МЕТОДЫ И АЛГОРИТМЫ ОТСЕВА СОСТАВНЫХ ДЛИННЫХ ЧИСЕЛ МЕРСЕННА

К.В. Сомик

1. Числа Мерсенна - особая форма натуральных чисел: $M(p) = 2^p - 1,\ p -$ простое число.

2. $M(p)$ - числа могут быть простыми и составными. Всего известно 47 простых чисел Мерсенна. Уже не одно столетие математики, а теперь и программисты всего мира осуществляют поиски очередного наибольшего простого числа Мерсенна. В настоящее время таковым является число $M[43112609]$, насчитывающее 12978189 разрядов. Оно было найдено в августе 2008 года в рамках проекта распределённых вычислений GIMPS. Открытие наибольшего простого числа Мерсенна – всегда событие мирового значения, которое демонстрирует возможности используемых вычислительных систем и алгоритмов. Но именно алгоритмическая часть этого проекта, в частности, тест Люка-Лемера – длительное время остается практически без изменения. Поскольку для отыскания каждого нового простого числа Мерсенна требуется анализ все большего количества супердлинных чисел, процесс превращается во все более длительную многомесячную процедуру, несмотря на то, что в распределенной сети GIMPS задействованы десятки тысяч компьютеров и волонтеров - программистов. По оценке американских экспертов, чтобы получить следующее простое число при нынешнем уровне алгоритмической и компьютерной базы потребуется несколько лет.

3. На решение вышеизложенной проблемы направлен проект «Триамер» (научный руководитель проекта и автор алгоритмов – Сомик К.В.). Он реализуется при поддержке МГИУ на базе суперкомпьютерного комплекса (СКК) МГУ (разделы «Ломоносов» и «ГрафИТ!»). Теоретические оценки показывают, что в диапазоне простых степеней $DM(43112609;50298029)$ с большой вероятностью имеются одно или несколько простых $M(p)$ - чисел. Всего в этом диапазоне – 406946 $M(p)$ - чисел, подавляющее большинство из которых – составные. Процесс решения проблемы поиска простого $M(p)$ - числа в диапазоне DM в рамках проекта «Триамер» включает два этапа:

3.1. Быстрый отсев большей части составных $M(p)$ - чисел в исследуемом диапазоне p с помощью параллельного выполнения на СКК новых высокопроизводительных алгоритмов Trimr.sv и Trimr.dvd.

3.2. Сертификация оставшегося подмножества псевдопростых чисел с помощью также нового алгоритма Trimr.str, который гораздо быстрее теста Люка-Лемера.

В предлагаемой статье излагаются основные идеи, положенные в основу алгоритмов Trimr.sv и Trimr.dvd, а также результаты отладки и выполнения отсева составных $M(p)$ - чисел в поддиапазоне $DM1:(49000027;50298029).$

4. В теоретической основе этих алгоритмов – три новации:

4.1. Наименьший возможный простой делитель x проверяемого составного числа $M(p) = x \bullet y$ выражается формулой: $x = 2 \bullet p \bullet t_x + 1$ (4.1), где:

$$t_x = a_i(p) + 12 \bullet t; (i = 1,2,3,4); (t = 0,1,...).$$

При этом параметр индекса делителя t_x: $a_i(p)$ может принимать одно из четырех значений, которые зависят от формы простого числа p, как это представлено в Таблице 1.

Таблица 1. Зависимость значений параметров $a_i(p)$ индекса делителя t_x от формы p

Форма p	$a_1(p)$	$a_2(p)$	$a_3(p)$	$a_4(p)$
6*s + 1	0	8	12 – *Mod*[18*s – 3),12]	*Mod*[6*s – 1),12]
6*s - 1	0	4	*Mod*[6*s – 3),12]	*Mod*[6*s + 1),12]

Справедливость выведенной формулы (4.1) вытекает из условия совместности следующих соотношений:

4.1.1. Поскольку числа p, x - простые, они должны иметь форму:

$$p = 6s \pm 1; x = 6k \pm 1.$$

4.1.2. Согласно Эйлеру любые множители $M(p)$ - чисел имеют форму:

$$x = 2 \bullet p \bullet t_x + 1 = 8 \bullet m \pm 1.$$

4.2. Метод редукции размерности делимого к размерности делителя - двухэтапная процедура. На первом этапе степень p проверяемого на делимость числа $M(p)$ представляется в виде списка ненулевых степеней числа 2 (список masp[j1]). На втором этапе в соответствии с этим списком, упорядоченным в порядке возрастания значений степеней двойки, осуществляется итеративное вычисление остатка от деления квадрата предыдущего остатка m на искомый делитель x. Поэтому на каждой итерации размерность делимого всегда меньше x^2, а число итераций не превышает Log[2, p]. Такой метод позволяет многократно ускорять процесс проверки делимости числа $M(p)$.

На основе п.п.4.1 и 4.2. разработан алгоритм параллельной проверки делимости Trimr.sv. Наряду с многократным ускорением за счет реализации метода редукции размерности делимого, данный алгоритм дает дополнительно 4-х кратное ускорение проверки делимости путем параллельной проверки 4 вариантов значений индекса t_x (4), а также параллельного исключения из обработки индексов, порождающих заведомо составные делители, что обеспечивает еще примерно 4-х кратное мультипликативное ускорение.

4.3. Метод локализации области нулевых значений целочисленного дифференциала частного k - го порядка основан на анализе характеристического Диофантова уравнения 2 – го порядка. Пусть t_{x0} - значение индекса делителя, вплоть до которого мы проверили и не обнаружили признаков делимости числа $M(p)$. Тогда имеем:

$$x = 24 * p * t_x + 2 * p * a + 1;\ y = 24 * p * t_y + 2 * p * b + 1;$$
$$x_0 = 24 * p * t_{x0} + 2 * p * a + 1;\ y_0 = 24 * p * t_{y0} + 2 * p * b + 1;$$
$$i = t_x - t_{x0};\ j = t_{y0} - t_y;$$
$$m_0 = \frac{(n - x_0 * y_0)}{24 * p}.$$

Отсюда получаем характеристическое Диофантово уравнение 2 – го порядка с двумя неизвестными i, j:

$$j * (x_0 + 24 * p * i) = (i * y_0 - m_0)\ (4.3.1).$$

Поскольку $y_0 = 24 * p * q_{p0} + m_{p0}$, получаем:

$$j = q_{p0} - z(i);$$
$$z(i) = \frac{d_0 - i * m_{p0}}{x_0 + 24 * p * i};\ d_0 = x_0 * q_{p0} + m_0.$$

Понятно, что если существует целое i, дающее целое частное $z(i)$, то число $M(p)$ - составное. Обозначим $x(h) = x_0 + 24 * p * h$ и рассмотрим возрастающую целочисленную последовательность значений h и убывающую последовательность соответствующих значений $z(h)$:

$$d_0 - m_{p0} = x(1) * z(1) + m(1);$$
$$d_0 - 2 * m_{p0} = x(2) * z(2) + m(2);$$
$$\dots$$
$$d_0 - h * m_0 = x(h) * z(h) + m(h);$$
$$\dots$$
$$d_0 - i * m_0 = x(i) * z(i) + m(i);$$
$$m(i) = 0; h \neq i \rightarrow m(h) \neq 0.$$

Очевидно, что целочисленная последовательность $z(h)$ ограничена сверху (q_{p0}) и монотонно убывает:

$$z(1) > z(2) > ... > z(h) > \dots > z(i).$$

Будем называть $dz^1(h)$ целочисленным дифференциалом первого порядка:

$$d^1 z(h) = z(h) - z(h+1) = \frac{24 * p * (d_0 - h * m_{p0}) + x_h * m_{p0}}{x_h * x_{h+1}} (4.3.2).$$

Аналогично определяется целочисленный дифференциал второго порядка:

$$d^2 z(h) = d^1 z(h) - d^1 z(h+1) = \frac{2 * 24 * p * [24 * p * (d_0 - h * m_{p0}) + x_h * m_{p0}]}{x_h * x_{h+1} * x_{h+2}}$$ и т.д.

Если $h = 1$, округляя до целого, получаем:

$$d^1 z(1) = IP[\frac{24 * p * (d_0 - m_{p0})}{x_1 * x_2}];$$
$$d^2 z(1) = IP[\frac{2 * 24^2 * p^2 * (d_0 - m_{p0})}{x_1 * x_2 * x_3}];$$
$$\dots$$
$$d^k z(1) = IP[\frac{k! * 24^k * p^k * (d_0 - m_{p0})}{x_1 * x_2 * x_3 * \dots * x_{k+1}}] (4.3.3).$$

Очевидно, что $\exists k_0$ такое, что при $k > k_0 \rightarrow d^k z(1) = d^k z(2) = \dots = 0.$ При этом соответствующее k_0 легко и быстро можно найти. В этом и заключается идея алгоритма Trimr.dvd, который практически без деления по рекурсии на основе предыдущего значения частного $z(h)$ с помощью небольшого количества операций вычитания (сложения) и сравнения находит текущее его значение $z(h+1)$ и опять-таки с помощью этих быстродействующих операций определяет остаток $m(h+1).$

Алгоритм Trimr.dvd будет использоваться на втором этапе отсева составных чисел Мерсенна, когда большая их часть уже отсеяна с помощью алгоритма Trimr.sv.

5. В ходе тестирования программы Trimr.sv на суперкомпьютере «Ломоносов» проводилась обработка 1000 $M(p)$- чисел. При этом $p > 49000000$, т.е. все они – больше рекордного простого числа $M(p)$. Изменяемым параметром при запуске программы Trimr.sv был коэффициент k диапазона значений индекса t делителя x: $k = IP[\frac{t}{p}] = [0{,}0001 \Leftrightarrow 4].$ При изменении k в указанном интервале верхняя граница индекса t

делителя изменялась, соответственно, примерно, в интервале: [5000—200000000]. Для каждого значения k измерялась доля отсева составных чисел $M(p)$ и время выполнения обработки τ. Все тесты выполнялись на 10 узлах, по 2 параллельных процесса на каждом. Полученные результаты приведены в табл.2 и на рис.1. Следует отметить, что с ростом количества задействованных узлов происходит практически линейное увеличение быстродействия обработки массива проверяемых $M(p)$ - чисел при заданном уровне параметра k.

Таблица 2. Результаты тестирования программы Trimr.sv

Диапазон k	Доля отсева	Время выполнения, сек	Среднее время обработки 1 числа,сек
0.0001	0.367	0,25	0,00015
0.001	0.424	1,675	0,0015
0.01	0.469	15,175	0,015
0.1	0.5	111,575	0,1125
0.5	0.523	628,1	0,625
1	0.533	1065,05	1
4	0.549	4224,85	4,17

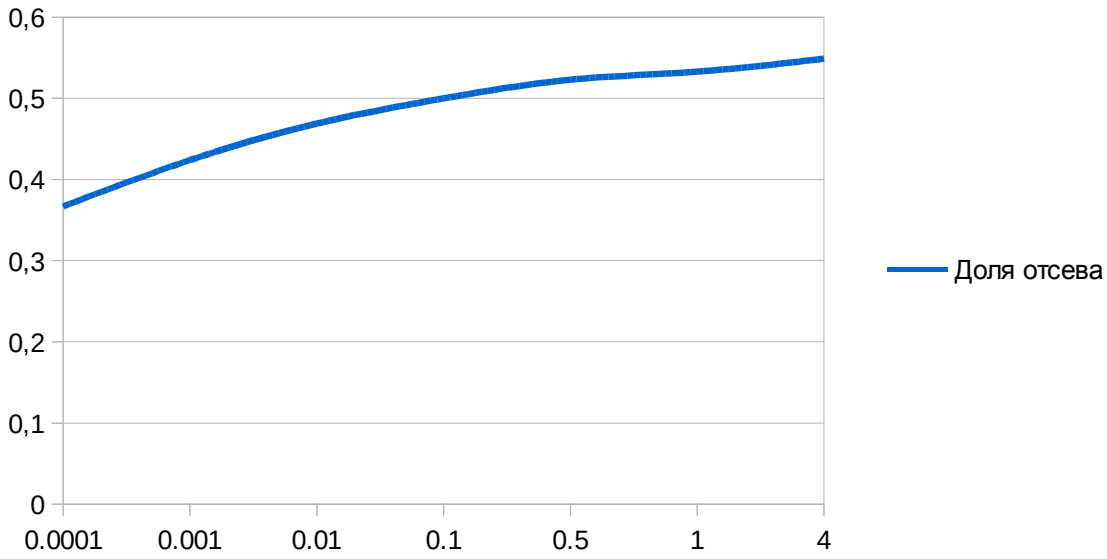

Рис.1. График зависимости доли отсева от диапазона проверки k

Исследование практического быстродействия и эффективности программы Trimr.sv показало, что в поддиапазоне $DM1$, содержащем 67398 $M(p)$ - чисел, при $0.5 < k < 1$ она обеспечивает отсев более половины составных чисел за вполне приемлемое время: несколько суток обработки на СКК МГУ.

6. Также было произведено сравнение быстродействия алгоритма Trimr.sv с другими алгоритмами аналогичной функциональности. Так, в пакете Математика (Wolfram Research, USA) в качестве одного из наиболее быстродействующих алгоритмов факторизации длинных чисел предлагается функция FactorIntegerECM. Данная функция основана на теории эллиптических кривых и алгоритме Х.Ленстра (H.W.Lenstra). Сравнение быстродействия данной функции и алгоритма Trimr.sv показывает, что мой алгоритм обеспечивает нахождение наименьшего делителя составных чисел Мерсенна с показателем в рекордном диапазоне за время чуть более 0,2 сек., тогда как функция FactorIntegerECM при обработке тех же чисел вообще не дает решения за приемлемое время. Представление о том, насколько быстрее работает Trimr.sv, можно получить на следующем примере: функция FactorIntegerECM находит наименьший множитель числа Мерсенна с показателем p = 86423 за 7613.94 сек., тогда как Trimr.sv для такого же числа делает это за 0,531 сек. (протоколы выполнения имеются). Эти данные получены на обычном персональном компьютере без эффекта

распараллеливания, который ускоряет обработку примерно в 16 раз, поэтому реально в этом примере новый алгоритм Trimr.sv обеспечивает ускорение по сравнению с FactorIntegerECM более чем в 200000 раз!

7. В настоящее время обработка поддиапазона $DM1$ завершена. Отсеяно около 60% составных чисел. Далее с помощью модифицированного варианта программы Trimr.dvd (в ней будет реализован метод п.4.3) планируется существенно увеличить долю отсева составных чисел. На заключительном этапе будет выполнена сертификация оставшихся $M(p)$ - чисел с помощью программы Trimr.str.

Таким образом, на базе суперкомпьютерного комплекса МГУ экспериментально установлена высокая эффективность новых методов и алгоритмов проверки делимости больших массивов чисел Мерсенна рекордной длины.

ЛИТЕРАТУРА:

1. Г. Дэвенпорт. Высшая арифметика. Введение в теорию чисел. – М: URSS, 2010, ISBN 978-5-397-01298-0.
2. О.Н. Василенко. Теоретико-числовые алгоритмы в криптографии. – М: МЦНМО, 2003, ISBN 5–94057–103-4.
3. К.В. Сомик. Триангулярная система счисления. – М., ВИНИТИ Деп.№ 622-В2005, 2005.
4. K. Somik. Triangular System of Numbers. Wolfram Technology Conference materials, October 12-14, Champaign, Illinois, USA, 2006.