Тема 8. Факториал. Метод Шёнхаге

С. Б. Гашков, И. С. Сергеев

Быстрый вариант алгоритма вычисления $n!$ был предложен Шёнхаге около 1994 г. — он имеет сложность $O(M(\log n!)) = O(M(n \log n))$ и основан на идее «деления пополам».

Обозначим через $\{p_i\}$ последовательность простых натуральных чисел в порядке возрастания. Известно, что число простых чисел, не превосходящих n, равно $\pi(n) \sim \frac{n}{\ln n}$ (Адамар, Валле Пуссен).

Метод Шёнхаге состоит в выполнении следующих вычислений в обратном порядке:

$$n! = 2^k x_0, \quad x_0 = x_1^2 y_1, \quad x_1 = x_2^2 y_2, \quad x_2 = x_3^2 y_3, \quad \ldots, \quad x_s = 1,$$

где y_i является произведением всех простых множителей, входящих в x_{i-1} в нечетной степени, и поэтому x_i является квадратом; k — степень вхождения двойки в $n!$. Например,

$$21! = 2^{18} x_0, \quad x_0 = 3^9 \cdot 5^4 \cdot 7^3 \cdot 11 \cdot 13 \cdot 17 \cdot 19,$$

$$x_1 = 3^4 \cdot 5^2 \cdot 7, \quad y_1 = 3 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19,$$

$$x_2 = 3^2 \cdot 5, \quad y_2 = 7.$$

$$x_3 = 3, \quad y_3 = 5,$$

$$x_4 = 1, \quad y_4 = 3.$$

Пусть $e_i(N)$ — степень, в которой число p_i входит в N. Известно (и легко проверяется), что

$$e_i(n!) = \lfloor n/p_i \rfloor + \lfloor n/p_i^2 \rfloor + \lfloor n/p_i^3 \rfloor + \ldots = \lfloor n/p_i \rfloor + e_i(\lfloor n/p_i \rfloor !). \qquad (1)$$

Принципиально вычисление факториала состоит из трех этапов: (I) поиск простых чисел $p_i \le n$, (II) вычисление показателей $e_i(n!)$ и $e_i(y_j)$, (III) непосредственное вычисление факториала при помощи действий вида $x^2 y$.

1 Этап III

Рассмотрим третий этап. Оценим величину чисел x_i и y_i.

Лемма 1. *Пусть* $n! < 2^{2^t}$. *Тогда* $x_i < 2^{2^{t-i}}$ *и если простое число* p *делит* y_i, *то* $p \le n/2^{i-2}$.

Доказательство. Первое неравенство следует из соотношений $x_i^2 \le x_{i-1}$ и $x_0 \le n!$. Докажем второе.

а) Если $p|y_i$, то $p|x_{i-1}$, $p^2|x_{i-2}$ и т.д. Окончательно, $p^{2^{i-1}}|x_0$.

б) Заметим, что $e_i(n!) < n/p_i + n/p_i^2 + \ldots = n/(p_i - 1)$, откуда $e_i(n!) \le n - 1$.

в) Следовательно, если $p^m|n!$, то $p \le 2n/m$. Действительно, если $p_i > 2n/m$, то

$$e_i(n!) = \lfloor n/p_i \rfloor + e_i(\lfloor n/p_i \rfloor !) \le 2\lfloor n/p_i \rfloor - 1 \le 2\lfloor m/2 \rfloor - 1 < m.$$

г) Поэтому из а) следует, что $p \le n/2^{i-2}$.

Оценим сложность вычисления y_j, если даны $e_i(y_j)$. Сложность перемножения 2^s чисел длины b не превосходит

$$2^{s-1}M(b) + 2^{s-2}M(2b) + \ldots + M(2^{s-1}b) = O(sM(2^s b)).$$

По лемме число y_j является произведением не более чем $\pi(n/2^{j-2})$ (простых) чисел длины $\log_2 n$, следовательно, вычисляется со сложностью $O(M(n/2^j)\log n)$.

Число x_{j-1}, если даны числа x_j и y_j вычисляется со сложностью $O(M(2^{t-j}))$, т.к. число x_j согласно лемме имеет длину не более 2^{t-j}, а y_j — не более, чем x_{j-1}, т.е. 2^{t-j+1}.

Суммируя сложность вычисления y_j и x_{j-1} по всем j и учитывая, что $2^t = O(n \log n)$, получаем для сложности этапа III оценку

$$\sum_{j=0}^{\log_2 n} O(M(n/2^j)\log n + M(2^t/2^j)) = O(M(n)\log n + M(2^t)) = O(M(n\log n)).$$

2 Этап II

Заметим, что достаточно вычислить только набор показателей $e_i(n!)$, т.к. для любого j показатель $e_i(y_j)$ совпадает с $(j-1)$-м разрядом числа $e_i(n!)$ (нумерация с нуля). Действительно, по построению:

$$e_i(n!) = e_i(y_1) + 2e_i(y_2) + \ldots + 2^{s-1}e_i(y_s).$$

При каждом i показатель $e_i(n!)$ вычисляется по формуле (1) за $O(\log n)$ делений и сложений $\log n$-разрядных чисел, т.е. со сложностью $O(M(\log n)\log n)$. Общая сложность, следовательно, не превосходит $\pi(n)O(M(\log n)\log n) = O(nM(\log n))$.

3 Этап I

Если вычисления выполняются схемой из функциональных элементов, то все необходимые простые числа p_i следует считать известными заранее. Однако при программной реализации целесообразно рассмотреть случай, когда эти простые числа тоже должны быть вычислены.

Далее мы без доказательства будем использовать известное соотношение

$$\ln\ln n < \sum_{i\le\pi(n)} \frac{1}{p_i} < \ln\ln n + C,$$

справедливое при любом $n \ge 2$.

Пусть нам даны простые числа, не превосходящие $\sqrt{n}$. Тогда остальные простые числа в интервале $[\sqrt{n},\, n]$ могут быть найдены методом «решета Эратосфена». Для этого последовательными сложениями вычисляются последовательности

$$p_i,\ 2p_i,\ \ldots,\ m_i p_i,$$

такие, что $m_i = \lfloor n/p_i \rfloor$. Всего эти последовательности состоят не более чем из $n\sum_{i\le\pi(\sqrt{n})} \frac{1}{p_i} = \Theta(n\log\log n)$ чисел. Таким образом, сложность их вычисления составляет $O(n\log n\log\log n)$.

Заметим, что два упорядоченных набора длины k и l могут быть соединены в один упорядоченный набор не более чем за $k+l-1$ операций сравнения элементов последовательностей. Как следствие, набор из m упорядоченных наборов суммарной длины N можно упорядочить за $O(N\log m)$ операций сравнения, если проводить попарные объединения в бинарном дереве. Операция сравнения чисел длины b имеет сложность $O(b)$.

Разобьем исходные последовательности на группы: в j-й группе — последовательности, соответствующие числам p_i, $\pi(n^{2^{-1-j}}) < i \le \pi(n^{2^{-j}})$, где $1 \le j < \log_2\log_2 n$.

По построению, j-я группа состоит из

$$n \sum_{\pi(n^{2^{-1-j}})<i\le\pi(n^{2^{-j}})} \frac{1}{p_i} = n\Theta(\log(2^{-j}/2^{-1-j})) = O(n)$$

чисел. При этом в j-й группе не более $\pi(n^{2^{-j}}) < n^{2^{-j}}$ последовательностей.

Таким образом, сложность упорядочивания чисел в j-й группе можно оценить как $O(n \log n^{2^{-j}}) = 2^{-j} O(n \log n)$ операций сравнения $\log_2 n$-разрядных чисел, т.е. $2^{-j} O(n \log^2 n)$. Суммарная сложность упорядочиваний по всем группам следовательно оценивается как $O(n \log^2 n)$.

Полученные $\log_2 \log_2 n$ упорядоченных наборов длины не более n упорядочиваются за $\log_2 \log_2 n$ операций объединения, при этом длина всех промежуточных упорядоченных наборов не превосходит n — сложность этого шага $O(n \log n \log \log n)$.

Если обозначить через $P(n)$ сложность генерации последовательности простых чисел, не превосходящих n, то получено соотношение

$$P(n) \leq P(\sqrt{n}) + O(n \log^2 n),$$

откуда следует $P(n) = O(n \log^2 n)$.

Окончательно для сложности программной реализации вычисления $n!$ получаем оценку $O(M(n \log n) + n \log^2 n)$.