Algo DS
26.03.2022

Horner-Schema

$$p(x) = 3{,}1x^4 + 5x^3 - \frac{1}{4}x^2 + x - 9$$

$$p(7) = 3{,}1 \cdot \underbrace{7 \cdot 7 \cdot 7 \cdot 7}_{\text{4 Faktoren}} + 5 \cdot \underbrace{7 \cdot 7 \cdot 7}_{\text{3 Fakt.}} - \frac{1}{4} \cdot \underbrace{7 \cdot 7}_{2} + 1 \cdot \underbrace{7}_{1} - 9$$

$n = \text{Grad}(p) = 4$

Multiplikationen: $n + (n-1) + \ldots + 2 + 1 = \frac{n \cdot (n+1)}{2} = \Theta(n^2)$

$$p(x) = \Big(\big((3{,}1x + 5)x + (-\tfrac{1}{4})\big)x + 1\Big)x - 9$$

4 Multiplik.

Multipl.: $n = \Theta(n)$

- Taylor-Reihe $\quad f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k$

$\sin x$: $f(x) = \sin x$ $\quad f'(x) = \cos x$ $\quad f''(x) = -\sin x$ $\quad f'''(x) = -\cos x$

$x = 0$: $f(0) = 0$ $\quad f'(0) = 1$ $\quad f''(0) = 0$ $\quad f'''(x) = -1$

$$\sin x = \frac{0}{0!} + \frac{1}{1!}x + \frac{0}{2!}x^2 - \frac{1}{3!}x^3 + \frac{0}{4!}x^4 + \frac{1}{5!}x^5 + \frac{0}{6!}x^6 - \frac{1}{7!}x^7 + \ldots$$

$$= x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \ldots$$

```
# Algo+DS, Aufgabe 3.2
# Polynome, Horner-Schema
#
# v0.2, time() durch process_time() ersetzt

from random import random
# from time import time
from time import process_time

def potenz (x,n):
    prod = 1
    for k in range(n):
        prod = prod*x
    return prod

def polynom_einfach (pol, x):
    summe = 0
    grad = len(pol)-1
    for k in range(grad+1):
        summe = summe + pol[k] * potenz(x,k)
    return summe

# Horner-Schema, wie in Cormen, S. 41
def polynom (pol, x):
    summe = 0
    grad = len(pol)-1
    for i in range(grad, -1, -1):
        summe = pol[i] + x * summe
    return summe



pol1 = [9.0, -0.33, 2.0, 1.5]

print ("Polynom: p1 =", pol1)
print ("p1(2) =", polynom(pol1, 2))


# Erzeuge grosses zufaelliges Polynom
pol2 = [random() for _ in range(10000)]

print ("Polynom: p2 =", pol2[:10], "...")

old_start = process_time()
ergebnis_old = polynom_einfach(pol2, 1.001)
old_zeit = process_time() - old_start

start = process_time()
ergebnis = polynom(pol2, 1.001)
zeit = process_time() - start

print ("einfach: p2(1.001) = %12.4f, Zeit = %10.8f" % (ergebnis_old, old_zeit))
print ("Horner:  p2(1.001) = %12.4f, Zeit = %10.8f" % (ergebnis,     zeit))
```

```
from math import factorial, sin

ANZAHL_KOEFFIZIENTEN = 100

koeff = []
for i in range(ANZAHL_KOEFFIZIENTEN):
    if i % 2 == 0:
        koeff.append(0)
    elif i % 4 == 1:
        koeff.append (1/factorial(i))
    else:
        koeff.append (-1/factorial(i))

def sin1 (x):
    sum = 0.0
    for k in range(ANZAHL_KOEFFIZIENTEN):
        sum += x**k * koeff[k]
        # print ("Addiere %100.95f" % (x**k / koeff[k]))
    return sum


for i in range(20):
    x  = (i-10)/1
    y  = sin(x)
    y1 = sin1(x)
    # print ("%5.2f %20.15f %20.15f %25.20f" % (x,y,y1,y-y1))
    print ("x=%8.4f sin1(x)=%10.5f Error: %30.20f" % (x,y1,abs(y-y1)))
```