Carlos Mc Gregor c.mcgregormuro@mail.utoronto.ca

CSC373S: Algorithm Design, Analysis & Complexity

Lecture 01

Friday January 06, 2017
based on notes by Denis Pankratov

Modular Exponentiation

Input: $a, b, m \in \mathbb{N}$
Output: $a^b m \in \{0, 1, \dots, m-1\}$

Measure of interest: number of modular multiplications

Brute-force: cost b
To keep the numbers small, we perform all intermediate computations modulo m by repeatedly multiplying by $a \mod m$:

$$1$$
$$a \mod m$$
$$a^2 \mod m \equiv (a \mod m)^2 \mod m,$$
$$a^3 \mod m \equiv ((a^2 \mod m) \times (a \mod m)) \mod m$$
$$\dots$$
$$a^b \mod m$$

Approach: Repeated Squaring
$a \mod m, a^2 \mod m, (a^2)^2 \mod m, \dots, a^{2k} \mod m$, where $2k$ gets to be equal to b

Algorithm:

```
RepeatedSquaring(a, b, m):
  t = 1
  x = a mod m
  p = b

  while p > 0:
    # When the current exponent, p, is odd, we simply multiply
    # the current value by a mod m, since:
    # tx^p = (tx)x^(p-1)
    # In this algo in particular, this allows us to keep
    # a mod m in t for a final use
    if p is odd:
      t = t * x mod m
```

```
            p -= 1
        # When p is even, we can just increasingly square it
        # until p = 1, keeping in mind that:
        # tx^p = t(x^2)^(p/2)
        else:
            x = x * x mod m
            p = p / 2

    return t mod m
```

Example Execution: Let us have $a = 2, b = 5, m = 4$ so that the initial variables are $t = 1, x = 2 \mod 4, p = 5, m = 4$.

<u>Step $p = 5$</u>

$$\begin{aligned}
t &= t \cdot x \mod m \\
&= 1 \cdot (2 \mod 4) \mod 4 = 2 \\
&\equiv 2 \mod 4 \\
p &= p - 1 \\
&= 5 - 1 = 4
\end{aligned}$$

<u>Step $p = 4$</u>

$$\begin{aligned}
x &= x \cdot x \mod m \\
&= (2 \mod 4)(2 \mod 4) \mod 4 \\
&= (2 \mod 4)^2 \mod 4 \\
&\equiv 2^2 \mod 4 \\
p &= \frac{p}{2} \\
&= \frac{4}{2} = 2
\end{aligned}$$

<u>Step $p = 2$</u>

$$\begin{aligned}
x &= x \cdot x \mod m \\
&= (2^2 \mod 4)(2^2 \mod 4) \mod 4 \\
&= (2^2 \mod 4)^2 \mod 4 \\
&\equiv 2^4 \mod 4 \\
p &= \frac{p}{2} \\
&= \frac{2}{2} = 1
\end{aligned}$$

Step $p = 1$

$$\begin{aligned} t &= t \cdot x \mod m \\ &= (2 \mod 4) \cdot (2^4 \mod 4) \mod 4 = 0 \\ &\equiv 2^5 \mod 4 \\ p &= p - 1 \\ &= 1 - 1 = 0 \end{aligned}$$

At the end of the loop, we return $t \mod m$, which is just t.
From this algorithm, the most computationally expensive operation is multiplication.
Loop Invariant (LI): statement that, if true prior to a given iteration, remains true after the iteration.

For this algorithm, our loop invariant is:

$$t \cdot x^p \mod m = a^b \mod m$$

Base Case:
$t = 1, x = a \mod m, p = b$

$$\begin{aligned} t \cdot x^p \mod m &= (1(a \mod m)^b) \mod m \\ &= a^b \mod m \end{aligned}$$

Termination Condition: $p = 0$
Since the LI is $t \cdot x^p \mod m = a^b \mod m$, when $p = 0$, the loop returns $t \mod m = a^b \mod m$.

Analysis of Efficiency:

1. One modular multiplication per iteration, so suffices to count iterations
2. Within any 2 consecutive iterations, p is decreased by a factor of 2

Let p_i be the value of p before the i^{th} iteration.

$$\begin{aligned} p_1 &= b \\ p_{1+2} &\leq \frac{p_1}{2} = \frac{b}{2} \\ p_{1+2+2} &\leq \frac{p_{1+2}}{2} = \frac{b}{2^2} \end{aligned}$$

By induction:

$$p_{1+2k} \leq \frac{b}{2^k}$$

What value of k makes $p_{1+2k} < 1$?

It suffices $^k/_{2^k} < 1$
It suffices $k = (\log_2 b) + 1$
In conclusion, the number of multiplications this algorithm performs is $\mathcal{O}(\log b)$