**Algorithm 1:** 适用于 P256 素数的模加算法

**Input:** 256-bit 大数 $a, b$，模 $p$，三个长度为 4 的数组
**Output:** $res = a + b \bmod p$

$carry = 0$;
**for** i from 0 by 1 to 4 **do**
  $sum_1 = a[i] + b[i]$ 并且更新 $carry1$  // 本次加法进位则 $carry1$ 置 1，否则置 0
  $sum_2 = sum_1 + carry$ 并且更新 $carry2$  // $carry2$ 的意义与 $carry1$ 一样
  $sum[i] = sum_2$ ;
  $carry = carry1 \mid carry2$ ;
**end**
$borrow = 0$;
$try[0] = sum[0] - p[0] - borrow$ 并且更新 $borrow1$  // 本次减法借位则 $borrow1$ 置 1，否则置 0
$borrow = borrow1$;
$try[1] = sum[1] - p[1] - borrow$ 并且更新 $borrow1$ ;
$borrow = borrow1$;
$try[2] = sum[2] - borrow$ 并且更新 $borrow1$ ;
$borrow = borrow1$;
$try[3] = sum[3] - p[3] - borrow$ 并且更新 $borrow1$ ;
$borrow = borrow1$;
$select\_mask1 = 0 - borrow$  // select_mask1 为 64-bit 数
$select\_mask2 = 0 - borrow$
**for** i from 0 by 1 to 4 **do**
  $res[i] = (!select\_mask1 \ \&\ try[i]) \mid ((select\_mask1 \ \&\ !select\_mask2) \ \&\ sum[i]) \mid$
  $((select\_mask1 \ \&\ select\_mask2) \ \&\ try[i])$
**end**
返回 $res$。

**Algorithm 2:** 适用于 P256 素数的模减算法

**Input:** 256-bit 大数 $a, b$，模 $p$，
  三个长度为 4 的数组
**Output:** $res = a - b \bmod p$
**Data:** $correction = 2^{256} - p$

$borrow = 0$ ;
**for** ifrom0by1to4 **do**
  $diff_1 = a[i] - b[i]$ 并且更新 $borrow1$  // 本次减法借位则 $borrow1$ 置 1，否则置 0 位
  $diff_2 = diff_1 - borrow$ 并且更新 $borrow2$  // $borrow2$ 的意义与 $borrow1$ 一样
  $diff[i] = sum_2$;
  $borrow = borrow1 \mid borrow2$ ;
**end**
$select\_mask = 0 - borrow$  // select_mask 为 64-bit 数
$borrow = 0$ ;
$res[0] = diff[0] - (select\_mask \& correction[i]) - borrow$ 并且更新 $borrow1$ ;
$borrow = borrow1$;
$res[1] = diff[1] - borrow$ 并且更新 $borrow1$ ;
$borrow = borrow1$;
$res[2] = diff[2] - (select\_mask \& correction[i]) - borrow$ 并且更新 $borrow1$ ;
$borrow = borrow1$;
$res[3] = diff[3] - (select\_mask \& correction[i]) - borrow$ 并且更新 $borrow1$ ;
返回 $res$。