Example 222. Consider again the elliptic curve E, described by $y^2 = x^3 - x + 9$.

(a) Determine $(0,3) \boxplus (1,3)$.

(b) Determine $(0,3) \boxplus (1,-3)$.

(c) Determine $4(0,3)$, which is short for $(0,3) \boxplus (0,3) \boxplus (0,3) \boxplus (0,3)$.

Solution. We let Sage do the work for us:

```
>>> E = EllipticCurve([-1,9])
```

```
>>> E(0,3) + E(1,3)
```

$(-1 : -3 : 1)$

```
>>> E(0,3) + E(1,-3)
```

$(35 : 207 : 1)$

```
>>> 4*E(0,3)
```

$$\left(-\frac{1677023}{60279696} : \frac{1406201395535}{468011559744} : 1\right)$$

We conclude that $(0,3) \boxplus (1,3) = (-1,-3)$ and $(0,3) \boxplus (1,-3) = (35,207)$ (one of the points mentioned in Example 220), while

$$4(0,3) = \left(-\frac{1677023}{60279696}, \frac{1406201395535}{468011559744}\right).$$

Comment. Note how Sage represents the point (x, y) as $(x : y : 1)$. These are **projective coordinates** which make it easier to incorporate the special point O which is represented by $(0 : 1 : 0)$.

https://en.wikipedia.org/wiki/Projective_coordinates

The following computation demonstrates that adding O doesn't do anything:

```
>>> E(0)
```

$(0 : 1 : 0)$

```
>>> E(0,3) + E(0)
```

$(0 : 3 : 1)$

Comment. Note that, starting with a single point such as $(0, -3)$, we can generate other points such as $2(0,-3) = \left(\frac{1}{36}, \frac{647}{216}\right)$ (one of the points mentioned in Example 220). If the initial point is rational then so are the points generated from it.

Advanced comment. If you want to dig deeper, you can try to translate the geometric description of the addition $P \boxplus Q$ into algebra by deriving equations for the coordinates of $P \boxplus Q = (x_r, y_r)$ in terms of the coordinates of $P = (x_p, y_p)$ and $Q = (x_q, y_q)$. For instance, for the elliptic curve $y^2 = x^3 + ax + b$, one finds that

$$\begin{aligned} x_r &= \lambda^2 - x_p - x_q, \\ y_r &= \lambda(x_p - x_r) - y_p, \end{aligned}$$

where $\lambda = (y_q - y_p)/(x_q - x_p)$ is the slope of the line connecting P and Q. If P and Q are the same point, then this line becomes the tangent line and the slope becomes $\lambda = (3x_p^2 + a)/(2y_p)$ instead. For more details:

https://en.wikipedia.org/wiki/Elliptic_curve

From these formulas, can you reproduce the computations we did in Sage?

Elliptic curves modulo primes

For cryptographic purposes, elliptic curves are usually considered modulo a (large) prime p.

Example 223. Let us consider $y^2 = x^3 - x + 9$ (the elliptic curve from the previous examples) modulo 7. List all points on that curve.

Solution. Note that, because we are working modulo 7, there are only 7 possible values for each of x and y. Hence, we can just go through all $7^2 = 49$ possible points (x, y) to find all points on the curve.

Or, better, we go through all possibilities for x (such as $x = 2$) and determine the corresponding possible values for y (if $x = 2$, then $y^2 = 2^3 - 2 + 9 = 15 \equiv 1 \pmod 7$ which has solutions $y \equiv \pm 1 \pmod 7$).

Doing so, we find 9 points: $O, (0, \pm 3), (\pm 1, \pm 3), (2, \pm 1)$.

[Recall that O is the special point "at ∞" which serves as the neutral element with respect to $\boxplus$.]

Comment. A theorem of Hasse–Weil says that the number of points on an elliptic curve modulo p is always close to p (this is indeed what we expect because, for each of the p choices for x, we get an equation of the form $y^2 \equiv a \pmod p$ which has 2 solutions if a is a nonzero quadratic residue [and for a random a the odds are about 50% that it is quadratic]). Moreover, we can compute the exact number of points very efficiently.

By taking everything modulo 7, we still have the previously introduced addition rule $\boxplus$.

For instance. $(0, 3) \boxplus (1, -3) = (35, 207) \equiv (0, -3) \pmod 7$

Here is how we can use Sage to list all points, as well as add any two of them:

```
>>> E7 = EllipticCurve(GF(7), [-1,9])
>>> E7.points()
  [(0: 1: 0), (0: 3: 1), (0: 4: 1), (1: 3: 1), (1: 4: 1), (2: 1: 1), (2: 6: 1), (6: 3: 1), (6: 4: 1)]
>>> E7(0,3) + E7(1,-3)
  (0: 4: 1)
>>> E7(1,-3) + E7(0,-3)
  (6: 3: 1)
```

Multiples of a point are simply denoted with nP. For instance, $3P = P \boxplus P \boxplus P$.

We then have a version of the **discrete logarithm** problem for elliptic curves:

(discrete logarithm) Given P, xP on an elliptic curve, determine x.
(computational Diffie–Hellman) Given P, xP, yP on an elliptic curve, determine $(xy)P$.

Comment. Interestingly, it appears that the computational Diffie–Hellman problem (CDH) is more difficult for elliptic curves modulo p than for regular multiplication modulo p. Indeed, suppose that p is an n-digit prime. Then the best known algorithms for regular CDH modulo p has runtime $2^{O(\sqrt[3]{n})}$, whereas the best algorithm for the elliptic curve CDH modulo p has runtime $\sqrt{p} \approx 2^{n/2} = 2^{O(n)}$.

As a consequence, it is believed that a smaller prime p can be used to achieve the same level of security when using elliptic curve Diffie–Hellman (ECDH). In practice 256bit primes are used, which is believed to provide security comparable to 2048bit regular Diffie–Hellman (DH); this makes ECDH about ten times faster in practice than DH.

Comment. On the other hand, due to that reduced bit size, quantum computing attacks on elliptic curve cryptography, if they become available, would be more feasible compared to attacks on ElGamal/RSA.