Algorithms

(Divide-and-Conquer)

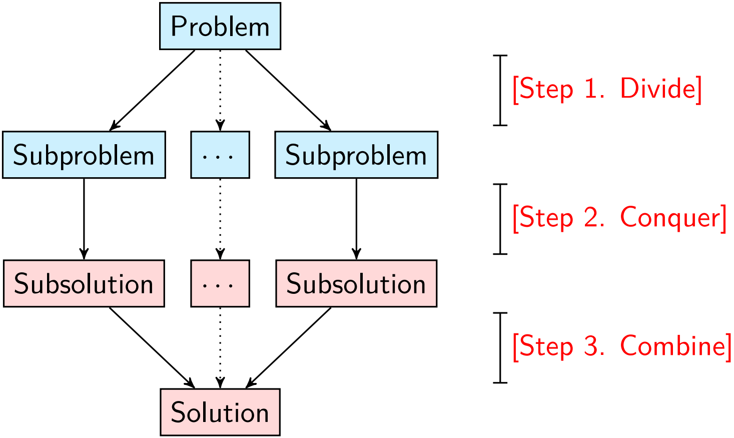
Pramod Ganapathi
Department of Computer Science
State University of New York at Stony Brook

April 27, 2021

Contents

- Merge Sort
- Quicksort
- Karatsuba's Integer Multiplication
- Strassen's Matrix Multiplication

Divide-and-conquer



Divide-and-conquer problem-solving template

5-step process:

Step 1. Problem

Step 2. Core idea

Step 3. Example

Step 4. Algorithm

Step 5. Complexity

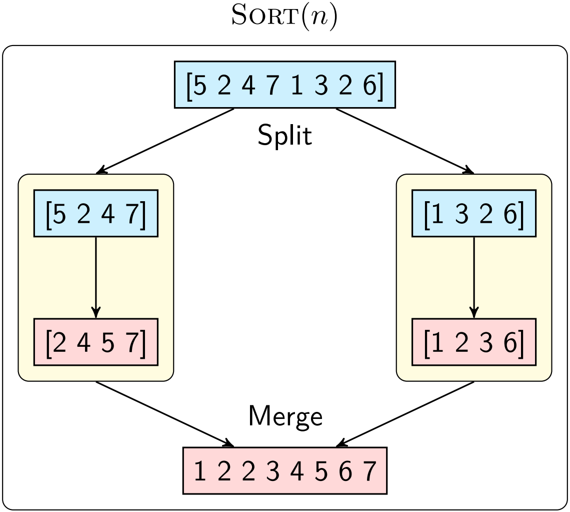
Merge sort

Step 1. Problem

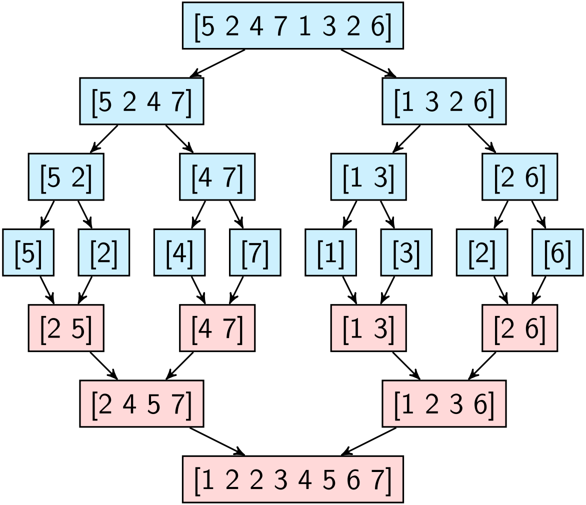
Problem

- Sort a given n-sized array in nondecreasing order.

Step 2. Core idea



Step 3. Example



Step 4. Algorithm

$\textsc{Mergesort}(A[0..(n-1)])$

Input: An array $A[0..(n-1)]$ of orderable elements
Output: Array $A[0..(n-1)]$ sorted in nondecreasing order
1. **if** $n > 1$ **then**
2. $\quad B[0..(\lfloor n/2 \rfloor - 1)] \leftarrow A[0..(\lfloor n/2 \rfloor - 1)]$
3. $\quad C[0..(\lceil n/2 \rceil - 1)] \leftarrow A[\lfloor n/2 \rfloor..(n-1)]$
4. $\quad \textsc{Mergesort}(B[0..(\lfloor n/2 \rfloor - 1))$
5. $\quad \textsc{Mergesort}(C[0..(\lceil n/2 \rceil - 1))$
6. $\quad \textsc{Merge}(A, B, C)$

Step 4. Algorithm

$\textsc{Merge}(A[0..(p+q-1)], B[0..(p-1)], C[0..(q-1)])$

Input: Arrays $B[0..(p-1)]$ and $C[0..(q-1)]$ both sorted
Output: Sorted array $A[0..(p+q-1)]$ of the elements of B and C

$i \leftarrow 0$; $j \leftarrow 0$; $k \leftarrow 0$
while $i < p$ **and** $j < q$ **do**
  if $B[i] \leq C[j]$ **then**
    $A[k] \leftarrow B[i]$
    $i \leftarrow i + 1$
  else
    $A[k] \leftarrow C[j]$
    $j \leftarrow j + 1$
  $k \leftarrow k + 1$
if $i = p$ **then**
  $A[k..(p+q-1)] \leftarrow C[j..(q-1)]$
else
  $A[k..(p+q-1)] \leftarrow B[i..(p-1)]$

Step 5. Complexity

- Time complexity.

$$T_{\text{Sort}}(n) = \begin{cases} 0 & \text{if } n = 1, \\ 2T_{\text{Sort}}(n/2) + T_{\text{Merge}}(n) & \text{if } n > 1. \end{cases}$$

$T_{\text{Merge}}(n) \in \Theta(n)$

Solving, $T_{\text{Sort}}(n) \in \Theta(n \log n)$

- Space complexity.
Extra space is $\Theta(n)$.

$$S_{\text{Sort}}(n) = \begin{cases} \mathcal{O}(1) & \text{if } n = 1, \\ 2S_{\text{Sort}}(n/2) & \text{if } n > 1. \end{cases}$$

Solving, total space $S_{\text{Sort}}(n) \in \Theta(n)$

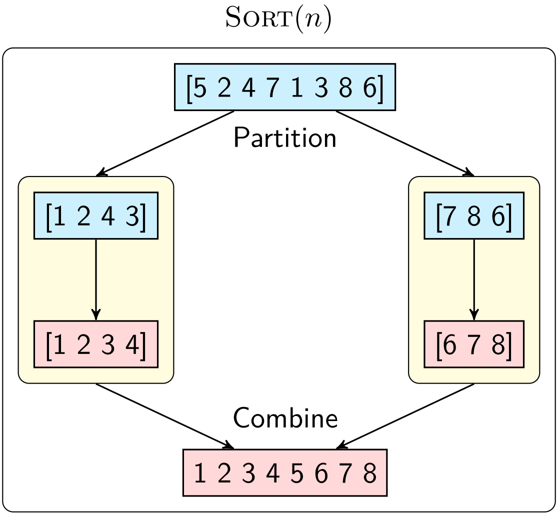
Quicksort

Step 1. Problem

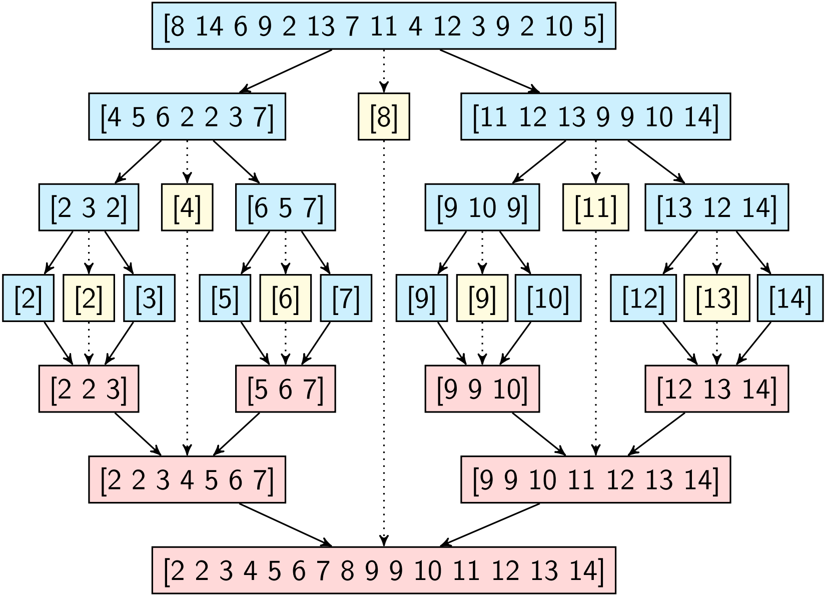
Problem

- Sort a given n-sized array in nondecreasing order.

Step 2. Core idea

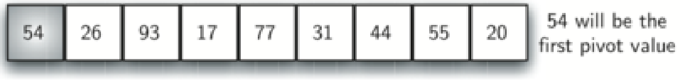


Step 3. Example

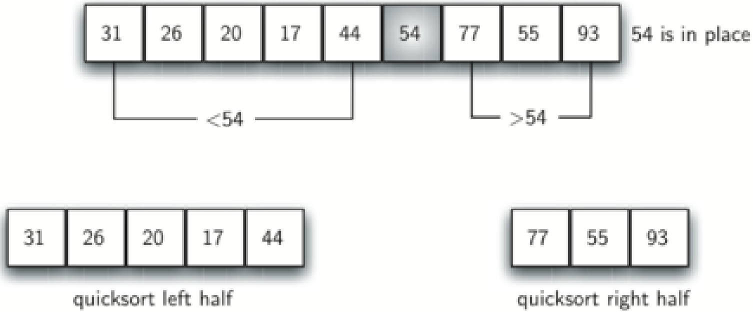


Step 3. Example

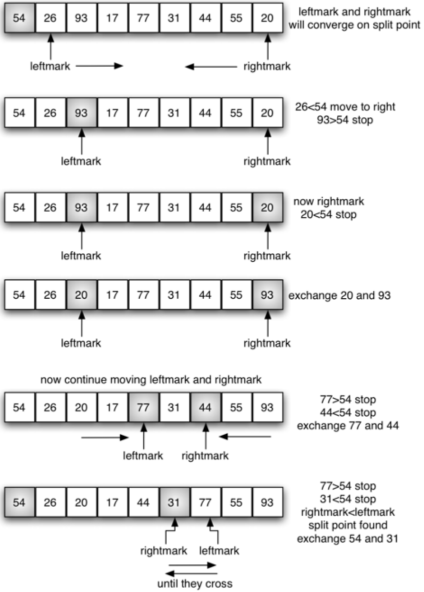
- Before partition



- After partition

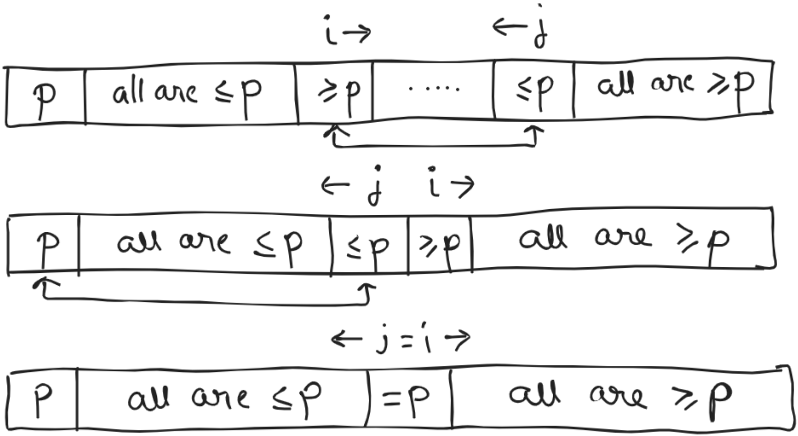


Step 3. Example



Source: `https://runestone.academy/runestone/books/published/pythonds/SortSearch/TheQuickSort.html`

Step 3. Example



Step 4. Algorithm

$\textsc{Quicksort}(A[\ell..r])$

Input: An array $A[\ell..r]$ of orderable elements
Output: Array $A[\ell..r]$ sorted in nondecreasing order

1. **if** $\ell < r$ **then**
2. $\quad s \leftarrow \textsc{Partition}(A[\ell..r])$ $\triangleright$ s is a split position
3. $\quad \textsc{Quicksort}(A[\ell..s-1])$
4. $\quad \textsc{Quicksort}(A[s+1..r])$

Step 4. Algorithm

PARTITION($A[\ell..r]$)

Partitions a subarray by Hoare's algorithm, using the first element as a pivot
Input: An array $A[\ell..r]$ of orderable elements
Output: Partition of $A[\ell..r]$, with the split position returned as this function's value

$p \leftarrow A[\ell]$
$i \leftarrow \ell$; $j \leftarrow r + 1$
repeat
  repeat $i \leftarrow i + 1$ **until** $A[i] \geq p$
  repeat $j \leftarrow j - 1$ **until** $A[j] \leq p$
  SWAP($A[i], A[j]$)
until $i \geq j$
SWAP($A[i], A[j]$) ▷ undo last swap when $i \geq j$
SWAP($A[\ell], A[j]$)
return j

Step 5. Complexity

- Time complexity.

$$T_{\textsc{Sort}}(n) = \begin{cases} \Theta(1) & \text{if } n = 2, \\ T_{\textsc{Sort}}(n-1) + T_{\textsc{Partition}}(n) & \text{if } n > 2. \end{cases}$$

$$T_{\textsc{Partition}}(n) = \Theta(n)$$

Solving, $T_{\textsc{Sort}}(n) \in \Theta\left(n^2\right)$

- Space complexity.
  Extra space is $\Theta(\log n)$ stack space for recursion.

Karatsuba's integer multiplication

Step 1. Problem

Problem

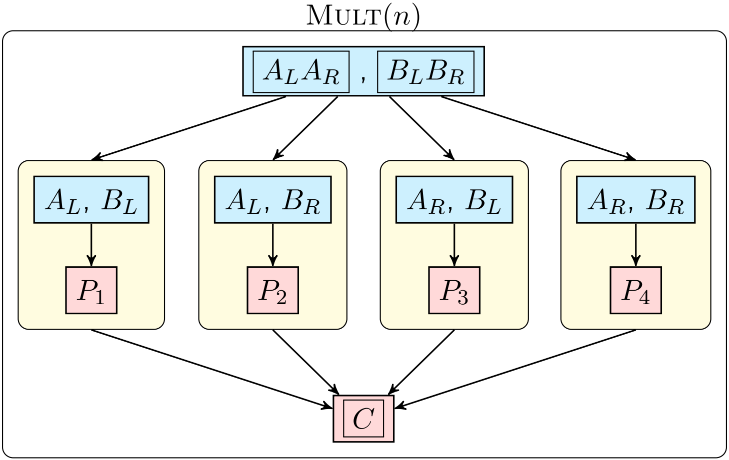
- Multiply two n-bit nonnegative binary numbers.
  For simplicity, we assume n is a power of 2.
- Formally, let A and B be n-bit binary numbers
  $A = A[n-1]A[n-2]\ldots A[0]$ and
  $B = B[n-1]B[n-2]\ldots B[0]$.
  Compute $C = C[2n-1]C[2n-2]\ldots C[0]$ such that

$$\begin{aligned}\boxed{C} &= \boxed{A} \times \boxed{B} \\ &= \boxed{A[n-1]\ldots A[0]} \times \boxed{B[n-1]\ldots B[0]} \\ &= \boxed{C[2n-1]\ldots C[0]}\end{aligned}$$

Step 2. Core Idea

$$
\begin{aligned}
A \times B &= (A_L A_R) \times (B_L B_R) \\
&= (A_L \cdot 2^{n/2} + A_R) \times (B_L \cdot 2^{n/2} + B_R) \\
&= (A_L \times B_L) \cdot 2^n + (A_L \times B_R + A_R \times B_L) \cdot 2^{n/2} \\
&\qquad + (A_R \times B_R)
\end{aligned}
$$

Step 2. Core idea



Step 3. Example

$$\begin{aligned}1100 \times 1001 &= (11)(00) \times (10)(01) \\ &= (11 \cdot 2^2 + 00) \times (10 \cdot 2^2 + 01) \\ &= (11 \times 10) \cdot 2^4 + (11 \times 01 + 00 \times 10) \cdot 2^2 + (00 \times 01)\end{aligned}$$

Step 4. Algorithm

$\textsc{Product}(A[h \dots \ell], B[h \dots \ell])$

Input: Two n-bit nonnegative binary numbers A and B, where h and ℓ are the higher and lower order bits and $n = h - \ell + 1$

Output: Product of nonnegative integers A and B

if $h = \ell$ **then**
  return $A[h] \times B[h]$
else
  $mid \leftarrow \lfloor (h + \ell)/2 \rfloor$; $n \leftarrow h - \ell + 1$
  $A_L \leftarrow A[h \dots mid]$, $A_R \leftarrow A[mid + 1 \dots \ell]$
  $B_L \leftarrow B[h \dots mid]$, $B_R \leftarrow B[mid + 1 \dots \ell]$
  $P_1 \leftarrow \textsc{Product}(A_L, B_L)$
  $P_2 \leftarrow \textsc{Product}(A_L, B_R)$
  $P_3 \leftarrow \textsc{Product}(A_R, B_L)$
  $P_4 \leftarrow \textsc{Product}(A_R, B_R)$
  return $(P_1 \cdot 2^n + (P_2 + P_3) \cdot 2^{n/2} + P_4)$

Step 5. Complexity

- Time complexity.

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 4T(n/2) + \Theta(n) & \text{if } n > 1. \end{cases}$$
$$\in \Theta\left(n^2\right)$$

- Space complexity.

$$S(n) \in \Theta(n)$$

Amazing idea

Problem

Is there is a strategy to perform multiplication of two complex numbers with only 3 multiplications?
$(a+ib)(c+id) = (ac-bd) + i(bc+ad)$

Solution 1

Let $x = bd$, $y = ac$, and $z = (a+b)(c+d)$.
Then, real part $= y - x$ and imaginary part $= z - x - y$.

Solution 2

Let $x = c(a+b)$, $y = a(d-c)$, and $z = b(c+d)$.
Then, real part $= x - z$ and imaginary part $= x + y$.

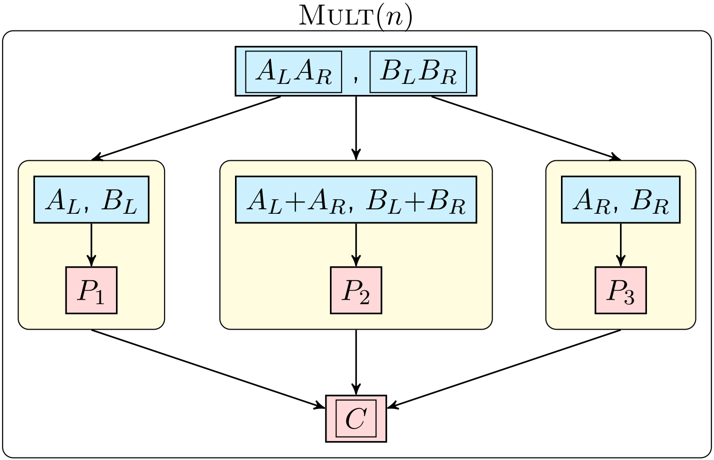
Solution 3

Let $x = c(a+b)$, $y = a(c-d)$, and $z = d(a-b)$.
Then, real part $= y + z$ and imaginary part $= x - y$.

Step 2. Core idea

$$
\begin{aligned}
A \times B &= (A_L A_R) \times (B_L B_R) \\
&= (A_L \cdot 2^{n/2} + A_R) \times (B_L \cdot 2^{n/2} + B_R) \\
&= (A_L \times B_L) \cdot 2^n + (A_L \times B_R + A_R \times B_L) \cdot 2^{n/2} \\
&\qquad + (A_R \times B_R) \\
&= (A_L \times B_L) \cdot 2^n \\
&\qquad + \left(\begin{array}{c} (A_L + A_R) \times (B_L + B_R) \\ -(A_L \times B_L) - (A_R \times B_R) \end{array} \right) \cdot 2^{n/2} \\
&\qquad + (A_R \times B_R)
\end{aligned}
$$

Step 2. Core idea



Step 3. Example

$$\begin{aligned}
1100 \times 1001 &= (11)(00) \times (10)(01) \\
&= (11 \cdot 2^2 + 00) \times (10 \cdot 2^2 + 01) \\
&= (11 \times 10) \cdot 2^4 + (11 \times 01 + 00 \times 10) \cdot 2^2 + (00 \times 01) \\
&= (11 \times 10) \cdot 2^4 + \left(\begin{array}{c} (11 + 00) \times (10 + 01) \\ -11 \times 10 - 00 \times 01 \end{array} \right) \cdot 2^2 \\
&\qquad + (00 \times 01)
\end{aligned}$$

Step 4. Algorithm

$\textsc{Karatsuba-Product}(A[h \ldots \ell], B[h \ldots \ell])$

Input: Two n-bit nonnegative binary numbers A and B, where h and ℓ are the higher and lower order bits and $n = h - \ell + 1$

Output: Product of nonnegative integers A and B

if $h = \ell$ **then**
  return $A[h] \times B[h]$
else
  $mid \leftarrow \lfloor (h + \ell)/2 \rfloor$; $n \leftarrow h - \ell + 1$
  $A_L \leftarrow A[h \ldots mid]$, $A_R \leftarrow A[mid + 1 \ldots \ell]$
  $B_L \leftarrow B[h \ldots mid]$, $B_R \leftarrow B[mid + 1 \ldots \ell]$
  $P_1 \leftarrow \textsc{Karatsuba-Product}(A_L, B_L)$
  $P_2 \leftarrow \textsc{Karatsuba-Product}((A_L + A_R), (B_L + B_R))$
  $P_3 \leftarrow \textsc{Karatsuba-Product}(A_R, B_R)$
  return $(P_1 \cdot 2^n + (P_2 - P_1 - P_3) \cdot 2^{n/2} + P_3)$

Step 5. Complexity

- Time complexity.

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 3T(n/2) + \Theta(n) & \text{if } n > 1. \end{cases}$$
$$\in \Theta\left(n^{\log 3}\right)$$

- Space complexity.

$$S(n) \in \Theta(n)$$

Strassen's matrix multiplication

Step 1. Problem

Example

$$\begin{bmatrix} 2 & 7 & 3 & 6 \\ 5 & 8 & 3 & 8 \\ 6 & 4 & 5 & 6 \\ 0 & 3 & 9 & 7 \end{bmatrix} \times \begin{bmatrix} 8 & 4 & 4 & 3 \\ 7 & 7 & 6 & 8 \\ 5 & 3 & 8 & 4 \\ 2 & 5 & 5 & 7 \end{bmatrix} = \begin{bmatrix} 92 & 96 & 104 & 116 \\ 127 & 125 & 132 & 147 \\ 113 & 97 & 118 & 112 \\ 80 & 83 & 125 & 109 \end{bmatrix}$$

- A's ith row $\times$ B's jth column $= C[i, j]$ cell
- E.g.: $5 \times 4 + 8 \times 6 + 3 \times 8 + 8 \times 5 = 132$

Definition

If A and B are $n \times n$ matrices consisting of real numbers, then the matrix product $C = A \times B$ is defined and computed as

$$C[i, j] = \sum_{k=1}^{n} A[i, k] \times B[k, j] \text{ for } i, j \in [1, n]$$

Step 2. Core idea

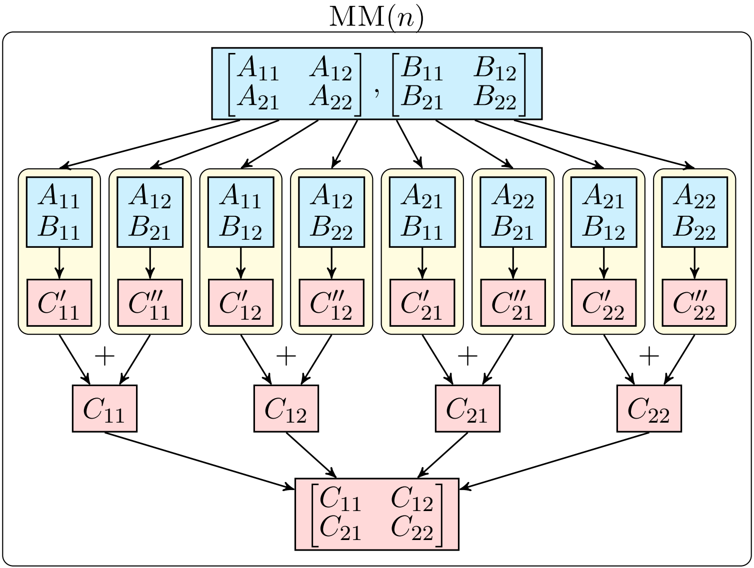
$$C = A \times B$$

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

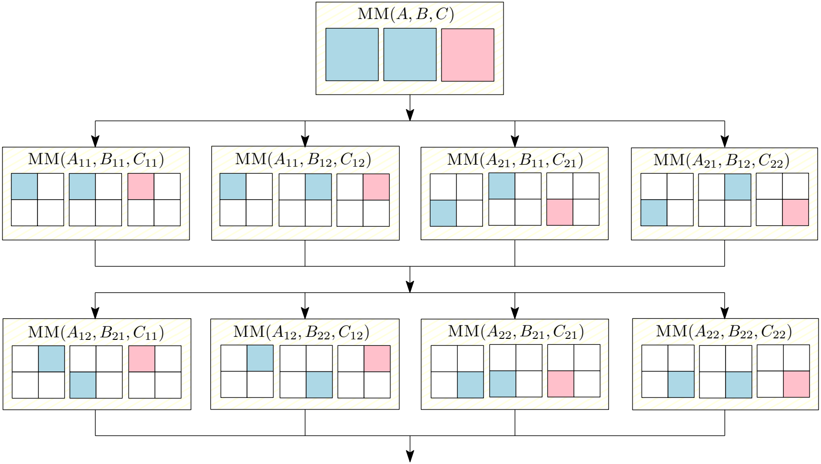
$$= \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

$$= \begin{bmatrix} A_{11}B_{11} & A_{12}B_{21} \\ A_{21}B_{11} & A_{22}B_{21} \end{bmatrix} + \begin{bmatrix} A_{11}B_{12} & A_{12}B_{22} \\ A_{21}B_{12} & A_{22}B_{22} \end{bmatrix}$$

Step 2. Core idea



Step 2. Core idea



Step 4. Algorithm

$\mathrm{MM}(A, B, C, n)$

if $n = 1$ **then**
  $\mathrm{MM\text{-}L{\small OOP}}(A, B, C, n)$
else
  $\mathrm{MM}(A_{11}, B_{11}, C_{11}, n/2)$
  $\mathrm{MM}(A_{12}, B_{21}, C_{11}, n/2)$
  $\mathrm{MM}(A_{11}, B_{12}, C_{12}, n/2)$
  $\mathrm{MM}(A_{12}, B_{22}, C_{12}, n/2)$
  $\mathrm{MM}(A_{21}, B_{11}, C_{21}, n/2)$
  $\mathrm{MM}(A_{22}, B_{21}, C_{21}, n/2)$
  $\mathrm{MM}(A_{21}, B_{12}, C_{22}, n/2)$
  $\mathrm{MM}(A_{22}, B_{22}, C_{22}, n/2)$

Step 5. Complexity

- Time complexity.

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T(n/2) + \Theta(1) & \text{if } n > 1. \end{cases}$$
$$\in \Theta\left(n^3\right)$$

- Space complexity.

$$S(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 4S(n/2) + \Theta(1) & \text{if } n > 1. \end{cases}$$
$$\in \Theta\left(n^2\right)$$

Amazing idea

Problem

Is there is a strategy to perform multiplication of two complex numbers with only 3 multiplications?
$(a+ib)(c+id) = (ac-bd) + i(bc+ad)$

Solution 1

Let $x = bd$, $y = ac$, and $z = (a+b)(c+d)$.
Then, real part $= y - x$ and imaginary part $= z - x - y$.

Solution 2

Let $x = c(a+b)$, $y = a(d-c)$, and $z = b(c+d)$.
Then, real part $= x - z$ and imaginary part $= x + y$.

Solution 3

Let $x = c(a+b)$, $y = a(c-d)$, and $z = d(a-b)$.
Then, real part $= y + z$ and imaginary part $= x - y$.

Step 2. Core idea

Problem	Traditional	Solution 1	Solution 2	Solution 3
Complex number mult.	4 mults 2 adds	3 mults 5 adds	3 mults 5 adds	3 mults 5 adds

$$C = A \times B$$

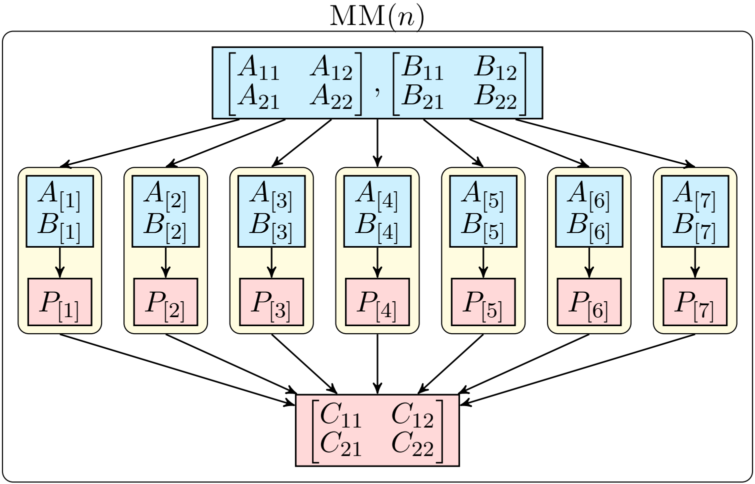
$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

$$= \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

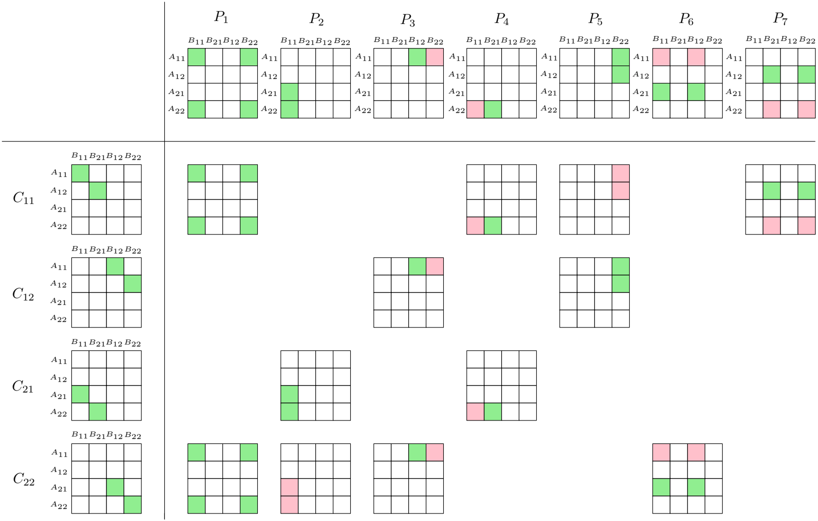
$$= \begin{bmatrix} A_{11}B_{11} & A_{12}B_{21} \\ A_{21}B_{11} & A_{22}B_{21} \end{bmatrix} + \begin{bmatrix} A_{11}B_{12} & A_{12}B_{22} \\ A_{21}B_{12} & A_{22}B_{22} \end{bmatrix}$$

Problem	Traditional	Strassen	Winograd
2×2 MM	8 mults 4 adds	7 mults 18 adds	7 mults 15 adds
$n \times n$ MM	n^3 mults $(n^3 - n^2)$ adds	$n^{\log_2 7}$ mults $(6n^{\log_2 7} - 6n^2)$ adds	$n^{\log_2 7}$ mults $(5n^{\log_2 7} - 5n^2)$ adds

Step 2. Core idea



Step 2. Core idea



Step 4. Algorithm

Strassen-MM(A, B, C, n)

Input: $n \times n$ matrices A and B
Output: $C \leftarrow A \times B$

1. **if** $n = 1$ **then** $C \leftarrow A \times B$
2. **else**

{Step 1. Divide} ....

3. $A_{[1]} \leftarrow A_{11} + A_{22}$, $B_{[1]} \leftarrow B_{11} + B_{22}$,
 $A_{[2]} \leftarrow A_{21} + A_{22}$, $B_{[2]} \leftarrow B_{11}$,
 $A_{[3]} \leftarrow A_{11}$, $B_{[3]} \leftarrow B_{12} - B_{22}$,
 $A_{[4]} \leftarrow A_{22}$, $B_{[4]} \leftarrow B_{21} - B_{11}$,
 $A_{[5]} \leftarrow A_{11} + A_{12}$, $B_{[5]} \leftarrow B_{22}$,
 $A_{[6]} \leftarrow A_{21} - A_{11}$, $B_{[6]} \leftarrow B_{11} + B_{12}$,
 $A_{[7]} \leftarrow A_{12} - A_{22}$, $B_{[7]} \leftarrow B_{21} + B_{22}$

{Step 2. Conquer} ....

4. **for** $i \leftarrow 1$ **to** 7 **do**
 Strassen-MM$(A_{[i]}, B_{[i]}, P_{[i]}, n/2)$

{Step 3. Combine} ....

5. $C_{11} \leftarrow P_{[1]} + P_{[4]} - P_{[5]} + P_{[7]}$, $C_{12} \leftarrow P_{[3]} + P_{[5]}$,
 $C_{21} \leftarrow P_{[2]} + P_{[4]}$, $C_{22} \leftarrow P_{[1]} - P_{[2]} + P_{[3]} + P_{[6]}$

Step 5. Complexity

- Time complexity.

$$T(n) = \begin{Bmatrix} \Theta\left(1\right) & \text{if } n = 1, \\ 7T(n/2) + \Theta\left(n^2\right) & \text{if } n > 1. \end{Bmatrix}$$
$$\in \Theta\left(n^{\log_2 7}\right)$$

- Space complexity.

$$S(n) = \begin{Bmatrix} \Theta\left(1\right) & \text{if } n = 1, \\ 7S(n/2) + \Theta\left(1\right) & \text{if } n > 1. \end{Bmatrix}$$
$$\in \Theta\left(n^{\log_2 7}\right)$$

Fast MM algorithms

Year	Discoverer	$T(n)$
–	Classical	$\mathcal{O}\left(n^3\right)$
1969	Volker Strassen	$\mathcal{O}\left(n^{2.808}\right)$
1973	Shmuel Winograd	$\mathcal{O}\left(n^{2.808}\right)$
1978	Victor Pan	$\mathcal{O}\left(n^{2.78017}\right)$
1979	Dario Bini et al.	$\mathcal{O}\left(n^{2.7799}\right)$
1979	Victor Pan	$\mathcal{O}\left(n^{2.6054}\right)$
1981	Arnold Schönhage	$\mathcal{O}\left(n^{2.5479}\right)$
1982	Don Coppersmith & Shmuel Winograd	$\mathcal{O}\left(n^{2.4955480}\right)$
1986	Volker Strassen	$\mathcal{O}\left(n^{2.4785}\right)$
1987	Don Coppersmith & Shmuel Winograd	$\mathcal{O}\left(n^{2.3754770}\right)$
2010	Andrew Stothers	$\mathcal{O}\left(n^{2.3737}\right)$
2014	Virginia Vassilevska Williams	$\mathcal{O}\left(n^{2.372873}\right)$
2014	François Le Gall	$\mathcal{O}\left(n^{2.3728639}\right)$