# CIS 3362 - 10/28/22

Calculating $a^{exp} \bmod n$

```
res = 1
for (int i = 0; i < exp; i++)
    res = (res * a) % n;
```

Run-time at least O(exp)
Problem exp will be very large!
Not viable!

## Key Idea

$$a^{1000000} = \left(a^{500,000}\right)^2 \bmod n$$

(500,000 steps)

$$= X^2 \bmod n$$

(1 step (huge savings!))

If exponent is even
Calculate $X = a^{exp/2} \bmod n$
return $(X * X) \bmod n$

else
Calculate $= a \times a^{exp-1} \bmod n$

Subproblem

BAD CASE
(only happens once in a row, never twice!)

# mults+mods is O(log exp)

Javer - modPow

Python - pow (base, exp, mod)

Hmw 43 101011

$3^{43} = 3^{32} \cdot 3^{8} \cdot 3^{2} \cdot 3^{1}$

powb = 3

$3^1, 3^2, (3^2)^2 = 3^4, (3^4)^2 = 3^8$