



Design and Analysis
of Algorithms I

Introduction

Karatsuba Multiplication

Example

$$\begin{array}{r} x = \overset{a}{5}\overset{b}{678} \\ y = \underset{c}{12}\underset{d}{34} \end{array}$$

Step 1: Compute $a \cdot c = 672$

Step 2: Compute $b \cdot d = 2652$

Step 3: Compute $(a+b)(c+d) = 134 \cdot 46 = 6164$

Step 4: Compute $\textcircled{3} - \textcircled{2} - \textcircled{1} = 2840$

Step 5:

$$\begin{array}{r} 6720000 \\ 2652 \\ 284000 \\ \hline 7006652 = (1234)(5678) \end{array}$$

A Recursive Algorithm

Write $x = 10^{n/2}a + b$ and $y = 10^{n/2}c + d$

Where a, b, c, d are $n/2$ -digit numbers.

[example: $a=56, b=78, c=12, d=34$]

$$\begin{aligned}\text{Then } x.y &= (10^{n/2}a + b)(10^{n/2}c + d) \\ &= (10^n ac + 10^{n/2}(ad + bc) + bd) \quad (*)\end{aligned}$$

Idea : recursively compute ac, ad, bc, bd , then
compute $(*)$ in the obvious way

Simple Base Case
Omitted

Karatsuba Multiplication

$$x.y = (10^n ac + 10^{n/2}(ad + bc) + bd$$

1. Recursively compute ac
2. Recursively compute bd
3. Recursively compute $(a+b)(c+d) = ac+bd+ad+bc$

Gauss' Trick : $(3) - (1) - (2) = ad + bc$

Upshot : Only need 3 recursive multiplications (and some additions)

Q : which is the fastest algorithm ?