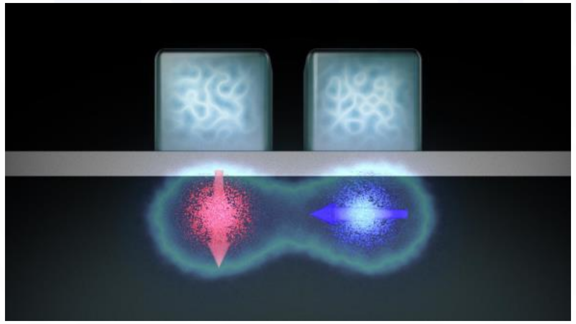
Introduction on Why Study Algorithms?



Design and Analysis of Algorithms I

Why Study Algorithms?

- important for all other branches of computer science

WHY STUDY ALGORITHMS?

- important for all other branches of computer science
- plays a key role in modern technological innovation

WHY STUDY ALGORITHMS?

- important for all other branches of computer science
- plays a key role in modern technological innovation
  - "Everyone knows Moore's Law – a prediction made in 1965 by Intel co-founder Gordon Moore that the density of transistors in integrated circuits would continue to double every 1 to 2 years....in many areas, performance gains due to improvements in algorithms have vastly exceeded even the dramatic performance gains due to increased processor speed."
    - Excerpt from *Report to the President and Congress: Designing a Digital Future,* December 2010 (page 71).

WHY STUDY ALGORITHMS?

- important for all other branches of computer science
- plays a key role in modern technological innovation
- provides novel "lens" on processes outside of computer science and technology
  - quantum mechanics, economic markets, evolution

WHY STUDY ALGORITHMS?

- important for all other branches of computer science
- plays a key role in modern technological innovation
- provides novel “lens” on processes outside of computer science and technology
- challenging

WHY STUDY ALGORITHMS?

- important for all other branches of computer science
- plays a key role in modern technological innovation
- provides novel "lens" on processes outside of computer science and technology
- challenging
- fun

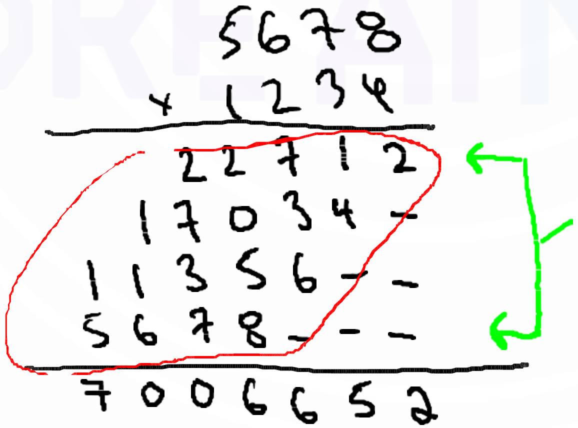
INTEGER MUTLIPLICATION

Input : 2 n digit numbers x and y

Output : product x*y

“Primitive Opera t i on” –add or multiply 2 single digit numbers

THE GRADE-SCHOOL ALGORITHM



Roughly n operations per row up to a constant

# of operations overall ~ constant* n^2

THE ALGORITHM DESIGNER'S MANTRA

"Perhaps the most important principle for the good algorithm designer is to refuse to be content."

-Ullman, *The Design and Analysis of Computer Algorithms*, 1974

CAN WE DO BETTER ?
[than the "obvious" method]

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] Then

$$x.y = (10^{n/2}a + b)(10^{n/2}c + d)$$
$$= (10^n ac + 10^{n/2}(ad + bc) + bd \qquad (*)$$

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 ?