\alpha + a_d \\
\alpha^d - a_1\alpha^{d-1} - a_2\alpha^{d-2} - \ldots - a_{d-1}\alpha - a_d &= 0
\end{aligned}$$

The equation is divided by α^{n-d} in the first step, and terms are rearranged in the second step. The last equation is called the *characteristic equation* of the recurrence. The characteristic equation of a recurrence can be read of very quickly; the coefficients of the equation are the same as the coefficients of the recurrence. In general, the characteristic equation will have d roots, where d is the order of the recurrence. The solution to the recurrence depends on whether all roots of the characteristic equation are distinct.

2.1 Simple Case: Distinct Roots

If the characteristic equation has d distinct roots $r_1, r_2, \ldots, r_d$, then the solution to the recurrence has the form:

$$f(n) = c_1 r_1^n + c_2 r_2^n + \ldots + c_d r_d^n$$

Note that the roots of the characteristic equation need not even be real! In the end, imaginary parts will cancel just as the $\sqrt{5}$'s canceled in the solution to the Fibonacci recurrence.

The constants $c_1, c_2, \ldots, c_d$ depend on the boundary conditions. Each boundary condition generates a linear equation in these constants. For example, the boundary conditions $f(0) = b_0$, $f(1) = b_1$, $\ldots$, $f(d-1) = b_{d-1}$ generate the linear equations:

$$\begin{aligned}
b_0 &= c_1 + c_2 + \ldots + c_d \\
b_1 &= c_1 r_1 + c_2 r_2 + \ldots + c_d r_d \\
\ldots & \ldots \ldots \\
b_{d-1} &= c_1 r_1^{d-1} + c_2 r_2^{d-1} + \ldots + c_d r_d^{d-1}
\end{aligned}$$

If there are d boundary conditions, then we obtain d linear equations in d unknowns. A neat linear algebra fact is that a system of equations of this particular type is never degenerate. This means that for every set of constants $b_0, b_1, \ldots b_{d-1}$ in the boundary conditions, we can find a unique set of constants $c_0, c_1, \ldots c_{d-1}$ that give a solution to the recurrence and satisfy the boundary conditions.

2.2 Tricky Case: Repeated Roots

If the characteristic equation of a recurrence has a repeated root, then the solution is slightly more complicated. The main tool is the following theorem, which we will not prove.

Theorem 2.1 If r is a root of the characteristic equation of a recurrence with multiplicity m, then all of $r^n, nr^n, n^2r^n, \ldots n^{m-1}r^n$ are solutions to the recurrence.

For example, if the characteristic equation of a recurrence has root r_1 with multiplicity 2, root r_2 with multiplicity 1, and root r_3 with multiplicity 3, then the solution to the recurrence has the form:

$$f(n) = c_1 r_1^n + c_2 n r_1^n + c_3 r_2^n + c_4 r_3^n + c_5 n r_3^n + c_6 n^2 r_3^n$$

As before, the constants depend on the boundary conditions. Each boundary condition generates a linear equation, and the constants are computed by solving the resulting system of linear equations. In this respect, solving linear recurrences is very similar to solving linear differential equations.

2.3 An Example of the Tricky Case

To understand recurrences of this tricky type, let's work an example.

Suppose that there is a type of plant that lives forever, but only reproduces during the first year of life. How fast will the plant population grow? (Notice that this is just the reverse of the graduate student job problem where faculty "reproduce" in every year except the first.)

Let $f(n)$ be the number of plants in year n. As boundary conditions, define $f(0) = 0$ and $f(1) = 1$. The plant population in year n is equal to the population from the year before plus the number of new plants. The population from the year before is $f(n-1)$. The number of new plants this year is equal to the number of new plants last year or $f(n-1) - f(n-2)$. We can now form a recurrence equation for the plant population:

$$\begin{aligned} f(n) &= f(n-1) + (f(n-1) - f(n-2)) \\ &= 2f(n-1) - f(n-2) \\ f(n) - 2f(n-1) + f(n-2) &= 0 \end{aligned}$$

The characteristic equation is $\alpha^2 - 2\alpha + 1 = 0$, which has the single root $\alpha = 1$ with multiplicity 2. Therefore, the solution to the recurrence has the form:

$$\begin{aligned} f(n) &= c_1(1)^n + c_2 n(1)^n \\ &= c_1 + c_2 n \end{aligned}$$

The boundary conditions, $f(0) = 0$ and $f(1) = 1$, generate the following linear equations:

$$\begin{aligned} 0 &= c_1 + c_2(0) \\ 1 &= c_1 + c_2(1) \end{aligned}$$

The solution to the linear system is $c_1 = 0$ and $c_2 = 1$. Therefore, the solution to the recurrence is:

$$\begin{aligned} f(n) &= c_1 + c_2 n \\ &= 0 + (1)n \\ &= n \end{aligned}$$

The answer turns out to be very simple! In year n, there are exactly n plants. Of course, we probably could have solved this problem more easily with Guess and Verify, but for many other examples, guessing is not so easy.

3 Inhomogeneous Recurrences

We can now solve all recurrences of the form:

$$f(n) - a_1 f(n-1) - \ldots - a_d f(n-d) = 0$$

Strictly speaking, this is the family of linear *homogeneous* recurrences. These recurrences are called homogeneous because the right side of the equation is 0. If the right side were 1 or n^2 or some arbitrary function $g(n)$, then the recurrence would be called *inhomogeneous*.

Definition An inhomogeneous linear recurrence has the form:

$$f(n) - a_1 f(n-1) - a_2 f(n-2) - \ldots - a_d f(n-d) = g(n)$$

where $a_1, a_2, \ldots a_d$ are constants.

There are three steps to solving an inhomogeneous linear recurrence.

1. Replace $g(n)$ by 0 and solve the resulting homogeneous recurrence. (Ignore boundary conditions for now; that is, do not solve for constants $c_1, c_2, \ldots, c_d$.) The solution to the homogeneous recurrence is called the *homogeneous solution.*
2. Restore $g(n)$ and find a single solution to the recurrence, again ignoring boundary conditions. This is called the *particular solution.* (This is the hard part!)
3. Add the homogeneous and particular solutions together to obtain the *general solution.* Now use the boundary conditions to determine the constant factors by the usual method of generating and solving a system of linear equations.

Notice that the method for solving inhomogeneous linear recurrences is closely analogous to the method for solving inhomogeneous linear differential equations!

3.1 An Example of an Inhomogeneous Recurrence

Let $f(n) = 4f(n-1) + 3^n$ and, as a boundary condition, let $f(1) = 1$. This is an inhomogeneous linear recurrence.

Step 1: Solve the Homogeneous Recurrence

The homogeneous recurrence is $f(n) - 4f(n-1) = 0$. The characteristic equation is $\alpha - 4 = 0$. The only root is $\alpha = 4$. Therefore, the homogeneous solution is $c_1 4^n$.

Step 2: Find a Particular Solution

We must find a single solution to the recurrence $f(n) = 4f(n-1) + 3^n$. This is usually the hard part of solving an inhomogeneous linear recurrence. Let's guess that there is a solution of the form $c3^n$. Substituting this into the recurrence gives:

$$\begin{aligned} f(n) &= 4f(n-1) + 3^n \\ c3^n &= 4c3^{n-1} + 3^n \\ 3c &= 4c + 3 \\ c &= -3 \end{aligned}$$

The equation was divided by 3^{n-1} in the first step. The second and third steps were only simplification. We now have a particular solution, $f(n) = -3 \cdot 3^n = -3^{n+1}$.

Step 3: Add Solutions and Find Constants

We now add the homogeneous solution and the particular solution to obtain the general solution:

$$f(n) = c_1 4^n - 3^{n+1}$$

The value of the constant is obtained from the boundary condition, $f(1) = 1$:

$$\begin{aligned} 1 &= c_1 4^1 - 3^{1+1} \\ &= 4c_1 - 9 \\ c_1 &= 5/2 \end{aligned}$$

The solution to the recurrence is $f(n) = \frac{5}{2}4^n - 3^{n+1}$!

Since we could easily have made a mistake, let's check to make sure that our solution at least works for $n = 2$. From the recurrence, $f(2) = 4f(1) + 3^2 = 13$. From our closed form, $f(2) = \frac{5}{2}4^2 - 3^3 = 40 - 27 = 13$. It looks right!

3.2 A Final Example

Let $f(n) = 4f(n-1) - 4f(n-2) + n$ and, as a boundary condition, let $f(0) = 4$ and $f(1) = 7$.

Step 1: Solve the Homogeneous Recurrence

The homogeneous recurrence is $f(n) - 4f(n-1) + 4f(n-2) = 0$. The characteristic equation is $\alpha^2 - 4\alpha + 4$. The only root is $\alpha = 2$ with multiplicity 2. Therefore, the homogeneous solution is $c_1 2^n + c_2 n 2^n$.

Step 2: Find a Particular Solution

Now we need a single solution to the recurrence $f(n) = 4f(n-1) - 4f(n-2) + n$. Let's guess a solution of the form $an + b$ and substitute into the recurrence:

$$\begin{aligned} f(n) &= 4f(n-1) - 4f(n-2) + n \\ an + b &= 4(a(n-1) + b) - 4(a(n-2) + b) + n \\ &= 4an - 4a + 4b - 4an + 8a - 4b + n \\ &= 4a + n \\ &\Rightarrow a = 1, b = 4a = 4 \end{aligned}$$

This gives the particular solution $f(n) = n + 4$.

Step 3: Add Solutions and Find Constants

Adding the homogeneous solution and the particular solution gives the general solution:

$$f(n) = c_1 2^n + c_2 n 2^n + n + 4$$

The values of the constants are obtained from the boundary conditions, $f(0) = 4$ and $f(1) = 7$.

$$\begin{aligned} 4 &= c_1 2^0 + 0 c_2 2^0 + 0 + 4 \\ &= c_1 + 4 \\ c_1 &= 0 \\ 7 &= c_1 2^1 + 1 c_2 2^1 + 1 + 4 \\ &= 2c_1 + 2c_2 + 5 \\ c_2 &= 1 \end{aligned}$$

The solution to the recurrence is $f(n) = n2^n + n + 4$.

3.3 Finding a Particular Solution

The hardest part in solving inhomogeneous recurrences is finding a particular solution. There are some good rules of thumb, however. Suppose the recurrence has the form:

$$f(n) - a_1 f(n-1) - a_2 f(n-2) - \ldots - a_d f(n-d) = g(n)$$

If $g(n)$ is a polynomial of degree r, then guess that $f(n)$ is a polynomial of degree r. For example, if $g(n) = n^2$, then guess $f(n) = an^2 + bn + c$ and determine a, b, and c by substituting and verifying. Similarly, if $g(n)$ is an exponential, then guess that $f(n)$ is an exponential. For example, if $g(n) = 2^n + 3^n$, then guess $f(n) = a2^n + b3^n$. In general, guess a particular solution with the same form as $g(n)$.