Recall: When A is a set, we say any subset of $A \times A$ is a (binary) **relation** on A. A relation R on a set A is called **reflexive** means $(a, a) \in R$ for every element $a \in A$. A relation R on a set A is called **symmetric** means $(b, a) \in R$ whenever $(a, b) \in R$, for all $a, b \in A$. A relation R on a set A is called **transitive** means whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$, for all $a, b, c \in A$. A relation is an **equivalence relation** means it is reflexive, symmetric, and transitive.

Definition: (*Rosen 9.5*) An **equivalence class** of an element $a \in A$ for an equivalence relation R on the set A is the set $\{s \in A | (a, s) \in R\}$. We write this as $[a]_R$.

Definition: Let $R_{(\mathbf{mod}\ n)}$ be the set of all pairs of integers (a, b) such that $(a \ \mathbf{mod}\ n = b \ \mathbf{mod}\ n)$. We say a is **congruent to** b **mod** n means $(a, b) \in R_{(\mathbf{mod}\ n)}$. A common notation is to write this as $a \equiv b (\mathbf{mod}\ n)$.

Modular arithmetic:

$(102 + 48)$ **mod** $10 =$ ____

$(7 \cdot 10)$ **mod** $5 =$ ____

(2^5) **mod** $3 =$ ____

Lemma (Section 4.1, page 241): For $a, b \in \mathbb{Z}$ and positive integer n, $(a, b) \in R_{(\mathbf{mod}\ n)}$ if and only if $n | a - b$.

Lemma (Section 4.1 Theorem 5): For $a, b \in \mathbb{Z}$ and positive integer n, if $a \equiv b (\mathbf{mod}\ n)$ and $c \equiv d (\mathbf{mod}\ n)$ then $a + c \equiv b + d (\mathbf{mod}\ n)$ and $ac \equiv bd (\mathbf{mod}\ n)$. **Informally**: can bring mod "inside" and do it first, for addition and for multiplication.

Application: Cryptography

Definition: Let a be a positive integer and p be a large[1] prime number, both known to everyone. Let k_1 be a secret large number known only to person P_1 (Alice) and k_2 be a secret large number known only to person P_2 (Bob). Let the **Diffie-Helman shared key** for a, p, k_1, k_2 be $(a^{k_1 \cdot k_2} \textbf{ mod } p)$.

Idea: P_1 can quickly compute the Diffie-Helman shared key knowing only a, p, k_1 and the result of $a^{k_2} \textbf{ mod } p$ (that is, P_1 can compute the shared key without knowing k_2, only $a^{k_2} \textbf{ mod } p$). Further, any person P_3 who knows neither k_1 nor k_2 (but may know any and all of the other values) cannot compute the shared secret efficiently.

Key Property: $\forall a \in \mathbb{Z}\, \forall b \in \mathbb{Z}\, \forall g \in \mathbb{Z}^+\, \forall n \in \mathbb{Z}^+ ((g^a \textbf{ mod } n)^b, (g^b \textbf{ mod } n)^a) \in R_{(\textbf{mod } n)}$

Modular Exponentation; Algorithm 5 in Section 4.2 (page 254)

```
procedure modular exponentiation(b: integer;
                 n = (a_{k-1}a_{k-2}...a_1a_0)_2 , m: positive integers)
x := 1
power := b mod m
for i:= 0 to k-1
   if a_i = 1 then x:= (x · power) mod m
   power := (power · power) mod m
return x {x equals b^n mod m}
```

Calculate 3^8 **mod** 7

Approach 1: Directly

3^1 **mod** $7 =$

3^2 **mod** $7 =$

3^3 **mod** $7 =$

3^4 **mod** $7 =$

3^5 **mod** $7 =$

3^6 **mod** $7 =$

3^7 **mod** $7 =$

3^8 **mod** $7 =$

How many multiplication operations did we use?

Approach 2: Using Algorithm 5

$b =$ ____, $n =$ __________, $k =$ ____, $m =$ ____

i	a_i	x	*power*
		1	b **mod** $m =$
0			
1			
2			
3			

How many multiplication operations did we use?

[1] We leave the definition of "large" vague here, but think hundreds of digits for practical applications. In practice, we also need a particular relationship between a and p to hold, which we leave out here. See more in Rosen, 4.6, p302.