CS711008Z Algorithm Design and Analysis

Lecture 5. Fast Division [1]

Dongbo Bu

Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China

[1]The slides are prepared based on Lecture 30 of The Design and Analysis of Algorithms (by D. C. Kozen)

Input: Two integers numerator (also called dividend) n and denominator (also called divisor)d with at most k bits;
Output: Two integers quotient q and remainder r such that $q = n/d$ and $r = n \mod d$.

Two types of division algorithms

Division algorithms fall into two main categories:

- Slow division algorithms: producing one digit of the final quotient per iteration.
- Fast division algorithms: starting with a close approximation to the final quotient, and producing twice as many digits of the final quotient on each iteration.

Grade school algorithm: Long Division

- Generating a digit at each iteration
- Time-complextiy: $O(k^2)$ addition
- Question: Is grade school algorithm optimal?

- Problem: Given two n-digit numbers s and t, to calculate $q = s/t$ and $r = s \mod t$.
- Methods:
  1. Calculate $x = 1/t$ using Newton's method first: $x_{i+1} = 2x_i - t \times x_i^2$.
  2. At most $\log n$ iterations are needed.
  3. Thus division is as fast as multiplication.

Details of FAST DIVISION: Newton's method

Objective: Calculate $x = 1/t$.

- x is the root of $f(x) = 0$, where $f(x) = (t - \frac{1}{x})$. (Why the form here?)
- Newton's method:

$$
\begin{aligned}
x_{i+1} &= x_i - \frac{f(x_i)}{f'(x_i)} && (1)\\
&= x_i - \frac{t - \frac{1}{x_i}}{\frac{1}{x_i^2}} && (2)\\
&= -t \times x_i^2 + 2x_i && (3)
\end{aligned}
$$

- Convergence speed: quadratic, i.e. $\epsilon_{i+1} \le M\epsilon_i^2$, where M is a supremum of a ratio, and ϵ_i denotes the distance between x_i and $\frac{1}{t}$. Thus the number of iterations is limited by $\log\log t = O(\log n)$.

FAST DIVISION: an example

Objective: to calculate $\frac{1}{13}$.

#Iteration	x_i	ϵ_i
0	0.0187	-0.0582231
1	0.032854	-0.044069
2	0.051676	-0.0252471
3	0.0686367	-0.00828638
4	0.0760304	-0.000892633
5	0.0769127	-1.03583e-05
6	0.0769231	-1.39483e-09
7	0.0769231	-2.77556e-17
8	0.0769231	0

Note: the quadratic convergence implies that the error ϵ_i has a form of $O(e^{2^i})$; thus the iteration number is limited by $\log\log(t)$.