

อัลกอริทึมการแบ่งแยกและเอาชนะ
(Divide and Conquer algorithm)
Part 2

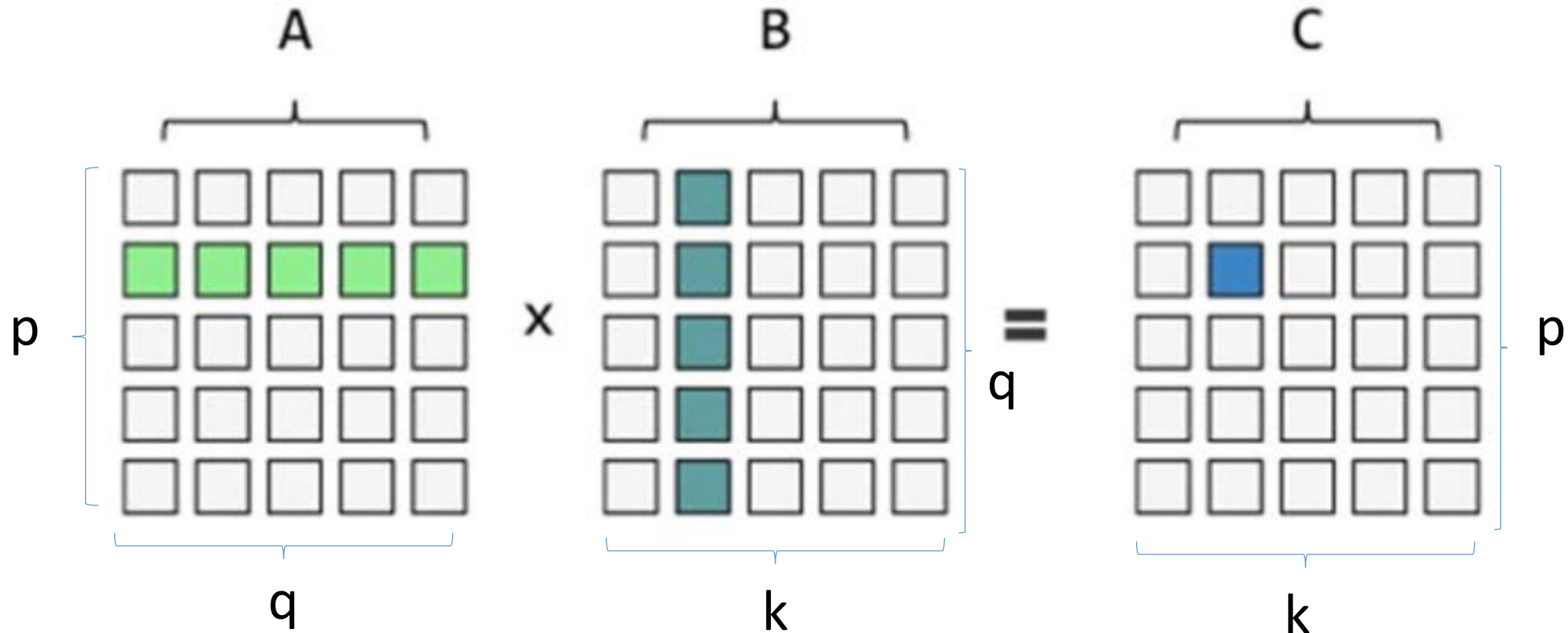
อ.ดร.ลือพล พิพานเมฆาภรณ์

luepol.p@sci.kmutnb.ac.th

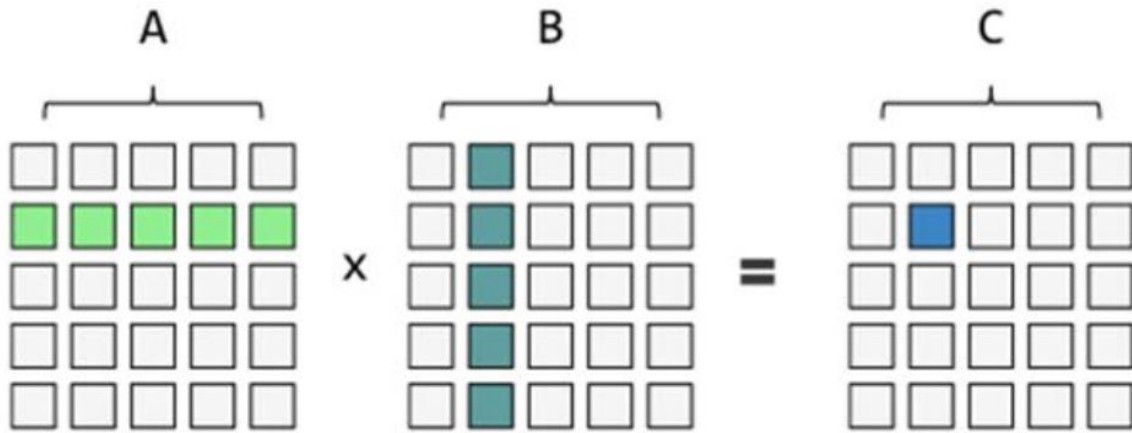
Matrix multiplication

Input: Matrix $A[p \times q]$, $B[q \times k]$

Output: $C = A * B$ where $C[p \times k]$



Matrix Multiplication



$$C_{i,j} = \sum_{k=0}^{N-1} A_{i,k} * B_{k,j}$$

```
for i=0 to N-1:
```

```
    for j=0 to N-1:
```

```
        for k = 0 to N-1:
```

```
            C[i][j] = A[i][k] * B[k][j]
```

$O(N^3)$

Example (N = 2)

$$A = \begin{bmatrix} 1 & 3 \\ 7 & 5 \end{bmatrix} \quad B = \begin{bmatrix} 6 & 8 \\ 4 & 2 \end{bmatrix}$$

$$AB = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

$$AB = \begin{bmatrix} 1 \times 6 + 3 \times 4 & 1 \times 8 + 3 \times 2 \\ 7 \times 6 + 5 \times 4 & 7 \times 8 + 5 \times 2 \end{bmatrix}$$

$$AB = \begin{bmatrix} 18 & 14 \\ 62 & 66 \end{bmatrix}$$

Matrix Multiplication (DAC)

Divide Step: when $N > 1$, split matrix A and B into 8 parts.

$$A = \begin{bmatrix} \begin{matrix} A_{11} & A_{12} \\ a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{matrix} \\ \begin{matrix} A_{21} & A_{22} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix} \end{bmatrix} \quad B = \begin{bmatrix} \begin{matrix} B_{11} & B_{12} \\ b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \end{matrix} \\ \begin{matrix} B_{21} & B_{22} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{matrix} \end{bmatrix}$$

$$C = \begin{bmatrix} C11 & C12 \\ C21 & C22 \end{bmatrix}$$

$$C_{1,1} = A_{1,1}B_{1,1} + A_{1,2}B_{2,1}$$

$$C_{1,2} = A_{1,1}B_{1,2} + A_{1,2}B_{2,2}$$

$$C_{2,1} = A_{2,1}B_{1,1} + A_{2,2}B_{2,1}$$

$$C_{2,2} = A_{2,1}B_{1,2} + A_{2,2}B_{2,2}$$

Combine

Matrix Multiplication (DAC)

Conquer Step: when $N = 1$, multiply $a * b$

$$A_{ij} = [a], \quad B_{ij} = [b] \quad C_{ij} = [a * b]$$

Matrix Multiplication with Recursive

SQUARE-MATRIX-MULTIPLY-RECURSIVE(*A*, *B*)

```
1  n = A.rows
2  let C be a new n × n matrix
3  if n == 1
4      c11 = a11 · b11
5  else partition A, B, and C as in equations (4.9)
6      C11 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A11, B11)
           + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A12, B21)
7      C12 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A11, B12)
           + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A12, B22)
8      C21 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A21, B11)
           + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A22, B21)
9      C22 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A21, B12)
           + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A22, B22)
10 return C
```

$$T(n) = \begin{cases} 8T\left(\frac{n}{2}\right) + n^2 & n > 1 \\ 1 & n = 1 \end{cases}$$

$O(n^3)$

Strassen's algorithm (1969)

$$A = \begin{bmatrix} A_{11} & A_{12} \\ a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ A_{21} & A_{22} \end{bmatrix} \quad B = \begin{bmatrix} B_{11} & B_{12} \\ b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ \hline b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \\ B_{21} & B_{22} \end{bmatrix}$$

ลดปัญหาการรวมคำตอบซึ่งต้องมีการ
คูณทั้งหมด **8** ครั้ง โดยนำเสนองการ
รวมคำตอบแบบใหม่

$$M_1 := (A_{1,1} + A_{2,2})(B_{1,1} + B_{2,2})$$

$$M_2 := (A_{2,1} + A_{2,2})B_{1,1}$$

$$M_3 := A_{1,1}(B_{1,2} - B_{2,2})$$

$$M_4 := A_{2,2}(B_{2,1} - B_{1,1})$$

$$M_5 := (A_{1,1} + A_{1,2})B_{2,2}$$

$$M_6 := (A_{2,1} - A_{1,1})(B_{1,1} + B_{1,2})$$

$$M_7 := (A_{1,2} - A_{2,2})(B_{2,1} + B_{2,2})$$

$$C_{1,1} = M_1 + M_4 - M_5 + M_7$$

$$C_{1,2} = M_3 + M_5$$

$$C_{2,1} = M_2 + M_4$$

$$C_{2,2} = M_1 - M_2 + M_3 + M_6$$

$$C = \begin{bmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}$$

Strassen's algorithm (N = 2)

$$A = \begin{bmatrix} 1 & 3 \\ 7 & 5 \end{bmatrix}$$

$$B = \begin{bmatrix} 6 & 8 \\ 4 & 2 \end{bmatrix}$$

$$M_1 = (1 + 5) \times (6 + 2) = 48$$

$$M_2 = (7 + 5) \times 6 = 72$$

$$M_3 = 1 \times (8 - 2) = 6$$

$$M_4 = 5 \times (4 - 6) = -10$$

$$M_5 = (1 + 3) \times 2 = 8$$

$$M_6 = (7 - 1) \times (6 + 8) = 84$$

$$M_7 = (3 - 5) \times (4 + 2) = -12$$

$$\mathbf{M}_1 := (\mathbf{A}_{1,1} + \mathbf{A}_{2,2})(\mathbf{B}_{1,1} + \mathbf{B}_{2,2})$$

$$\mathbf{M}_2 := (\mathbf{A}_{2,1} + \mathbf{A}_{2,2})\mathbf{B}_{1,1}$$

$$\mathbf{M}_3 := \mathbf{A}_{1,1}(\mathbf{B}_{1,2} - \mathbf{B}_{2,2})$$

$$\mathbf{M}_4 := \mathbf{A}_{2,2}(\mathbf{B}_{2,1} - \mathbf{B}_{1,1})$$

$$\mathbf{M}_5 := (\mathbf{A}_{1,1} + \mathbf{A}_{1,2})\mathbf{B}_{2,2}$$

$$\mathbf{M}_6 := (\mathbf{A}_{2,1} - \mathbf{A}_{1,1})(\mathbf{B}_{1,1} + \mathbf{B}_{1,2})$$

$$\mathbf{M}_7 := (\mathbf{A}_{1,2} - \mathbf{A}_{2,2})(\mathbf{B}_{2,1} + \mathbf{B}_{2,2})$$

Strassen's algorithm (N = 2)

$$C_{1,1} = M_1 + M_4 - M_5 + M_7$$

$$C_{1,2} = M_3 + M_5$$

$$C_{2,1} = M_2 + M_4$$

$$C_{2,2} = M_1 - M_2 + M_3 + M_6$$

$$\begin{aligned} AB &= \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} 48 - 10 - 8 - 12 & 6 + 8 \\ 72 - 10 & 48 - 72 + 6 + 84 \end{bmatrix} \\ &= \begin{bmatrix} 18 & 14 \\ 62 & 66 \end{bmatrix} \end{aligned}$$

Strassen's algorithm

Input: $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ and $B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \in \mathbb{R}^{n \times n}$

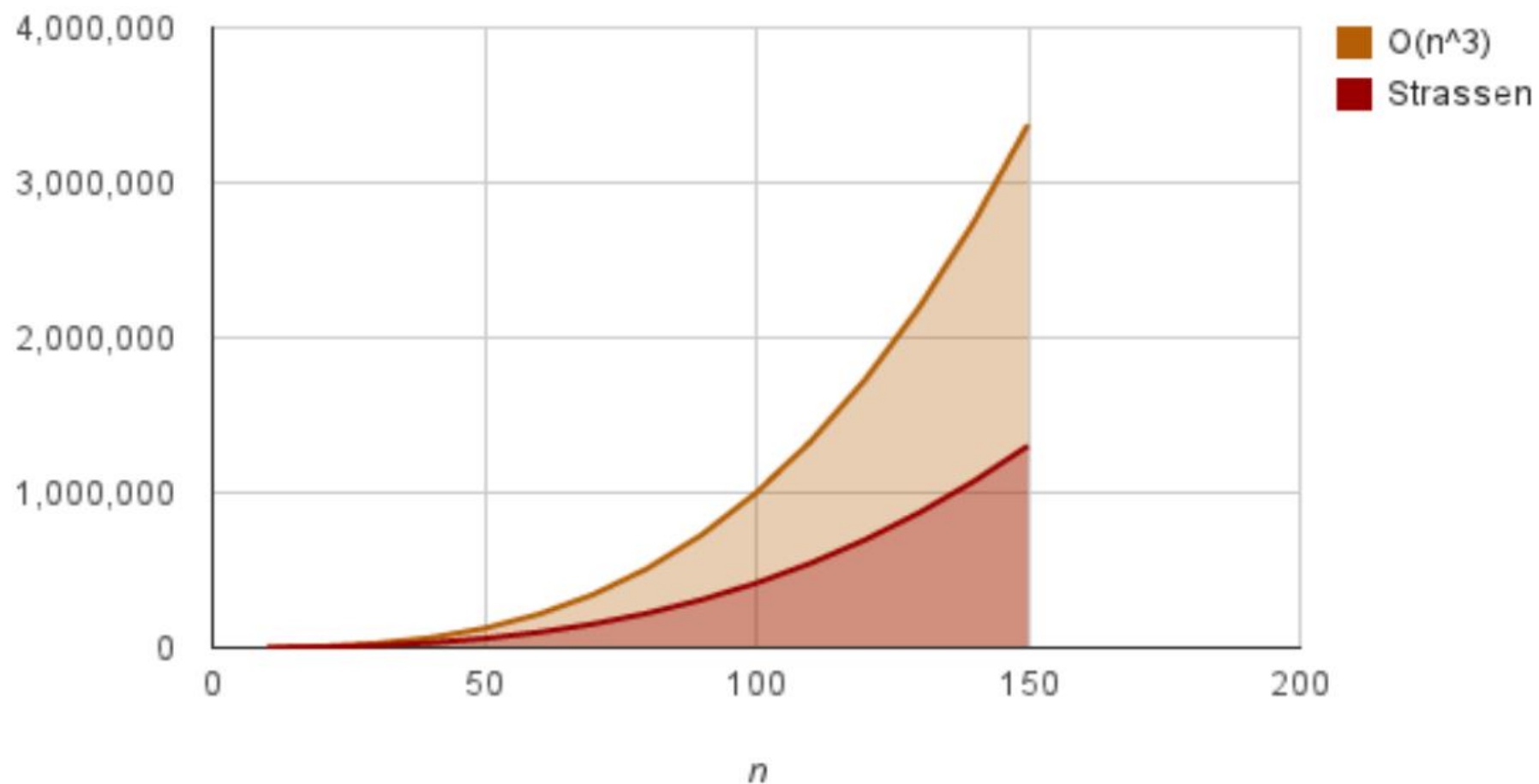
```
1:  if  $n = 1$  then
2:     $C = A \cdot B$ 
3:  else
4:     $M_1 = (A_{11} + A_{22}) \cdot (B_{11} + B_{22})$ 
5:     $M_2 = (A_{21} + A_{22}) \cdot B_{11}$ 
6:     $M_3 = A_{11} \cdot (B_{12} - B_{22})$ 
7:     $M_4 = A_{22} \cdot (B_{21} - B_{11})$ 
8:     $M_5 = (A_{11} + A_{12}) \cdot B_{22}$ 
9:     $M_6 = (A_{21} - A_{11}) \cdot (B_{11} + B_{12})$ 
10:    $M_7 = (A_{12} - A_{22}) \cdot (B_{21} + B_{22})$ 
11:    $C_{11} = M_1 + M_4 - M_5 + M_7$ 
12:    $C_{12} = M_3 + M_5$ 
13:    $C_{21} = M_2 + M_4$ 
14:    $C_{22} = M_1 - M_2 + M_3 + M_6$ 
```

$$T(n) = \begin{cases} 7T\left(\frac{n}{2}\right) + n^2 & n > 1 \\ 1 & n = 1 \end{cases}$$

$$O(n^{2.378})$$

Strassen's Matrix Multiplication

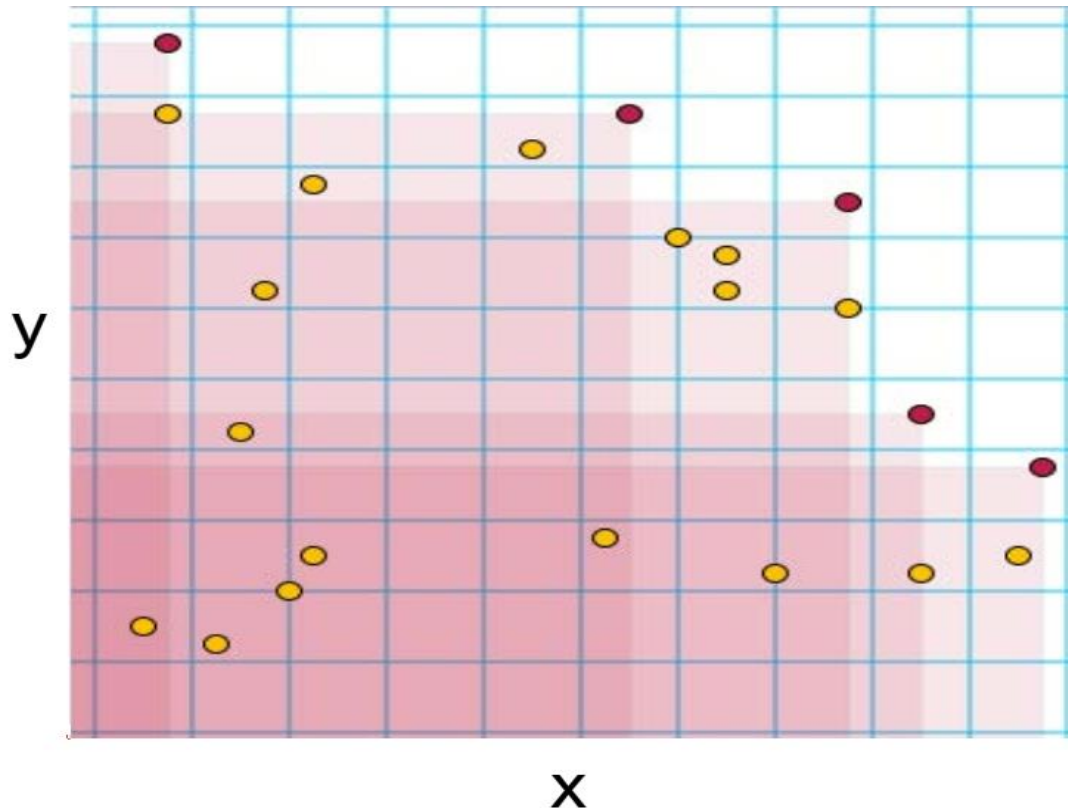
เปรียบเทียบจำนวนครั้งในการคูณเมื่อ n เพิ่มขึ้น



$$T = n^{2.378}$$

Maxima set problem

กำหนดให้ระนาบ X, Y มีจุดทั้งหมด n จุด โดย p คือ maxima หากไม่มีจุด q ใดๆ ที่ครอบคลุม p ได้
 หมายเหตุ จุด p จะครอบคลุมโดย q ก็ต่อเมื่อ $p.x < q.x$ และ $p.y < q.y$

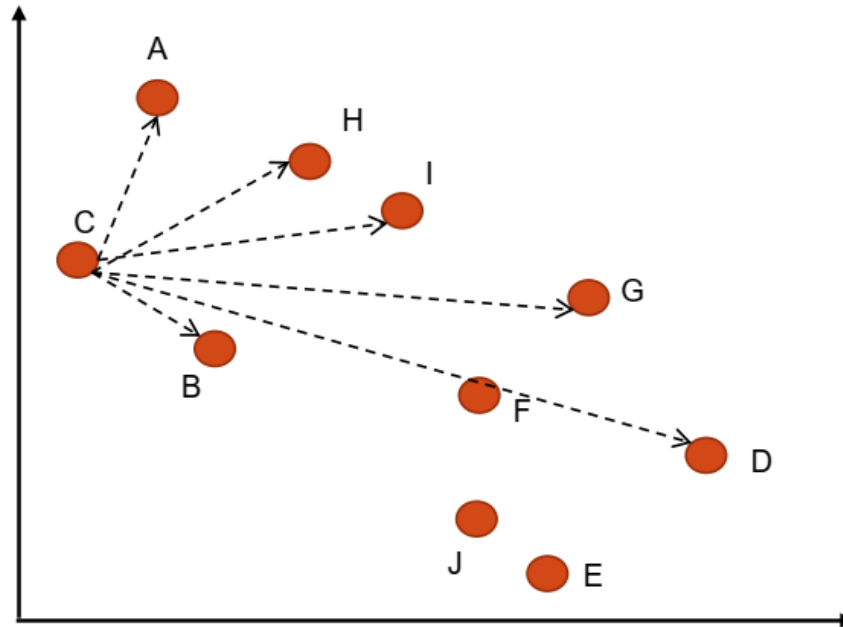


จุดสีแดง คือ maxima set

Finding maxima set (Brute force)

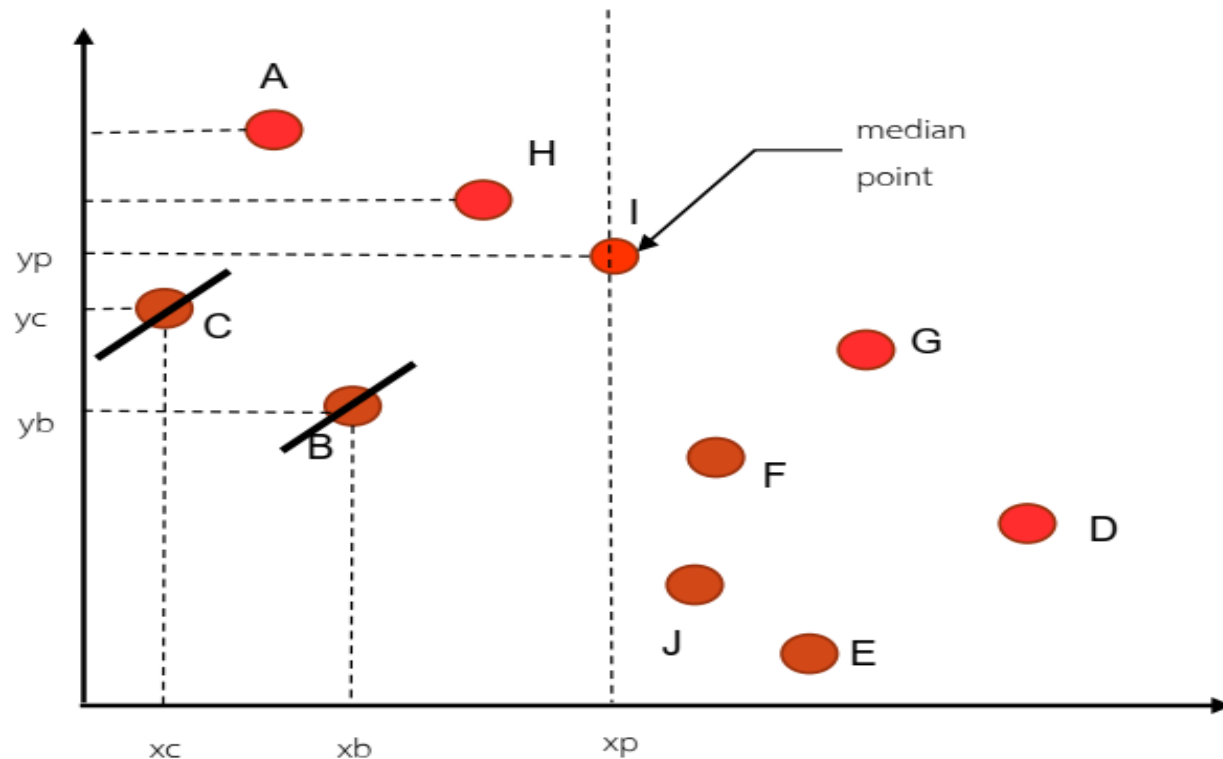
หลักการ brute force ก็คือไล่เช็คทีละจุด โดยหากพบว่ามีย่านน้อย 1 จุดที่ถูกครอบคลุมโดยจุดใดๆ ก็จะได้ถือว่าจุดดังกล่าวไม่ได้อยู่ในเซตคำตอบ

ดังนั้นหากมีทั้งหมด n จุด คาดหวังได้ว่าจำนวนจุดที่ต้องเปรียบเทียบทั้งหมดคือ n^2



Finding maxima set (DAC)

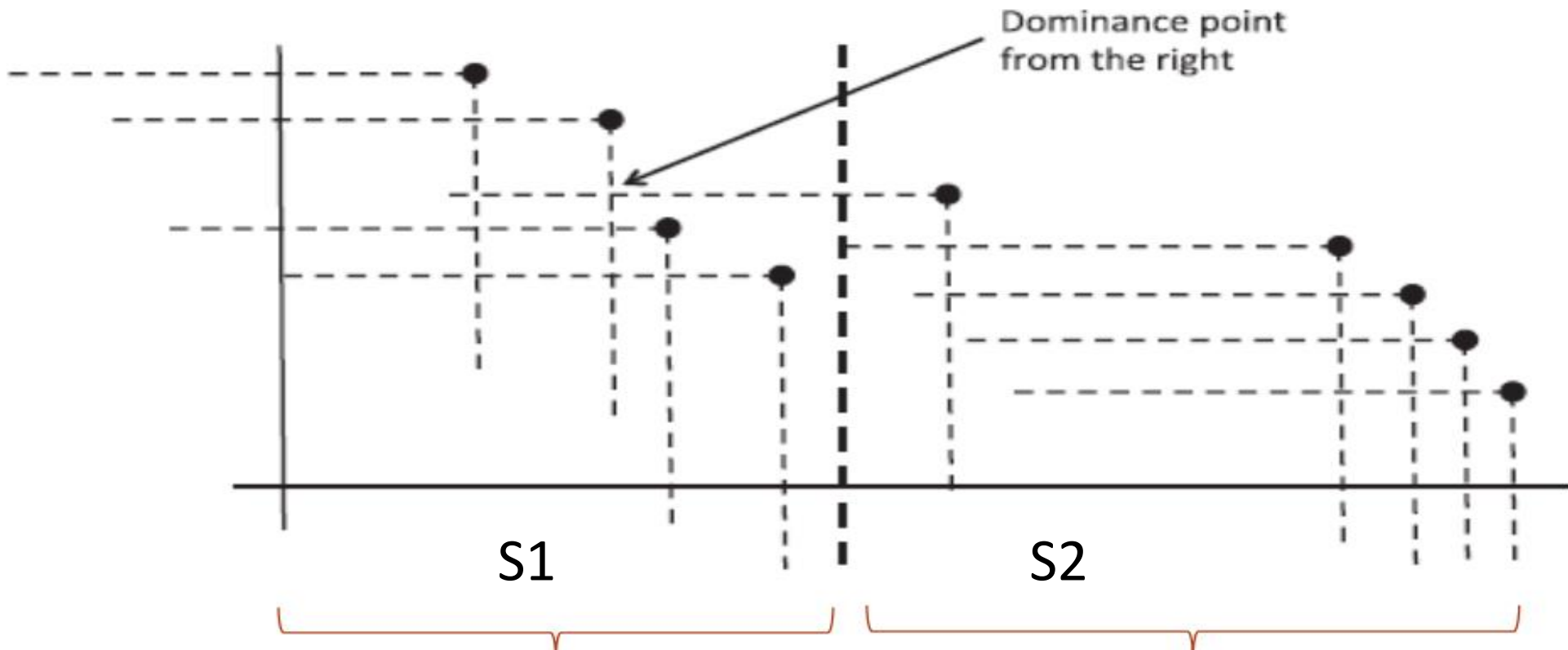
1. เรียงลำดับจุดทั้งหมดสอดคล้องสัมพันธ์กับค่า x (lexicographic order)
2. ค้นหาจุดกลาง (median point) เป็นจุดอ้างอิงและเป็นตัวแบ่งกลุ่มออกเป็น 2 กลุ่ม
3. ทุกๆ จุดด้านซ้ายที่มีค่า y มากกว่า y_p คือ maxima point
4. ทุกๆ จุดด้านขวาเป็น maxima point ยกเว้นแต่จะมีบางจุดด้านขวาที่มีค่า y มากกว่า



Finding maxima set (DAC)

อย่างไรก็ตามอาจมีบางจุดที่เป็น maxima ด้านซ้าย S1 ถูกครอบคลุมโดยบางจุดที่เป็น maxima ด้านขวา S2

วิธีการแก้ปัญหาก็คือทุกครั้งที่มีการ add จุดเข้าไปที่ S2 จะต้องนำค่า y ของจุดนั้นไปเช็คกับทุกๆ จุด maxima ใน S1 และกำจัดทิ้งหากพบว่าค่า y ของจุดใน S1 มีค่าน้อยกว่า



Divide and Conquer

ประมวลผลเบื้องต้น เรียงลำดับจุดทั้งหมดสอดคล้องสัมพันธ์กับค่า x (lexicographic order)

Base case: ถ้า $n = 1$ เพิ่มจุดเข้าไปในเซตคำตอบ S

Divide: ให้ p เป็นจุดกลาง (median) ดังนั้นแบ่งจุดที่เหลือออกเป็น 2 กลุ่มคือ $S1$ และ $S2$
โดยทุกๆ จุดใน $S1$ จะมีค่า x น้อยกว่า x_p มิฉะนั้นจะถูกเก็บไว้ใน $S2$

Combine:

** ให้ q เป็นจุดที่มีค่า x ต่ำสุดใน $S2$ จากนั้นกำจัดทุกๆ จุด ใน $S1$ ที่มีค่า y น้อยกว่า y_q
เซตคำตอบคือ $S = S1 \cup S2$

Algorithm MaximaSet(S):

Input: A set, S , of n points in the plane

Output: The set, M , of maxima points in S

if $n \leq 1$ **then**

return S

Let p be the median point in S , by lexicographic (x, y) -coordinates

Let L be the set of points lexicographically less than p in S

Let G be the set of points lexicographically greater than or equal to p in S

$M_1 \leftarrow \text{MaximaSet}(L)$

$M_2 \leftarrow \text{MaximaSet}(G)$

Let q be the lexicographically smallest point in M_2

for each point, r , in M_1 **do**

if ~~$x(r) \leq x(q)$~~ **and** $y(r) \leq y(q)$ **then**

 Remove r from M_1

return $M_1 \cup M_2$

ตัวอย่าง

กำหนดให้เซตของจุดมีดังนี้ $\{(1,4), (2,6), (3,1), (4,5), (5,7), (6,9), (7,2), (8,6), (9,3)\}$ จงหาเซต maxima

$\{(1,4), (2,6), (3,1), (4,5), (5,7), (6,9), (7,2), (8,6), (9,3)\}$

$(5,7)$

$(6,9), (8,6), (9,3)$

$(1,4) (2,6), (3,1) (4,5) (5,7)$

$(6,9) (7,2) (8,6) (9,3)$

$(2,6), (3,1)$

$(5,7)$

$(6,9), (7,2)$

$(8,6), (9,3)$

$(1,4) (2,6), (3,1)$

$(4,5) (5,7)$

$(6,9) (7,2)$

$(8,6) (9,3)$

$(2,6)$

$(3,1)$

$(4,5)$

$(5,7)$

$(6,9)$

$(7,2)$

$(8,6)$

$(9,3)$

$(1,4) (2,6)$

$(3,1)$

$(4,5)$

$(5,7)$

$(6,9)$

$(7,2)$

$(8,6)$

$(9,3)$

$(1,4) (2,6)$

$(3,1)$

$(4,5)$

$(5,7)$

$(6,9)$

$(7,2)$

$(8,6)$

$(9,3)$