Math component: Elliptic curves and Lenstra's Algorithm

Inverses via Bezout.

(1) Analyze the code for `inverse`. What is the output? Why does it work?

Addition of points on elliptic curves.

(1) In your own words, describe the meaning of a *pseudo elliptic curve modulo n*.
(2) Complete the provided code for `EC_addition`.
(3) Describe the exact point at which this function confronts the possibility that the input elliptic curve may be, in fact, a pseudo elliptic curve. What is the output in this case? More precisely, what is the significance of `witness` in this context? What does it witness?

An analog of `FastExp` for elliptic curves via successive doubling.

Consider a point $P \neq \mathcal{O}$ on an elliptic curve E defined modulo an integer. Given a positive integer e, our purpose here is to describe a faster way to compute the multiple eP, which we define to be the outcome of adding P to itself e times. It will be analogous to the `fastExp` function that you have already written, which speeds up exponentiation modulo n via successive squaring.

(1) Consider the base 2 (binary) expansion of e, i.e, the unique expression

$$e = e_0 + e_1 2 + e_2 2^2 + \cdots + e_k 2^k.$$

Describe how to produce the list $[e_0, e_1, \ldots, e_k]$ using a function from GE 2.
Note: You are being asked to write the code here.
(2) Write a function `doublesList` whose input is $[E, P, k]$ and whose output is the list

$$[P, 2P, (2^2)P, \ldots, (2^k)P].$$

When doing so, feel free to assume that E is actually an EC, as opposed to a pseudo EC.
Note: You are being asked to write the code here.
(3) Write a simple function `EC_multiple` whose input is $[E, P, e]$ and that outputs eP. When doing so, feel free to assume that E is actually an EC, as opposed to a pseudo EC.
Note: You are being asked to write the code here.
(4) Compare your function with the one that I wrote. They should be almost identical. What, if any, are the significant differences? More precisely, what is the purpose of the `if` statements appearing in my code? What does this have to do with pseudo elliptic curves?

Some applications. Use the programs you wrote above to answer the following questions. Make sure to phrase your responses in complete sentences!

(1) Using a loop, compute the order of $P = (1, 3)$ on $E(41)$ defined by $y^2 = x^3 + 8$.
(2) Suppose that Alice and Bob agree to use the EC Diffie-Hellman key exchange with the prime $p = 2671$, the elliptic curve $E(2671)$ given by $y^2 = x^3 + 171x + 853$, and the point $P = (1980, 431)$ on $E(2671)$. Furthermore, suppose that Bob's secret (private) number is $b = 1943$, and that during the exchange, Alice passes Bob the point $A = (2110, 543)$. What point B should Bob pass to Alice? What point represents their shared key?
(3) Consider the point $P = (2, 3)$ on the elliptic curve $E(557)$ given by $y^2 = x^3 - 10x + 21$. Verify that $189P = \mathcal{O}$, but that neither $63P$ nor $27P$ equals $\mathcal{O}$. Using only this information, explain how you can conclude that the order of P actually equals 189. Next, what does this tell you about the possibilities for $|E(557)|$? Combine this with the Hasse-Bound[1] to compute, by hand, the exact value of $E(557)$. Was P a primitive root?

[1]Recall that this says that $p + 1 - 2\sqrt{p} \leq |E(p)| \leq p + 1 + 2\sqrt{p}$ for an elliptic curve $E(p)$

Analysis of the function `EC_random`.

(1) Read the code provided for `EC_random`. What is the output?
(2) Provide a mathematical argument to justify why `EC_random` outputs what it claims to.

Lenstra's Algorithm. Consider a positive integer n, and the pseudo elliptic curve $E(n)$ defined by the equation $y^2 \equiv x^3 + Ax + B$ modulo n. As noted in lecture, if n is prime, then $E(n)$ is a group, but otherwise, there will be points in $E(n)$ whose sum is not defined.

Note that if P and Q are points on $E(n)$ whose sum is not defined, then the "slope" M that appears in our calculations is not defined, precisely because its "denominator", though nonzero modulo n, is not a unit, and hence, can't be a valid denominator. By taking the gcd of this denominator with n, we should obtain a factor of n. Our goal here is to utilize this observation to implement an elliptic curve analog of Pollard's $p-1$ factoring algorithm, due to H. Lenstra.

Lenstra's EC factoring algorithm (with factorials): Consider a positive integer n, a pseudo elliptic curve $E(n)$, and a point $P \in E(n)$. We must also specify a positive integer b, which, informally, tells us how hard to try.

- Efficiently (iteratively) attempt to compute the multiples $P_k = (k!)P$
- For the first P_k that is not defined, determine the "slope" M at this step whose "denominator" is not defined modulo n. Output the gcd of this with n.
- If each of the multiples $P_1, \ldots, P_b$ is defined, try increasing the value of b.
- You may also start over entirely, by either updating P or $E(n)$.

(1) Complete the code for `factorial_lenstra` and `random_factorial_lenstra`.
Note: The multiples P_k should be computed recursively as $P_1 = P$ and then $P_k = kP_{k-1}$ for all higher k. This observation will be helpful when completing these functions.
(2) Describe, in words, the significance of the output of `lenstra` when a factor is found.
(3) Apply `factorial_lenstra` to factor $n = 363982776557$ using the point $P = (2, 5)$ on the curve $E(n)$ given by $y^2 = x^3 + 3x + 11$. You'll have to experiment with different b values. What is the first P_k that cannot be computed? What factor of n is obtained?
(4) Apply `random_factorial_lenstra` to factor $n = 59865879235697$.
(5) Pick a few really large integers, try to factor them using the random version of your function, and report the results.
(6) Describe how to use `doublesList` to define a different version of (random) Lenstra. If you choose to do actually implement this, experiment and compare it to the factorial version, and report back.