EECS 70 Discrete Math and Probability
Spring 2020 UC Berkeley

Midterm 1

Exam solutions

Section 0: Pre-exam questions (3 points)

1. **What other courses are you taking this term? (1 pt)**

   Solutions: Insert courses here... Typical choices: 61C, Physics 7B.

2. **Think back to remember a moment where something in some course really clicked for you. Describe it. How did you feel? (2 pts)**

   Solutions: Lots of 70 related answers possible: The vector perspective on the CRT was particularly popular. Lots of 16A related answers as well.

PRINT your name and student ID: _____

3. Short Answer (8 pts)

These are all short answer questions. Write your final answers inside the appropriate boxes. You don't need to justify your answer, just show whatever work that you actually had to do outside the box.

(a) (2 pts) **Fill in the truth-table for** $\neg Q \implies P$:

Solutions:

P	Q	$\neg Q \implies P$
T	T	T
T	F	T
F	T	T
F	F	F

(b) (1 pt) **Find the multiplicative inverse of 18 mod 73.** Your answer should be an integer in $\{0,1,2,\ldots,72\}$. To spare you some calculations, note that: $73-4*18=1$

Solutions: Taking the equation given mod 73, we see that $-4 \cdot 18 \equiv 1 \pmod{73}$. *Thus*, $-4 \equiv 69$ is the multiplicative inverse of 18 mod 73.

(c) (3 pts) **Find** 42^{1601} mod 17. Your answer should be an integer in $\{0,1,2,\ldots,16\}$.

Solutions:

$$42^{1601} \equiv 42^{16*100+1} \equiv (42^{16})^{100}42 \equiv 42 \equiv 8 \pmod{17}$$

(d) (2 pts) For distinct primes p and q, **how many numbers in the set** $\{0,1,\ldots,pq-1\}$ **do not have a multiplicative inverse modulo** pq**?**

Solutions: $p+q-1$. There are q multiples of p in the set: $0,p,2p,\ldots(q-1)p$. There are also p multiples of q in the set: $0,q,2q,(p-1)q$. However, we counted 0 twice, so we subtract 1.

PRINT your name and student ID: _____

4. A proof (6 pts)

Suppose x, y are integers. **Prove that if xy and $x+y$ are both even, then both x and y must be even.**

Solutions: This can be done as a proof by contraposition. The negation of "both x and y must be even" is "at least one of x or y is odd."

The negation of "xy and $x+y$ are both even" is "xy is odd or $x+y$ is odd."

So we want to prove: "If least one of x or y is odd, then xy is odd or $x+y$ is odd."

Starting with x odd, we have that there exists an integer k so that $x = 2k+1$. Now, let's consider cases for y: it is either even or odd.

- Suppose y is even. Then there exists j so that $y = 2j$. In this case $x+y = 2k+1+2j = 2(k+j)+1$ which is odd. And we are done.
- Suppose y is odd. Then there exists j so that $y = 2j+1$. In this case $xy = (2k+1)(2j+1) = 4kj+2k+2j+1 = 2(2kj+k+j)+1$ which is odd. And we are done.

This leaves starting with y odd. So there is an integer j so that that $y = 2j+1$. While in principle there are cases here as well, we have already done the x is odd case above. So this just leaves the x is even case. This means there is an integer k so that $x = 2k$. At this point, we can compute $x+y = 2k+2j+1 = 2(k+j)+1$ which is odd. And we are done.

Since the contrapositive has been proved, the original statement has been proved.

PRINT your name and student ID: _____

5. Multi-GCD (8 pts)

In this problem we quickly explore a generalization of the GCD and EGCD algorithms to more than two numbers at a time. For natural numbers $n_1, n_2, \ldots, n_k \in \mathbb{N}$, let $\text{GCD}(n_1, n_2, \ldots, n_k)$ denote the largest natural number which divides each of $n_1, n_2, \ldots, n_k$.

One thing that we observe is that $\text{GCD}(a,b,c) = \text{GCD}(a, \text{GCD}(b,c))$.

This fact also allows us to naturally extend the EGCD algorithm to more than two integers. **Find** $a,b,c \in \mathbb{Z}$ **such that** $21a + 15b + 35c = 1$**.**

Here are some calculations that might be helpful: $35 - 2*15 = 5$ and $21 - 4*5 = 1$.

Solutions: Using the spirit of the hint, start with the observation that $\text{GCD}(35,15) = 5$. Then we can use EGCD to find $b', c' \in \mathbb{Z}$ such that $15b' + 35c' = 5$. Proceeding,

$$\begin{aligned} 35 &= 15(0) + 35(1) \\ 15 &= 15(1) + 35(0) \\ 5 &= 15(-2) + 35(1) \end{aligned}$$

so we can set $b' = -2$, $c' = 1$.

(This didn't need to be done since we just gave you those computations to save you time.)

The observation in the problem about the nature of the GCD tells you that what we want to do is consider the $\text{GCD}(21,5)$ next.

Note that $15(-2k) + 35(k) = 5k$, so if we find $a,k \in \mathbb{Z}$ such that $21a + 5k = 1$, then our final answer will be $a = a, b = -2k, c = k$. Proceeding as before,

$$\begin{aligned} 21 &= 21(1) + 5(0) \\ 5 &= 21(0) + 5(1) \\ 1 &= 21(1) + 5(-4), \end{aligned}$$

as the hint we gave you showed. Putting everything together treating $k = -4$ and $a = 1$, we get $a = 1, b = 8, c = -4$ is a possible solution.

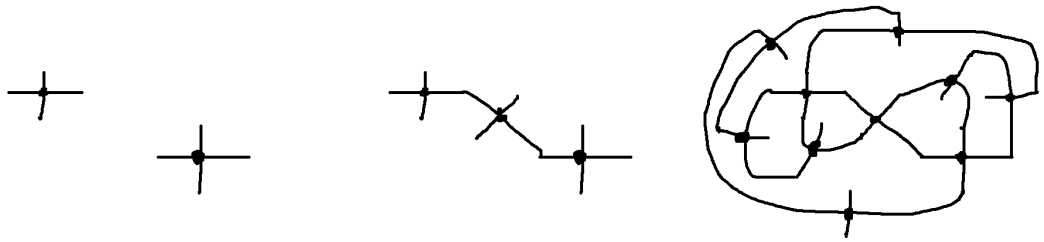
Of course, this is not the unique solution and other solutions can also work.

PRINT your name and student ID: _____

6. Graph Game (21 pts)

Alice and Bob are playing a game where they take turns drawing on a piece of paper. Alice goes first and Bob goes second. The rules of the game are as follows:

- We start off with n crosses, i.e. spots with four free ends.
- During their turn, a player must connect any two free ends with a curve, provided the curve does not intersect any existing curves or crosses, and then add a stroke across the curve to create two more free ends.
- If any player cannot move, they lose the game.



Above, we have an example of states of a game with $n = 2$ at the start, after one move, and after the game has concluded.

(a) (6 pts) Note that the graph formed by the curves is always planar. Assume that the center of each cross is a vertex and any intersection of a curve with a stroke is also a vertex. The curves connecting two vertices are edges. *Note that each move in the game therefore is adding two edges even though it is described as drawing a single curve.*

After m moves have been made, how many vertices, edges, and free ends are there on the paper? Recall that at $m = 0$, we start with $v(0) = n$ vertices, $e(0) = 0$ edges, and $f_e(0) = 4n$ free ends.

Solutions: Each move adds one vertex to the graph $v(m) = n + m$. Each move adds two edges to the graph $e(m) = 2m$. After each move we make two new free ends and use two of the previous free ends so the number of free ends remain constant so $f_e(m) = 4n$.

(b) (3 pts) When the game concludes, the graph must be connected and planar. A planar graph subdivides the plane into faces. **How many free ends can be in a face of this concluding planar graph?**

(HINT: Remember, no more moves can be made when the game is over.)

Solutions: There can be one free end in a face since if there are 2 free ends in a face we can make another move.

(c) (12 pts) Since Alice goes first and Bob and Alice alternate turns, Alice plays odd numbered moves and Bob plays even numbered ones. **For which n does Alice win and for which n does Bob win?** Justify your answer.

(HINT: Euler's formula says that $v + f = e + 2$ for connected planar graphs. The answers from the previous parts can help here.)

Solutions: Each move adds one vertex to the graph $v(m) = n + m$. Each move adds two edges to the graph $e(m) = 2m$. After each move we make two new free ends and use two of the previous free ends so the number of free ends remain constant so $f_e(m) = 4n$.

There can be one free end in a face since if there are 2 free ends in a face we can make another move.

Let m be the number of moves. $v = m + n$, because the game starts with n vertices (one for each cross) and each move adds one vertex. $e = 2m$ because each move adds two edges, one going to each free end connected to the added vertex. $f = 4n$, because the number of free ends does not change throughout the game and at the end each face of the graph contains one and only one free end. Plugging the obtained values into Euler's formula, we have

$$\begin{aligned} v + f - e &= 2 \\ m + n + 4n - 2m &= 2 \\ m &= 5n - 2 \end{aligned}$$

Then if n is even, m is even so Bob, the player that goes second wins. Similarly, if n is odd, m is odd, so Alice, the player that goes first, wins.

PRINT your name and student ID: _____

7. Toy Stability (38 pts)

Edward is babysitting n kids. He has n toys and wants to distribute them to the kids so that every kid gets a toy. Every kid has a preference list towards the n toys. Edward's goal is to reach a **stable pairing**, which is defined to be a pairing in which *there doesn't exist a rogue group of kids who can exchange their toys among each other with everyone in the group getting a more preferred toy.*

Note: in this problem, toys are inanimate and so have no preferences of their own.

(a) (6 pts) Let $n = 3$ and let the preference list of the 3 kids (1, 2, 3) towards 3 toys (A, B, C) be as follows:

Kid	Preference List
1	$A > B > C$
2	$A > C > B$
3	$B > A > C$

Determine whether each of the following is a stable pairing. Fill in the appropriate bubble.

i. (1, A), (2, B), (3, C) ◯ **Stable** ◯ **Not Stable**
Solutions: Not stable. 2 and 3 can exchange toy and both get a more preferred toy.

ii. (1, A), (2, C), (3, B) ◯ **Stable** ◯ **Not Stable**
Solutions: Stable. Note that two kids get their most preferred toy so they don't want to participate in exchanging. So there is no way exchange could happen.

iii. (1, B), (2, C), (3, A) ◯ **Stable** ◯ **Not Stable**
Solutions: Not stable. 1 and 3 can exchange toys.

(b) (10 pts) Edward comes up with an algorithm to distribute the toys via a game. It proceeds in the following steps:

i. Start the epoch counter, a global variable for this algorithm, at 1.

ii. Initially pair each kid with a toy randomly, ask them to hold their tentative toy, and place all kids (holding their tentative toy) in the center of the room. All these kids and toys in the center of the room are considered to be in the game.

iii. Ask each kid still in the game to point to the person in the game who holds their most preferred toy among all the toys **that are still in the game**. Treat this as a directed graph in which the kids in the game are vertices and the edges represent who they are pointing to.

iv. If there is a directed cycle in the graph (including potentially a cycle of length 1 where a person is just pointing to themselves), then arbitrarily pick such a cycle. This is called a "trading cycle."

v. For every kid in the picked trading cycle, assign them the toy that they are pointing to. In addition, give each of these kids a card that says the current epoch number. Send these kids outside the room, thereby removing them and their assigned toys from the game. All the kids still in the game cross off the removed toys from their preference lists.

vi. If there are still kids in the game, increment the epoch number by one and repeat from step iii. Otherwise, the game has concluded and every kid has been assigned a toy.

Execute the algorithm for the following preference list for $n = 6$ starting from the initial pairing $(1,A),(2,B),(3,C),(4,D),(5,E),(6,F)$. You only need to provide the final pairing in the box.

Kid	Preference List
1	$C > B > D > A > F > E$
2	$C > E > F > A > D > B$
3	$C > A > D > E > F > B$
4	$B > E > F > D > A > C$
5	$A > C > B > D > E > F$
6	$B > D > E > F > A > C$

Solutions:

Iteration 1: 3 forms a length 1 cycle. Remove kid 3 and toy C from the game. The resulting pairing doesn't change. Iteration 2: 1's most preferred toy that remains in the game is in 2's hand, 2's is in 5's hand and 5's is in 2's hand. So kid 1, 2 and 5 forms a preference cycle. Implement the exchange and remove the three kids from the game. The resulting pairing is now (1, B), (2, E), (3, C), (4, D), (5, A), (6, F). Iteration 3: Only 4 and 6 are left in the game. They have each other's most preferred toy that remains in the game. They exchange toys and the algorithm terminates as all kids are removed from the game. The resulting pairing is (1, B), (2, E), (3, C), (4, F), (5, A), (6, D).

(c) (8 pts) For the general algorithm given, **prove that if there are still kids in the game, there must be at least one trading cycle (including possibly a cycle of length 1).**

(HINT: Remember, the kids are vertices and the directed edges represent which kid/toy they are pointing at. What must happen as we walk along these edges?)

Solutions: Build a graph with kids as vertices. There exists a directed edge from vertex i to vertex j if i's most preferred toy remaining in the game is possessed by j. Note we allow self-pointing edge (i,i). We start from an arbitrary vertex and walk the graph following the directed edges. Since every vertex has an edge going out from it and there are $|V|$ vertices, we must visit a vertex that's already seen after $|V|$ steps. Therefore, a cycle must exist.

(d) (4 pts) **Prove that the algorithm must terminate if started with a finite number n of kids and toys.** Feel free to assume the previous part, even if you didn't get the proof.

Solutions: We know from part (c) that at least 1 kid must exit the room at each epoch, meaning that the there will be no kids left in at most n epochs. Since n is finite, the algorithm is guaranteed to terminate in a finite time.

(e) (10 pts) **Show that the algorithm produces a stable pairing.**

(HINT: The kids' epoch numbers and the well-ordering principle might be helpful.)

Solutions: We induct on number of iterations. We show that the kids remove from iteration m cannot participate in any mutually beneficial exchange of toys (an exchange of toys that benefits all participants). As all kids are removed from the game at some iteration, this is equivalent to showing we have a stable pairing.

Base case: m = 1 The kids removed from the game in the first iteration has the top toy on their preference list after the exchange. So they cannot participate in any further mutually beneficial exchanges.

Inductive Hypothesis: Assume the kids removed from the game in iteration $i \leq k$ cannot participate in any further mutually beneficial exchanges.

Inductive Step: Want to show that the kids removed in iteration $k+1$ cannot participate in any further mutually beneficial exchanges. Let kid P be an arbitrary kid that is removed in iteration $k+1$. To participate in an exchange, the kid can either get toy from a kid that has already been removed in a previous iteration $i <= k$ or from a kid that remains after iteration k. The first case is impossible by inductive hypothesis. The second case is also impossible: since kid P is removed in iteration $k+1$,

they get their most preferred toy held by kids that remain after iteration k by definition of preference cycle. Therefore, getting a more preferred toy from kids that remain after iteration k is impossible.

Therefore, kids removed in iteration $k+1$ cannot participate in any further mutually beneficial exchanges.

PRINT your name and student ID: _____

8. RSA with three primes, two exponents, and a glitch (15 pts)

Suppose you have three distinct primes p,q,r and positive natural numbers e_1 and e_2 that are both coprime with $p-1, q-1$, and $r-1$.

Suppose further that x is a natural number from $1,2,\ldots,pqr-1$. Let $y_1 = x^{e_1} \bmod pqr$ and $y_2 = x^{e_2} \bmod pqr$ be two different encryptions of x.

There was a glitch and you lose both y_1 and y_2. Suppose you only have access to $y_p = y_1 \bmod p$ and $y_q = y_1 \bmod q$ from the first encryption, and $y_r = y_2 \bmod r$ from the second encryption.

Give an explicit way to recover x from these three numbers y_p, y_q, y_r, given knowledge of p,q,r,e_1,e_2. Describe all computations that you would have to do. You may invoke egcd as a subroutine freely, as well as standard mod operations of addition, multiplication, and exponentiation.

(HINT: You might want to recover x_p, x_q, x_r first.)

Solutions:

Let

$$x_p = x \bmod p$$
$$x_q = x \bmod q$$
$$x_r = x \bmod r$$

We follow the hint, since we know that once we have x_p, x_q, x_r as defined above, we can recover the unique value of $x \bmod pqr$ by using the Chinese Remainder Theorem.

To recover x_p, observe that we have access to $y_p = y_1 \bmod p = x^{e_1} \bmod p$. If we compute $d_p = e_1^{-1} (\bmod p-1)$ by using $egcd(e_1, p-1)$, then we can undo the exponentiation (this is valid since e_1 is coprime to $p-1$). The important thing here is to choose the positive number less than $p-1$ as our representative for the inverse $d_p = e_1^{-1}$ here. That is $d_p e_1 = 1 + i(p-1)$ for some natural number i. (So that exponentiation will mean exactly what we think it does.)

Namely,

$$\begin{aligned} x_p &= y_p^{d_p} \bmod p = (x^{e_1})^{d_p} \bmod p \\ &= x^{e_1 d_p} \bmod p = x^{1+i(p-1)} \bmod p \\ &= x(x^{p-1})^i = x \bmod p \end{aligned}$$

where the last equality uses Fermat's Little Theorem. Note that this computation is valid even if $d_1 = e_1^{-1} (\bmod (p-1)(q-1)(r-1))$, since we can still manipulate the exponent and apply Fermat's Little Theorem.

Similarly, for x_q and x_r, we have $d_q = e_1^{-1} (\bmod q-1)$ and $d_r = e_2^{-1} (\bmod r-1)$ such that

$$\begin{aligned}
x_q &= y_q^{d_q} \bmod q = (x^{e_1})^{d_q} \bmod q \\
&= x^{e_1 d_q} \bmod q = x^{1+j(q-1)} \bmod q \\
&= x(x^{q-1})^j = x \bmod q \\
x_r &= y_r^{d_r} = (x^{e_2})^{d_r} \bmod r \\
&= x^{e_2 d_r} \bmod r = x^{1+k(r-1)} \bmod r \\
&= x(x^{r-1})^k = x \bmod r
\end{aligned}$$

Finally, we have a system of modular equations for x:

$$\begin{aligned}
x &= x_p \pmod p \\
x &= x_q \pmod q \\
x &= x_r \pmod r
\end{aligned}$$

We know by the Chinese Remainder Theorem that the x can be uniquely recovered $\bmod pqr$ as follows:

$$x = (x_p(qr)[(qr)^{-1} \bmod p] + x_q(pr)[(pr)^{-1} \bmod q] + x_r(pq)[(pq)^{-1} \bmod r]) \bmod pqr.$$

Alternatively, you could have solved using two person RSA for an appropriately defined x_{pq} and then CRT combined with x_r to get the answer.

PRINT your name and student ID: _____

9. Secret Sharing (30 pts)

There is a 100 bit-long secret s that we want to distribute among ten people so that if any two of them get together, they can reconstruct the secret but any one of them on their own learns essentially nothing about the secret. We number the ten people $0, 1, \ldots, 9$.

Instead of using the exact scheme described in lecture and homework, this problem asks you to investigate some variant schemes.

(a) (10 pts) We pick a prime number q that is 101 bits long. We use this q to work within $GF(q)$. We roll a q-sided die to get a number f from $0, 1, \ldots, q-1$. We use Lagrange interpolation to construct a degree 1 polynomial $P(x)$ with coefficients from $GF(q)$ so that $P(x)$ passes through the points $P(10) = s$ and $P(11) = f$. To person i, we distribute the secret share $y_i = P(i)$.

Describe exactly how you would recover the secret from any two people getting together. Everyone knows the scheme that is being used, and in particular, knows their own number, the prime number q, and the fact that $P(x)$ must be a degree 1 polynomial.

Solutions: This problem follows the standard secret-sharing paradigm of having a polynomial whose evaluations are distributed among the people. The twist is just that instead of putting the secret as the constant term in the polynomial, the secret is the evaluation of the polynomial $P(x)$ at $x = 10$.

Getting two evaluations of the polynomial at distinct points allows us to recover the underlying polynomial.

Suppose that person i and person j get together. Given $(i, y_i = P(i))$ and $(j, y_j = P(j))$, we know that we can use Lagrange interpolation to recover the degree 1 polynomial. In particular, consider $\Delta_i(x) = (x-j)(i-j)^{-1}$ and $\Delta_j(x) = (x-i)(j-i)^{-1}$ where the multiplicative inverses are computed in $GF(q)$. (For example, by computing the EGCD of $(i-j)$ and q to get $1 = (i-j)^{-1}(i-j) + bq$ since q is prime.)

Then $P(x) = y_i\Delta_i(x) + y_j\Delta_j(x)$ and so the secret is just

$$\begin{aligned} s &= P(10) && (1)\\ &= (y_i\Delta_i(10) + y_j\Delta_j(10)) \bmod q && (2)\\ &= (y_i(10-j) - y_j(10-i))(i-j)^{-1} \bmod q. && (3) \end{aligned}$$

You could also have worked explicitly with the linear form of $P(x) = \alpha_1 x + \alpha_0$ to solve this problem.

(b) (8 pts) For the scheme described in the previous part, **explain why any one person on their own has learned essentially nothing about the secret that they didn't already know before they got their secret share.**

(HINT: You need to show that based on what they know, they cannot rule out any possible value for the secret s.)

Solutions: The secret to the security of secret-sharing is that any group in posession of insufficient shares cannot rule out any value for the secret s.

In this case, this group consists of a single individual. Call this person i who has $y_i = P(i)$.

Here, the easiest way to see the answer is to suppose that there was a hypothetical person who held the "eleventh point" $P(10)$ wherein s itself is held. Clearly, this hypothetical eleventh person can come with any possible value for s and we'd still be able to use Lagrange interpolation to reconstruct a degree-1 polynomial $P_s(x)$ from (i, y_i) and $(10, s)$. This polynomial would have an evaluation at 11 that would be $P_s(11)$ and this would be some number in the finite field $GF(q)$. Since f came from the

roll of a q-sided die, this particular value of $P(11)$ is as likely as any other, and so we cannot rule out this value of s based on our knowledge of the scheme and (i, y_i). (Note: simply saying that we have a possible polynomial is not enough — it has to be a polynomial that could've been constructed by our scheme. This is why checking the 11th evaluation is important.)

Since we can't rule any values for s out, we have learned nothing about s.

(c) (12 pts) Here is an alternative scheme. We pick a collection of twelve prime numbers $q_0, q_1, \ldots, q_{11}$ that are each 101 bits long (they are all actually bigger than 2^{100}) and arranged in decreasing order $q_0 > q_1 > \cdots > q_{11}$. We roll a q_{11} sided die to get a number f from $0, \ldots, q_{11} - 1$. We then compute the number $p = (sv_{10} + fv_{11}) \bmod q_{10}q_{11}$ where v_{10} is the smallest natural number such that $v_{10} \bmod q_{10} = 1$ and $v_{10} \bmod q_{11} = 0$, and similarly v_{11} is the smallest natural number such that $v_{11} \bmod q_{10} = 0$ and $v_{11} \bmod q_{11} = 1$.

To person i, we distribute the secret share $y_i = p \bmod q_i$. Note that each share is at most 101 bits long.

Describe exactly how you would recover the secret from any two people getting together. You can assume that everyone knows all the primes q_i — this list of primes isn't secret.

(HINT: What is $p \bmod q_{10}$?)

Solutions: This problem is clearly referencing the relationship between the CRT and Lagrange Interpolation that you've seen in earlier homework problems.

We know that $p \bmod q_{10} = s$ since

$$\begin{aligned} p \bmod q_{10} &= ((sv_{10} + fv_{11}) \bmod q_{10}q_{11}) \bmod q_{10} && (4)\\ &= (sv_{10} + fv_{11}) \bmod q_{10} && (5)\\ &= s(v_{10} \bmod q_{10}) + f(v_{11} \bmod q_{10}) \bmod q_{10} && (6)\\ &= s \bmod q_{10} && (7)\\ &= s. && (8) \end{aligned}$$

Above we used the key property in (**??**) that modding by pq and then by q is the same as just modding by q. From there, we just use basic mod arithmetic and the property that $v_{10} \bmod q_{10} = 1$ and $v_{11} \bmod q_{10} = 0$. The final line is true because s is a 100-bit number while $q_{10} > 2^{100}$.

This means that getting p suffices for recovering s.

Suppose that person i and person j get together. We can use the CRT to get p from the pair $(i, y_i), (j, y_j)$ since this pair sets up a pair of equations:

$$\begin{aligned} p &\equiv y_i \pmod{q_i}\\ p &\equiv y_j \pmod{q_j} \end{aligned}$$

which we know we can solve up to a multiple of q_iq_j using the CRT. Namely: $p = y_iv_i + y_jv_j \bmod q_iq_j$ where $v_i = q_j(q_j)^{-1}$ where the multiplicative inverse is computed mod q_i and $v_j = q_i(q_i)^{-1}$ where this multiplicative inverse is computed mod q_j. Alternatively, v_i and v_j can be directly obtained from the EGCD of q_i, q_j since if $aq_i + bq_j = 1$, then we can consider $v_j = aq_i$ and $v_i = bq_j$.

Notice that $q_iq_j > q_{10}q_{11}$ and so there is no ambiguity in p which we know to be a natural number less than $q_{10}q_{11}$ and hence a natural number less than q_iq_j as well.

Finally, we take the resulting p and compute $s = p \bmod q_{10}$ to complete the recovery of the secret. Putting everything together $s = (y_iv_i + y_jv_j \bmod q_iq_j) \bmod q_{10}$.

Contributors:

- Edward Im.
- Jonathan Lin.
- Khalil Sarwari.
- Neil Sharma.
- Kevin Zhang.
- Yiming Ding.

