

## ¿Que es el algoritmo de Horner?

El algoritmo de Horner es un algoritmo reconocido por su eficiencia para evaluar polinomios utilizando un número mínimo de operaciones (sumas y productos). Aunque la solución de un polinomio para un valor específico de  $x$  es una tarea sencilla el algoritmo reduce la cantidad de operaciones necesarias para llegar al resultado lo que la convierte en una técnica más eficiente y más deseable a la hora de programarla.

## ¿Como funciona?

Supongamos que se tiene el polinomio

$$2 + 4x - 5x^2 + 2x^3 - 6x^4 + 8x^5 + 10x^6$$

Primero notemos que, dado  $x$ , el método usual para evaluar  $x^k$  requiere  $k$  productos:

$$x^k = x \cdot x \cdot \dots \cdot x$$

Ahora bien, si separamos el polinomio en términos de la forma

$$2 + x(4 + x(-5 + x(2 + x(-6 + x(8 + 10x))))$$

Es facil ver que ahora se requieren 6 sumas y 6 productos para evaluar el polinomio en  $x$ .

Si el grado de  $p(x)$  es  $n$ , con esté método se requieren  $n$  productos.

Si  $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ , con  $a_n \neq 0$ , la evaluación de  $p(x)$  en  $x = z$  se realiza con los siguientes pasos:

$$\begin{aligned} b_{n-1} &= a_n \\ b_{n-2} &= a_{n-1} + z \cdot b_{n-1} \\ b_{n-3} &= a_{n-2} + z \cdot b_{n-2} \\ &\vdots \\ b_0 &= a_1 + z \cdot b_1 \\ p(z) &= a_0 + z \cdot b_0 \end{aligned}$$

## Ejemplo

Supongamos que se quiere evaluar el polinomio  $p(x) = 2 + 4x - 5x^2 + 2x^3$  en  $x = 3$ .

Entonces, se tiene que

$$\begin{aligned} b_3 &= 2 \\ b_2 &= -5 + 3 \cdot 2 = 1 \\ b_1 &= 4 + 3 \cdot 1 = 7 \\ p(3) &= 2 + 3 \cdot 7 = 23 \end{aligned}$$

## Implementación en Pseudo-Código

**input**  $n; a_i, i = 0, \dots, n; z$

$b_{n-1} \leftarrow a_n$

**for**  $k = n - 1$  **to**  $0$  **step**  $-1$ , **do**

$$b_{k-1} \leftarrow a_k + z * b_k$$

**end do**

**output**  $b_i, i = -1, \dots, n - 1$

**end**

Notar que  $b_{-1} = p(z)$

## Ventajas

- Reduce la cantidad de operaciones necesarias para evaluar un polinomio.
- Es más eficiente que el método usual.
- Es más fácil de programar.

## Implementación en Python

En python el código para evaluar un polinomio en un valor específico de  $x$  es el siguiente:

```
#Valor de x
valorX=8;
#Coeficientes del polinomio
coeficientes=[4,7,3,6,2];
resultado=0;
#Recorrer los coeficientes
for i in range(0,len(coeficientes)):
    #Multiplicar al valor parcial el valor de x más el coeficiente
    resultado= resultado * valorX + coeficientes[i]
print("Resultado:"+str(resultado))
```

Fig. 1: Algoritmo de Horner Python.

## Implementación en C

En C el código para evaluar un polinomio en un valor específico de  $x$  es el siguiente:

```
#include<stdio.h>

int main(int argc, char** argv){

    //Valor de x
    float valorX=8;
    //Coeficientes del polinomio
    float coeficientes[]={4,7,3,6,2};
    float resultado=0;
    int i;

    //Recorrer los coeficientes
    for(i=0;i<(sizeof(coeficientes)/sizeof(float));i++){
        //Multiplicar al valor parcial el valor de x más el coeficiente
        resultado= resultado * valorX + coeficientes[i];
    }

    printf("Resultado %f\n",resultado);
    return 0;
}
```

Fig. 2: Algoritmo de Horner C.