Fall 2024 CIS 3362 Homework #7: Elliptic Curves Point Generation
Check WebCourses for the due date

The slow but relatively simple way of generating all of the points on an elliptic curve is to loop through all possible values of potential points (x, y) and check if they satisfy the equation

$$y^2 \equiv (x^3 + ax + b) \ mod\ p$$

The run-time of this algorithm is $O(p^2)$ since the algorithm tries p^2 potential points.

Since we know that there are roughly O(p) points on the curve, it would be ideal if we could generate all of the points on the curve in closer to O(p) time. There is an algorithm that is within the scope of this class (but one I haven't taught) that can achieve close to this run-time that is described in the following link:

https://dummit.cos.northeastern.edu/docs/numthy_7_elliptic_curves.pdf

They describe an algorithm in detail that works for primes p such that $p \equiv 3\ (mod\ 4)$. For this assignment, you'll write a program that executes this algorithm to generate all of the points on a given elliptic curve. Your program will be run on several input cases (stored in separate files, but your program will read from standard input and write to standard output.)

For convenience, the algorithm is discussed in detail here:

Step 1: Determining which values of x have matching solutions for y.
For each value of x, from x = 0 through x = p – 1, there is either 0, 1 or 2 values of y that satisfy the equation. The middle case is easy to detect. If $(x^3 + ax + b) \equiv 0\ (mod\ p)$, then the only solution for y is y = 0. Otherwise, there is no solution for y or there are two solutions.

The Legendre symbol is defined as follows for odd primes p:

$\left(\frac{c}{p}\right) = 1$, if there exists a value of y for which $y^2 \equiv c\ (mod\ p)$ and

$\left(\frac{c}{p}\right) = -1$, if there **does not** exists a value of y for which $y^2 \equiv c\ (mod\ p)$

Furthermore, for odd primes, p, Euler determined a straight-forward method for calculating the value of the Legendre symbol:

$$\left(\frac{c}{p}\right) = c^{\frac{p-1}{2}}\ (mod\ p)$$

As we previously studied, the expression on the right always evaluates to either 1 or -1, which is consistent with the values of the Legendre symbol.

Thus, to detect if there are 0 or 2 solutions for y, we can use the Euler's Criterion for determining the Legendre symbol. First, for a give value of x, calculate $(x^3 + ax + b) \equiv c \ (mod\ p)$, Next, compute the remainder when $c^{\frac{p-1}{2}}$ is divided by p. If this remainder is 1, then we know there are two values of y on the curve for this particular value of x. (If we get p – 1, then we can skip over this value of x.)

Step 2: Determining the matching value of y for a given x that has one.

Note: This step ONLY works if $p \equiv 3 \ (mod\ 4)$. Apparently this step is more difficult if that's not the case. Thus, all of the input cases for this program will guarantee that the primes chosen leave a remainder of 3 when divided by 4.

Given that $p \equiv 3 \ (mod\ 4)$, and that $\left(\frac{c}{p}\right) = 1$, we can calculate that one solution to

$$y^2 \equiv c \ (mod\ p)$$

Then, one value of y which satisfies this equation is $y \equiv c^{\frac{p+1}{4}} (mod\ p)$. Notice that this exponent is an integer since $p \equiv 3 \ (mod\ 4)$. To see that this is indeed true, consider calculating $(c^{\frac{p+1}{4}})^2$. This definitely yields $c^{\frac{p+1}{2}}$. Consider this expression mod p:

$$c^{\frac{p+1}{2}} = c^{\frac{p-1}{2}} \times c \equiv 1 \times c \equiv c \ (mod\ p)$$

The substitution was made because we know the Legendre symbol $\left(\frac{c}{p}\right) = 1$, which implies that $c^{\frac{p-1}{2}} \equiv 1 \ (mod\ p)$ via Euler's Criterion.

To get the other value of y, just subtract the original solution for y above from p.

The Problem

Given, p, a and b specifying an elliptic curve, output every point on the curve in lexicographical order, sorted by x, breaking ties by y.

Input (from standard input)

The first and only line of input will contain three space-separated positive integers, $\boldsymbol{p}$ ($10 < \boldsymbol{p} < 2\text{x}10^6$, $\boldsymbol{p} \equiv 3\ (mod\ 4)$), $\boldsymbol{a}$ ($0 < \boldsymbol{a} < \boldsymbol{p}$) and $\boldsymbol{a}$ ($0 < \boldsymbol{b} < \boldsymbol{p}$), representing the values of $\boldsymbol{p}$, $\boldsymbol{a}$ and $\boldsymbol{b}$ defining an elliptic curve.

Output (to standard output)

Output $\boldsymbol{n}$ lines, where $\boldsymbol{n}$ represents the number of points on the elliptic curve described in the input. On each line, output a single value of $\boldsymbol{x}$ followed by a space and a single value of $\boldsymbol{y}$ ($0 \leq \boldsymbol{x}, \boldsymbol{y} < \boldsymbol{p}$) representing that the point ($\boldsymbol{x}, \boldsymbol{y}$) is on the curve. Output the points in sorted order by x (from low to high), breaking ties by y (from low to high).

Sample Input

```
23 1 1
```

Sample Output

```
0 1
0 22
1 7
1 16
3 10
3 13
4 0
5 4
5 19
6 4
6 19
7 11
7 12
9 7
9 16
11 3
11 20
12 4
12 19
13 7
13 16
17 3
17 20
18 3
18 20
19 5
19 18
```

Grading Details

Writing the $O(p^2)$ algorithm to solve this problem is very, very easy (it's 5 lines of code in Python). Thus, very few execution or code points will be given to this solution.

Rather, a majority of points will be awarded for implementing the algorithm described above.

It's better to attempt the steps of the algorithm and have them not work at all on any test cases than to turn in the simple code that works (fast enough) for cases upto a few thousand.

Deliverables

Please submit **a single source file**, either **`ecpoints.py`** or **`ecpoints.java`** with your solution to the problem.