Использование вычетов для разбиения числа на простые множители

Программа включает в себя ряд методик, которые используют вычеты для получения p & q из n, где n=pq. Наиболее интересны методики серии “базис A”, т. к. обладают большей эффективностью в сравнении с другими.

Дельта – Правый (базис A).

Самый простой метод основан на получении вычетов $k_i = n \bmod 2^i$, где $i = 1, 2 \ldots \log_2 n$; затем полученные вычеты складываются начиная с первого; на каждом шаге происходит проверка является ли полученная сумма делителем n; если сумма чётна, то n делится на сумму вычетов минус 1 – в обратном случае просто на сумму вычетов.

Дельта – Левый (базис A).

Его принцип работы следующий:

Шаг 1 :

$\frac{n}{k_1}$

$\frac{n}{k_1 + k_0}$

Шаг 2 :

$\frac{n}{k_2}$

$\frac{n}{k_2 + k_1}$

$\frac{n}{k_2 + k_1 + k_0}$

****

Шаг i :

$\frac{n}{k_i}$

$\frac{n}{k_i + k_{i-1}}$

. . . . . .

$\frac{n}{k_i + k_{i-1} + \ldots + k_0}$

При этом, как и в первом методе, может проводиться проверка на чётность/нечётность суммы вычетов.

Дельта – Декриз (базис A).

Принцип работы следующий:

Шаг 1 :

$\frac{n}{k_1}$

$\frac{n}{k_1 - k_0}$

Шаг 2 :

$\frac{n}{k_2}$

$\frac{n}{k_2 - k_1}$

$\frac{n}{k_2 - k_1 - k_0}$

****

Шаг i :

$\frac{n}{k_i}$

$\frac{n}{k_i - k_{i-1}}$

. . . . . .

$\frac{n}{k_i - k_{i-1} - \ldots - k_0}$

Прошу извинить меня за некоторое несоответствие: в пошаговых примерах индексация идёт с нуля, а в формулах с единицы.

Расширенные версии.

Получают $q = n \bmod \sum_i k_i$, *а затем* $p = \frac{n}{q}$.

Dee..ep (basis A).

Обрабатывает случаи, когда $qt = n \bmod qr$ $(1 \le t < r)$, где qr – сумма вычетов. Является более совершенной версией “Delta – LeFt eXt (basis A)”, но более медленной.

То, на чём методики ломают зубы.

n устойчиво ко всем методам серии “базис А”, если из полученных вычетов нельзя получить сумму равную qr, где $1 \le r < p$.

Замечания по второй версии эппа.

Добавлены два метода (BS & DF--) по эффективности сравнимые с Dee..ep - о них я расскажу немного позже, и исправлен баг в Dee..ep’е.

Два способа малость увеличить эффективность алгоритмов.

1. Возвести n в какую либо степень: например расширенная версия Декриза не “бьёт” 25 – зато без проблем дробит 625 на 5 и 125; ещё одно число 35 – при возведении в куб ломается на ура (кстати, качайте самую последнюю версию эппа, т. к. я из – за потери сна на всех этих вопросах, допускаю в проге досадные траблы, впрочем, я не жалуюсь – это того стоит☺☺).
2. Умножить n на z, правда, я, на данный момент, не обладаю быстрой методикой по подбору z, т. ч. с практической точки зрения она пока выглядит весьма сомнительной.

P. S.

Мной разрабатываются и другие методы разбиения n, но пока они довольно “сыры”, чтобы о них рассказывать.

P.P. S.

Буду благодарен за участие в проекте, как по вопросам программирования, математики – так и финансовой поддержке, к тому же, надо заметить, что работаю и в других направлениях имеющих не меньшее прикладное значение, чем вышеизложенное.
e-mail: xft_turbo@mail.ru
Web: http://xproject-all.narod.ru/prgsale.htm
Пример программы без арифметики больших чисел:
http://xproject-all.narod.ru/catcher_of_secret_key_ver2.zip