Homework 8

Chris Powell

A. Silverman 16.1. Use the method of successive squaring to compute each of the following powers.

(a) $5^{13} \pmod{23}$

First, we find the binary expansion of the exponent 13:

$$\begin{aligned} 13 &= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 8 + 4 + 0 + 1. \end{aligned}$$

Then we compute $5^k \pmod{23}$ for each $k \in \{2^0, \ldots, 2^3\}$:

$$\begin{aligned} 5^1 &\equiv 5 \pmod{23} \\ 5^2 &\equiv 25 \equiv 2 \pmod{23} \\ 5^4 &\equiv (5^2)^2 \equiv 2^2 \equiv 4 \pmod{23} \\ 5^8 &\equiv (5^4)^2 \equiv 4^2 \equiv 16 \pmod{23}. \end{aligned}$$

Therefore, by Algorithm 16.1, $5^{13} \equiv 16^1 \cdot 4^1 \cdot 2^0 \cdot 5^1 \equiv 320 \equiv 21 \pmod{23}$.

(b) $28^{749} \pmod{1147}$

First, we find the binary expansion of the exponent 749:

$$\begin{aligned}749 &= 1\cdot 2^9 + 0\cdot 2^8 + 1\cdot 2^7 + 1\cdot 2^6 + 1\cdot 2^5 \\ &\quad + 1\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 \\ &= 512 + 128 + 64 + 32 + 8 + 4 + 1.\end{aligned}$$

Then we compute $28^k \pmod{1147}$ for each $k \in \{2^0, \ldots, 2^9\}$:

$$\begin{aligned}
28^1 &\equiv 28 \pmod{1147}\\
28^2 &\equiv 784 \pmod{1147}\\
28^4 &\equiv (28^2)^2 \equiv 784^2 \equiv 614656 \equiv 1011 \pmod{1147}\\
28^8 &\equiv (28^4)^2 \equiv 1011^2 \equiv 1022121 \equiv 144 \pmod{1147}\\
28^{16} &\equiv (28^8)^2 \equiv 144^2 \equiv 20736 \equiv 90 \pmod{1147}\\
28^{32} &\equiv (28^{16})^2 \equiv 90^2 \equiv 8100 \equiv 71 \pmod{1147}\\
28^{64} &\equiv (28^{32})^2 \equiv 71^2 \equiv 5041 \equiv 453 \pmod{1147}\\
28^{128} &\equiv (28^{64})^2 \equiv 453^2 \equiv 205209 \equiv 1043 \pmod{1147}\\
28^{256} &\equiv (28^{128})^2 \equiv 1043^2 \equiv 1087849 \equiv 493 \pmod{1147}\\
28^{512} &\equiv (28^{256})^2 \equiv 493^2 \equiv 243049 \equiv 1032 \pmod{1147}.
\end{aligned}$$

Therefore, by Algorithm 16.1,

$$\begin{aligned}28^{749} \equiv& 1032^1 \cdot 493^0 \cdot 1043^1 \cdot 453^1 \cdot 71^1 \\ & \cdot 90^1 \cdot 144^0 \cdot 1011^1 \cdot 784^0 \cdot 28^1 \\ \equiv& 289 \pmod{1147}.\end{aligned}$$

B. Silverman 16.2c. The method of successive squaring described in the text allows you to compute $a^k \pmod{m}$ quite efficiently, but it does involve creating a table of powers of a modulo m.

(c) Use your program to compute the following quantities:

(i) $2^{1000} \pmod{2379}$

```
def expmod(a, k, m):
    """compute a^k mod m"""
    b = 1
    while k:
        if k % 2 == 1:
            b = (b * a) % m
```

```
        a, k = (a ** 2) % m, k // 2
    return b
```

$2^{1000} \equiv 562 \pmod{2379}$

(ii) $567^{1234} \pmod{4321}$

$567^{1234} \equiv 3214 \pmod{4321}$

(iii) $47^{258008} \pmod{1315171}$

$47^{258008} \equiv 1296608 \pmod{1315171}$

C. Silverman 16.3.

(a) Compute $7^{7386} \pmod{7387}$ by the method of successive squaring. Is 7387 prime?

Since $7^{7386} \equiv 702 \not\equiv 1 \pmod{7387}$, Fermat's Little Theorem implies that the modulus 7387 is not prime.

(b) Compute $7^{7392} \pmod{7393}$ by the method of successive squaring. Is 7393 prime?

Since $7^{7392} \equiv 1 \pmod{7393}$, Fermat's Little Theorem implies that the modulus 7393 is prime.

D. Silverman 16.4. Ignore the second paragraph. To generate random numbers, put import random at the top of your file, then call random.randint(a,b) to get a random number between a and b inclusive. Write a program to check if a number n is composite or probably prime as follows. Choose 10 randoms numbers $a_1, \ldots, a_{10}$ between 2 and $n-1$ and compute $a_i^{n-1} \pmod{n}$ for each a_i. If $a_i^{n-1} \not\equiv 1 \pmod{n}$ for any a_i, return the message "n is composite." If $a_i^{n-1} \equiv 1 \pmod{n}$ for all the a_i's, return the message "n is probably prime"

```
import random


def expmod(a, k, m):
    """compute a^k mod m"""
    b = 1
```

```
    while k:
        if k % 2 == 1:
            b = (b * a) % m
        a, k = (a ** 2) % m, k // 2
    return b


def probablyprime(n):
    """Returns whether integer n is likely prime or composite
    """
    A = []
    for i in range(10):
        A += [random.randint(2, n-1)]
    for i in A:
        if expmod(A[i], n-1, n) % n != 1:
            return str(n) + " is composite"
    return str(n) + " is prime"
```

E. 1. Show that if $\gcd(a, n) = 1$ and $r \equiv s \pmod{\varphi(n)}$, then $a^r \equiv a^s \pmod{n}$.

Proof. As $\gcd(a, n) = 1$, Euler's formula implies $a^{\varphi(n)} \equiv 1 \pmod{n}$. Since $r \equiv s \pmod{\varphi(n)}$, we have $r = \varphi(n)k + s$ for some integer k. Take $r, s > 1$. Then

$$a^r \equiv a^{\varphi(n)k+s} \equiv (a^{\varphi}(n))^k \cdot a^s \equiv 1^k \cdot a^s \equiv a^s \pmod{n}.$$

$\square$

2. Show that if $\gcd(a, n) \neq 1$, the above is not necessarily true.

Consider $(a, n, r, s) = (2, 8, 13, 1)$. Then $\gcd(2, 8) = 2 \neq 1$, $1 \equiv 13 \pmod{4}$, and $\varphi(8) = \varphi(2^3) = 2^3 - 2^2 = 4$. But $2^1 \not\equiv 2^{13} \pmod{8}$.

F. Silverman 17.2

(a) Solve the congruence $x^{113} \equiv 347 \pmod{463}$.

By the Euclidean algorithm, we know $\gcd(347, 463) = 1$. Also, by the Linear Equation Theorem, we know there exists $u, v \in \mathbb{Z}_{>0}$ satisfying

$$113u - \varphi(463)v = \gcd(113, 462).$$

But since 463 is prime, we know $\varphi(463) = 463^1 - 463^0 = 462$. By applying the extended Euclidean algorithm, we find

$$\gcd(113, 462) = 1, \quad (u, v) = (323, 79).$$

Now observe that

$$\begin{aligned}
\left(347^{323}\right)^{113} &= 347^{323\cdot 113} \\
&= 347^{1+\varphi(463)(79)} \\
&= 347 \cdot \left(347^{\varphi(463)}\right)^{79} \\
&\equiv 347 \cdot 1^{79} && \text{(Euler's formula)} \\
&\equiv 347 \pmod{463}.
\end{aligned}$$

Using successive squaring, we compute

$$347^{113} \equiv 37 \pmod{473}.$$

Hence $x = 37$ satisfies $x^{113} \equiv 347 \pmod{463}$.

(b) Solve the congruence $x^{275} \equiv 139 \pmod{588}$.

By the Euclidean algorithm, we know $\gcd(139, 588) = 1$. Also, by the Linear Equation Theorem, we know there exists $u, v \in \mathbb{Z}_{>0}$ satisfying

$$275u - \varphi(588)v = \gcd(275, 588).$$

But

$$\begin{aligned}\varphi(588) &= \varphi(2^2)\varphi(3^1)\varphi(7^2) \\ &= (2^2 - 2^1)(3^1 - 3^0)(7^2 - 7^1) \\ &= 168.\end{aligned}$$

By applying the extended Euclidean algorithm, we find

$$\gcd(275, 168) = 1, \quad (u, v) = (11, 18).$$

Now observe that

$$\begin{aligned}\left(139^{11}\right)^{275} &= 139^{11\cdot 275} \\ &= 139^{1+\varphi(588)(18)} \\ &= 139 \cdot \left(139^{\varphi(588)}\right)^{18} \\ &\equiv 139 \cdot 1^{18} && \text{(Euler's formula)} \\ &\equiv 139 \pmod{463}.\end{aligned}$$

Using successive squaring, we compute

$$139^{11} \equiv 559 \pmod{588}.$$

Thus $x = 559$ satisfies $x^{275} \equiv 139 \pmod{588}$.

G. Silverman 17.4 Our method for solving $x^k \equiv b \pmod{m}$ is first to find integers u and v satisfying $ku - \varphi(m)v = 1$, and then the solution is $x \equiv b^u \pmod{m}$. However, we only showed that this works provided that $\gcd(b, m) = 1$, since we used Euler's formula $b^{\varphi(m)} \equiv 1 \pmod{m}$.

(a) If m is a product of distinct primes, show that $x \equiv b^u \pmod{m}$ is always a solution to $x^k \equiv b \pmod{m}$, even if $\gcd(b, m) > 1$.

Proof. Assume m has prime factorization $m = \prod_{i=1}^{r} p_i$, where each p_i is distinct. Then

$$\varphi(m) = \varphi\left(\prod_{i=1}^{r} p_i\right) = \prod_{i=1}^{r} \varphi(p_i) = \prod_{i=1}^{r}(p_i - 1)$$

which implies $p_i - 1 \mid \varphi(m)$ for all i. Thus for each i, $\varphi(m) = (p_i - 1)k$ for some $k \in \mathbb{Z}$. We know $\prod_{i=1}^{r} p_i \mid (b^u)^k - b$ as $(b^u)^k \equiv b \pmod{m}$. We claim $p_i \mid (b^u)^k - b$ for all i.
Now either $p_i \mid b$ for all i or there is some p_i for which $p_i \nmid b$. If $p_i \mid b$, then $p_i \mid (b^u)^k - b$. Suppose there is some p_i which does not divide b. We know there exists $u, v \in \mathbb{Z}$ satisfying $ku - \varphi(m) = 1$. But this implies

$$ku = 1 - \varphi(m)v = 1 + ((p_j - 1)k)\, v.$$

Thus,

$$\begin{aligned}
(b^u)^k &= b^{uk} \\
&= b^{1+(p_j-1)kv} \\
&= b \cdot b^{(p_i-1)kv} \\
&= b \cdot \left(b^{p_i-1}\right)^{kv} \\
&\equiv b \cdot 1^{kv} \pmod{p_i} \qquad \text{(Fermat's Little Theorem)} \\
&\equiv b \pmod{p_i}.
\end{aligned}$$

So $p_i \mid (b^u)^k - b$. Thus every p_i divides $(b^u)^k - b$, as claimed. The result follows. □

(b) Show that our method does not work for the congruence $x^5 \equiv 6 \pmod{9}$.

Note that $\gcd(6, 9) = 3 \neq 1$. By applying the extended Euclidean algorithm, we find that $(u, v) = (5, 4)$ satisfies $5u - 6v = 1$. Now see that $6^5 \equiv 0 \pmod{9}$, yet $0^5 \equiv 6 \pmod{9}$. Hence, the given congruence admits no solutions.