Homework 7: QAP

Problem 1

Let ϕ be the transformation of a column vector into a polynomial like we discussed in class (using lagrange interpolation over the x values [0, 1, ..., n] and the y values being the values in the vector).

Use Python compute:

$$\phi(c \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}) = c \cdot \phi(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix})$$

Test out a few vectors to convince yourself this is true in general.

In English, what is the above equality stating?

Problem 2: QAP by hand

Convert the following R1CS into a QAP **over real numbers, not a finite field**

```
import numpy as np import random # Define the matrices A =
np.array([[0,0,3,0,0,0], [0,0,0,0,1,0], [0,0,1,0,0,0]]) B =
np.array([[0,0,1,0,0,0], [0,0,0,1,0,0], [0,0,0,5,0,0]]) C =
np.array([[0,0,0,0,1,0], [0,0,0,0,0,1], [-3,1,1,2,0,-1]]) # pick
values for x and y x = 100 y = 100 # this is our orignal formula out =
3 ∗ x ∗ x ∗ y + 5 ∗ x ∗ y − x− 2∗y + 3# the witness vector with the
intermediate variables inside v1 = 3∗x∗x v2 = v1 ∗ y w = np.array([1,
out, x, y, v1, v2]) result = C.dot(w) ==
np.multiply(A.dot(w),B.dot(w)) assert result.all(), "result contains
an inequality"
```

You can use a computer (Python, sage, etc) to check your work at each step and do the Lagrange interpolate, but you must show each step.

Be sure to check the polynomials manually because you will get precision loss when interpolating over floats/real numbers.

Check your work by seeing that the polynomial on both sides of the equation is the same.

Problem 3: QAP over a finite field

Refer to the code here: https://www.rareskills.io/post/r1cs-to-qap

Do the same operation R1CS above but convert it to a QAP over a finite field. Don’t do it by hand, use Python. If you pick GF79 like the article does, you’ll need to find the congruent element in the field since some of the scalar values for the R1CS above will be negative or be larger than 79.