**ВЫЧИСЛЕНИЯ «ПО-ВЗРОСЛОМУ»**

0. Разрядная сетка компьютеров ограничена. Когда-то использовались 8-разрядные процессоры, затем – 16-, 32-, 64-разрядные. Для бытовых целей чисел такой разрядности достаточно – как для представления целых чисел, так и для вещественных вычислений. Однако существует много приложений, где требуются вычисления с большими числами (например, криптография, астрономия). Также зачастую нужно выполнять точные вычисления (с целыми числами) (например, в финансовой области) либо вычисдения с произвольно заданной точностью.

Ясно, что напрямую такие вычисления выполнить невозможно, и в мире создано достаточно большое количество библиотек, облегчающих труд программистов в этой области. Как правило, большие числа представляются в этих библиотеках в виде структур. Рассмотрим основные библиотеки, позволяющие выполнять вычисления с большими целыми числами, а также с вещественным числами с произвольной точностью.

1. Прежде всего, отметим, что возможность работы с большими числами появилась в среде программирования .NET 4.0. Для представления больших чисел здесь используется специальный тип BigInteger, об особенностях устройства которого можно прочитать, например, здесь: <http://habrahabr.ru/post/207754/> Уможение, деление и другие операции выполняются как с обычными числами (т.е., соответствующие операторы перегружаются). Ни о какой оптимизации, по всей видимости, речи не идет. Какого-либо типа «BigDouble» тоже пока нет.

Кстати, несколько слов об оптимизации умножения. Легко увидеть, что умножение «столбиком» имеет сложность вычисления О(n^2), где n – количество знаков. Выдающийся математик академик А.Н.Колмогоров выдвинул гипотезу о том, что это – не только верхняя граница, но и асимптотическая нижняя граница. Однако его студент, а впоследствии тоже крупный математик А.А.Карацуба опроверг учителя, разработав алгоритм умножения, имеющий сложность О(nlog23). Это привело, увы, к закрытию семинара по кибернетике, проводимом А.Н.Колмогоровым ☺ Метод Карацуба и его обобщения изучаются нынче в вузах и нашли применения во многих пакетах математических программ.

 2. Наиболее известная библиотека с открытым исходным кодом для выполнения вычислений с произвольной точностью – GMP: <https://gmplib.org/> Библиотека написана на Си и ассемблере, предназначена для Unix-систем, однако имеется много портов под Windows. Как указано на сайте библиотеки, вычисление миллиарда знаков числа pi с помощью ее функций занимает полчаса времени на обычном компьютере. В библиотеке релизовано много численных методов.

Функции библиотеки GMP можно разделить на несколько категорий:

1. Высокоуровневые функции целочисленной арифметики со знаком (mpz). Порядка 150 арифметических и логических функций.
2. Высокоуровневые функции операций с дробями (mpq). В этой категории имеется 35 функций.
3. Высокоуровневые функции арифметики с плавающей точкой (mpf, mprf). Эти 70 функций предназначены для достижения произвольной точности при вычислениях.
4. Интерфейс к вышеперечисленным функциям в виде классов языка С++.
5. Низкоуровневые функции, работающие с положительными целыми числами и, в общем случае, не предназначенные для непсоредственного использования Библиотека mpn, порядка 70 функций.

Для умножения длиных целых чисел в библиотеке GMP реализовано 8 функций, в том числе, и упоминавшийся алгоритм Карацубы.

3. Другая заслуживающая упоминания библиотека – MPFR/MPIR: <http://www.holoborodko.com/pavel/mpfr/#download> Библиотека написана на С++, предназначена для Ubuntu и Debian. Есть порты на Windows. Библиотека используется в библиотеке Boost для С++, также имеется версия для Matlab, которая позволяет выполнять вычисления в этом пакете с произвольной точностью. Последняя библиотека разработана автором страницы, ссылку на которую приведена, - Павлом Голобородько, который в настоящее время востребован в Японии.

4. Для программистов, пишущих на С# есть библиотека работы с длинными числами IntX: <https://github.com/devoyster/IntXLib>

По утверждению автора, его функции работают существенно быстрее, чем функции .Net 4.0, хотя, конечно, с GMP эту библиотеку сравнивать нельзя.

5. Ссылки на несколько важных для криптографии библиотек, позволяющих выполнять вычисления с длиннными числами с произвольной точностью, приведен на странице европейского конкурса по криптографии: <https://www.cosic.esat.kuleuven.be/nessie/call/mplibs.html>