21 Деление на малые простые

Задача. Боб разрабатывает программу генерации простых чисел для криптосистемы RSA. Первым этапом теста простоты большого натурального числа a, заданного $n \geq 8$ 32-разрядными словами, является проверка его делимости на малые простые val. Для этого Боб использует следующую функцию языка Си, которая находит остаток $a \bmod val$:

```
uint32 zzModVal(const uint32 a[], size_t n, uint32 val)
{
  uint32 r = 0;
  uint64 divisor;
  while (n--)
    divisor = r,
    divisor <<= 32,
    divisor += a[n],
    r = (uint32)(divisor % val);
  return r;
}
```

Для определения остатка требуется выполнить n делений `uint64 % uint32` и еще $2n$ сложений и сдвигов. Программа Боба будет использоваться в шифровальной машине `Amgine`. Деление на этой машине выполняется неэффективно, в 8–10 раз медленнее умножения.

Перепишите функцию `zzModVal` так, чтобы в ней использовалось не более 4 делений `uint32 % uint32`, не более $2n + 2$ умножений `uint32 * uint32`, а суммарное число сложений и сдвигов увеличилось не более чем в 2 раза.

Решение. Будем считать, что $val < 2^{16}$. Пусть $B = 2^{32}$ и пусть $a = (a_{n-1} \dots a_1 a_0)_B$ — запись числа a в системе счисления по основанию B: $a = \sum_{i=0}^{n-1} a_i B^i$, $0 \leq a_i < B$.

Вот новая функция:

```
uint32 zzModVal2(const uint32 a[], size_t n, uint32 val)
{
  uint32 r0 = 0;
  uint64 r1 = 0;
  uint32 b = (uint32)-1 % val + 1;
  if (b == val)
    return a[0] % val;
  while (n--)
    r1 *= b,
    r1 += r0,
    r1 *= b,
    r1 += a[n],
    r0 = (uint32)r1,
    r1 >>= 32;
  while (r1 != 0)
    r1 *= b,
    r1 += r0 % val,
    r0 = (uint32)r1,
    r1 >>= 32;
  return r0 % val;
}
```

В этой функции сначала определяется значение $b = B \mod val$. Если $b = 0$, т. е. $val \mid B$, то возвращается $a_0 \bmod val = a \bmod val$. Если же $b \neq 0$, то определяется и приводится по модулю val сумма

$$r = \sum_{i=0}^{n-1} a_i b^i \equiv \sum_{i=0}^{n-1} a_i B^i = a \pmod{val}.$$

Реализован следующий алгоритм:

1. $r = (r_1 r_0)_B \leftarrow 0$.
2. Для $i = n-1, \ldots, 0$:
   (a) $r \leftarrow (r_1 b + r_0)b + a_i$.
3. Пока $r_1 \neq 0$:
   (a) $r \leftarrow r_1 b + (r_0 \bmod val)$.
4. Возвратить $r_0 \bmod val$.

После каждой итерации 2a:

$$r \leq (B-1)(1+b+b^2) \leq (B-1)(val^2 - val + 1) < (B-1)(B+1) < B^2.$$

После первой итерации 3a:

$$r \leq (B-1)(val-1) + (val-1) = B(val-1).$$

После второй итерации 3a:

$$r \leq (val-1)(val-1) + (val-1) = val(val-1) < B.$$

Таким образом, будет выполнено не более двух итераций 3a, и требуемые в постановке задачи условия на число операций выполняются. □