18.783 Elliptic Curves **Spring 2015**
Problem Set #12 **Due: 05/08/2015**

Description

These problems are related to the material covered in Lectures 22 and 23. As usual, the first person to spot each non-trivial typo/error will receive 1–3 points of extra credit.

Instructions: Either solve **one** of Problems 1 and 2 or **both** or Problems 3 and 4. Then complete Problem 5, which is a survey. This Sage worksheet contains modular polynomials and helper functions from previous problem sets that you may find useful.

Problem 1. Isogeny volcanoes (100 points)

For the purposes of this problem, an isogeny volcano is an ordinary component of an ℓ-isogeny graph $G_\ell(\mathbb{F}_q)$ that does not contain 0 or 1728, where $\ell \nmid q$. This is a bi-directed graph that we regard as an undirected graph.

(a) Use the CM method to explicitly construct isogeny volcanoes that meet each of the following sets of criteria:

- **(i)** $\ell = 2$, $d = 3$, V_0 is a 5-cycle;
- **(ii)** $\ell = 3$, $d = 2$, V_0 contains a single edge;
- **(iii)** $\ell = 7$, $d = 1$, V_0 contains a single vertex with two self-loops.

In your answers, specify the finite field used, the discriminant of the order $\mathcal{O}_0$ corresponding to V_0, and list each bi-directed edge just once, as a pair (v_1, v_2) of j-invariants corresponding to a horizontal or descending edge.

(b) Use the CM method to construct a single ordinary elliptic curve $E/\mathbb{F}_q$ that simultaneously satisfies all of the following criteria:

- **(i)** $j(E)$ is on the floor of its 2-volcano, which has depth 6.
- **(ii)** $j(E)$ is on the surface of its 3-volcano, which has depth 3.
- **(iii)** $j(E)$ is on the middle level of its 5-volcano, which has depth 2.
- **(iv)** $j(E)$ is on the floor of its 7-volcano, which as depth 5.
- **(v)** $j(E)$ is one of exactly two vertices in its 11-volcano.
- **(vi)** $j(E)$ is the only vertex in its 13-volcano.

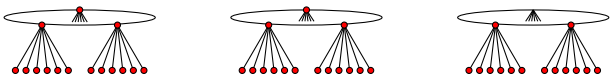
In your answer, specify the finite field $\mathbb{F}_q$, the j-invariant $j(E)$, and the discriminant D of the order $\mathcal{O} \simeq \mathrm{End}(E)$.

(c) Prove that the cardinality of a 2-isogeny volcano with an odd number of vertices must be a Mersenne number (an integer of the form $2^n - 1$). Give an explicit example of a 2-isogeny volcano with 15 vertices.

(d) Prove that every ordinary elliptic curve $E/\mathbb{F}_q$ is isogenous to an elliptic curve $E'/\mathbb{F}_q$ for which $E'(\mathbb{F}_q)$ is a cyclic group.

Problem 2. Computing modular polynomials (100 points)

As we have seen, the modular polynomials $\Phi_\ell(X,Y)$ play a key role in many theoretical and practical applications of elliptic curves. One can compute them using the q-expansions of the modular functions $j(z)$ and $j(\ell z)$, but this is approach is difficult to implement efficiently and extremely memory intensive. In this problem you will implement an easier and faster algorithm using isogeny volcanoes. The strategy is to use a CRT approach, with primes p carefully selected to achieve a configuration of ℓ-volcanoes similar to that depicted below:



Here we have a configuration of three ℓ-volcanoes, with $\ell = 7$, each of depth $d = 1$. There are a total of $\ell + 2$ vertices on the surface (any value greater than $\ell + 1$ suffices).

Provided we have completely "mapped" this configuration of ℓ-volcanoes, meaning that we know the j-invariants of every vertex in the figure, we can compute $\Phi_\ell(X,Y)$ as follows. For any particular j-invariant j_i on the surface, we know the values of all the roots of $\phi_i(Y) = \Phi_\ell(j_i, Y)$, since we know the $\ell + 1$ neighbors of j_i in $G_\ell(\mathbb{F}_p)$. We can therefore compute each ϕ_i as the product of its linear factors. If we then consider the coefficient of Y^k in ϕ_i, we know (at least) $\ell+2$ values c_{ik} of this coefficient, corresponding to $\ell + 2$ distinct j_i. This suffices to uniquely determine the polynomial $\psi_k(X)$ of degree at most $\ell + 1$ for which $\psi_k(j_i) = c_{ik}$, via Lagrange interpolation:

$$\psi_k(X) = \sum_{i=1}^{\ell+2} c_{ik} \prod_{m \neq i} \frac{(X - j_m)}{(j_i - j_m)}$$

(a) Prove that $\Phi_\ell(X,Y) = \sum_{k=0}^{\ell+1} \psi_k(X) Y^k$.

To make things simpler, we will use a configuration with (at least) $\ell+2$ isomorphic ℓ-volcanoes, each with one vertex on the surface and $\ell+1$ neighbors on the floor. This can be achieved using a fundamental discriminant D with $(\frac{D}{\ell}) = -1$ and $h(D) \geq \ell + 2$. The vertex on the surface of each ℓ-volcano will have endomorphism ring equal to the maximal order for $K = \mathbb{Q}(\sqrt{D})$, and the vertices on the floor will then have endomorphism ring equal to the order $\mathcal{O}'$ with discriminant $\ell^2 D$ (note that $\mathcal{O}'$ has index ℓ in $\mathcal{O}$). For convenience, we will choose D so that both $\mathrm{cl}(D)$ and $\mathrm{cl}(\ell^2 D)$ are cyclic groups generated by prime forms of norm $\ell_0 = 3$ (so we can use ℓ_0-volcanoes of depth 0; see part 4 of problem 2). This idealized setup is not always achievable, but it will work for our example using $\ell = 17$ and $D = -2339$, with class number $h(D) = 19$.

The key challenge is to map our set of ℓ-volcanoes without using the polynomial Φ_ℓ. Mapping the surface is easy: the vertices on the surface of our set of ℓ-volcanoes are the roots of the Hilbert class polynomial H_D (each root constitutes the surface of its own volcano). The vertices on the floor are the roots of the Hilbert class polynomial $H_{\ell^2 D}$, but this polynomial is much larger than H_D and we don't want to compute it, since it would take time $\widetilde{O}(\ell^4)$. Instead we will use Vélu's formulas (see [5, §12.3]) to compute a descending isogeny from each vertex on the surface. The kernel of this isogeny is a cyclic subgroup of $E[\ell]$, and Vélu's formulas require us to enumerate the points in the

kernel, which may lie in an extension field of degree as large as $\ell^2 - 1$ (the degree of the ℓ-division polynomial). But we will choose primes $p \equiv 1 \bmod \ell$ that satisfy the norm equation $4p = t^2 - \ell^2 D$. This ensures that the elliptic curves $E/\mathbb{F}_p$ with endomorphism ring $\mathcal{O}_K$ have rational ℓ-torsion (provided we choose the correct twist); in this situation Vélu's formulas are very efficient.

(b) With $\ell = 17$ and $D = -2339$, find a prime $p \equiv 1 \bmod \ell$ that satisfies $4p = t^2 - \ell^2 D$. Note that this requires $t \equiv \pm 2 \bmod \ell$, and with $t \equiv 2 \bmod \ell$ we will have $p + 1 - t$ divisible by ℓ^2. Use Sage to compute the Hilbert class polynomial $H_D(X)$ and find the roots of $H_D \bmod p$. For each of the roots $j_1, \ldots, j_h$ of H_D, construct an elliptic curve E_i with j-invariant j_i, and attempt to find a point $P_i \in E(\mathbb{F}_p)$ with order ℓ by computing random $P_i = mP$ with $m = (p + 1 - t)/\ell^2$. If you find $P_i \neq 0$ and $\ell P_i \neq 0$ then you will need to replace E_i with a quadratic twist $y^2 = x^3 + d^2 A + d^3 B$, where d a not a square in $\mathbb{F}_p$

As a sanity check, you may want to pick one of the E_i and use Sage to verify that $E_i(\mathbb{F}_p)$ has ℓ-rank 2, but this is not part of the algorithm.

We are now ready to apply Vélu's formulas to each pair (E_i, P_i) to obtain an ℓ-isogenous curve E_i'. Since every curve E_i' that is ℓ-isogenous to E_i lies on the floor, it does not matter which P_i we choose, any point of order ℓ will work. Below is a simplified algorithm that implements Vélu's formulas for the case where we have a cyclic subgroup generated by a point P of odd order on an elliptic curve given in short Weierstrass form $y^2 = x^3 + Ax + B$ over a finite field $\mathbb{F}_p$ with $p > 3$.

1. Set $t \leftarrow 0$, $w \leftarrow 0$, and $Q \leftarrow P$.
2. Repeat $(l - 1)/2$ times:
   a. Set $s \leftarrow 6Q_x^2 + 2A$, and then set $u \leftarrow 4Q_y^2 + sQ_x$.
   b. Set $t \leftarrow t + s$, $w \leftarrow w + u$, and $Q \leftarrow Q + P$.
3. Set $A' = A - 5t$ and $B' = B - 7w$.
4. Output the curve $E'/\mathbb{F}_p$ defined by $y^2 = x^3 + A'x + B'$.

In the description above Q_x and Q_y are the affine coordinates $(x.y)$ of the point Q.

(c) Implement the above algorithm and use it to compute elliptic curves E_i' that are ℓ-isogenous to the curves E_i you computed in step 2. Let $j_1', \ldots, j_h'$ be the corresponding j-invariants.

Now comes the interesting part. We want to enumerate the vertices on the floor of our our ℓ-volcano, but there are no horizontal ℓ-isogenies between vertices on the floor! Instead, we must go up to the surface and back down, which amounts to computing an isogeny of degree ℓ^2. If we return to the same vertex this is just the multiplication-by-ℓ map (the composition of an ℓ-isogeny with its dual), but otherwise it is a cyclic isogeny of degree ℓ^2, corresponding to the CM action of a proper $\mathcal{O}'$-ideal of norm ℓ^2.

(d) For $(\frac{D}{\ell}) = -1$, show that there are ℓ inequivalent integral primitive positive definite binary quadratic forms (ℓ^2, b, c) of discriminant $\ell^2 D$ (in our example these will all be reduced forms). These forms generate a cyclic subgroup G of $\mathrm{cl}(\ell^2 D)$ of order $\ell + 1$. For $\ell = 17$ and $D = -2339$, determine a generator $f = (a, b, c)$ for G.

Now we certainly don't want to use Φ_{ℓ^2} to compute the action of f (we don't even know Φ_ℓ yet!). But as in problem 1 of Problem Set 11, we can compute the action of an $\mathcal{O}'$-ideal of large norm using the action $\mathcal{O}'$-ideals of much smaller norm. In our example, we can use an $\mathcal{O}'$-ideal of norm $\ell_0 = 3$ to enumerate all the vertices on the floor of our set of volcanoes, and then determine the action of f by computing a discrete logarithm in $\mathrm{cl}(\ell^2 D)$. Recall that we chose D so that a prime form of norm 3 generates $\mathrm{cl}(\ell^2 D)$, so this is easy.

(e) Use $\Phi_{\ell_0} = \Phi_3$ to enumerate all the vertices on the floor as a cycle of 3-isogenies.

(f) Compute the discrete logarithm k of the form f from part 4 with respect to a prime form of norm $\ell_0 = 3$ in $\mathrm{cl}(\ell^2 D)$. There is no need to distinguish inverses, and you should find that $(\ell+1)k \equiv 0 \bmod h(\ell^2 D)$. Feel free to use brute force (a linear search); the time will be dominated by later steps in any case. Knowing k, you can now identify the subsets in the enumeration of part 5 that correspond to cosets of G. Each of these subsets will contain exactly one the j-invariants j_i' that you computed in step 3 and corresponds to the $\ell+1$ "children" of j_i (its neighbors on the floor).

(g) For each of $\ell+2$ vertices j_i on the surface, compute the polynomial $\phi_i(Y) = \Phi_\ell(j_i, Y) = \prod_n (Y - j_{im})$, where the j_{im} range over the $\ell+1$ children of j_i that you identified in part 6. Then, for k ranging from 0 to $\ell+1$, interpolate the unique polynomial $\psi_k(X)$ of degree at most $\ell+1$ for which $\psi_k(j_i)$ is equal to the coefficient of Y^k in $\phi_i(Y)$. You can do this with Sage: first create the polynomial ring `R.<X>=PolynomialRing(GF(p))`, and then use

```
R.lagrange_polynomial([(x0,y0),(x1,y1),...,(xn,yn)])
```

to compute the unique polynomial $f(X)$ of degree at most n for which $f(x_i) = y_i$. Note that $\psi_{\ell+1}(X)$ must be the constant polynomial 1.

Finally, compute $\Phi_\ell(X,Y) = \sum_{k=0}^{\ell+1} f_k(X) Y^k \bmod p$.

You have now computed $\Phi_{17}(X,Y) \bmod p$. As a sanity check, verify that the coefficients are symmetric: $\Phi_\ell(X,Y) = \Phi_\ell(Y,X)$. If you need to debug your algorithm, you may find it helpful to ask Sage to compute the Hilbert class polynomial $H_{\ell^2 D}(X)$ and then verify that the j-invariants j_i' that you computed in step 3 are actually roots of $H_{\ell^2 D} \bmod p$.

Now to convince ourselves that we have really computed $\Phi_{17} \bmod p$, let's use it to enumerate the roots of a completely *different* Hilbert class polynomial.

(h) Using the same prime p, pick a different discriminant D^* for which $4p = t^2 - v^2 D^*$ with v not divisible by 17 and $(\frac{D^*}{\ell}) = 1$, and such that $h(D^*) > 4$. Use Sage to compute all the roots of the Hilbert class polynomial $H_{D^*}(X) \bmod p$, and then use the polynomial $\Phi_{17}(X,Y) \bmod p$ to organize the roots into cycles of 17-isogenies (there will may well be only one 17-cycle, this depends on the D you pick).

Note that $\Phi_{17}(X,Y) \bmod p$ must permute the roots of $H_{D^*}(X) \bmod p$, and this permutation must be a product of cycles, each with length equal to the order of the prime forms of norm 17 in $\mathrm{cl}(D^*)$ (since the roots of H_{D^*} are a $\mathrm{cl}(D^*)$-torsor).

Provided that $D = O(\ell^2)$ and $\ell_0 = O(\log \ell)$, one can show that the algorithm you have implemented takes time $O(\ell^2 \log^3 p \log\log p)$, which is nearly optimal, since it is quasi-linear in the size of $\Phi_\ell \bmod p$. By applying the same algorithm to a sufficiently large set of suitable primes p_i (it suffices to have $\sum \log p_i > 6\ell \log \ell + 18\ell$), one can then use the Chinese remainder theorem (as in problem 1 of Problem Set 11) to compute the coefficients of $\Phi_\ell \in \mathbb{Z}[X, Y]$. Under the GRH, the total time to compute Φ_ℓ over $\mathbb{Z}$ is $O(\ell^3 \log^3 \ell \log\log \ell)$; see [3]. In practical terms, this algorithm can be used to compute Φ_ℓ even when ℓ is well into the thousands and Φ_ℓ is many gigabytes.

Problem 3. Atkin-Morain ECPP (50 points)

The bottleneck in the Goldwasser-Kilian elliptic curve primality proving algorithm is the time spent counting points on randomly generated elliptic curves in the hope of finding one with a suitable number of points (namely, the product of a large prime and a smooth co-factor). Atkin and Morain proposed an alternative approach in [1] that uses the CM method to construct an elliptic curve that is guaranteed to have a suitable number of points. This yields a much faster algorithm, with a heuristic running time of $\widetilde{O}(n^4)$ that makes it the method of choice for general purpose primality proving. While its running time is not provably polynomial time, in practice it is faster than even randomized versions of the AKS algorithm that run in $\widetilde{O}(n^4)$ expected time [2].

Given a smoothness bound B and probable prime p, the algorithm proceeds as follows:

1. Select a fundamental discriminant D for which $4p = t^2 - v^2 D$ has a solution (t, v) such that $m = p + 1 \pm t$ can be factored as cq, where $c > 1$ is B-smooth and $q > (p^{1/4} + 1)^2$ is a probable prime.[1]

2. Find a root j of $H_D \bmod p$ and use it to construct an elliptic curve $E/\mathbb{F}_p$ in Weierstrass form $y^2 = x^3 + Ax + B$, where $A = 3j(1728 - j)$ and $B = 2j(1728 - j)^2$. If unable to find a root of $H_D \bmod p$ within, say, twice the expected amount of time, perform a Miller-Rabin test on p. If it fails then report that p is not prime and otherwise repeat this step.

3. Generate a random $Q \in E(\mathbb{F}_p)$ with $P = cQ \neq 0$ and verify that $qP = 0$. If not, replace E with a quadratic twist $\tilde{E}\colon y^2 = x^3 + d^2Ax + d^3B$, for some non-residue d, and repeat this step. If the verification $qP = 0$ fails for E and its twist, or if anything else goes wrong (e.g., a square-root computation or inversion fails), report that p is not prime.

4. Output the certificate (p, A, B, x, y, q), where $P = (x, y)$.

As with the Goldwasser-Kilian algorithm, for $q > B$ one then proceeds to construct a primality certificate for q using the same algorithm, producing a chain of primality certificates that terminates with a prime $q \leq B$ whose primality is verified by trial division (see Lecture 13).

As described above, the algorithm does not achieve a heuristic running time of $\widetilde{O}(n^4)$. The computation is (perhaps surprisingly) dominated by the cost of solving the norm equations in step 1, which is in turn dominated by the time spent computing square-roots of D modulo p in Cornacchia's algorithm. In order to achieve a heuristic time of

[1] For $D = -3, -4$ there are other possible curve orders that can be used, but we will ignore them.

$\widetilde{O}(n^4)$ one selects a set of primes S and restricts the discriminants D to products of pairs of primes in S. By computing square-roots modulo p for all the primes in S, one can then efficiently compute the square-root modulo p of any such D; see [6] for details. For the purposes of this problem, you do not need to implement this optimization, it only becomes worthwhile when p is very large.

(a) Implement the Atkin-Morain ECPP algorithm described above in Sage and use it to construct a primality proof for the least prime greater than $2^{500}N$, where N is the last 4 digits of your student ID, using the smoothness bound $B = 2^{20}$ (you may want to tweak this to optimize the time spent searching for candidate curve orders, a smaller B means less time spent trial-dividing candidate m's but a lower probability of success) You can use the `norm_equation` function in the Sage worksheet linked to above to solve the norm equations in step 1. In your implementation, create the finite field $\mathbb{F}_p$ in Sage using `GF(p,proof=false)` to prevent Sage from trying to prove that p is prime by itself. You can use the `is_pseudoprime` function in Sage to test whether q is a probable prime after using trial-division to remove the B-smooth factor c. You needn't implement the Miller-Rabin test in step 2, as it is extremely unlikely to be necessary.

In your write-up, do not list all the primality certificates in full. Just give a table that lists just the discriminant D and the prime q used for each certificate, as well as the time spent constructing each certificate.

Problem 4. Supersingular isogeny graphs (50 points)

Let p be and ℓ be distinct primes. Recall from Theorem 14.16 that the j-invariant of every supersingular elliptic curve over $\overline{\mathbb{F}}_p$ lies in $\mathbb{F}_{p^2}$. In this problem you will explore some properties of the supersingular components of $G_\ell(\mathbb{F}_{p^2})$.[2]

(a) Compute the graph of the component of $G_2(\mathbb{F}_{97^2})$ containing the supersingular j-invariant 1. You may wish to draw the graph on paper, but in your write-up just give a complete list of directed edges.

(b) Prove that every supersingular vertex in $G_\ell(\mathbb{F}_{p^2})$ has out-degree $\ell+1$, and conclude that no supersingular component of $G_\ell(\mathbb{F}_{p^2})$ is an ℓ-volcano. Show by example that the in-degree need not be $\ell+1$.

(c) Design an efficient *Las Vegas* algorithm that, given an arbitrary j-invariant in $\mathbb{F}_{p^2}$, determines whether it lies in an ordinary or supersingular component of $G_\ell(\mathbb{F}_{p^2})$ by detecting the difference between these components as abstract graphs. Prove that if $\ell = O(1)$ then the expected running time of your algorithm is $\tilde{O}(n^3)$, where $n = \log p$.[3]

The fastest known algorithms for computing the trace of Frobenius all have complexity $\Omega(n^4)$, so your algorithm provides a way to determine whether a given elliptic curve over a finite field is ordinary or supersingular that is asymptotically more efficient than checking whether the trace of Frobenius is divisible by p, and in practice, it should be *much* faster.

[2] There is in fact only one supersingular component of $G_\ell(\mathbb{F}_{p^2})$, see [4, Cor. 78], but won't use this.

[3] As usual, the soft $\tilde{O}$-notation ignores factors that are polylogarithmic in n.

(d) By applying your algorithm to $G_2(\mathbb{F}_{p^2})$, determine which of the following j-invariants is supersingular. List the running time of your algorithm in each case.

(i) $p = 2^{64} + 81$:

```
p=2^64+81
R.<t> = PolynomialRing(GF(p))
F.<a> = GF(p^2, modulus=t^2+5)
j1=8326557536028784306*a + 13186271742734526835
j2=17095442389470987916*a + 5391379569813173462
j3=8201451720284342414*a + 1239990603471114829
j4=3832397532494683106*a + 3456346199771023610
j5=6995663267023152807*a + 5118305496003400382
```

(ii) $p = 2^{498}(2^{17} - 1) + 5^2 \cdot 11^2$:

```
p=2^498*(2^17-1)+5^2*11^2
F.<a>=GF(p^2)
j1=F(106873030904038253717857935791831574043723767360 1\
463652829906969943912262397017489359233817667235136 33\
617314116677847252974815762274295992015602852450016138)
j2=F(930783763888948580286413088959734211243124071761 7\
797432031465706705768740738818194689422900467626903 25\
811223608385837361515252894508396542189580901879014 80)
```

Be patient, it may take several minutes for your program to run on the last two examples (but it should not take more than 10 or 20 minutes).

Problem 5. Survey

Complete the following survey by rating each problem you attempted on a scale of 1 to 10 according to how interesting you found it (1 = "mind-numbing," 10 = "mind-blowing"), and how difficult you found it (1 = "trivial," 10 = "brutal"). Also estimate the amount of time you spent on each problem to the nearest half hour.

	Interest	Difficulty	Time Spent
Problem 1			
Problem 2			
Problem 3			
Problem 4			

Also, please rate each of the following lectures that you attended, according to the quality of the material (1="useless", 10="fascinating"), the quality of the presentation (1="epic fail", 10="perfection"), the pace (1="way too slow", 10="way too fast", 5="just right") and the novelty of the material (1="old hat", 10="all new").

Date	Lecture Topic	Material	Presentation	Pace	Novelty
4/30	Main theorem of CM				
5/4	Isogeny volcanoes				

Please feel free to record any additional comments you have on the problem sets or lectures, in particular, ways in which they might be improved.

References

[1] A.O.L. Atkin and F. Morain, *Elliptic curves and primality proving*, Mathematics of Computation **61** (1993), 29–68.

[2] D.J. Bernstein, *Proving primality in essentially quartic random time*, Mathematics of Computation **76** (2007), 398–403.

[3] R. Bröker, K. Lauter, and A.V. Sutherland, *Modular polynomials via isogeny volcanoes*, Mathematics of Computation **81** (2012), 1201–1231.

[4] David Kohel, *Endomorphism rings of elliptic curves over finite fields*, PhD thesis, University of California at Berkeley, 1996.

[5] L. C. Washington, *Elliptic curves: Number theory and cryptography*, second edition, CRC Press, 2008.

[6] F. Morain, *Implementing the asymptotically fast version of the elliptic curve primality proving algorithm*, Mathematics of Computation **76** (2007), 493–505.