EX.3.1.1.a, Sauer3

Use Lagrange interpolation to find a polynomial that passes through the points

a. $(0, 1)$, $(2, 3)$, $(3, 0)$.

EX.3.1.1.a, Sauer3, solution, Langou

Colab: `https://colab.research.google.com/drive/1PTbCPu8PGH-2AtLUg6vfdEgAJ_aKTOrt`

For the points $(0, 1)$, $(2, 3)$, $(3, 0)$, we have

$$\begin{aligned}
\ell_1(x) &= \frac{(x-2)(x-3)}{(0-2)(0-3)} = \frac{1}{6}(x-2)(x-3); \\
\ell_2(x) &= \frac{(x-0)(x-3)}{(2-0)(2-3)} = -\frac{1}{2}x(x-3). \\
\ell_3(x) &= \frac{(x-0)(x-2)}{(3-0)(3-2)} = \frac{1}{3}x(x-2);
\end{aligned}$$

This leads to

$$\begin{aligned}
p(x) &= 1 \cdot \ell_1(x) + 3 \cdot \ell_2(x) + 0 \cdot \ell_3(x) \\
&= \frac{1}{6}(x-2)(x-3) - 3\frac{1}{2}x(x-3) \\
&= \frac{1}{6}(x^2 - 5x + 6 - 9x^2 + 27x) \\
&= \frac{1}{6}(-8x^2 + 22x + 6) \\
&= -\frac{4}{3}x^2 + \frac{11}{3}x + 1
\end{aligned}$$

$$\boxed{p(x) = -\tfrac{4}{3}x^2 + \tfrac{11}{3}x + 1}$$

What is below is not needed for full credit

We can check that

$$\begin{aligned}
p(0) &= -\tfrac{4}{3}(0)^2 + \tfrac{11}{3}(0) + 1 = 1 & \checkmark \\
p(2) &= -\tfrac{4}{3}(2)^2 + \tfrac{11}{3}(2) + 1 = -\tfrac{16}{3} + \tfrac{22}{3} + 1 = 3 & \checkmark \\
p(3) &= -\tfrac{4}{3}(3)^2 + \tfrac{11}{3}(3) + 1 = -12 + 11 + 1 = 0 & \checkmark
\end{aligned}$$

What is below is not needed for full credit

```python
import numpy as np
x = np.array([0., 2., 3.])
y = np.array([1., 3., 0.])
```

```
# use np.polynomial.polynomial.polyfit to get the answer
# remember that `numpy` prints the coefficients from degree 0
# to degree 2
c = np.polynomial.polynomial.polyfit(x,y,2)
print(c)

# check that the answer is correct, so evaluate p at x and we check that
# we get y back.
yy = np.polynomial.polynomial.polyval(x,c)
print(yy)
```

```
[ 1.          3.66666667 -1.33333333]
[ 1.00000000e+00  3.00000000e+00 -1.55431223e-15]
```

```
# enter the Lagrange polynomials l1, l2, and l3
l1 = lambda x : (x-2.)*(x-3.)/6.
l2 = lambda x : -x*(x-3.)/2.
l3 = lambda x : x*(x-2.)/3.

# check that l1(0)=1 and l1(2)=0 and l1(3)=0
# check that l2(0)=0 and l2(2)=1 and l2(3)=0
# check that l3(0)=1 and l3(2)=0 and l3(3)=1
print("l1( [ 0, 2, 3 ] ) =", l1(x) )
print("l2( [ 0, 2, 3 ] ) =", l2(x) )
print("l3( [ 0, 2, 3 ] ) =", l3(x) )

# compute the polynomial p(x) that interpolates the points
# (0,1), (2,3), (3,0) in Langrange form
p = lambda x : l1(x) + 3. * l2(x)

# check that the answer is correct, so evaluate p at x and we check that
# we get y back.
print(" p( [ 0, 2, 3 ] ) =", p(x) )

# after some algebra, get p(x) = -(4/3) x^2 + (11/3) x + 1
# check that this polynomial interpolates the points
# (0,1), (2,3), (3,0)
q = lambda x : - (4./3.) * x**2 + (11./3.) * x + 1.
print(" q( [ 0, 2, 3 ] ) =", q(x) )

print("c=", np.array([ 1., (11./3.), - (4./3.) ] ) )
```

```
l1( [ 0, 2, 3 ] ) = [ 1. -0.  0.]
l2( [ 0, 2, 3 ] ) = [ 0.  1. -0.]
l3( [ 0, 2, 3 ] ) = [-0.  0.  1.]
 p( [ 0, 2, 3 ] ) = [1. 3. 0.]
 q( [ 0, 2, 3 ] ) = [1. 3. 0.]
c= [ 1.          3.66666667 -1.33333333]
```