Problem 1

Answer: 4

This value was computed in one of the lectures. The order is $r = 4$, since $7^4 \bmod 15 = 1$.

Problem 2

Answer:
▷ $7^{20} \bmod 55 = 1$
▷ $7^{10} \bmod 55 \neq 1$
▷ $(7^{10} + 1) \bmod 55 \neq 0$, and it has a common factor with 55.
▷ $(7^{10} - 1) \bmod 55$ has a common factor with 55 since $(7^{10} + 1) \bmod 55 \neq 0$.

Since $r = 20$ is the order, the following must hold:

$$7^{20} \bmod 55 = 1. \tag{1}$$

Next, consider the options related to $7^{10} \bmod 55$. It's straightforward to show that

$$7^{10} \bmod 55 = 34. \tag{2}$$

Clearly, we have $7^{10} \bmod 55 \neq 1$. Moreover, 34 has no factor in common with 55. After all, the corresponding prime factorizations are $34 = 2 \cdot 17$ and $55 = 5 \cdot 11$.
With the aid of our last result, we can analyze the remaining options. First notice that

$$\begin{aligned} \left(7^{10} + 1\right) \bmod 55 &= 35, \\ \left(7^{10} - 1\right) \bmod 55 &= 33. \end{aligned} \tag{3}$$

The first equation tells us that 55 does not divide $7^{10} + 1$. In this case, these numbers have a common factor given by

$$\begin{aligned} d &= \gcd\left(7^{10} + 1, 55\right) \\ &= \gcd\left(55, \left(7^{10} + 1\right) \bmod 55\right) \\ &= \gcd(55, 35) \\ &= 5. \end{aligned} \tag{4}$$

This result allows us to find another non-trivial factor of $N = 55$:

$$\frac{N}{d} = \frac{55}{5} = 11. \tag{5}$$

This is one of the prime factors of 33. Therefore, $(7^{10} - 1) \bmod 55$ has a common factor with 55.
The last remaining option is **wrong**. $(a^{r/2} - 1) \bmod 55$ has a common factor with 55 only when this number does not divide $a^{r/2} + 1$.

Problem 3

Answer:
▷ p
▷ q

The problem statement describes the case in which n and $a^{r/2} + 1$ have a non-trivial common factor. This factor is given by $\gcd(a^{r/2} + 1, n)$. Since n has only two factors, p and q, these are the only possible results for the GCD.

Problem 4

Answer: $O(m^3)$: $2m$ multiplications of m bit numbers and $2m$ modulo operations.

For every bit in the exponent k, we need to perform at most 2 multiplication + modulo operations. In total, we perform at most $2m$ such operations. Since the time cost of a single operation is $O(m^2)$, the complexity of modular exponentiation is $O(m^3)$.

Problem 5

Answer:

▷ Modular exponentiation is about computing $a^k \bmod n$ once for a given k. However, order finding repeatedly needs to compute $a^k \bmod n$ for various k until the result is 1.

▷ Repeated squaring yields $a \bmod n, a^2 \bmod n, a^4 \bmod n, \ldots, a^{2^k} \bmod n$. However the actual order r such that $a^r \bmod n = 1$ need not be a power of 2.