Divide-and-Conquer (분할 정복)

such as mergesort

Steps in Divide-and-Conquer-and-Combine

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

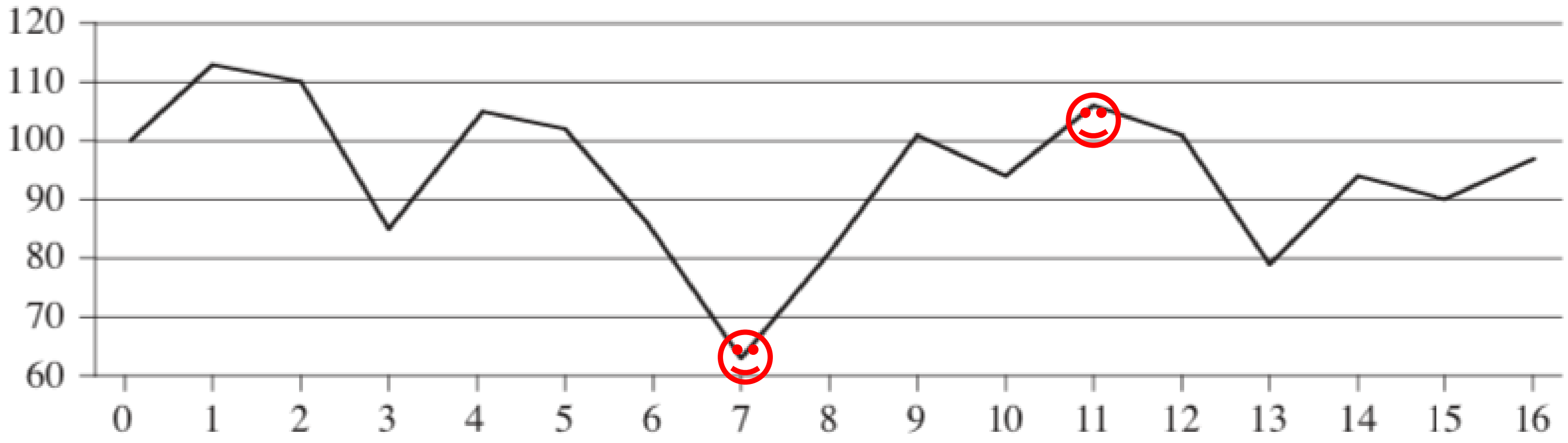
Conquer the subproblems by solving them recursively.
Base case: If the subproblems are small enough, just solve them by brute force.

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

Example 1: Maximum-subarray problem

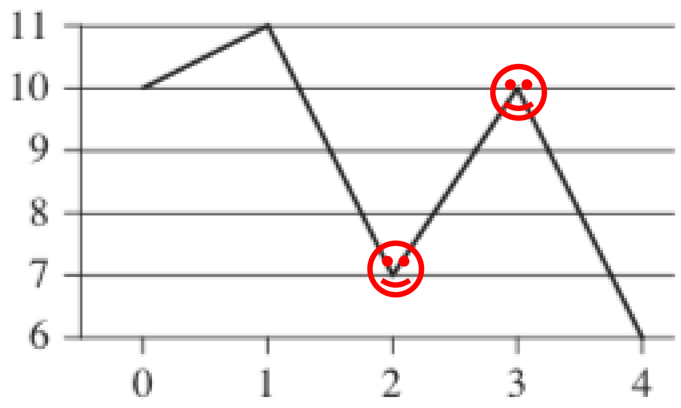
- price of stock in 17 day-period

(미래를 알 수 없는 실제 주식시장과는 다름)



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 (주먹구구) solution

for buy_date = 0 to 15

  for sell_date = buy_date+1 to 16

    find maximum price[sell_date] – price[buy_date]

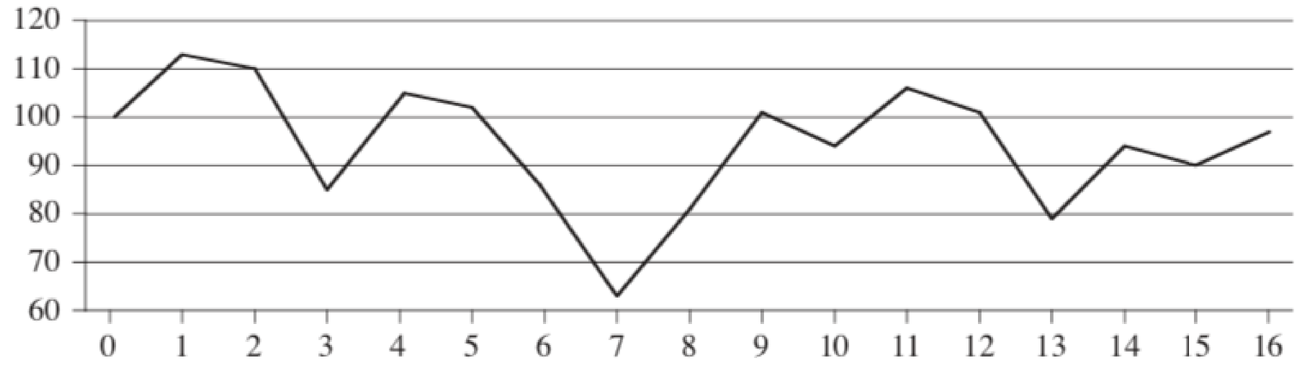
$\Theta(n^2)$

Find a maximum-subarray in the following array

- 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



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

Definition of maximum-subarray problem

Input: An array $\boxed{A[1..n]}$ of numbers. *[Assume that some of the numbers are negative, because this problem is trivial when all numbers are nonnegative.]*

Output: Indices $\boxed{i \text{ and } j}$ such that $A[i..j]$ has the greatest sum of any nonempty, contiguous subarray of A, along with the $\boxed{\text{sum}}$ of the values in $A[i..j]$.

Brute-force (주먹구구) solution

	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

```
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
```

$\Theta(n^2)$

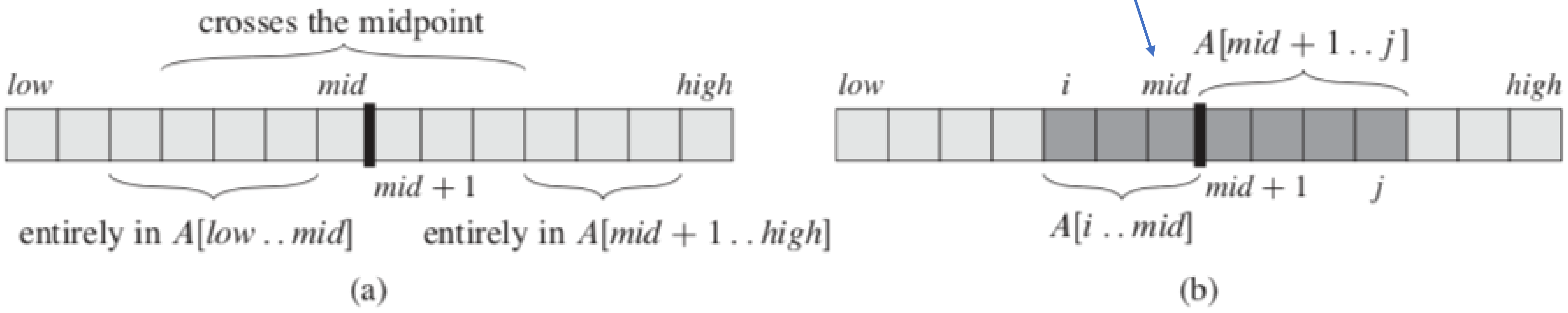
Divide-and-conquer solution

1. Divide : MaxSubarray(A[low…high]) 를 MaxSubarray(A[low…mid]) 과 MaxSubarray(A[mid+1…high]) 으로 나눈다.

2. Conquer : MaxSubarray(A[low…mid]) 과 MaxSubarray(A[mid+1…high]) 를 재귀적으로 푼다.

3. Combine : MaxSubarray(A[low…mid]) ,MaxSubarray(A[mid+1…high]),

MaxCrossingSubarray(A[low…high]) 중에서 최대값을 고른다.

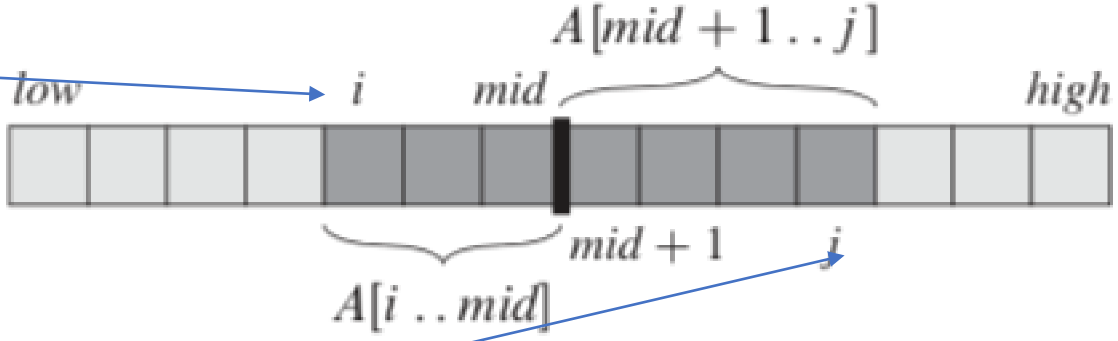


```
FIND-MAXIMUM-SUBARRAY(A, low, high)
if high == low                max_i    max_j    max   : 함수의 return 값이 여러 개일 수 있다.
    return (low, high, A[low])                // base case: only one element
else mid = ⌊(low + high)/2⌋
    (left-low, left-high, left-sum) =
        FIND-MAXIMUM-SUBARRAY(A, low, 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)
```

FIND-MAX-CROSSING-SUBARRAY$(A, low, mid, high)$

$=\Theta(n)$

```
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 of FIND-MAXIMUM-SUBARRAY

- assume n is a power of two.
- base case : $T(1) = \Theta(1)$ (상수 시간)
- recursive case : $$\begin{aligned} T(n) &= \Theta(1) + 2T(n/2) + \Theta(n) + \Theta(1) \\ &= 2T(n/2) + \Theta(n) . \end{aligned}$$

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

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

Example 2: matrix multiplication

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 matrix multiplication

SQUARE-MATRIX-MULTIPLY(A, B)

$n = A.rows$
let C be a new $n \times n$ matrix
for $i = 1$ **to** n
    for $j = 1$ **to** n
        $c_{ij} = 0$
        for $k = 1$ **to** n
            $c_{ij} = c_{ij} + a_{ik} \cdot b_{kj}$
return C

$\Theta(n^3)$

Divide-and-Conquer matrix multiplication

1. Divide : A,B,C 행렬을 ¼ 씩 나눈다.

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

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} 1 & 2 \\ 5 & 6 \end{pmatrix} & \begin{pmatrix} 3 & 4 \\ 7 & 8 \end{pmatrix} \\ \begin{pmatrix} 9 & 10 \\ 13 & 14 \end{pmatrix} & \begin{pmatrix} 11 & 12 \\ 15 & 16 \end{pmatrix} \end{pmatrix}$$

2. 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.

3. Combine :

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

Divide-and-Conquer matrix multiplication

```
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 of SQUARE-MATRIX-MULTIPLY-RECURSIVE

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

행렬 덧셈에 걸리는 시간

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

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

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

Strassen’s (Divide-and-Conquer) matrix multiplication

1. Divide : A,B,C 행렬을 ¼ 씩 나눈다.

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

2. Conquer : $S_1 \sim S_{10}$, $P_1 \sim P_7$ 을 계산한다. recursive call
3. 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}

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

- 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}

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

- C_{22}

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

Analysis of Strassen's matrix multiplication

- assume n is a 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

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