COMPSCI 575/MATH 513

Combinatorics and Graph Theory

Lecture #24: Recurrences: D&C and Linear
(Tucker Sections 7.2, 7.3, and 7.4)
David Mix Barrington
7 November 2016

Recurrences

- Systems of Recurrences
- Divide and Conquer Recurrences
- The CLRS Master Theorem
- Linear Recurrences
- Solving Linear Recurrences
- Inhomogeneous Recurrences
- Compound Inhomogeneous Terms

I am in an Opera!

- Valley Light Opera is doing Gilbert and Sullivan's *Ruddigore* November 12, 13, 18, 19, and 20 at the Academy of Music in Noho.
- Preview show Friday 11th at 7:30 costs $5 instead of $15 or $10 college rush at other shows: see vlo.org.

Systems of Recurrences

- Sometimes we need more than one recurrence to solve a counting problem.
- Consider strings over {a,b,c} with an even number of b's and an odd number of c's.
- If $f(n)$ is the number of such strings of length n, we have that $f(n) = f(n-1) + g(n-1) + h(n-1)$, where $g(n)$ is the number with odd numbers of both b's and c's, and $h(n)$ the number with even numbers of each.

Systems of Recurrences

- Then g(n), for example, is g(n-1) + f(n-1) + i(n-1), where i(n) is the number with an even number of b's and an odd number of c's.
- Each of the four functions is defined by a recurrence using itself and two of the others.
- By induction on n, assuming we define f(0), g(0), h(0), and i(0) to each be 1, we have well-defined and correct values f(n), g(n), h(n), and i(n) for each n.

Divide and Conquer

- Many algorithms take a divide and conquer approach, reducing a problem to similar problems with smaller parameters. Much of COMPSCI 311 is spent analyzing the resources used by such algorithms, and recurrences are a key tool in this analysis.
- If a_n is the number of steps to solve a problem of size n, we often get a recurrence of the form $a_n = ca_{n/2} + f(n)$, where c is a constant and f(n) is the time to split and merge the subproblems.

Simple D&C Examples

- If $c = 1$ and $f(n)$ is constant, we have $a_n = a_{n/2} + d$, which solves to $a_n = d\log_2(n) + A$, where A is a constant chosen to fit the initial conditions. We assume here that n is a power of 2, to avoid ceilings and floors.
- If $c = 2$ and $f(n)$ is constant, we have $a_n = 2a_{n/2} + d$, which solves to $a_n = An - d$. Our $3n/2 - 2$ steps to find max and min fits into this case.
- If $c = 2$ and $f(n) = dn$, we have $a_n = 2a_{n/2} + dn$, which solves to $a_n = dn(\log_2 n + A)$.

Fast Multiplication

- Normally multiplying two n-bit numbers would require $O(n^2)$ bit multiplications.
- By adding some cheaper additions, though, we can do it with fewer multiplications.
- Write the numbers w_1 and w_2 as u_1v_1 and u_2v_2, where the u's and v's are n/2 bit numbers. Then $w_1 \times w_2 = (u_1 \times u_2)2^n + [(u_1 \times v_2) + (v_1 \times u_2)]2^{n/2} + (v_1 \times v_2)$. We have four products of n/2-bit numbers.

Fast Multiplication

- $(u_1 \times u_2)2^n + [(u_1 \times v_2) + (v_1 \times u_2)]2^{n/2} + (v_1 \times v_2)$ has four products of n/2-bit numbers.
- But if we compute $u_1 \times u_2$, $v_1 \times v_2$, and $(u_1+v_1) \times (u_2+v_2)$, using only three multiplications, we can get all three terms we need by addition.
- Our number of multiplications satisfies the recurrence $a_n = 3a_{n/2}$, which turns out to solve to $a_n = n^{\log 3} = n^{1.585\ldots}$, much better than n^2. Of course there are complications like the time for the additions.

The CLRS Master Theorem

- In COMPSCI 311 we learn a theorem called the Master Theorem in the popular CLRS textbook. It gives a solution to the recurrence $a_n = ca_{n/k} + f(n)$, which applies when we divide the size-n problem into c problems of size n/k each, with f(n) overhead to split the problems and merge the solutions.
- The solutions are given in big-O form, befitting a course where we usually regard resource bounds this way.

The CLRS Master Theorem

- We have $a_n = ca_{n/k} + f(n)$.
- The result depends on the relationship between $f(n)$ and $g(n) = n^{\log c}$, where the log is base k. The statement below is approximate.
- If $f(n) = o(g(n))$, then $a_n = \Theta(g(n))$.
- If $f(n) = \Theta(g(n))$, then $a_n = \Theta(g(n)\log n)$.
- If $f(n) = \omega(g(n))$, then $a_n = \Theta(f(n))$.

Linear Recurrences

- A **linear recurrence** is one where the new term a_n is given by a linear combination of the r most recent terms, by a rule of the form $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_r a_{n-r}$.
- Since a_k is not defined for negative k, we have to give **initial conditions** $a_1, \ldots, a_{r-1}$ as well as the usual a_0.
- There's a general solution for these, which is reminiscent of the general solution for linear differential equations.

Solving Linear Recurrences

- It turns out that every such equation has a set of solutions that are themselves linear combinations of sequences of the form α^n, for some fixed numbers α.
- If α is going to lead to such a solution, we need to have $\alpha^n = c_1\alpha^{n-1} + \ldots + c_r\alpha^{n-r}$, which we can reduce to $\alpha^r = c_1\alpha^{r-1} + \ldots + c_r$, by dividing the first equation by $\alpha^{n-r.}$

Solving Linear Recurrences

- So α must satisfy the equation $\alpha^r - c_1\alpha^{r-1} - c_2\alpha^{n-2} - \ldots - c_r = 0$, which is called the **characteristic equation** of the recurrence.
- Over the complex numbers, at least, this equation of degree r has exactly r roots, counting multiplicity. (We'll assume for the time being that all the roots are distinct.)
- If $\alpha_1,\ldots,\alpha_r$ are the roots, any function of the form $A_1\alpha_1^n + \ldots + A_r\alpha_r^n$ will be a solution to the recurrence, and these are all of them.

Dealing With Initial Conditions

- More precisely, any linear combination of the functions α_i^n will satisfy the rule of the recurrence. In order to satisfy the initial conditions as well, we need to set the A_i's.
- If $a_0', \ldots, a_{r-1}'$ are the values of $a_0, \ldots, a_{r-1}$ given by the initial conditions, then for every k with $0 \leq k \leq r-1$, we must have $A_1\alpha_1^k + A_2\alpha_2^k + \ldots + A_r\alpha_r^k = a_k'$.
- These are r equations in r unknowns, and have exactly one solution.

Example: Exponential Rabbits

- If we start at time 0 with six rabbits, and the population doubles each year, how many do we have after n years?
- The recurrence is $a_n = 2a_{n-1}$, with initial condition $a_0 = 6$. (Since $r = 1$ here, we need only one initial condition.) The characteristic equation is $\alpha^1 - 2 = 0$, solving to $\alpha = 2$.
- So any function of the form $A2^n$ meets the rule, and to have $A2^0 = 6$, we choose $A = 6$.

Example: Fibonacci Rabbits

- The Fibonacci function was also originally designed to model rabbit populations, with each rabbit producing one offspring in every generation except its first. We don't model any rabbit deaths.
- So the population a_n after n generations is the a_{n-1} from the previous generation, plus one more for each of the a_{n-2} rabbits that are more than one generation old.

Example: Fibonacci Rabbits

- So the characteristic equation is $\alpha^2 - \alpha - 1 = 0$, which by the quadratic formula has two roots, $\alpha_1 = (1+\sqrt{5})/2$ and $\alpha_2 = (1-\sqrt{5})/2$.
- Any function of the form $A_1\alpha_1^n + A_2\alpha_2^n$ follows the recursive rule. Solving the pair of equations $A_1 + A_2 = a_0' = 1$ and $A_1\alpha_1 + A_2\alpha_2 = a_1' = 1$ gives us $A_1 = \alpha_1/\sqrt{5}$ and $A_2 = -\alpha_2/\sqrt{5}$.
- It's perhaps surprising that these irrational coefficients and bases of powers give us the familiar sequence 1, 1, 2, 3, 5, 8, 13, 21,…

Complex Roots

- How would we get complex numbers in the solution to a linear recurrence? Let $a_n = -a_{n-1} - a_{n-2}$, with initial conditions $a_0 = 0$ and $a_1 = 1$.
- We get a sequence 0, 1, -1, 0, 1, -1, 0, 1, -1,…, which doesn't look exponential at all.
- But in fact the characteristic equation $\alpha^2 + \alpha + 1 = 0$ has two roots $(-1+\sqrt{-3})/2$, the two complex cube roots of unity. The correct linear combination of powers of these gives the real numbers of our periodic sequence.

Multiple Roots

- What about a rule like $a_n = 4a_{n-1} - 4a_{n-2}$, with characteristic equation $\alpha^2 - 4\alpha + 4 = 0$, which has a double root of $\alpha = 2$?
- If we start with $a_0 = 0$ and $a_1 = 1$, the sequence goes on 4, 12, 32, 80, 192, which we might recognize as $a_n = n2^{n-1}$. Where did this come from?
- The function 2^n satisfies the rule for the recurrence, but it turns out that $n2^n$ does as well, as $n2^n = 4(n-1)2^{n-1} - 4(n-2)2^{n-2}$.

Multiple Roots

- If α is a root of the characteristic equation with multiplicity m, then it turns out that $n\alpha^n$, $n^2\alpha^n$,…, $n^{m-1}\alpha^n$ all satisfy the rule, and any function that satisfies the rule is a linear combination of these functions and α^n itself.
- We won't prove this here, but on HW#6 you'll show the m = 3 case (Exercise 7.3.9, not in the back of the book).
- Note that for a characteristic equation of degree r, we still have exactly r functions, so that there will be one linear combination meeting the initial conditions.

Inhomogeneous Linears

- For example, consider a recurrence of degree 1, so that $a_n = ca_{n-1} + f(n)$. The h-part is "ca_{n-1}" and the i-part is $f(n)$.
- If we can find any function z such that $z_n = cz_{n-1} + f(n)$, then any function of the form $a_n = Ac^n + z_n$ will satisfy the recurrence, and we can use the initial conditions to find A as before.
- A special case is when $c = 1$, so that $a_n = a_{n-1} + f(n)$. This has the solution $z_n = f(1) + \ldots + f(n)$, so that $a_n = a_0 + f(1) + f(2) + \ldots + f(n)$.

Inhomogeneous Linears

- To solve $a_n = ca_{n-1} + f(n)$ with $c \neq 1$, we can use some standard general solutions, that we will justify with generating functions next lecture.
- If f(n) is a constant d, the particular solution is another constant B.
- If $f(n) = dn$, we have $B_1n + B_0$ for two constants B_0 and B_1.
- For $f(n) = dn^2$ we have $B_2n^2 + B_1n + B_0$, and for $f(n) = ed^n$ we have Bd^n.

Compound Inhomogeneous

- Example 3 in Section 7.4 has the recurrence $a_n = 3a_{n-1} - 4n + 3 \cdot 2^n$, and asks for a general solution.
- The difficulty here is that the i-part is the sum of two functions, but we can proceed by finding particular solutions for each of the two functions, and the adding them to $A3^n$.
- To get $y_n = 3y_{n-1} - 4n$, we look for a solution of the form $B_1 n + B_0$, and get $2n + 3$ by taking the equation $B_1 n + B_0 = 3(B_1 n + B_0) - 4n$ and solving for B_0 and B_1.

Compound Inhomogeneous

- Example 3 in Section 7.4 has the recurrence $a_n = 3a_{n-1} - 4n + 3 \cdot 2^n$, and asks for a general solution.
- To get $y_n = 3y_{n-1} + 3(2^n)$, we look for a solution of the form $B2^n$, and get $6(2^n)$ by taking the equation $B2^n = 3B2^{n-1} + 3(2^n)$ and solving for B.
- Our general solution is $a_n = A3^n + 2n + 3 + 6(2^n)$.