Divide and Conquer

- Steps

Divide the problem into a number of subproblems that are smaller instances of the same problem.

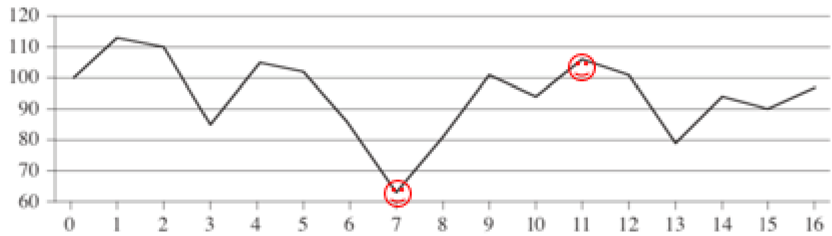
Conquer the subproblems by solving them recursively.

Base case: If the subproblems are small enough, just solve them by brute force.

Combine the subproblem solutions to give a solution to the original problem.

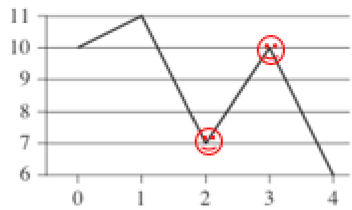
Maximum-subarray Problem

- Price of stock in 17 day-period



Day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Price	100	113	110	85	105	102	86	63	81	101	94	106	101	79	94	90	97
Change		13	−3	−25	20	−3	−16	−23	18	20	−7	12	−5	−22	15	−4	7

- 최대 이익은 최소값이나 최대값과 상관이 없다



Day	0	1	2	3	4
Price	10	11	7	10	6
Change		1	−4	3	−4

- Brute Force Solution : $\Theta(n^2)$

```
for buy_date = 0 to 15
    for sell_date = buy_date+1 to 16
        find maximum price[sell_date] - price[buy_date]
```

Change in stock prices

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	13	−3	−25	20	−3	−16	−23	18	20	−7	12	−5	−22	15	−4	7

maximum subarray

- Definition
  - **Input** : An array **A[1..n]** of numbers (음수가 존재한다고 가정)
  - **Output** : Indices **i and j** such that A[i..j] has the greatest sum of any nonempty, contiguous subarray of A, along with the **sum** of the values in A[i..j]

- Brute Force Solution : $\Theta(n^2)$

```
max = A[1]; max_i = 1; max_j = 1
for i = 1 to 16
    profit = 0
    for j = i to 16
        profit = profit + A[j]
        if (profit > max)
            max = profit; max_i = i, max_j = j
```

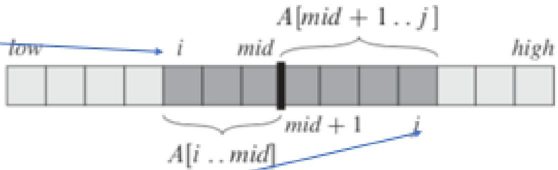
- Divide and Conquer Solution
  - **Divide** : MaxSubarray(A[low..high])를 low~mid, mid+1~high 두 subarray로 나눈다
  - **Conquer** : 두 subarray를 재귀적으로 푼다
  - **Combine** : 세 subarray 중 최대값을 고른다

max_i max_j max : 함수의 return 값이 여러 개일 수 있다.

```
FIND-MAXIMUM-SUBARRAY(A, low, high)
if high == low
    return (low, high, A[low])            // base case: only one element
else mid = ⌊(low + high)/2⌋
    (left-low, left-high, left-sum) =
        FIND-MAXIMUM-SUBARRAY(A, low, mid)
    (right-low, right-high, right-sum) =
        FIND-MAXIMUM-SUBARRAY(A, mid + 1, high)
    (cross-low, cross-high, cross-sum) =
        FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
    if left-sum ≥ right-sum and left-sum ≥ cross-sum
        return (left-low, left-high, left-sum)
    elseif right-sum ≥ left-sum and right-sum ≥ cross-sum
        return (right-low, right-high, right-sum)
    else return (cross-low, cross-high, cross-sum)
```

$=\Theta(n)$

```
FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
left-sum = -∞
sum = 0
for i = mid downto low
    sum = sum + A[i]
    if sum > left-sum
        left-sum = sum
        max-left = i
right-sum = -∞
sum = 0
for j = mid + 1 to high
    sum = sum + A[j]
    if sum > right-sum
        right-sum = sum
        max-right = j
return (max-left, max-right, left-sum + right-sum)
```



- Analysis
  - assume **n** is a power of two
  - base case : $T(1) = \Theta(1)$

- recursive case

$$\begin{aligned} T(n) &= \Theta(1) + 2T(n/2) + \Theta(n) + \Theta(1) \\ &= 2T(n/2) + \Theta(n) . \end{aligned}$$

$$\rightarrow \quad T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 , \\ 2T(n/2) + \Theta(n) & \text{if } n > 1 . \end{cases}$$

$\rightarrow \quad T(n) = \Theta(n \lg n)$ by master method, $< \quad \Theta(n^2)$

Matrix Multiplication

- Definition

Input: Two $n \times n$ (square) matrices, $A = (a_{ij})$ and $B = (b_{ij})$

Output: $n \times n$ matrix $C = (c_{ij})$, where $C = A \cdot B$, i.e.,

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

for $i, j = 1, 2, \ldots, n$.

- Brute Force Solution

```
SQUARE-MATRIX-MULTIPLY(A, B)
n = A.rows
let C be a new n × n matrix
for i = 1 to n
    for j = 1 to n
        c_ij = 0
        for k = 1 to n
            c_ij = c_ij + a_ik · b_kj
return C
```

$\Theta(n^3)$

- Divide and Conquer Solution
  - **Divide** : A,B,C 행렬을 1/4씩 나눈다

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \quad C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} 1 & 2 \\ 5 & 6 \end{pmatrix} & \begin{pmatrix} 3 & 4 \\ 7 & 8 \end{pmatrix} \\ \begin{pmatrix} 9 & 10 \\ 13 & 14 \end{pmatrix} & \begin{pmatrix} 11 & 12 \\ 15 & 16 \end{pmatrix} \end{pmatrix}$$

  - **Conquer**

$A_{11}B_{11}$, $A_{12}B_{21}$,$A_{11}B_{12}$, $A_{12}B_{22}$,$A_{21}B_{11}$, $A_{22}B_{21}$,$A_{21}B_{12}$, $A_{22}B_{22}$ 를 계산한다.

  - **Combine**

$$\begin{aligned} C_{11} &= A_{11} \cdot B_{11} + A_{12} \cdot B_{21} \\ C_{12} &= A_{11} \cdot B_{12} + A_{12} \cdot B_{22} \\ C_{21} &= A_{21} \cdot B_{11} + A_{22} \cdot B_{21} \\ C_{22} &= A_{21} \cdot B_{12} + A_{22} \cdot B_{22} \end{aligned}$$

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

- Analysis
  - assume n is a power of two
  - base case : $T(1) = \Theta(1)$
  - recursive case

행렬 덧셈에 걸리는 시간

$$\begin{aligned} T(n) &= \Theta(1) + 8T(n/2) + \Theta(n^2) \\ &= 8T(n/2) + \Theta(n^2) . \end{aligned}$$

$$\rightarrow \quad T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ 8T(n/2) + \Theta(n^2) & \text{if } n > 1 \end{cases}$$

$\rightarrow \quad T(n) = \Theta(n^3).$ by master method

- Strassen's matrix multiplication
  - **Divide** : A,B,C 행렬을 1/4씩 나눈다

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \quad C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

  - **Conquer**

S_1~S_{10} , P_1~P_7 을 게산한다. recursive call

  - **Combine**

$$\begin{aligned} C_{11} &= P_5 + P_4 - P_2 + P_6 , \\ C_{12} &= P_1 + P_2 , \\ C_{21} &= P_3 + P_4 , \\ C_{22} &= P_5 + P_1 - P_3 - P_7 . \end{aligned}$$

$$
\begin{aligned}
S_1 &= B_{12} - B_{22} \\
S_2 &= A_{11} + A_{12} \\
S_3 &= A_{21} + A_{22} \\
S_4 &= B_{21} - B_{11} \\
S_5 &= A_{11} + A_{22} \\
S_6 &= B_{11} + B_{22} \\
S_7 &= A_{12} - A_{22} \\
S_8 &= B_{21} + B_{22} \\
S_9 &= A_{11} - A_{21} \\
S_{10} &= B_{11} + B_{12}
\end{aligned}
$$

$$
\begin{aligned}
P_1 &= A_{11} \cdot S_1 &&= A_{11} \cdot B_{12} - A_{11} \cdot B_{22} , \\
P_2 &= S_2 \cdot B_{22} &&= A_{11} \cdot B_{22} + A_{12} \cdot B_{22} , \\
P_3 &= S_3 \cdot B_{11} &&= A_{21} \cdot B_{11} + A_{22} \cdot B_{11} , \\
P_4 &= A_{22} \cdot S_4 &&= A_{22} \cdot B_{21} - A_{22} \cdot B_{11} , \\
P_5 &= S_5 \cdot S_6 &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} , \\
P_6 &= S_7 \cdot S_8 &&= A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22} , \\
P_7 &= S_9 \cdot S_{10} &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{12} - A_{21} \cdot B_{11} - A_{21} \cdot B_{12} .
\end{aligned}
$$

Strassen's matrix multiplication

$$
\begin{aligned}
C_{11} &= P_5 + P_4 - P_2 + P_6 , \\
C_{12} &= P_1 + P_2 , \\
C_{21} &= P_3 + P_4 , \\
C_{22} &= P_5 + P_1 - P_3 - P_7 .
\end{aligned}
$$

$$
\begin{aligned}
P_1 &= A_{11} \cdot S_1 &&= A_{11} \cdot B_{12} - A_{11} \cdot B_{22} , \\
P_2 &= S_2 \cdot B_{22} &&= A_{11} \cdot B_{22} + A_{12} \cdot B_{22} , \\
P_3 &= S_3 \cdot B_{11} &&= A_{21} \cdot B_{11} + A_{22} \cdot B_{11} , \\
P_4 &= A_{22} \cdot S_4 &&= A_{22} \cdot B_{21} - A_{22} \cdot B_{11} , \\
P_5 &= S_5 \cdot S_6 &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} , \\
P_6 &= S_7 \cdot S_8 &&= A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22} , \\
P_7 &= S_9 \cdot S_{10} &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{12} - A_{21} \cdot B_{11} - A_{21} \cdot B_{12} .
\end{aligned}
$$

$$
\begin{aligned}
C_{11} &= P_5 + P_4 - P_2 + P_6 , \\
C_{12} &= P_1 + P_2 , \\
C_{21} &= P_3 + P_4 , \\
C_{22} &= P_5 + P_1 - P_3 - P_7 .
\end{aligned}
$$

- C_{11}

$$
\begin{array}{l}
A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} \\
\qquad - A_{22} \cdot B_{11} \qquad + A_{22} \cdot B_{21} \\
\quad - A_{11} \cdot B_{22} \qquad\qquad - A_{12} \cdot B_{22} \\
\qquad\qquad - A_{22} \cdot B_{22} - A_{22} \cdot B_{21} + A_{12} \cdot B_{22} + A_{12} \cdot B_{21} \\
\hline
A_{11} \cdot B_{11} \qquad\qquad\qquad + A_{12} \cdot B_{21}
\end{array}
$$

- C_{12}

$$
\begin{array}{l}
A_{11} \cdot B_{12} - A_{11} \cdot B_{22} \\
\qquad + A_{11} \cdot B_{22} + A_{12} \cdot B_{22} \\
\hline
A_{11} \cdot B_{12} \qquad + A_{12} \cdot B_{22}
\end{array}
$$

$$
\begin{aligned}
P_1 &= A_{11} \cdot S_1 &&= A_{11} \cdot B_{12} - A_{11} \cdot B_{22} , \\
P_2 &= S_2 \cdot B_{22} &&= A_{11} \cdot B_{22} + A_{12} \cdot B_{22} , \\
P_3 &= S_3 \cdot B_{11} &&= A_{21} \cdot B_{11} + A_{22} \cdot B_{11} , \\
P_4 &= A_{22} \cdot S_4 &&= A_{22} \cdot B_{21} - A_{22} \cdot B_{11} , \\
P_5 &= S_5 \cdot S_6 &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} , \\
P_6 &= S_7 \cdot S_8 &&= A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22} , \\
P_7 &= S_9 \cdot S_{10} &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{12} - A_{21} \cdot B_{11} - A_{21} \cdot B_{12} .
\end{aligned}
$$

$$
\begin{aligned}
C_{11} &= P_5 + P_4 - P_2 + P_6 , \\
C_{12} &= P_1 + P_2 , \\
C_{21} &= P_3 + P_4 , \\
C_{22} &= P_5 + P_1 - P_3 - P_7 .
\end{aligned}
$$

- C_{21}

$$
\begin{array}{l}
A_{21} \cdot B_{11} + A_{22} \cdot B_{11} \\
\qquad - A_{22} \cdot B_{11} + A_{22} \cdot B_{21} \\
\hline
A_{21} \cdot B_{11} \qquad + A_{22} \cdot B_{21}
\end{array}
$$

- C_{22}

$$
\begin{array}{l}
A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} \\
\qquad - A_{11} \cdot B_{22} \qquad\qquad + A_{11} \cdot B_{12} \\
\qquad\qquad - A_{22} \cdot B_{11} \qquad\qquad - A_{21} \cdot B_{11} \\
- A_{11} \cdot B_{11} \qquad\qquad - A_{11} \cdot B_{12} + A_{21} \cdot B_{11} + A_{21} \cdot B_{12} \\
\hline
\qquad\qquad A_{22} \cdot B_{22} \qquad\qquad + A_{21} \cdot B_{12}
\end{array}
$$

- Analysis
  - Assume n is power of two

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ 7T(n/2) + \Theta(n^2) & \text{if } n > 1 \end{cases}$$

행렬 덧셈에 걸리는 시간

→ $T(n) = \Theta(n^{\lg 7})$ where 2.80< lg7< 2.81

→ $T(n) = O(n^{2.81}) \quad < \quad \Theta(n^3)$