8

Horner's Algorithm

Horner's algorithm is an excellent example of an algorithm. It is a simple and useful algorithm, but it does not have much to do with discrete mathematics. However, it is inherently interesting enough to justify inclusion here.

The purpose of Horner's algorithm is to efficiently evaluate polynomials. The rationale of Horner's algorithm is quite simple. Suppose, for example, that we want to evaluate the polynomial $p(x) = 4x^5 - 3x^4 + 7x^3 + 6x^2 + 3x + 9$ at $x = 3$. The usual method of evaluation is to evaluate each product (such as $4{\cdot}3^5$ or $7{\cdot}3^3$) separately and then add. The drawback is that to evaluate any power of x, we go through all of the previous powers. Clearly, a simpler approach would be to compile a list of the powers of x, computed recursively, $x^n = x{\cdot}x^{n-1}$, and then use them. This is illustrated in **Figure 1**.

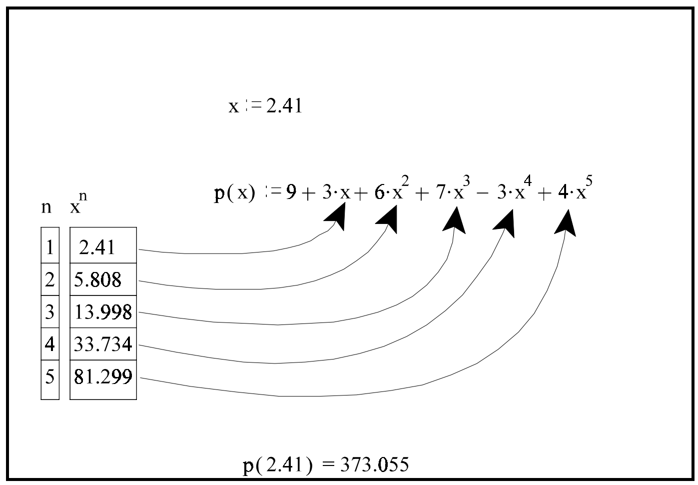

Figure 1 Evaluation of p(x) at x = 2.41

Horner's algorithm is even more efficient than the method just advocated. Let us denote a polynomial of degree n by the following notation: $p(X) = C_nX^n + C_{n-1}X^{n-1} + ... + C_1X + C_0$.

The purpose of the algorithm is to compute p(α) where α is a constant. The algorithm works as follows:

Horner's Algorithm

1. Set u ← n (where n is the degree of the polynomial).
2. Set Result ← C_n.
3. If u = 0 stop. Answer is Result.
4. Compute Result ← Result × α + C_{u-1}
5. u ← u -1.
6. Go to step 3.

In the above instance of computing $p(x) = 4x^5 - 3x^4 + 7x^3 + 6x^2 + 3x + 9$ at x = 2.41, the algorithm proceeds like this:

- u ← 5
- Result ← 4
- Result ← 4×2.41 + (-3) = 6.64
- u ← 4
- Result ← 6.64×2.41 + 7 = 23.0024
- u ← 3
- Result ← 23.0024×2.41 + 6 = 61.435784
- u ← 2
- Result ← 61.435784×2.41 + 3 = 151.0602394
- u ← 1
- Result ← 151.0602394×2.41 + 9 = 373.0551770
- u ← 0
- Stop. Answer is 373.0551770

Note that Horner's algorithm only requires twice as long to evaluate a 20'th degree polynomial as a 10'th degree polynomial. Another way of motivating Horner's Algorithm is to rewrite the polynomial $p(X) = C_nX^n + C_{n-1}X^{n-1} + ...$ as $(...((C_nX + C_{n-1})X + C_{n-2})X....)X + C_0$. The above example, $p(x) = 4x^5 - 3x^4 + 7x^3 + 6x^2 + 3x + 9$ can be rewritten as ((((4x-3)x+7)x+6)x+3)x+9.

☐ **Exercise 1** Write Horner's algorithm without a *go-to* statement.

1. Horner's Algorithm

Set u ← n (where n is the degree of the polynomial).
Set Result ← C_n.
While u ≠ 0
  u ← u -1
  Compute Result ← Result × α + C_u
Answer = Result.

Note that in this version, the computation u ← u -1 has been moved and that changes the form of the main operation.