大整数模余运算举例说明

- 3185^{2753} mod3233 = $(-48)^{2753}$ mod3233
- =- (3^{2753} mod3233* 2^{4*2753} mod3233) mod 3233
- 由于这里的模数3233太大，指数2753也太大，没有什么特殊情况可以利用。所以计算比较困难。
- 一种减负的做法是，利用模指数运算（参见下一页）。
- 2753 = $(101011000001)_2$，借助计算器或者计算机，利用这个分别将每项的模余求出来，再求乘积的模余。

Modular Exponentiation 模指数运算 (自己看看)

- In cryptography it is important to be able to find b^n **mod** m efficiently, where b, n, and m are large integers.
- It is impractical to first compute b^n and then find its remainder when divided by m because b^n will be a huge number. Instead, we can use an algorithm as follows.
- *Assume* $n = (a_{k-1} \ldots a_1 a_0)_2$, we can get

$$b^n = b^{a_{k-1}\cdot 2^{k-1}+\cdots+a_1\cdot 2+a_0} = b^{a_{k-1}\cdot 2^{k-1}} \cdots b^{a_1\cdot 2} \cdot b^{a_0}$$

- This shows that to compute b^n, we need only compute the values of b, b^2, $(b^2)^2 = b^4$, $(b^4)^2 = b^8$, . . . , b^{2^k}. Once we have these values, we multiply the terms b^{2^j} in this list, where $a_j = 1$.
- The algorithm successively finds b **mod** m, b^2 **mod** m, b^4 **mod** m, . . . , $b^{2^{k-1}}$ **mod** m and multiplies together those terms b^{2^j} **mod** m where $a_j = 1$,