

取模 mod

模运算的性质：

- ▼ C |
- 1 运算
 - 2 1. 模p加法: $(a + b) \% p = (a \% p + b \% p) \% p$
 - 3
 - 4 2. 模p减法: $(a - b) \% p = (a \% p - b \% p) \% p$
 - 5
 - 6 3. 模p乘法: $(a * b) \% p = ((a \% p) * (b \% p)) \% p$
 - 7
 - 8 4. 幂模p : $(a^b) \% p = ((a \% p)^b) \% p$
 - 9
 - 10 5. 模运算满足结合律、交换律和分配律。

好像不需要下面这个双重 mod：

直接计算就可以

▼ Python |

```
1 if val >= mod:
2     val = val % mod
3 elif val < 0:
4     val = (val + mod) % mod
```

在Python中，整数溢出不会像某些其他语言那样导致错误，而是会回绕。

于是 elif 中采用了双重 mod 保证溢出数据的正确求模。

▼ Python |

```
1 if val >= mod:
2     val = val % mod
3 elif val < 0:
4     val = (val % mod + mod) % mod
```

举个例子：

```
1  mod = 1000000007
2  val = 1000000077
3
4  # 此时val在python中判定为负数
5  # 进入elif中
6  # val = (val % mod + mod) % mod
7  val % mod = 70
8  val % mod + mod = 1000000077
9  (val % mod + mod) % mod = 70
10 # 可以看到，我们保证了溢出数据的正确求模
11
```