Нека $C = 10^6$.

**Решение за 5 точки**

$brute\ force$.

Сложност: $O(q * (n + log_2(C)))$

**Решение за 10 точки, $\boldsymbol{a} = 0$**

$gcd(b * v_l + c, \dots, b * v_r + c) =$

$gcd(b * v_l + c, b * v_{l+1} + c - b * v_l - c, \dots, b * v_r + c - b * v_{r-1} - c) =$

$gcd(b * v_l + c, b * gcd(v_{l+1} - v_l, \dots, v_r - v_{r-1}))$

Второто може да се пресметне със сегментно дърво.

Сложност: $O(q * (log_2(n) + log_2(C)))$

**Решение за 15 точки, $\boldsymbol{l = 1, r = n}$**

Пълното решение за един интервал.

**Решение за 20 точки**

Пълното решение с коренова декомпозиция по интервали.

**Решение за 70 точки**

Бавна реализация на пълното решение.

**Решение за 100 точки**

Разбиваме всеки интервал като $merge\ sort\ tree$, остава за всяка заявка да решим $O(log_2(n))$ пъти случая за цял интервал.

За всеки такъв интервал подреждаме $b * v_i + a * v_i^2 + c$ като коефициенти $b * L + a * R$.

Докато имаме $k > 0$ оставащи коефициента:

Сортираме по $L$, запомняме $(L_0, R_0)$ и вадим съседните, получаваме нов списък $(L_1 - L_0, R_1 - R_0), \dots, (L_{k-1} - L_{k-2}, R_{k-1} - R_{k-2})$ и сега

$gcd(b * L_0 + a * R_0, \dots, b * L_{k-1} + a * R_{k-1}) =$

$g(b * L_0 + a * R_0, b * (L_1 - L_0) + a * (R_1 - R_0), \dots, b * (L_{k-1} - L_{k-2}) + a * (R_{k-1} - R_{k-2}))$

Ако някъде получим коефициент $L' = 0$, то сме останали само с $R' * a$, махаме го и пазим $gcd$ на всички такива за интервала.

Накрая отговорът е $gcd$ на запомнените (за първото пресмятаме $b * L + a * R + c$, за останалите $b * L + a * R$) и $gcd$ на числата от вида $R * a$.

Можем да докажем, че има $\leq log_2(C)$ такива нива и за ниво $i$ (номерирането започва от 0), $|R| \leq C^2 * 2^i$, затова трябва да използваме бавно умножение по модул да намерим $gcd$ на намереното досега $gcd$ и $b * L + a * R$, което обаче може да се ограничи от $\sum_{i=0}^{\lfloor log_2(C) \rfloor} \lceil log_{\lfloor \frac{ULLONG\_MAX}{C^2 * 2^i} \rfloor}(C) \rceil = 42$ като ако искаме $x * y$ по модул $m$, на съответната стъпка базата е $\lfloor ULLONG\_MAX \ / \ x \rfloor$ вместо стандартното 2. На практика е $\sim levels$, където $levels$ е броят описани нива.

Дължината на интервалите намалява бързо и строенето за един интервал е $O(n * log_2(n))$.

Също можем да минаваме ръчно достатъчно малките интервали вместо да правим описаното по-горе.

Сложност: $O((n + q) * log_2(n) * 42)$

Автор: Мартин Копчев