Efficient Way of evaluating Polynomials

Suppose $P(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$
then we can rewrite as

n additions
n multiplications
n-2 exponentiations

$$P(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + \dots a_n x)))\dots)$$

This has n additions and n multiplications.

Example $P_3(x) = x^3 - 4x^2 + 5x - 2$

$$P_3(x) = -2 + x(5 + x(-4 + x(1)))$$

BTW, this is equivalent to "synthetic division"

Divided Difference Form

$$P(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-x_2) + \dots + a_n(x-x_0)(x-x_1)\dots(x-x_{n-1}).$$

We can rewrite in a similar way

$$P(x) = a_0 + (x-x_0)(a_1 + (x-x_1)(a_2 + (x-x_2)(a_3 + \dots (x-x_{n-1})(a_n)))\dots)$$

Input: $\vec{a}$ - vector of coefficients (length n+1)
$\vec{x}$ - vector of x data points (length n)
x - value

Output: $P(x)$

```
Set  temp = a(n+1)
for  j = n, n-1, n-2, ..., 1
    Set  temp = a(j) + (x - x(j)) * temp
end
Output  temp
```