АКАДЕМИЯ НАУК СССР
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

ВОПРОСЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ И ТЕОРИИ АЛГОРИТМОВ

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР АН СССР
МОСКВА 1988

С.А. АБРАМОВ, С.Н. РЫБИН

ОБОБЩЕНИЕ БИНАРНОГО АЛГОРИТМА ВЫЧИСЛЕНИЯ НАИБОЛЬШЕГО ОБЩЕГО ДЕЛИТЕЛЯ ЦЕЛЫХ ЧИСЕЛ

Наряду с широко известным алгоритмом Евклида нахождения наибольшего общего делителя (НОД) двух чисел используется бинарный алгоритм. Его описание имеется, в частности, в [1], там же показано, что число шагов в обоих алгоритмах имеет порядок количества двоичных цифр тех чисел, к которым они применяются. Однако в бинарном алгоритме, в отличие от алгоритма Евклида, нет трудоемких операций деления.

Ниже речь будет идти об обобщении бинарного алгоритма. При этом под обобщением некоторого алгоритма нахождения НОД двух натуральных чисел u и v понимается включение в этот алгоритм дополнительных операций, которые в итоге обеспечивают нахождение вместе с НОД (u, v) таких целых чисел α и β, что $\alpha u + \beta v = \text{НОД}(u, v)$.

В [1] утверждается, что бинарный алгоритм обобщить нельзя (с. 368). Однако ниже будет описано совершенно естественное обобщение бинарного алгоритма.

Бинарный алгоритм нахождения НОД (u, v) состоит в построении последовательности пар $(u_0, v_0), \ldots, (u_i, v_i), \ldots, (u_m, v_m)$. Для $i = 0, 1, 2, \ldots, m$ по крайней мере одно из чисел u_i, v_i нечетно; $u_0 = u/2^w$, $v_0 = v/2^w$, где w – наибольшее неотрицательное целое такое, что u и v делятся на 2^w. Если $u_{i-1} \neq v_{i-1}$, то u_i и v_i вычисляются следу-

ющим образом:

1) если u_{i-1} четное, то $u_i = u_{i-1}/2$, $v_i = v_{i-1}$,

2) если v_{i-1} четное, то $u_i = u_{i-1}$, $v_i = v_{i-1}/2$,

3) если u_{i-1} и v_{i-1} нечетные и $u_{i-1} > v_{i-1}$, то $u_i = u_{i-1} - v_{i-1}$, $v_i = v_{i-1}$,

4) если u_{i-1} и v_{i-1} нечетные и $u_{i-1} < v_{i-1}$, то $u_i = u_{i-1}$, $v_i = v_{i-1} - u_{i-1}$.

После того, как получена пара (u_m, v_m), в которой $u_m = v_m$, выполнение алгоритма прекращается, и $2^w u_m$ объявляется искомым наибольшим общим делителем.

Предлагаемое обобщение бинарного алгоритма состоит в построении четверок $(\alpha_i, \beta_i, \gamma_i, \delta_i)$ таких, что

$$u_i = \alpha_i u_0 + \beta_i v_0, \quad v_i = \gamma_i u_0 + \delta_i v_0, \quad i = 0, 1, \ldots, m.$$

Очевидно, что можно положить

$$\alpha_0 = \delta_0 = 1, \quad \beta_0 = \gamma_0 = 0.$$

Для каждого из случаев 1) – 4) укажем способ вычисления $(\alpha_i, \beta_i, \gamma_i, \delta_i)$ по $(\alpha_{i-1}, \beta_{i-1}, \gamma_{i-1}, \delta_{i-1})$, $i = 1, \ldots, m$.

Достаточно привести формулы для α_i и β_i, так как γ_i и δ_i могут быть вычислены аналогичным образом. Для определенности считаем, что v_0 нечетно, т.е. $u = 2^k u_*$, $v = 2^w v_0$, $k \geq w$, u_* и v_0 нечетны.

Если u_i получено из u_{i-1} способом 1), т.е. $u_i = u_{i-1}/2$, возможны две ситуации:

а) α_{i-1} четно, тогда в силу нечетности v_0 число β_{i-1} также четно, $\alpha_i = \alpha_{i-1}/2$, $\beta_i = \beta_{i-1}/2$, = /2;

б) α_{i-1} нечетно, это наименее тривиальный момент. Полагаем

$$\alpha_i = (\alpha_{i-1} - v_0)/2, \quad \beta_i = (\beta_{i-1} + u_0)/2.$$

Если u_i получено из u_{i-1} способом 3), т.е.

$$u_i = u_{i-1} - v_{i-1} = (\alpha_{i-1} - \gamma_{i-1})u_0 + (\beta_{i-1} - \delta_{i-1})v_0,$$

то

$$\alpha_i = \alpha_{i-1} - \gamma_{i-1},\ \beta_i = \beta_{i-1} - \delta_{i-1}.$$

Если v_i получается из v_{i-1} способом 4), то следует действовать аналогично 3).

Пусть теперь v_i получено из v_{i-1} способом 2). Здесь нет полной аналогии с тем случаем, когда u_i получается из u_{i-1} способом 1), потому что мы не можем исходить из нечетности u_0. Однако случай, когда α_{i-1}, β_{i-1} оба являются четными, по-прежнему решается просто

$$\alpha_i = \alpha_{i-1}/2,\ \beta_i = \beta_{i-1}/2;$$

случай, когда α_{i-1} четно, а β_{i-1} нечетно, невозможен, случай же нечетного α_{i-1} и четного β_{i-1} возможен только при четном u_0, а случай одновременно нечетных α_{i-1}, β_{i-1} – только при нечетном u_0. Последние два случая решаются уже применявшимся приемом

$$\alpha_{i+1} = (\alpha_i - v_0)/2,\ \beta_{i+1} = (\beta_i + u_0)/2.$$

Следует заметить, что деление на 2 не является трудоемкой операцией в отличие от деления с остатком в алгоритме Евклида, так как деление на 2 представляет собой простое удаление нуля из набора двоичных цифр или сдвиг.

Переходы от

$$(\alpha_{i-1}, \beta_{i-1}, \gamma_{i-1}, \delta_{i-1}) \text{ о } (\alpha_i, \beta_i, \gamma_i, \delta_i),$$

возникающие в тех случаях, когда в самом бинарном алгоритме отыскания наибольшего общего делителя применяется способом 1) или 2), оставляют справедливыми соотношения

$$|\alpha_i| < u_0,\ |\beta_i| < v_0,\ |\gamma_i| < u_0,\ |\delta_i| < v_0 \qquad (1)$$

(значение i при этом, конечно, увеличивается на единицу). Если же, например, был применен способ 3) и если оказалось, что

$$|\alpha_{i-1} - \gamma_{i-1}| \geq v_0 \text{ или } |\beta_{i-1} - \delta_{i-1}| \geq u_0,$$

то можно взять

$$\alpha_i = \alpha_{i-1} - \gamma_{i-1} \pm v_0, \quad \beta_i = \beta_{i-1} - \delta_{i-1} \pm u_0,$$

при этом перед дополнительным слагаемым берется знак минус, если рассматриваемая разность $\alpha_{i-1} - \gamma_{i-1}$ или $\beta_{i-1} - \delta_{i-1}$ положительна; перед другим дополнительным слагаемым выбирается знак плюс. Этими добавочными действиями мы добиваемся постоянного выполнения неравенств (1). Поэтому в результате применения обобщенного бинарного алгоритма Евклида, в котором предусмотрены указанные меры по предупреждению роста коэффициентов, будут получены α и β такие, что $\alpha u + \beta v = $ НОД (u, v) и $|\alpha| \leq v$, $|\beta| \leq u$, а если $u > 1$, $v > 1$, то $|\alpha| < v$, $|\beta| < u$.

Если коэффициенты α, β отличаются от тех, которые дает обычный алгоритм Евклида, то эти последние имеют вид $\alpha \pm v$, $\beta \pm u$, где знак минус выбирается для положительного α или β, а в другом случае берется знак плюс.

Уступая по трудоемкости, алгоритму Шенхаге [2], предложенный алгоритм обладает, однако, тем преимуществом, что базируется на совершенно простых процедурах – сдвигах, сложении, вычитании. В отличие от алгоритма Шенхаге, он не требует привлечения очень сложно устроенных сверхбыстрых алгоритмов умножения (так как умножения ему вообще не нужны) и не требует запоминания большого количества промежуточных величин. Перечисленные особенности являются ценным обстоятельством при оперировании с очень большими числами, занимающими в памяти компьютера несколько машинных слов.

Литература

1. Кнут Д. Искусство программирования для ЭВМ. Т. 2. М.: Мир, 1977. – 723 с.

2. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979. – 536 с.