Math 3159A Group Assignment 2 Fall 2020

Instructions

- This assignment is due on Tuesday, October 13, 2020 at 2:00 PM EDT. Late submissions will **not** be accepted.
- This assignment consists of one problem with two parts. You must submit both parts to receive full credit.
- Your solution needs to be formatted using the LaTeX template available on OWL. Note that there are different templates available for regular assignments and group assignments. You should use the one for group assignments.
- All group members are expected to be working on the solution and every member should attend all group meetings.
- The Scribe will be submitting the assignment on behalf of the group. It is assumed that every member of the group has proofread the submission.
- All solutions must be written in full sentences.
- You are not allowed to use online resources and should only discuss the solution with members of your group.
- This assignment is worth 5 points.

Part 1.

1. Let p be a prime and a an integer not divisible by p. Show that the congruence
$$x^2 \equiv a \bmod p$$
has either two or no solutions in $\mathbb{Z}/p$.

2. Let p and q be distinct primes and a an integer such that $\gcd(a, pq) = 1$. Show that the congruence
$$x^2 \equiv a \bmod pq$$
has either four or no solutions in $\mathbb{Z}/pq$.

3. Based on your proof above, describe a polynomial-time algorithm that finds all solutions to the congruence
$$x^2 \equiv 1 \bmod pq.$$

4. Describe a polynomial-time algorithm that given a natural number N of the form $N = pq$ (with p and q are primes) and four elements of $a_1, a_2, a_3, a_4 \in \mathbb{Z}/N$ such that $a_i^2 \equiv 1 \bmod N$, finds p and q.

Math 3159A Group Assignment 2 Fall 2020

Part 2.

In this exercise, we will be implementing questions 3 and 4 of Part 1 (Note: You will write two solve functions).

1. Write a function in Python3 called `solve1` that, given two distinct primes p, q, returns all solutions to the congruence $x^2 \equiv 1 \bmod pq$.

   - Download the file `generate_input1.py` from OWL, use it to obtain three pairs (p, q) by running

     ```
     python generate_input1.py [last three digits of your
                                student number]
     ```

     and run your program on these three inputs. Here, we use the last three digits of the Programmer's student number.

2. Write a function in Python3 called `solve2` that, given a natural number of the form $N = pq$ (for p, q primes) and four roots of unity $a_1, a_2, a_3, a_4 \in \mathbb{Z}/N$, returns the primes p, q.

   - Download the file `generate_input2.py` from OWL, use it to obtain three tuples $(N, (a_1, a_2, a_3, a_4))$ by running

     ```
     python generate_input2.py [last three digits of your
                                student number]
     ```

     and run your program on these three inputs. Here, we use the last three digits of the Programmer's student number.

For both `solve1` and `solve2`, please include:

1. the *Python code* implementing your solution;

2. and the three *inputs you generated*, and the *output of your program* run on these three inputs.

Examples

Here are some examples of what your functions `solve1` and `solve2` should do:

```
>>> solve1(3,5)
 (1, 14, 4, 11)
>>> solve1(7,11)
 (1, 76 ,43, 34)
>>> solve1(31,101)
 (1, 3130, 807, 2324)
```

```
 >>> solve2(15,1,14,4,11)
 (3, 5)
>>> solve2(26069,1,26068,5372,20697)
 (131, 199)
>>> solve2(5723,1,5722,2715,3008)
 (59, 97)
```

Notes

- You may not use any trivial brute-force algorithms such as one that computes all squares of x modulo pq. You must implement the mathematics you developed in Part 1 of the assignment. You will receive no credit if you use this type of algorithm for either `solve1` or `solve2`.

- The files `generate_input1.py` and `generate_input2.py` are written in Python3, and so should be your solution. Make sure you are using a 64bit version of Python3.

- Your code should not make use of any external libraries such as `numpy` or `math`. All the auxiliary functions should be implemented by you, and should be included in your submission. You should only use the most basic arithmetic operations such as `+`, `-`, `*`, `//`, `%`.

- Comments in the code are not mandatory. However in the case of an incorrect solution, the comments can provide grounds for partial credit.