Avaliação de um polinômio $p_n(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$

Entrada: vetor a = [a0,a1,a2,...,an] de inteiros (* coeficientes do polinômio *);

n: inteiro (* grau do polinômio *); x: inteiro (* argumento do polinômio *)

Primeiro método: Potenciação

Utiliza a função potência: x^i

Algoritmo Pot;

Início

P <- a0;

Para i <- 1 até n Faça

P <- p+(ai.x^i) ;

Fim.

Considerando a complexidade da função potência x^i : i-1 multiplicações :

$T(n) = 2n + 1 + \sum_{i=1,\dots n} (i+1)$

$T(n) = 2n + 1 + n + (n.(n+1))/2$

Portanto T(n) é $O(n^2)$.

Segundo método: Produto

Armazena o valor da potência em uma variável auxiliar realizando somente um produto por iteração

Algoritmo Prod;

Início

P <- a0; y <- x; z <- n-1

Para i <- 1 até z Faça

P <- p + ai.y;   y <- y.x;

Fim_Para;

p <- p + an.y;

Fim.

T(n) = 1 + (2n – 1) + 3(n-1) + 2 = 5n – 1

Portanto T(n) é O(n).

Terceiro método: Horner

Considera a forma do polinômio: $p_n(x) = (((a_nx + a_{n-1})x + ... + a_2).x + a_1)x + a_0$

Pode ser implementado recursivamente.

Algoritmo Horner;

Inicio

p <- an; Z <- n-1;

Para i <- z até 0 Faça

P <- ai + x.p;

Fim.

T(n) = 1 + (2n + 1) + 2.n = 4n + 2

Portanto T(n) é O(n).

Qual método é melhor ?

Fazendo:

5n -1 = 4n + 2 => n0 = 3.

Portanto para n ≥ 3 o método Horner domina assintoticamente o método Prod.