Written problems

1. Textbook exercise 4.7.

Solution.

If S_1, S_2 are as in the problem statement, then using the fact that $A \equiv g^a \pmod p$,

$$\begin{aligned} A^{S_1} S_1^{S_2} &\equiv g^{aS_1} \left(g^i A^j\right)^{S_2} \pmod p \\ &\equiv g^{aS_1 + iS_2 + ajS_2} \pmod p \\ &\equiv g^{aS_1 - ij^{-1}S_1 - aS_1 jj^{-1}} \pmod p \ \left(\text{since } S_2 \equiv -S_1 j^{-1} \pmod{p-1}\right) \\ &\equiv g^{\cancel{aS_1} - ij^{-1}S_1 - \cancel{aS_1}} \pmod p \\ &\equiv g^{-ij^{-1}S_1} \pmod p. \end{aligned}$$

We can now see the wisdom in the use of the number j here: it's presense in both S_1 and S_2 is perfectly calibrated to cause the *terms with the secret key a to cancel*. This is what allows Eve to find her document D without needing to know anything about a. In particular, the choice $D \equiv -ij^{-1}S_1 \pmod{p-1}$ ensures that $A^{S_1} S_1^{S_2} \equiv g^D \pmod p$.

Observe that D depends on i and j is a rather complicated way – both appear as both coefficients and exponents in the expression for D. Therefore it is hopeless for Eve to try to rig them up to give her a specific document D.

2. Textbook exercise 5.24.

Solution.

We can interpret the data in the problem statement as probabilities as follows.

$$\begin{aligned} \Pr(\text{Urn 1}) = \Pr(\text{Urn 2}) &= \frac{1}{2} \\ \Pr(\text{Pencil}|\text{Urn 1}) &= \frac{7}{10} \\ \Pr(\text{Pencil}|\text{Urn 2}) &= \frac{4}{12} = \frac{1}{3} \end{aligned}$$

These will be the inputs needed to calculate the probabilities in the problem.

(a) Using equation 5.20 from the textbook:

$$\begin{aligned} \Pr(\text{Pencil}) &= \Pr(\text{Pencil}|\text{Urn 1}) \Pr(\text{Urn 1}) + \Pr(\text{Pencil}|\text{Urn 2}) \Pr(\text{Urn 2}) \\ &= \frac{7}{10} \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{2} \\ &= \frac{31}{60} \end{aligned}$$

(b) Using Bayes' formula and part (a):

$$\begin{aligned}\Pr(\text{Urn 1}|\text{Pencil}) &= \Pr(\text{Pencil}|\text{Urn 1})\frac{\Pr(\text{Urn 1})}{\Pr(\text{Pencil})} \\ &= \frac{7}{10}\frac{1/2}{31/60} \\ &= \frac{7}{10}\cdot\frac{30}{31} \\ &= \frac{21}{31}.\end{aligned}$$

(c) We can write the following conditional probabilities involving two pencils, by counting outcomes in each of the two cases.

$$\begin{aligned}\Pr(\text{Two pencils}|\text{Urn 1}) &= \binom{7}{2}/\binom{10}{2} \\ &= \frac{21}{45} \\ &= \frac{7}{15} \\ \Pr(\text{Two pencils}|\text{Urn 2}) &= \binom{4}{2}/\binom{12}{2} \\ &= \frac{6}{66} \\ &= \frac{1}{11}\end{aligned}$$

Therefore, proceeding as in part (a):

$$\begin{aligned}\Pr(\text{Two pencils}) &= \Pr(\text{Two pencils}|\text{Urn 1})\Pr(\text{Urn 1}) + \Pr(\text{Two pencils}|\text{Urn 2})\Pr(\text{Urn 2}) \\ &= \frac{7}{15}\frac{1}{2} + \frac{1}{11}\frac{1}{2} \\ &= \frac{92}{330}\end{aligned}$$

3. Textbook exercise 5.28.

Solution.

Once m is chosen, the outcomes of each run of the algorithm are independent events. Therefore:

$$\begin{aligned}\Pr(\text{No } N \text{ times}|m \text{ has property } A) &\leq (1-p)^N \\ \Pr(\text{No } N \text{ times}|m \text{ doesn't have property } A) &= 1\end{aligned}$$

Therefore, applying Bayes' formula gives the following.

$$\begin{aligned}\Pr(A^c|N \text{ times}) &= \Pr(N \text{ times}|A^c)\frac{\Pr(A^c)}{\Pr(N \text{ times}|A)\Pr(A)+\Pr(N \text{ times}|A^c)\Pr(A^c)} \\ &\geq 1\cdot\frac{\delta}{(1-p)^N(1-\delta)+1\cdot\delta} \\ &\geq \frac{\delta}{(1-p)^N(1-\delta)+\delta}\end{aligned}$$

4. Textbook exercise 5.29.

   Solution.

   (a) Using the formula from the previous problem, this probability is at least $\frac{9/10}{(\frac{1}{4})^{25}\frac{1}{10}+\frac{9}{10}} = \frac{9}{10/4^{25}+9}$.

   (b) This new probability would be $\frac{9}{10/4^{100}+9}$. This might as well be equal to 1.

   (c) The probability that m does not have property A, given N negative test results, is at least $\frac{99/100}{\frac{1}{100\cdot 2^N}+\frac{99}{100}} = \frac{99}{2^{-N}+99}$. In fact, whenever $N \geq 0$ this is at least 99/100, so in fact you are 99% confident before doing any experiments at all (this makes sense, since your prior assumption is that there is a 99% chance that m does not have property A, and negative test results will only increase the certainty of your belief).

   (d) We must solve the inequality $\frac{99}{2^{-N}+99} \geq 1-\frac{1}{1{,}000{,}000}$. This is equivalent to $\frac{1}{1{,}000{,}000} \geq \frac{2^{-N}}{2^{-N}+99} = \frac{1}{1+99\cdot 2^N}$, i.e. $1+99\cdot 2^N \geq 1{,}000{,}000$. The minimum such N is $\log_2(999{,}999/99) = \log_2(10{,}101) \approx 13.3$. So $N = 14$ is sufficient to ensure that the odds of a million to one that m does not have property A.

5. Textbook exercise 5.39.

   Solution.

   The number 304 is a collision: it is both g^{234} and $hg^{399} \pmod p$. Therefore $h \equiv g^{234-399} \pmod{811}$, so the discrete logarithm is $-165 \equiv 645 \pmod{810}$. Indeed, $10^{645} \equiv 106 \pmod{811}$.

6. Suppose that n is an odd integer greater than 1, and that we are selecting an element $a \in \mathbf{Z}/n$ uniformly at random. This problem will investigate the probability that a is a unit. Let $p_1, p_2, \cdots, p_\ell$ be the distinct prime factors of n (e.g. if $n = 45$, $\ell = 2$ and the primes are 3 and 5; note that 3 occurs only once in the list even though 3^2 divides n).

   (a) Let E_i denote the event that p_i does not divide a, and let F denote the event that a is a unit modulo n. Prove that $F = E_1 \cap E_2 \cap \cdots \cap E_\ell$.

   Solution. The number a is a unit if and only if it has no common factor with n. This is equivalent to having to common prime factor. This is equivalent to saying that for each i, $p_i \nmid a$. Therefore F holds if and only if each E_i holds, i.e. F is the intersection of the events E_i.

(b) Prove that the probability of F is equal to the product of the probabilities of the E_i (a stronger fact is that the events E_i are independent events, which is one way to prove this).

Solution.

Let $P = p_1 p_2 \cdots p_\ell$. Since $P = p_1 p_2 \cdots p_\ell$ divides n, it follows that all of the resides classes (mod P) are equally likely to occur as the residue of a when a is selected uniformly at random. Now, the resides $a\%P$ is uniquely determined by the residues $a_i = a\%p_i$, and every possible choice of this list $a_1, a_2, \cdots, a_\ell$ occurs exactly once, by the Chinese Remainder Theorem. Therefore we may regard the ℓ values $a_1, a_2, \cdots, a_\ell$ as uniformly distributed *independent* random variables. The event E_i is precisely the event that $a_i \neq 0$, hence all of these events are independent. It follows that $\Pr(E_1 \cap \cdots \cap E_\ell) = \Pr(E_1)\Pr(E_2)\cdots\Pr(E_\ell)$.

(c) Compute the probability of E_i and deduce a formula for the probability of F.

Solution.

The probability that $a_i = 0$ is $\frac{1}{p_i}$, so the probability of E_i is $1 - \frac{1}{p_i}$, and the probability of F is

$$\Pr(F) = \prod_{i=1}^{\ell} \left(1 - \frac{1}{p_i}\right).$$

(d) Use this formula to deduce a formula for $\phi(n)$ in terms of n and the primes p_i.

Solution.

$$\phi(n) = n \prod_{i=1}^{\ell} \left(1 - \frac{1}{p_i}\right).$$

Another common way to write this is $\phi(n) = \prod_{i=1}^{\ell} p_i^{e_i - 1}(p_i - 1)$, where e_i is the multiplicity of p_i in the prime factorization of n.

Remark. At various times while discussing RSA, we allowed ourselves to assume that the message transmited is a unit modulo N. Part (c) of this problem justifies why this is a reasonable assumption, given the size of the primes in RSA.

7. Suppose that n and p_i are as in the previous problem. Suppose that we are choosing an integer a uniformly at random from the unit group $(\mathbf{Z}/n)^\times$. This problem will investigate the probability that a is a square, i.e. that $x^2 \equiv a \pmod{n}$ has a solution.

(a) Let E_i denote the event that $x^2 \equiv a \pmod{p_i}$ has a solution and let F denote the event that $x^2 \equiv a \pmod{n}$ has a solution. Prove that $F \subseteq E_1 \cap \cdots \cap E_\ell$. (*Note.* In fact, these events are identical, so we could write $=$ instead of $\subseteq$, but this is more difficult to prove.)

Solution.

Suppose that $a \in F$, i.e. a is a square modulo n. It suffices to show that $a \in E_i$ for each i, which will imply that a lies in the intersection of these sets, by definition. Now, if $a \in F$, then there exists x such that $a \equiv x^2 \pmod{n}$. But being congruent modulo n implies beign congruenct modulo any factor of n (by the transitivity of divisibility), so

in fact $a \equiv x^2 \pmod{p_i}$ for all i (with the same value of x). This means that event E_i holds. Hence $a \in \bigcap E_i$. It follows that $F \subseteq \bigcap E_i$.

(b) Assume that the probability of E_i is $\frac{1}{2}$ (I encourage you to try to prove it, but you are free to assume it without proof). Prove that

$$\Pr(F) \leq \frac{1}{2^\ell}.$$

(you should proceed in a similar manner as in the previous problem).

Solution.

Define $a_i = a\%p_i$ as in the solution of part (b) of the previous problem. Then as before, these are *independent* random variables, each uniformly distributed among the units modulo p. Now, the event E_i depends only on the variable a_i, since it is a property of $a \pmod{p_i}$. Hence these events are independent, and $\Pr(\bigcap E_i) = \prod \Pr(E_i)$. By assumption, each of these $\Pr(E_i)$ is $\frac{1}{2}$, so $\Pr(\bigcap(E_i)) = \frac{1}{2^\ell}$. By part (a), $\Pr(F) \leq \Pr(\bigcap E_i)$, hence $\Pr(F) \leq \frac{1}{2^\ell}$.

Addendum: here is a proof of the assumption. We first claim that if p is an odd prime and $a \not\equiv 0 \pmod{p}$, then $x^2 \equiv a \pmod{p}$ has either 2 or 0 solutions. To see this, observe that x_0 is one solution, then any other solution x satisfies $x^2 \equiv x_0^2 \pmod{p}$, hence $(x - x_0)(x + x_0) \equiv 0 \pmod{p}$, so p divides either $x - x_0$ or $x + x_0$, meaning that x is either x_0 or $-x_0 \pmod{p}$. This shows that there are at *most* two solutions. But conversely, if x_0 is one solution, then $-x_0$ is a second solution (these are not equal since $p \neq 2$; this is the only place where we must assume that $p \neq 2$). This proves the claim (we could also cite the result of exercise 3.2 in the textbook, i.e. Problem Set 5, problem 2). Now, consider the function $f(x) = x^2\%p$. Then as x varies over $1, 2, \cdots, p-1$, this function achieves each element in its range exactly twice, by the claim. But this means that the total number of elements in the range is one half of the number of elements in the domain. Therefore the size of the range is $(p-1)/2$. Thus the number of square units modulo p is exactly half the number of units modulo p.

Remark. We discussed a collision algorithm in class that can factor n if two different integers a, b are found such that $a^2 \equiv b^2 \pmod{n}$. This algorithm is more likely to succeed if there are relatively few squares modulo n (hence greater odds of a collision). This problem shows that this algorithm is most effective when n has many distinct prime factors, since a smaller fraction of the units are squares. In particular, this algorithm poses little danger to RSA keys.

8. Suppose that p, q are two prime numbers such that $p \equiv 1 \pmod{q}$. Suppose that we choose an element a from $(\mathbf{Z}/p)^\times$ uniformly at random.

   (a) What is the probability that $\mathrm{ord}_p(a) = q$?

   (b) What is the probability that $\mathrm{ord}_p(a^{(p-1)/q}) = q$?

   Remark. This shows why the suggested approach to finding DSA parameters (as in PSet 6 number 12) is better than choosing a at random until you find one that is order q.

Solution.

By problem set 5, problem 2 (problem 3.2 in the textbook), the number of solutions to the equation $x^e \equiv 1 \pmod{p}$ is equal to $\gcd(e, p-1)$ (the problem cited guarantees that this is the number of solutions whenever there are any solutions at all, and we know that one solution exists, namely $x = 1$). In particular, assuming that $e \mid (p-1)$, we deduce that there are exactly e solutions. This is the key fact in both calculations.

(a) The order of a is q if and only if $a^q \equiv 1 \pmod{p}$ but $a \not\equiv 1 \pmod{p}$ (since q is prime, the only possible orders of an element such that $a^q \equiv 1$ are 1 and q, so we only need to rule out 1). Since $q \mid (p-1)$, there are q elements a such that $a^q \equiv 1 \pmod{p}$, and one of them is $a = 1$. So there are $q-1$ elements of order q; the probability of finding one is $\frac{q-1}{p-1}$. For q, p as in DSA, this probability is negligible since p is many orders of magnitude larger than q; this is not a good way to find order-q elements.

(b) Since $(a^{(p-1)/q})^q \equiv 1 \pmod{p}$ (by Fermat's little theorem), we know that the order of $a^{(p-1)/q}$ is either 1 or q. Therefore $a^{(p-1)/q} \pmod{p}$ has order q if and only if it is not 1. Now, since $\frac{p-1}{q}$ is a factor of $p-1$, the number of a such that $a^{(p-1)/q} \equiv 1 \pmod{p}$ is exactly $\frac{p-1}{q}$, hence the probability that $a^{(p-1)/q} \equiv 1 \pmod{p}$ is exactly $\frac{1}{q}$ when a is chosen uniformly among all $p-1$ units. Therefore the probability that $a^{(p-1)/q} \pmod{p}$ has order equal to q is $1 - \frac{1}{q}$. In contrast to part (a), this is essentially a sure thing, given that q is typically chosen to be at least on the order of 2^{160}.

9. Suppose that Samantha signs M different documents with the same DSA public key. Suppose the number of possible ephemeral keys is N. We have seen that she must be sure not to use the same ephemeral key twice.

(a) Suppose that $k_1, \cdots, k_M$ are the ephemeral keys that Samantha chooses. Assume that each was chosen uniformly at random from all N possible choices. Calculate the expected value of the number of pairs $\{i, j\}$ of two distinct indices such that $k_i = k_j$.

Solution.

There are $\binom{M}{2}$ pairs $\{i, j\}$, and the probability that any particular pair gives $k_i = k_j$ is $\frac{1}{N}$. By linearity of expectation, the expected value of the number of such pairs is therefore $\binom{M}{2}/N$.

(b) Prove that the probability that *some two* keys are the same is less than your answer from part (a).

Solution.

One equivalent way to express the expected value from part (a) is formula 5.27 form the textbook, which says in this case:

$$\mathbf{E}(\text{equal pairs}) = \sum_{n=0}^{\binom{M}{2}} n \cdot \Pr(\text{there are exactly } n \text{ matching pairs})$$

Now, the events "there are exactly n matching pairs" are disjoint events. Therefore we can obtain the probability that there is *at least one* matching pair as their sum (for $n > 0$):

$$\Pr(\text{at least one matching pair}) = \sum_{n=1}^{\binom{M}{2}} \Pr(\text{there are exactly } n \text{ matching pairs})$$

But we can now compare these formulas, by noting simply that the coefficient n in the first is at least 1 whenever $n \neq 0$.

$$\begin{aligned}
\mathbf{E}(\text{equal pairs}) &= \sum_{n=0}^{\binom{M}{2}} n \cdot \Pr(\text{there are exactly } n \text{ matching pairs}) \\
&= \sum_{n=1}^{\binom{M}{2}} n \cdot \Pr(\text{there are exactly } n \text{ matching pairs}) \text{ (the } n = 0 \text{ term is equal to 0)} \\
&> \sum_{n=1}^{\binom{M}{2}} 1 \cdot \Pr(\text{there are exactly } n \text{ matching pairs}) \\
&> \Pr(\text{at least one matching pair})
\end{aligned}$$

It follows that the probability that *some* two keys are equal is less than $\binom{M}{2}/N$. In fact, the probability is very close to this expected value when M is relatively small, because the contributions to the expected value coming from the possibility that there are more than one matching pair is extremely small.

(c) Suppose that $M = 2^{32}$ (this is what M would be if Samantha signs one document per second for 120 years). Suppose that the parameters (p, q) used for DSA are such that p is 512 bits long and q is 160 bits long (as in the original DSA standard from 1991). Show that the probability that Samantha ever chooses the same ephemeral key twice is less than $\frac{1}{2^{96}}$.

Solution.

From the previous part, this probability is less than $\binom{M}{2}$, which is less than $\frac{1}{2}M^2 = 2^{63}$. Since q is 160 bits long, we know that $q > 2^{159}$, so $q - 1 \geq 2^{159}$. The number of possible ephemeral keys is $q - 1$, so $N \geq 2^{159}$. It follows that $\binom{M}{2}/N < 2^{63}/2^{159} = 2^{-96}$, which therefore furnishes a rather strong upper bound on the probaility of this mistake. For all practical purposes, this probability is zero.

Remark. These odds are a lot worse for Samantha if she doesn't choose a good random number generator.

10. Textbook exercise 6.1.

Solution.

(a) We wish to compute $(0, 2) \oplus (3, -5)$.

Secant slope is:

$$\begin{aligned} \lambda &= \frac{(2)-(-5)}{(0)-(3)} \\ &= -7/3 \end{aligned}$$

The third intersection point is:

$$\begin{aligned} x_3 &= (-7/3)^2 - (0) - (3) \\ &= 22/9 \\ y_3 &= (-7/3)\cdot(22/9) + (2) \\ &= --100/27 \\ \Rightarrow (0,2)\oplus(3,-5) &= (22/9, 100/27) \end{aligned}$$

(b) First, we wish to compute $P \oplus P = (0,2) \oplus (0,2)$.

Tangent slope is:

$$\begin{aligned} \lambda &= \frac{3\cdot(0)^2 + -2}{2\cdot(2)} \\ &= -1/2 \end{aligned}$$

The third intersection point is:

$$\begin{aligned} x_3 &= (-1/2)^2 - (0) - (0) \\ &= 1/4 \\ y_3 &= (-1/2)\cdot(1/4) + (2) \\ &= 15/8 \\ \Rightarrow (0,2)\oplus(0,2) &= (1/4, -15/8) \end{aligned}$$

Next, we wish to compute $Q \oplus Q = (3,-5) \oplus (3,-5)$.

Tangent slope is:

$$\begin{aligned} \lambda &= \frac{3\cdot(3)^2 + -2}{2\cdot(-5)} \\ &= -5/2 \end{aligned}$$

The third intersection point is:

$$\begin{aligned} x_3 &= (-5/2)^2 - (3) - (3) \\ &= 1/4 \\ y_3 &= (-5/2)\cdot(1/4) + (5/2) \\ &= 15/8 \\ \Rightarrow (3,-5)\oplus(3,-5) &= (1/4, -15/8) \end{aligned}$$

This is not a typo; in fact $P \oplus P = Q \oplus Q$, which is an unusual coincidence.

(c) To find $P \oplus P \oplus P$, use the answer to part (b); we wish to compute $(1/4, -15/8) \oplus (0, 2)$.

$$\begin{aligned}
\text{Secant slope is:} & \\
\lambda &= \frac{(-15/8) - (2)}{(1/4) - (0)} \\
&= -31/2 \\
\text{The third intersection point is:} & \\
x_3 &= (-31/2)^2 - (1/4) - (0) \\
&= 240 \\
y_3 &= (-31/2) \cdot (240) + (2) \\
&= --3718 \\
\Rightarrow (1/4, -15/8) \oplus (0, 2) &= (240, 3718)
\end{aligned}$$

To find $Q \oplus Q \oplus Q$, use the answer to part (b) again; we wish to compute $(1/4, -15/8) \oplus (3, -5)$.

$$\begin{aligned}
\text{Secant slope is:} & \\
\lambda &= \frac{(-15/8) - (-5)}{(1/4) - (3)} \\
&= -25/22 \\
\text{The third intersection point is:} & \\
x_3 &= (-25/22)^2 - (1/4) - (3) \\
&= -237/121 \\
y_3 &= (-25/22) \cdot (-237/121) + (-35/22) \\
&= 845/1331 \\
\Rightarrow (1/4, -15/8) \oplus (3, -5) &= (-237/121, -845/1331)
\end{aligned}$$

11. Textbook exercise 6.2.

Solution.

(a) To find $P \oplus Q$, we wish to compute $(-1, 4) \oplus (2, 5)$.

Secant slope is:

$$\begin{aligned}\lambda &= \frac{(4)-(5)}{(-1)-(2)} \\ &= 1/3\end{aligned}$$

The third intersection point is:

$$\begin{aligned}x_3 &= (1/3)^2-(-1)-(2) \\ &= -8/9 \\ y_3 &= (1/3)\cdot(-8/9)+(13/3) \\ &= 109/27 \\ \Rightarrow (-1,4)\oplus(2,5) &= (-8/9,-109/27)\end{aligned}$$

To find $P \ominus Q$, note that $\ominus Q = (2,-5)$. So we wish to compute $(-1,4)\oplus(2,-5)$.

Secant slope is:

$$\begin{aligned}\lambda &= \frac{(4)-(-5)}{(-1)-(2)} \\ &= -3\end{aligned}$$

The third intersection point is:

$$\begin{aligned}x_3 &= (-3)^2-(-1)-(2) \\ &= 8 \\ y_3 &= (-3)\cdot(8)+(1) \\ &= --23 \\ \Rightarrow (-1,4)\oplus(2,-5) &= (8,23)\end{aligned}$$

(b) To find $2P = P \oplus P$, we wish to compute $(-1,4)\oplus(-1,4)$.

Tangent slope is:

$$\begin{aligned}\lambda &= \frac{3\cdot(-1)^2+0}{2\cdot(4)} \\ &= 3/8\end{aligned}$$

The third intersection point is:

$$\begin{aligned}x_3 &= (3/8)^2-(-1)-(-1) \\ &= 137/64 \\ y_3 &= (3/8)\cdot(137/64)+(35/8) \\ &= 2651/512 \\ \Rightarrow (-1,4)\oplus(-1,4) &= (137/64,-2651/512)\end{aligned}$$

To find $2Q = Q \oplus Q$, we wish to compute $(2,5) \oplus (2,5)$.

$$\begin{aligned}
\text{Tangent slope is:} & \\
\lambda &= \frac{3 \cdot (2)^2 + 0}{2 \cdot (5)} \\
&= 6/5 \\
\text{The third intersection point is:} & \\
x_3 &= (6/5)^2 - (2) - (2) \\
&= -64/25 \\
y_3 &= (6/5) \cdot (-64/25) + (13/5) \\
&= --59/125 \\
\Rightarrow (2,5) \oplus (2,5) &= (-64/25, 59/125)
\end{aligned}$$

For the bonus, here is a complete list of integral points:

$$(-2,3), (-1,4), (2,5), (4,9), (8,23), (43,282), (52,375), (5234,378661)$$

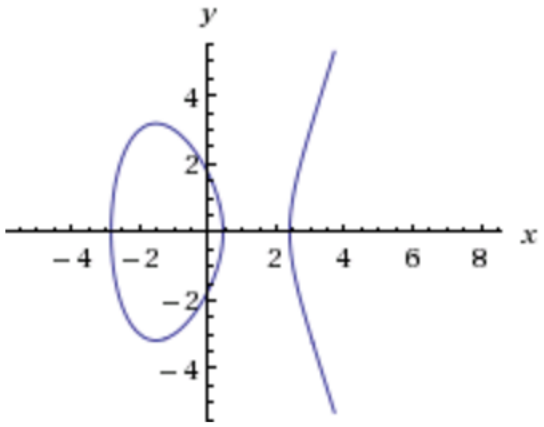
and their inverses:

$$(-2,-3), (-1,-4), (2,-5), (4,-9), (8,-23), (43,-282), (52,-375), (5234,-378661)$$

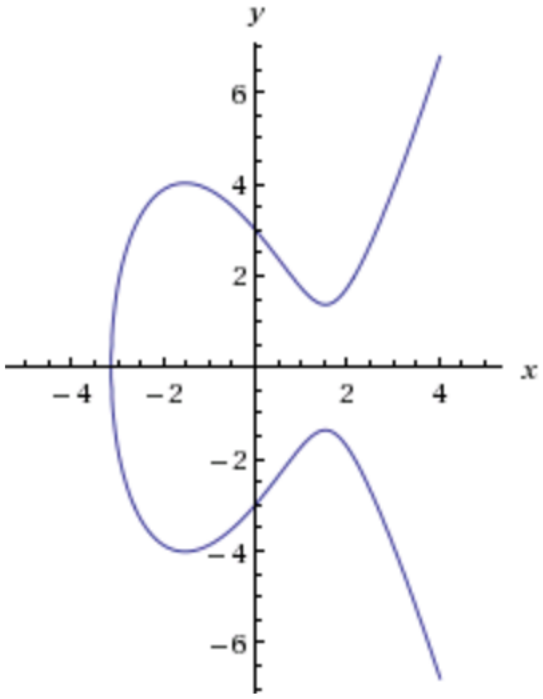
12. Textbook exercise 6.4. (You can use whatever computing equipment you wish to help sketch the pictures).

Solution.

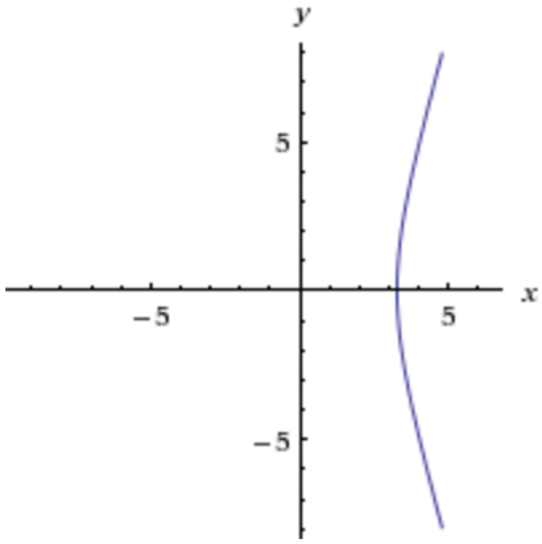
Here are some graphs from Wolfram Alpha. Note that the last two are singular; they are the so-called *nodal cubic* and *cuspidal cubic* curves. They are not suitable for cryptography; although you can define a group law on them (after removing the singular point), the discrete logarithm problem is too easy to solve for them (specifically, they can both be given a *rational parameterization*, which makes the discrete logarithm problem easy to solve afterward).



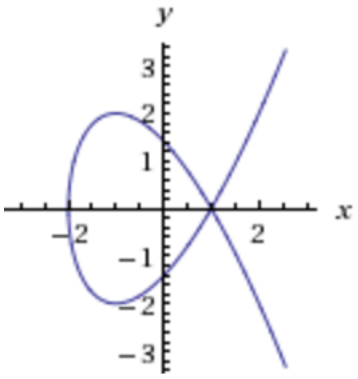
(a)



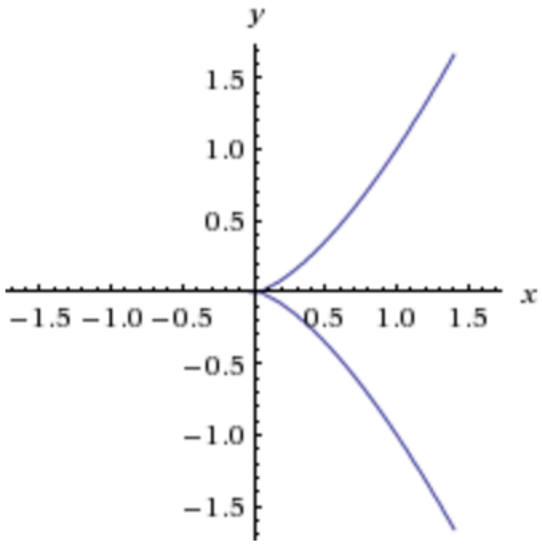
(b)



(c)



(d)



(e)

Programming problems

13. Given integers A, B, and p, and two points P, Q on the elliptic curve $Y^2 = X^3 + AX + B$ over $\mathbf{F}_p$, compute the sum $P \oplus Q$.

Solution.

We must implement the formulas from the textbook. One tricky aspect is to make sure that the edge cases are dealt with properly. Below I've shown one way to deal with them: first check whether one of the two points is $\mathcal{O}$, then compute the numerator and denominator of the slope of the line through P and Q (which may be a tangent line if $P = Q$), return $\mathcal{O}$ if the denominator would be 0 (i.e. the line, secant or tangent, is vertical), and otherwise computing and inverting the third intersection point.

I omit the source for the `inv_mod(a,m)` function, since it has been shown several times before.

```
### Omitted: code for inv_mod(a,m) (modular inverse)

def add(P,Q,A,B,p):
        if P==0: return Q
        if Q==0: return P
        if P==Q:
                x = P[0]
                y = P[1]
                dy = (3*x*x + A) % p
                dx = (2*y) % p
        else:
                dy = (P[1]-Q[1]) % p
                dx = (P[0]-Q[0]) % p
        if dx == 0: # Vertical tangent/secant line
                return 0
        sl = (dy*inv_mod(dx,p)) % p #Slope of line
        it = (P[1] - P[0]*sl) % p #y-intercept of line
        x = (sl**2 - P[0]-Q[0]) % p
        y = (-(sl*x+it)) % p
        return (x,y)


### I/O
def read_point():
  ls = map(int,raw_input().split())
  if len(ls) == 2: return tuple(ls)
  else: return 0

A,B,p = map(int,raw_input().split())
P = read_point()
Q = read_point()
R = add(P,Q,A,B,p)
```

```
if R==0: print 0
else: print R[0],R[1]
```

14. Given A, B, p as before, one point P on the elliptic curve, and an integer n (positive or negative), compute the point nP.

    For this, we essentially duplicate the code of the fast-powering algorithm from modular arithmetic, replacing modular multiplication with Elliptic curve addition.

```
### Omitted: soure for all functions from the previous problem.

def inv_pt(P,A,B,p):
        if P == 0: return 0
        return (P[0],(-P[1])%p)

def mult(P,n,A,B,p):
        if n<0:
                P = inv_pt(P,A,B,p)
                n = -n
        res = 0
        while n > 0:
                if n%2 == 1:
                        res = add(res,P,A,B,p)
                n /= 2
                P = add(P,P,A,B,p)
        return res

### I/O
def read_point():
  ls = map(int,raw_input().split())
  if len(ls) == 2: return tuple(ls)
  else: return 0

A,B,p = map(int,raw_input().split())
P = read_point()
n = int(raw_input())

res = mult(P,n,A,B,p)
if res == 0: print 0
else: print res[0],res[1]
```

15. Given A, B, p as before, determine the number of points on the elliptic curve, and print the "trace of Frobenius" t_p, as defined on page 309 of the textbook.

    Solution.

For p of the size in the test cases, we can use the following procedure.

- Make a list of all of the squares modulo p, by squaring each possible residue class and storing them in a set.
- Trying each possible value of x. If $x^3 + Ax + B$ is 0, there is one point $(x, 0)$ with this x. If $x^3 + Ax + B$ is a nonzero square, there are two points $(x, \pm y)$, which if it is a non-square, there are zero.
- Add one for the point $\mathcal{O}$ at infinity.

We can compute the size of the group in this way; subtracting from $p + 1$ gives the trace of Frobenius. Here is an implementation.

```
def count_tf(A,B,p):
        sq = set()
        for a in range(1,p): sq.add(a*a%p)
        pts = 1
        for x in range(p):
                rhs = (x**3 + A*x + B)%p
                if rhs == 0: pts += 1
                if rhs in sq: pts += 2
        return p+1 - pts

### I/O
A,B,p = map(int,raw_input().split())
print count_tf(A,B,p)
```

Remark. There are very efficient algorithms for this problem, sufficient for primes of the size needed for cryptography, but they are beyond the scope of this course. For this problem, p will be no more than 15 bits in length, and a more naive method will suffice. See the online problem statement for a further hint.

16. Given parameters p, q, g for DSA and a public key A, find a document D with a valid signature (S_1, S_2). Note that you will *not* be able to decide on D in advance (since DSA is secure, as far as we know), but instead compute it along with a signature. See exercise 4.7 in the textbook for a suggestion; you will need to adapt the method in that exercise from ElGamal to DSA.

Solution.

We can attempt to begin as in the ElGamal version, by choosing i and j at random and setting

$$S_1 = g^i A^j \% p \% q.$$

Since the order of g is the second prime q in DSA, was can choose i and j from $\mathbf{Z}/q$.

Now, to decide how to write down S_2 and D, we can consider the verification equation that we wish to satisfy.

$$g^{S_2^{-1}D} A^{S_2^{-1}S_1} \% p \% q = S_1$$

It is very hard to work with this equation: the $\%p\%q$ term rules doing virtually any manipulation of the exponent. However, we know that since $S_1 = g^i A^j \pmod p$, *it is sufficient* to solve the following $\pmod p$ congruence instead.

$$g^{S_2^{-1}D} A^{S_2^{-1}S_1} \equiv g^i A^j \pmod p$$

This congruence can be transformed from a $\pmod p$ congruence into a $\pmod q$ congruence by expressing both sides as powers of g (and recalling that g has order q).

$$\begin{aligned} g^{S_2^{-1}D + aS_2^{-1}S_1} &\equiv g^{i+aj} \pmod p \\ \Leftrightarrow S_2^{-1}D + aS_2^{-1}S_1 &\equiv i + aj \pmod q \end{aligned}$$

Now, the essential difficulty in solving this congruence, for Eve, is that *she doesn't know what a is*. So she needs to somehow solve a congruence that includes a completely unknown quantity. The most foolproof way to do this is to choose the variables in such a way that the desired congruence is true *no matter what a is*. There turns out to be a way to do this: make sure that all of the a terms cancel. This can be arranged by making sure that the following congruences hold.

$$\begin{aligned} S_2^{-1}S_1 &\equiv j \pmod q \text{ (to force the } a \text{ terms to cancel} \\ S_2^{-1}D &\equiv i \pmod q \text{ (to ensure that the remaining terms match)} \end{aligned}$$

Remember that we've already chosen i, j, and S_1, so we just solve for S_2 and D as follows.

$$\begin{aligned} S_2 &\equiv j^{-1}S_1 \pmod q \\ D &\equiv iS_2 \pmod q \\ &\equiv ij^{-1}S_1 \pmod q \end{aligned}$$

These choices of D, S_1, S_2 will indeed satisfy the verification equation, hence form a valid signed documnet. Note that in the end, this was virtually the same as the solution to problem 1, except that we remove two minus signs and replace $\pmod{p-1}$ with $\pmod q$. An implementation is below.

```
### Omitted: code for inv_mod(a,m) (modular inversion)

import random
random.seed()
```

```
def forge(p,q,g,A):
        i = random.randrange(1,q)
        j = random.randrange(1,q)
        s1 = pow(g,i,p)*pow(A,j,p) % p % q
        s2 = inv_mod(j,q)*s1 % q
        d = i * s2 % q
        return d,s1,s2

### I/O
p,q,g,A = map(int,raw_input().split())
d,s1,s2 = forge(p,q,g,A)
print d,s1,s2
```