Quizz on Modular Exponentiation

Angelo Kyrilov

October 27, 2017

1 Introduction

As part of the Number Theory portion of the Discrete Mathematics course, we have covered the concept of modular exponentiation, and studied an efficient algorithm for performing it. Lab 4 asked students to implement the algorithm in Python.

Upon inspection of submitted solutions for lab 4, it became apparent that many students have copied code found on the Internet for the exercise on modular exponentiation. What made this clear is the fact that the submitted code differed substantially from what was presented in lectures, and a large number of students had the exact same solution.

Many students may have been unaware of this, but such actions constitute plagiarism. In addition, there is very little to no learning opportunities when one copies and pastes code fragments found on websites. The lab exercise was assigned so that students get practical experience with this very important algorithm, and improve their understanding of it as well as get exposure to a very powerful Computer Science technique known as dynamic programming.

Since many students presented a solution other than their own, the instructional team is not able to accurately assess students' knowledge of the material, so there will be a quiz on modular exponentiation during the lecture time on Tuesday, October 31, 2017, at 9 am, in COB2 130. This quiz will be part of the grade for lab 4.

2 Grading

The grade for lab 4 will be computed as the geometric mean of the grade for the Python coding exercises, call it p, and the grade for the quiz, call it q. The overall grade for lab 4 will then be $\sqrt{pq}$. This means that a score of 0 for one of the components, results in a score of 0 for the lab overall.

3 Preparation

Modular exponentiation is an important technique in Computer Science, particularly in cryptography, which is covered in CSE15. Modular exponentiation is the process of finding the remainder when an integer b is raised to an exponent, e, and divided by an integer n.

3.1 Basic Idea

Example Compute $3^{17} \pmod 5$.

Solution: We can just compute $3^{17} = 129140163$, and then reduce it modulo 5, so we have $3^{17} \pmod 5 = 3$. □

This method is ineffective for large exponents. Attempting to compute $3^{9876543210} \pmod 5$ would not complete in a reasonable time. We can however make use of the following property:

$$ab \bmod n = [(a \bmod n)(b \bmod n)] \bmod n \tag{1}$$

In the example above, we can rewrite 3^{17} as $3^{10+7} = 3^{10}3^7$. By property (1), we have

$$\begin{aligned}
3^{17} \bmod 5 &= (3^{10} \bmod 5)(3^7 \bmod 5) \bmod 5 \\
&= [(59049 \bmod 5)(2187 \bmod 5)] \bmod 5 \\
&= 4 \cdot 2 \bmod 5 \\
&= 8 \bmod 5 \\
&= 3
\end{aligned}$$

Using this method, we had to carry out more steps but we did not have to work with numbers as large as we did in the original solution, 59049 compared to 129140163.

3.2 Binary Expansions

In the last section we saw how we can reduce the size of the exponents we are working with by expressing the exponent as a sum of integers. This allowed us to raise our base to smaller exponents, reduce each one modulo n and multiply the results together, again reducing modulo n.

If we represent the exponent as a binary expansion, that is a sum of powers of 2, we notice a very desirable property, which will allow us to compute our results quickly.

We start by representing the exponent 17 as a binary expansion, using the change of base algorithm, which tells us to keep dividing 17 by 2 and keep track of the remainders. Executing the algorithm gives the following:

$$\begin{aligned}
17 \text{ div } 2 &= 8 \text{ rem } 1 \\
8 \text{ div } 2 &= 4 \text{ rem } 0 \\
4 \text{ div } 2 &= 2 \text{ rem } 0 \\
2 \text{ div } 2 &= 1 \text{ rem } 0 \\
1 \text{ div } 2 &= 0 \text{ rem } 1
\end{aligned}$$

Therefore $17 = 10001_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 2^4 + 2^0$. Substituting this into the original problem, we get:

$$\begin{aligned} 3^{17} \bmod 5 &= 3^{2^4+2^0} \bmod 5 \\ &= 3^{2^4} \cdot 3^{2^0} \bmod 5 \end{aligned}$$

The property that we can use here is the following:

$$b^{2^k} = \left(b^{2^{k-1}}\right)^2 \tag{2}$$

With property (2) we can compute each factor by simply squaring the previous one, and reducing modulo n.

We need to compute 3^{2^4} mod 5, and 3^{2^0} mod 5, so we start with the smallest power of 2 and work our way up. The first thing we do is compute the base case.

$$3^{2^0} \bmod 5 = 3^1 \bmod 5 = 3 \bmod 5 = 3$$

To compute 3^{2^1} mod 5, we simply square the result of 3^{2^0} mod 5, and reduce modulo 5. Therefore:

$$\begin{aligned} 3^{2^1} \bmod 5 &= \left(3^{2^0}\right)^2 \bmod 5 = 3^2 \bmod 5 = 9 \bmod 5 = 4 \\ 3^{2^2} \bmod 5 &= \left(3^{2^1}\right)^2 \bmod 5 = 4^2 \bmod 5 = 16 \bmod 5 = 1 \\ 3^{2^3} \bmod 5 &= \left(3^{2^2}\right)^2 \bmod 5 = 1^2 \bmod 5 = 1 \bmod 5 = 1 \\ 3^{2^4} \bmod 5 &= \left(3^{2^3}\right)^2 \bmod 5 = 1^2 \bmod 5 = 1 \bmod 5 = 1 \end{aligned}$$

We now have all the factors we need to compute the result we need:

$$3^{17} \bmod 5 = 3^{2^4} \cdot 3^{2^0} \bmod 5 = 1 \cdot 3 \bmod 5 = 3$$

Another interesting thing to note here is that once we have computed the row for 3^{2^2} mod 5, which equals 1, we know that all remaining rows will evaluate to 1. That is because we are squaring the result of each row, and $1^2 \bmod n = 1$.

3.3 Worked Examples

Example Compute 5^{123} (mod 7).

Solution: We should first represent the exponent 123 as a binary expansion.

$$\begin{aligned}
123 &= 2 \cdot 61 + 1 \\
61 &= 2 \cdot 30 + 1 \\
30 &= 2 \cdot 15 + 0 \\
15 &= 2 \cdot 7 + 1 \\
7 &= 2 \cdot 3 + 1 \\
3 &= 2 \cdot 1 + 1 \\
1 &= 2 \cdot 0 + 1
\end{aligned}$$

The sequence of remainders above gives the binary digits in order from right to left, so $123_{10} = 1111011_2 = 2^6 + 2^5 + 2^4 + 2^3 + 2^1 + 2^0$. Therefore:

$$\begin{aligned}
5^{123} (\text{mod } 7) &= 5^{2^6+2^5+2^4+2^3+2^1+2^0} (\text{mod } 7) \\
&= 5^{2^6} \cdot 5^{2^5} \cdot 5^{2^4} \cdot 5^{2^3} \cdot 5^{2^1} \cdot 5^{2^0} (\text{mod } 7)
\end{aligned}$$

We now compute the factors we need, starting with 5^{2^0} mod 7:

$$\begin{aligned}
5^{2^0} \text{ mod } 7 &= 5^1 \text{ mod } 7 = 5 \text{ mod } 7 = 5 \\
5^{2^1} \text{ mod } 7 &= \left(5^{2^0}\right)^2 \text{ mod } 7 = 5^2 \text{ mod } 7 = 25 \text{ mod } 7 = 4 \\
5^{2^2} \text{ mod } 7 &= \left(5^{2^1}\right)^2 \text{ mod } 7 = 4^2 \text{ mod } 7 = 16 \text{ mod } 7 = 2 \\
5^{2^3} \text{ mod } 7 &= \left(5^{2^2}\right)^2 \text{ mod } 7 = 2^2 \text{ mod } 7 = 4 \text{ mod } 7 = 4 \\
5^{2^4} \text{ mod } 7 &= \left(5^{2^3}\right)^2 \text{ mod } 7 = 4^2 \text{ mod } 7 = 16 \text{ mod } 7 = 2 \\
5^{2^5} \text{ mod } 7 &= \left(5^{2^4}\right)^2 \text{ mod } 7 = 2^2 \text{ mod } 7 = 4 \text{ mod } 7 = 4 \\
5^{2^6} \text{ mod } 7 &= \left(5^{2^5}\right)^2 \text{ mod } 7 = 4^2 \text{ mod } 7 = 16 \text{ mod } 7 = 2
\end{aligned}$$

Therefore we have the following:

$$\begin{aligned}
5^{123} (\text{mod } 7) &= 5^{2^6} \cdot 5^{2^5} \cdot 5^{2^4} \cdot 5^{2^3} \cdot 5^{2^1} \cdot 5^{2^0} \ (\text{mod } 7) \\
&= 2 \cdot 4 \cdot 2 \cdot 4 \cdot 4 \cdot 5 \ (\text{mod } 7) \\
&= 1280 \ (\text{mod } 7) \\
&= 6
\end{aligned}$$

Again, we could have written the last three rows of the table directly, following the established pattern of alternating 2 and 4.

Example Compute $3^{199} \pmod{17}$.

Solution: We first represent the exponent as a binary expansion:

$$\begin{aligned} 199 &= 2 \cdot 99 + 1 \\ 99 &= 2 \cdot 49 + 1 \\ 49 &= 2 \cdot 24 + 1 \\ 24 &= 2 \cdot 12 + 0 \\ 12 &= 2 \cdot 6 + 0 \\ 6 &= 2 \cdot 3 + 0 \\ 3 &= 2 \cdot 1 + 1 \\ 1 &= 2 \cdot 0 + 1 \end{aligned}$$

Therefore we have $199_{10} = 11000111_2 = 2^7 + 2^6 + 2^2 + 2^1 + 2^0$, and

$$3^{199} \pmod{17} = 3^{2^7} \cdot 3^{2^6} \cdot 3^{2^2} \cdot 3^{2^1} \cdot 3^{2^0} \pmod{17}$$

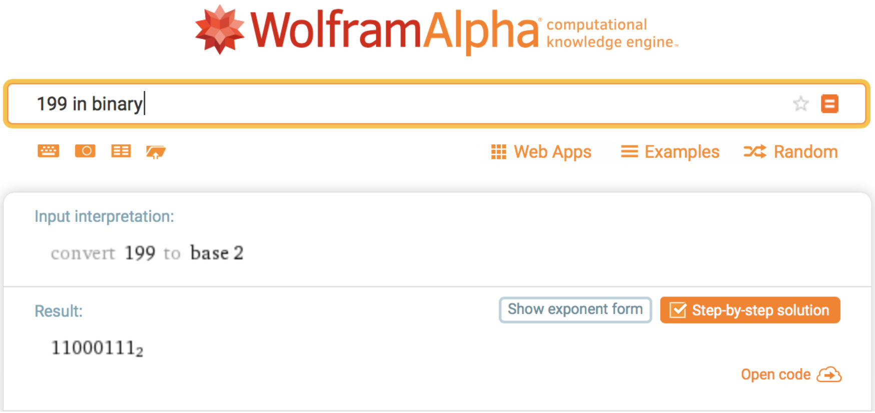
$$\begin{aligned} 3^{2^0} \pmod{17} &= 3^1 \pmod{17} = 3 \pmod{17} = 3 \\ 3^{2^1} \pmod{17} &= \left(3^{2^0}\right)^2 \pmod{17} = 3^2 \pmod{17} = 9 \pmod{17} = 9 \\ 3^{2^2} \pmod{17} &= \left(3^{2^1}\right)^2 \pmod{17} = 9^2 \pmod{17} = 81 \pmod{17} = 13 \\ 3^{2^3} \pmod{17} &= \left(3^{2^2}\right)^2 \pmod{17} = 13^2 \pmod{17} = 169 \pmod{17} = 16 \\ 3^{2^4} \pmod{17} &= \left(3^{2^3}\right)^2 \pmod{17} = 16^2 \pmod{17} = 265 \pmod{17} = 1 \\ 3^{2^5} \pmod{17} &= 1 \\ 3^{2^6} \pmod{17} &= 1 \\ 3^{2^7} \pmod{17} &= 1 \end{aligned}$$

$$\begin{aligned} \text{Therefore } 3^{199} \pmod{17} &= 3^{2^7} \cdot 3^{2^6} \cdot 3^{2^2} \cdot 3^{2^1} \cdot 3^{2^0} \pmod{17} \\ &= 1 \cdot 1 \cdot 13 \cdot 9 \cdot 3 \pmod{17} \\ &= 351 \bmod 17 \\ &= 11 \end{aligned}$$

3.4 Practice

In the quizz, you will be asked to compute $b^x \pmod{n}$, where $2 < b < 10$, $100 < x < 200$, and $10 < n < 20$. You are encouraged to make up problems and try to solve them, and see if you get them right.

The first step is to write a binary expansion of the exponent. Use Wolfram Alpha `http://www.wolframalpha.com/`, to verify your work. Figure 1 is a screenshot of Wolfram Alpha used to verify the binary expansion produced in the last example.



199 in binary

Input interpretation:

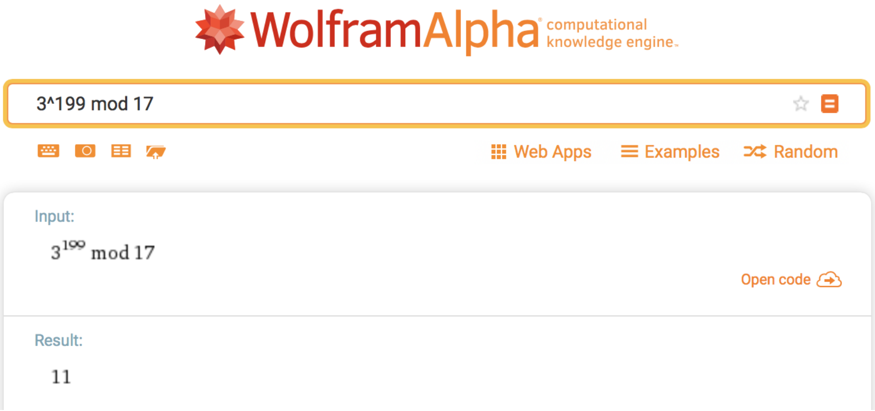
convert 199 to base 2

Result:

11000111_2

Figure 1: Wolfram Alpha with query: 199 in binary

Wolfram Alpha can also be used to verify the overall computation, as shown in Figure 2.



3^199 mod 17

Input:

$3^{199} \bmod 17$

Result:

11

Figure 2: Wolfram Alpha with query: 3^199 mod 17

With the two verifications above, we can be certain that our computations have been performed correctly.