Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr

WS 1998/99
Information 3
11. November 1998

Effiziente Algorithmen und Datenstrukturen

Some Hints for Solving Recurrences

When designing an algorithm for some problem it often happens that we can derive a solution for any instance with, say, relatively little effort if we only have the solutions to a few other, smaller instances of the same problem, or of instances of some other problem which we know how to solve. In the second case it is immediate to determine the complexity of an algorithm for the first problem which uses this reduction, from the known complexity of the second algorithm and the additional cost for the reduction. When we wish to determine the complexity of an algorithm constructed according to the first case, however, we usually end up with a relation of the following form for the complexity C of our algorithm:

$$C(n) = F_1(C(n^{(1)}), \ldots, C(n^{(i)})) + F_2(n),$$

where F_1 describes the dependency on the complexities of the smaller problem instances of size $n^{(1)}, \ldots, n^{(i)}$, and F_2 represents the additional cost incurred to form a solution for the original problem instance of size n. We shall now discuss some techniques for solving such recurrence relations, i.e., finding closed-form expressions for the function C when the function F_1 is sufficiently simple. Nonetheless, these methods provide a systematic way to deal with a fairly large class of recurrence equations (also see [Lue80]). The techniques are, in turn, *multipliers*, *characteristic polynomials*, *generating functions*, and *domain* and *range transformations*. We shall demonstrate each technique by means of an example where, for notational convenience, we write f_n for $f(n)$.

0.4.1 Multipliers

Suppose that the following recurrence relation is given:

$$\begin{aligned} f_n &= 2f_{n-1} + n\,, \qquad \text{for } n \geq 2, \\ f_1 &= 1\,. \end{aligned}$$

That is

$$\begin{array}{rcll} f_n &=& 2f_{n-1} + n\,, & / \cdot 1 \\ f_{n-1} &=& 2f_{n-2} + (n-1)\,, & / \cdot 2 \\ f_{n-2} &=& 2f_{n-3} + (n-2)\,, & / \cdot 4 \\ & \vdots & & \\ f_{n-i} &=& 2f_{n-i-1} + (n-i)\,, & / \cdot 2^i \\ & \vdots & & \\ f_2 &=& 2f_1 + 2\,. & / \cdot 2^{n-2} \end{array}$$

If we multiply each equation with the multiplier indicated to its right, then sum up all equations and cancel identical terms on both sides, we obtain

$$\begin{aligned}
f_n &= 2^{n-1} f_1 + \sum_{i=0}^{n-2} (n-i) 2^i \\
&= \sum_{i=0}^{n} (n-i) 2^i \qquad\qquad \text{because } f_1 = 1 \\
&= \sum_{i=0}^{n-1} \sum_{j=0}^{i} 2^j = 2^{n+1} - n - 2\,.
\end{aligned}$$

The last line follows by elementary transformations (see also below). Hence we obtain

$$f_n = 2^{n+1} - n - 2\,.$$

0.4.2 Characteristic Polynomials

As an example, suppose we are given

$$\begin{aligned}
f_n &= f_{n-1} + f_{n-2}\,; \qquad\qquad \text{for } n \geq 2 \\
f_1 &= 1, \quad f_0 = 0\,.
\end{aligned}$$

Such a recurrence is called a *homogeneous linear recurrence with constant coefficients.*
If we try to set $f_n = a^n$ for some a still to be determined we get the following *characteristic equation* for a (assuming that $a \neq 0$):

$$a^2 - a - 1 = 0, \text{ and hence } a_{1,2} = \frac{1 \pm \sqrt{5}}{2}\,.$$

It is clear that every linear combination

$$c_1 a_1^n + c_2 a_2^n$$

is then a solution to the recurrence relation $f_n - f_{n-1} - f_{n-2} = 0$ for $n \geq 2$. The boundary conditions for f_0 and f_1 determine specific values for c_1 and c_2, in our example $c_1 = 1/\sqrt{5} = -c_2$, and hence

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

The question is whether by this method we can obtain all solutions to recurrences like the one above. It is not hard to see that this is true if all roots of the characteristic polynomial are distinct. For the general case, with multiple roots allowed, we cite the following theorem (for a proof see [Liu68]):

Theorem 1 *Let $p(x)$ be the characteristic polynomial of the recurrence*
(*) $$p_0 f_n + p_1 f_{n-1} + \ldots + p_k f_{n-k} = 0\,; \qquad \text{for } n \geq k$$

with constant coefficients p_i. Let the roots of p, over the complex numbers, be r_i, $i = 1, \ldots, m$, each with multiplicity q_i. Then all solutions of (*) are given by

$$f_n = \sum_{i=1}^{m} (r_i^n \sum_{j=0}^{q_i - 1} c_{ij} n^j)\,,$$

where the c_{ij} are arbitrary constants.

If the given linear recurrence relation (with constant coefficients) is *inhomogeneous* we first try to derive a homogeneous recurrence using appropriate multipliers, and then apply the method discussed above. A more formal treatment can be found in [Lue80].

0.4.3 Generating Functions

For a sequence $(f_n)_{n\geq 0}$, its *generating function* $F(z)$ is defined as

$$F(z) = \sum_{n\geq 0} f_n z^n\,.$$

Just using calculus, we can establish the following table:

generating function	n-th element of the corresponding sequence
cF	cf_n
$F + G$	$f_n + g_n$
$F \cdot G$	$\sum_{i=0}^{n} f_i g_{n-i}$
$z^k F$	**if** $n < k$ **then** 0 **else** f_{n-k}
$F(z)/(1-z)$	$\sum_{i=0}^{n} f_i$
$z\frac{dF}{dz}(z)$	nf_n
$\int_0^z F(t)dt$	**if** $n = 0$ **then** 0 **else** f_{n-1}/n
$F(cz)$	$c^n f_n$

Hence, if we put

$$\begin{aligned} F(z) &= \sum_{n\geq 0} 2^n z^n = \frac{1}{1-2z}\,; \\ G(z) &= \sum_{n\geq 0} n z^n = \frac{z}{(1-z)^2}\,, \end{aligned}$$

we obtain

$$F(z)\cdot G(z) = \frac{z}{(1-z)^2(1-2z)} = \sum_{n\geq 0}\sum_{i=0}^{n}(n-i)2^i z^n\,.$$

Decomposing $z/((1-z)^2(1-2z))$ into partial fractions we get

$$\frac{z}{(1-z)^2(1-2z)} = \frac{-2}{1-z} - \frac{z}{(1-z)^2} + \frac{2}{1-2z}\,,$$

and thus the table above provides the answer

$$\sum_{i=0}^{n}(n-i)2^i = 2^{n+1} - n - 2\,.$$

0.4.4 Domain and Range Transformations

A real-valued sequence $(f_n)_{n\geq 0}$ is a mapping from $\mathbf{N}$ to the reals. Thus a transformation on the values of the sequence is called a *range transformation* and a transformation on the indices a *domain transformation*. We first show an example of a range transformation. Suppose we are to solve the recurrence

$$\begin{aligned} f_n &= f_{n-1}\cdot f_{n-2}, \qquad \text{for } n\geq 2\,,\\ f_1 &= 2, \quad f_0 = 1\,. \end{aligned}$$

If we let

$$g_n = \log f_n\,,$$

we may rewrite the recurrence as

$$\begin{aligned} g_n &= g_{n-1} + g_{n-2}, \qquad \text{for } n\geq 2\,,\\ g_1 &= 1, \quad g_0 = 0\,. \end{aligned}$$

For this recurrence, we know the solution (see above; g_n is the n-th *Fibonacci* number F_n), and substituting back yields

$$f_n = 2^{F_n}\,.$$

A domain transformation is conveniently used in the following example. Let

$$\begin{aligned} f_n &= 3f_{n/2} + n, \qquad \text{for } n = 2^k > 1\,,\\ f_1 &= 1\,. \end{aligned}$$

Here we substitute the index and define

$$g_k = f_{2^k}\,,$$

from which we get the following secondary recurrence for the g_k:

$$\begin{aligned} g_k &= 3g_{k-1} + 2^k, \qquad \text{for } k\geq 1\,,\\ g_0 &= 1\,, \end{aligned}$$

which can be solved with methods discussed earlier. We obtain $g_k = 3^{k+1} - 2^{k+1}$, and therefore

$$f_n = 3n^{\log 3} - 2n\,.$$

We finally give a closed form solution for a large class of very common recurrence relations. We leave the straightforward proof of the formulas as an easy exercise to the reader.

Let the function T be defined by the following recurrence:

$$\begin{aligned} T(n) &= sT(\frac{n}{m}) + an^e, \qquad \text{if } n = m^k > 1\,, \\ T(1) &= d \end{aligned}$$

(where $a > 0$, $e \geq 0$, $s \geq 1$, and m an integer ≥ 2). Then

$$T(n) = \begin{cases} \Theta(n^e), & \text{if } s < m^e \\ \Theta(n^e \log_m n), & \text{if } s = m^e \\ \Theta(n^{\log_m s}), & \text{if } s > m^e. \end{cases}$$

For a proof, assume first that n is a power of m.
This concludes our discussion of several very useful techniques to solve recurrence relations for sequences. Additional material can be found in [Mil60,Rei77,Rio58,Rio68], among many other references.

References:

[AHU74] Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading, Mass., 1974

[Knu76] Knuth, D.E.: Big Omicron and Big Omega and Big Theta. ACM Sigact News **8**, 2 (1976), p. 18

[Lue80] Lueker, G.S.: Some Techniques for Solving Recurrences. ACM Computing Surveys **12**, 4 (1980), p. 419–436

[Liu68] Liu, C.L.: Introduction to Combinatorial Mathematics. McGraw-Hill, New York, N.Y., 1968

[Mil60] Milne-Thomson, L.M.: The Calculus of Finite Differences. Macmillan, London, 1960

[Rei77] Reingold, E.M., Nievergelt, J., Deo, N.: Combinatorial Algorithms: Theory and Practice. Prentice-Hall, Englewood Cliffs, N.J., 1977

[Rio58] Riordan, J.: An Introduction to Combinatorial Analysis. J. Wiley, New York, N.Y., 1958

[Rio68] Riordan, J.: Combinatorial Identities. J. Wiley, New York, N.Y., 1968

[Ver94] Verma, R.M.: A general method and a master theorem for divide-and-conquer recurrences with applications. J. of Algorithms **16** (1994), p. 67–79

[Ver97] Verma, R.M.: General techniques for analyzing recursive algorithms with applications. SIAM J. Comput. **26** (1997), p. 568–581