**Advanced Numerical Methods** September 21, 23 and 30, 2011

# Lecture Notes 3: Strassen's algorithm

*Lecturer: Che-Rung Lee* *Scribe: Po-Yi Hsu 100062647*

## 1.4 Strassen's algorithm

### 1.4.1 Complex number multiplication

- Let $x = a + bi, y = c + di$. Then $xy = (ac - bd) + (ad + bc)i$. To compute $xy$, we use total 4 multiplications and 3 additions.
- Can we trade multiplications with additions?

$$s = (a + b)(c + d) = ac + bc + ad + bd \tag{1}$$
$$t = (a - b)(c - d) = ac - bc - ad + bd \tag{2}$$
$$w = (a - b)(c + d) = ac - bc + ad - bd \tag{3}$$
$$z = (a + b)(c - d) = ac + bc - ad - bd \tag{4}$$

- We can choose between 4 equations from (1) to (4) and form new equations. They all work fine. If we choose (1), we get

$$xy = (ac - bd) + (s - ac - bd)i$$

- Since equation (1) need 1 multiplication and 2 additions, each $ac$, $bd$ needs 1 multiplication. Now we need only 3 multiplications and 5 additions to compute $xy$. So we succeed to reduce the number of multiplications.

### 1.4.2 Polynomial multiplication

- Let

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0$$
$$q(x) = b_n x^n + b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots + b_2x^2 + b_1x + b_0$$

  Then

$$r(x) = p(x)q(x) = c_{2n}x^{2n} + c_{2n-1}x^{2n-1} + \cdots + c_1x + c_0$$

  which

$$c_k = \sum_{i,j=0,i+j=k}^{k} a_i b_j$$

- e.g. If $p(x) = 3x^2+2x+1$, $q(x) = 4x^2+5x+6$. Then $r(x) = 12x^4+23x^3+32x^2+17x+6$
- If we multiply two polynomials directly, we need total $n^2$ multiplications and $n^2 - 1$ additions. The time complexity is $O(n^2)$.

- Since polynomial multiplication is expensive, we try to trade polynomial multiplications with polynomial additions.
- Let

$$p(x) = \underbrace{a_n x^n + a_{n-1}x^{n-1} + \cdots + a_{\frac{n+1}{2}} x^{\frac{n+1}{2}}}_{p_1(x)} + \underbrace{a_{\frac{n-1}{2}} x^{\frac{n-1}{2}} + \cdots + a_1 x + a_0}_{p_2(x)}$$

$$q(x) = \underbrace{b_n x^n + b_{n-1}x^{n-1} + \cdots + b_{\frac{n+1}{2}} x^{\frac{n+1}{2}}}_{q_1(x)} + \underbrace{b_{\frac{n-1}{2}} x^{\frac{n-1}{2}} + \cdots + b_1 x + b_0}_{q_2(x)}$$

$$\begin{aligned}
p_1(x) &= a_n x^n + \cdots + a_{\frac{n+1}{2}} x^{\frac{n+1}{2}} = (a_n x^{\frac{n-1}{2}} + \cdots + a_{\frac{n+1}{2}}) x^{\frac{n+1}{2}} \\
p_2(x) &= a_{\frac{n-1}{2}} x^{\frac{n-1}{2}} + \cdots + a_0 \\
q_1(x) &= b_n x^n + \cdots + b_{\frac{n+1}{2}} x^{\frac{n+1}{2}} = (b_n x^{\frac{n-1}{2}} + \cdots + b_{\frac{n+1}{2}}) x^{\frac{n+1}{2}} \\
q_2(x) &= b_{\frac{n-1}{2}} x^{\frac{n-1}{2}} + \cdots + b_0
\end{aligned}$$

- Then

$$\begin{aligned}
p(x) &= p_1(x) x^{\frac{n+1}{2}} + p_2(x) \\
q(x) &= q_1(x) x^{\frac{n+1}{2}} + q_2(x)
\end{aligned}$$

- So we can rewrite

$$\begin{aligned}
r(x) &= p(x)q(x) \\
&= (p_1(x) x^{\frac{n+1}{2}} + p_2(x))(q_1(x) x^{\frac{n+1}{2}} + q_2(x)) \\
&= \underbrace{\underbrace{p_1(x)q_1(x)x^{n+1}}_{T(\frac{n}{2})} + (\underbrace{p_1(x)q_2(x)}_{T(\frac{n}{2})} + \underbrace{p_2(x)q_1(x)}_{T(\frac{n}{2})}) x^{\frac{n+1}{2}} + \underbrace{p_2(x)q_2(x)}_{T(\frac{n}{2})}}_{\underline{4T(\frac{n}{2})}} \qquad (5)
\end{aligned}$$

- Now we try to compute it's time complexity.

$$\begin{aligned}
T(n) &= \underline{4T(\frac{n}{2})} + n \\
&= 4(4T(\frac{n}{4}) + \frac{n}{2}) + n \\
&= 16T(\frac{n}{4}) + 2n + n \\
&= 16(4T(\frac{n}{8}) + \frac{n}{4}) + 2n + n \\
&= 64T(\frac{n}{8}) + 4n + 2n + n \\
&\quad \vdots \\
&= 4^{\lg n} T(1) + (2^{\lg n} n + \cdots + 4n + 2n + n) \\
&= n^2 + O(n^2) \\
&= O(n^2)
\end{aligned}$$

So the time complexity is still $O(n^2)$, we need to try other methods.

- We can rewrite equation (5) to $r(x) = r_1(x)x^{2m} + r_2(x)x^m + r_3(x)$.
  Let $s = (p_1+p_2)(q_1+q_2) = p_1q_1+p_2q_1+p_2q_1+p_2q_2$ (for convenience: $p_1, p_2, q_1, q_2, r_1, r_2, r_3$ represent $p_1(x), p_2(x), q_1(x), q_2(x), r_1(x), r_2(x), r_3(x)$). Then

$$\begin{aligned} r_1 &= p_1q_1 \\ r_3 &= p_2q_2 \\ r_2 &= s - r_1 - r_3 \end{aligned} \tag{6}$$

- From equation (6), we need total 3 polynomial multiplications and 4 polynomial additions. Now the time complexity is

$$\begin{aligned} T(n) &= 3T(\frac{n}{2}) + 4 \cdot \frac{n}{2} \\ &= 3T(\frac{n}{2}) + 2n \\ &= 3(3T(\frac{n}{4}) + 2 \cdot \frac{n}{2}) + 2n \\ &= 9T(\frac{n}{4}) + 3n + 2n \\ &= 9(3T(\frac{n}{8}) + 2 \cdot \frac{n}{4}) + 3n + 2n \\ &= 27T(\frac{n}{8}) + \frac{9}{4} \cdot 2n + \frac{3}{2} \cdot 2n + 2n \\ &\vdots \\ &= C_1T(1) + C_2 \end{aligned}$$

$$\begin{aligned} C_1 &= 3^{\log_2 n} = n^{\log_2 3} \doteqdot n^{1.585} \\ C_2 &= 2n(1 + \frac{3}{2} + \frac{9}{4} + \cdot + (\frac{3}{2})^{(\log_2 n)-1}) \\ &= 2n(\frac{(\frac{3}{2})^{\log_2 n} - 1}{\frac{3}{2} - 1}) \\ &= 4n(\frac{3}{2})^{\log_2 n} - 4n \\ &= 4n\frac{3^{\log_2 n}}{n} - 4n \\ &= 4 \cdot 3^{\log_2 n} - 4n \\ &= 4C_1 - 4n \end{aligned}$$

- So the total time complexity reduce to $O(n^{\log_2 3})$, down from $O(n^2)$.