Solutions to Midterm 2

1. (a) We first run the Extended GCD algorithm on 5 and 36.

$$
\begin{array}{ll}
\gcd(36,5) & 1\cdot 36 - 7\cdot 5 = 1. \\
= \gcd(5,1) & 0\cdot 5 + 1\cdot 1 = 1. \\
= \gcd(1,0) = 1 & 1\cdot 1 + 0\cdot 0 = 1.
\end{array}
$$

Since we get that $36 - 7\cdot 5 = 1$, taking everything modulo 35 gives us that $(-7)\cdot 5 = 1 \bmod 35$, and hence $5^{-1} = (-7) = 29 \bmod 35$.

(b) If $5x + 19 = 35 \bmod 36$, then $5x = 35 - 19 = 16 \bmod 36$, and $x = 16\cdot 5^{-1} \bmod 36$. By the previous part, $5^{-1} = 29$, and $x = 16\cdot 29 = 32 \bmod 36$.

(c) No, $6x + 19 = 35 \bmod 36$ does not have a solution. Note that finding such an x is equivalent to finding x such that $6x = 16 \bmod 36$. Since $\gcd(6,36) = 6 \neq 1$, 6 does not have an inverse modulo 36, and the only values $6x$ can take modulo 36 are $0, 6, 12, 18, 24, 30$.

Note that the mere fact that $\gcd(36,6) \neq 1$ is *not* sufficient to claim that the equation has no solution. For example, the equation $6x + 5 = 35 \bmod 36$ *does* have a solution. The fact that no solution exists requires that $35 - 19 = 16$ is not a multiple of $6 (\bmod 36)$. (If $\gcd(36,6)$ were equal to 1, then every integer $\bmod 36$ would be some multiple of 6.)

2. (a) The polynomial is

$$
2\frac{(x-1)(x-3)}{(0-1)(0-3)} + 6\frac{(x-0)(x-3)}{(1-0)(1-3)} + 20\frac{(x-0)(x-1)}{(3-0)(3-1)} = x^2 + 3x + 2.
$$

Indeed, $f(0) = 2, f(1) = 6$ and $f(3) = 20$.

(b) Let us assume there was a lower degree polynomial, which would have to be a linear polynomial $f(x) = ax + b$. Substituting values, we get $f(0) = b = 2$, and also $f(1) = a = 4$. However, we also have $f(3) = 3a = 18$, which is not consistent. Hence there is no lower degree polynomial that satisfies the given points.

Another (much simpler) way to see this is to use the result from class that there is a *unique* polynomial of degree *at most* 2 that passes through the given points. Since f is one such polynomial, there does not exist any other polynomial of degree 0, 1 or 2.

One more thing to notice: a few answers claimed that the points were not on a line and hence could not belong to a linear polynomial. Though this is generally the right general idea (though not a proof) when working over the reals, this intuition can be dangerously wrong when working modulo some number.

(c) One way of coming up with such a polynomial is to take $g(2) = 0$ (say), and then using Lagrange to get the cubic polynomial $6x^3 - 23x^2 + 21x + 2$. Another (much simpler) way is to take $g(x) = f(x) + h(x)$, where h is a polynomial that is 0 at the points $x = 0, x = 1, x = 3$. E.g., taking $h(x) = x(x-1)(x-3)$ would give us $x^3 - 3x^2 + 6x + 2$, which satisfies the given constraints.

Note that the theorem about uniqueness of the polynomial does not preclude the existence of degree 3 polynomials .

3. (a) If $q = \lfloor k/(p-1) \rfloor$, then $k = q(p-1) + (k \bmod (p-1))$. Now

$$\begin{aligned} a^k \bmod p &= a^{q(p-1)+(k \bmod (p-1))} \bmod p \\ &= a^{q(p-1)} \cdot a^{k \bmod (p-1)} \bmod p \\ &= 1 \cdot a^{k \bmod (p-1)} \bmod p. && \text{(By Fermat's Little Thm.)} \end{aligned}$$

(b) To evaluate $a^{(b^c)} \bmod p$, we first find $d = b^c \bmod (p-1)$. Using the algorithm for exponentiation given in class, we can evaluate d in polynomial time in the length of the representation of $b, c, (p-1)$.

Now, using the result of part (a), we know that $a^{(b^c)} = a^{(b^c) \bmod (p-1)} = a^d \bmod p$. we can again evaluate this in time polynomial in the representations of a, d, p. However, since d is at most $(p-1)$, this is polynomial in the representations of a and p. (This last step in the argument was something a lot of people missed.)

4. (a) To decrypt the message he has received, Bob just computes $c^d \bmod n = m^{ed} \bmod n = m$.

(b) Given the primes p, q and the public key e, we can now compute $d' = e^{-1} \bmod (p-1)(q-1)$. Note that since e is Bob's public key, this inverse must exist. But now we can compute $c^{d'} = m^{ed'} = m$. (Since inverses are unique, it must also be the case that $d' = d$, but we do not need this for our procedure.)

Note that all the operations can be done efficiently in the representation of p and q. The inverse can be computed by the (polynomial time) extended GCD algorithm and the exponentiation can also be done in polynomial time.