

Computación

Aula Virtual: <https://famaf.aulavirtual.unc.edu.ar/course/view.php?id=747>

Resguardo tutoriales: <https://www.famaf.unc.edu.ar/~moreschi/docencia/Computacion/>

Tutorial Problema 1 de la Guía N° 3

Problema 1:

Un polinomio de grado n se puede expresar por:

$$p(x) = c_0 + c_1x + c_2x^2 + \dots + c_nx^n.$$

Definiendo en PYTHON a la lista c por:

$$c = [c_0, c_1, c_2, \dots, c_n]$$

confeccione un programa, en un archivo, que calcule el polinomio para un valor de x , por medio de dos funciones que tengan como argumento c y x . Una de ellas que realice el método usual y la otra función que utilice el método de Horner.

Defina a c por:

```
c=[]
for i in range(100):
c.append(np.sin((100 - i)*2*np.pi/100))
```

y evalúe el polinomio para $x = 7$ y $x = 0.7$.

Tome el tiempo para cada evaluación, e imprima el tiempo de ejecución para cada evaluación del polinomio en pantalla.

Tutorial:

- Guarde en el archivo p1-g3.py las siguientes instrucciones:

```
1 import numpy as np
2 from timeit import default_timer as timer
3
4 # Usamos las funciones que fueron presentadas en el teórico
5
6 def evalpol(c,x):
7     """ Función para evaluar un polinomio.
8     Para un polinomio p(x)= c_n x**n+c_{n-1} x**{n-1}+ ...+c_0
9     c=[c_0, c_1, ..., c_n], vector de dimensión n+1 """
10    p = 0
11    n = len(c)
12    for i in range(n):
13        p = p + c[i] * x**i
14    return p
15
16 def horner(c,x):
```

```

17     """ Función que implementa el método de Horner """
18     n = len(c) - 1
19     p = c[n]
20     for i in reversed(range(n)):
21         p = p * x + c[i]
22     return p
23
24 # definimos la lista de coeficientes
25 c=[]
26 for i in range(100):
27     c.append(np.sin((100 - i)*2*np.pi/100))
28
29 print()
30 # -----
31 # calculamos con la fórmula usual y tomamos el tiempo
32 start= timer()
33 pol = evalpol(c,7)
34 print(' pol(7) =',pol)
35 end = timer()
36
37 print("    Tiempo de ejecución usando fórmula usual:", end-start)
38 print()
39 # -----
40 # calculamos con la fórmula usual y tomamos el tiempo
41 start= timer()
42 pol = horner(c,7)
43 print(' pol(7) =',pol)
44 end = timer()
45
46 print("Tiempo de ejecución usando fórmula de Horner:", end-start)
47 print()
48
49 # -----
50 # calculamos con la fórmula usual y tomamos el tiempo
51 start= timer()
52 pol = evalpol(c,0.7)
53 print(' pol(0.7) =',pol)
54 end = timer()
55
56 print("    Tiempo de ejecución usando fórmula usual:", end-start)
57 print()
58 # -----
59 # calculamos con la fórmula usual y tomamos el tiempo
60 start= timer()
61 pol = horner(c,0.7)
62 print(' pol(0.7) =',pol)
63 end = timer()
64
65 print("Tiempo de ejecución usando fórmula de Horner:", end-start)
66 print()

```

- Desde la terminal ejecute:
python3 p1-g3.py
e interprete el resultado.