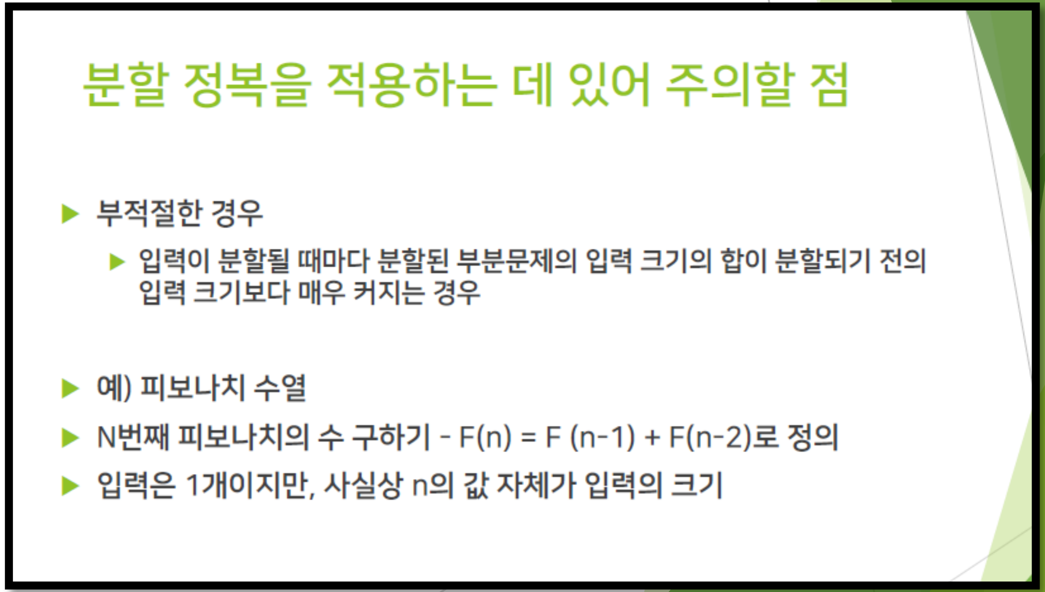
동적 프로그래밍 방법_1

2020년도 2학기 최 희석

목차

- 동적 프로그래밍 방법이란
- 피보나치 수열 문제
- 연쇄 행렬 곱셈 문제



동적 프로그래밍 방법이란

동적 프로그래밍 방법이란

- 문제의 크기가 작은 소문제에 대한 해를 저장해 놓고, 이를 이용하여 크기가 보다 큰 문제의 해를 점진적으로 만들어가는 상향식 접근 bottom-up 방법
  - 각 작은 문제는 원래의 문제와 동일한 문제이지만 입력의 크기만 작음
  - 입력의 크기가 아주 작은 단순한 문제가 되면 쉽게 해를 구할 수 있고, 이를 **테이블**에 저장
  - 이후 해당 소문제의 해가 필요할 때마다 테이블에 저장된 결과를 바로 이용
- 동적 "프로그래밍" dynamic programming ⇒ **동적 계획법**
  - 컴퓨터에서의 프로그램과는 무관, 해를 구축하는 테이블을 이용한다는 의미
- 최소값/최대값을 구하는 최적화 문제에 적용

동적 프로그래밍 방법이란

- 최적화 문제에 동적 프로그래밍 방법을 적용하려면?
  - **최적성의 원리** (principle of optimality)를 반드시 만족시켜야 함
    - "주어진 문제에 대한 최적해는 주어진 문제의 소문제에 대한 최적해로 구성된다"

- 동적 프로그래밍 방법의 처리 과정
  - 주어진 문제에 대해서 최적해를 제공하는 점화식을 도출
  - 가장 작은 소문제부터 점화식의 해를 구한 뒤 이를 테이블에 저장
  - 테이블에 저장되어 있는 소문제의 해를 이용하여 점차적으로 큰 상위 문제의 해를 구함

동적 프로그래밍 방법이란

분할 정복 방법	동적 프로그래밍 방법
하향식 접근 방법	상향식 접근 방법
상위 레벨의 큰 문제를 순환적으로 부분 배열로 분할하고, 이들의 해를 결합해서 원래 문제의 해결하는 방법	입력 크기가 작은 소문제들을 모두 해결하여 해를 테이블에 저장한 후, 이 해들을 이용하여 보다 큰 크기의 문제를 해결하는 방법
분할된 작은 문제들은 서로 독립적	소문제들이 서로 독립적이지 않고, 중복되는 부분이 존재
이진 탐색, 합병 정렬, 퀵 정렬, 선택 문제	피보나치 수열, 연쇄 행렬 곱셈, 스트링 편집 거리, 모든 정점 간의 최단 경로(플로이드 알고리즘), 저울 문제

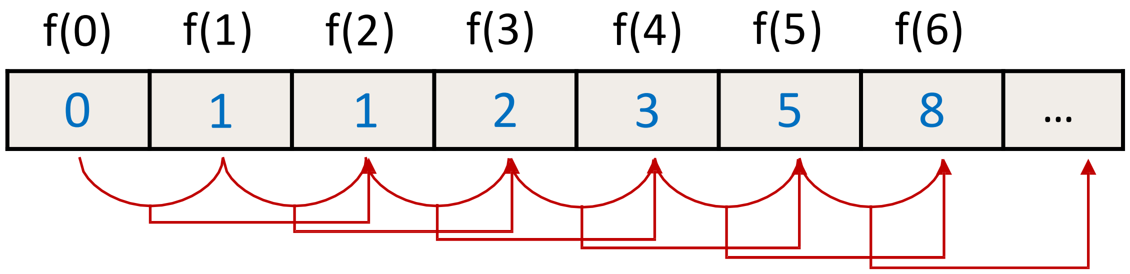
피보나치 수열 문제

피보나치 수열 문제

- 피보나치 수열의 개념과 원리

$$f(n) = \begin{cases} f(n-1) + f(n-2), & n \geq 2 \\ f(0) = 0, \quad f(1) = 1 \end{cases}$$

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, …



피보나치 수열 문제

```
Fibo(n)
{
   f[0]=0; f[1]=1;
   for (i=2; i<=n; i++)
   f[i] = f[i-1] + f[i-2];
   return f[n];
}
```

O(n)

피보나치 수열 문제

분할 정복 방법	동적 프로그래밍 방법
하향식 접근 방법	상향식 접근 방법
상위 레벨의 큰 문제를 순환적으로 부분 배열로 분할하고, 이들의 해를 결합해서 원래 문제의 해결하는 방법	입력 크기가 작은 소문제들을 모두 해결하여 해를 테이블에 저장한 후, 이 해들을 이용하여 보다 큰 크기의 문제를 해결하는 방법
분할된 작은 문제들은 서로 독립적	소문제들이 서로 독립적이지 않고, 중복되는 부분이 존재
이진 탐색, 합병 정렬, 퀵 정렬, 선택 문제	피보나치 수열, 연쇄 행렬 곱셈, 스트링 편집거리, 모든 정점 간의 최단 경로(플로이드 알고리즘), 저울 문제

- 분할정복 방법을 적용하면?
- 소문제가 독립이 아니기 때문에 분할정복 방법 적용 X
- 중복된 계산이 필요 -> 매우 비효율적

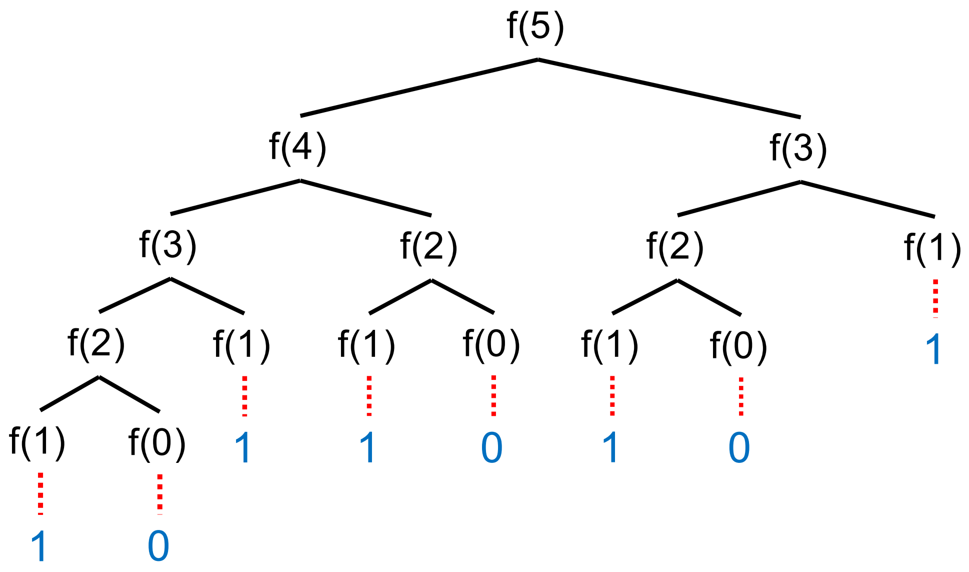
$$f(n)=\begin{cases} f(n-1)+f(n-2), & n\geq 2 \\ f(0)=0, \quad f(1)=1 & \end{cases}$$

순환 형태의 알고리즘

```
int f(n) {
  if (n<=0) return 0;
  if (n==1) return 1;
  else return( f(n-1) + f(n-2) );
}
```

피보나치 수열 문제

순환 형태를 이용한 피보나치 수열 f(5) 계산의 구조도



연쇄 행렬 곱셈 문제

연쇄 행렬 곱셈 문제의 개념과 원리

- n개의 행렬($M_1,M_2,\ldots,M_n$)을 연쇄적으로 곱하는 경우
  - 결합법칙 성립
    - $M_1 \cdot M_2 \cdot M_3 \rightarrow ((M_1 \cdot M_2) \cdot M_3) = (M_1 \cdot (M_2 \cdot M_3))$
    - 여러 가지 다른 곱셈 순서가 존재 → 곱셈의 횟수가 달라짐
- 연쇄 행렬 곱셈 문제?
  - n개의 행렬을 연쇄적으로 곱할 때 최적의 곱셈 순서를 구하는 문제
    - ⇒ 최소의 기본 곱셈 횟수를 가진 행렬의 곱셈 순서를 구하는 문제

연쇄 행렬 곱셈 문제

- 기본 곱셈
  - 행렬 원소끼리의 곱셈

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix} \times \begin{bmatrix} 9 & 11 & 13 \\ 10 & 12 & 15 \end{bmatrix} = \begin{bmatrix} 29 & 35 & 41 \\ 67 & 81 & 95 \\ 105 & 127 & 149 \\ 143 & 173 & 203 \end{bmatrix}$$

4×2   2×3   4×3

$$\underset{m\times n}{M_i} \times \underset{n\times r}{M_j} \xrightarrow[m\times n\times r]{\text{곱셈횟수}} \underset{m\times r}{M_i \cdot M_j}$$

연쇄 행렬 곱셈 문제

- M_1 : 30×5 M_2 : 5×20 M_3 : 20×15 M_4 : 15×10

$M_1(M_2(M_3M_4))$	$20 \times 15 \times 10 + 5 \times 20 \times 10 + 30 \times 5 \times 10 = 5,500$
$(M_1M_2)(M_3M_4)$	$30 \times 5 \times 20 + 20 \times 15 \times 10 + 30 \times 20 \times 10 = 12,000$
$M_1((M_2M_3)M_4)$	$5 \times 20 \times 15 + 5 \times 15 \times 10 + 30 \times 5 \times 10$ = 3,750
$((M_1M_2)M_3)M_4$	$30 \times 5 \times 20 + 30 \times 20 \times 15 + 30 \times 15 \times 10 = 16,500$
$(M_1(M_2M_3))M_4$	$5 \times 20 \times 15 + 30 \times 5 \times 15 + 30 \times 15 \times 10 = 6,900$

연쇄 행렬 곱셈 문제의 최적성의 원리

- n개의 행렬을 곱하는 최적의 순서는 n개 행렬의 어떤 부분집합을 곱하는 최적의 순서를 포함

7개 행렬을 곱하는 최적의 순서 $(M_1M_2)((((M_3M_4)M_5)M_6)M_7)$

→ M_3, M_4, M_5를 곱하는 최적의 순서 $((M_3M_4)M_5)$

부분 문제들의 최적해로 n개 행렬을 곱하는 최적의 순서를 구할 수 있음

최적성의 원리가 성립

동적 프로그래밍 방법으로 해결 가능

연쇄 행렬 곱셈 문제의 점화식

- n개의 행렬 Mi ($d_{i-1} X\ d_i$ 차원)(1≤i≤n)

M_1	×	M_2	×	M_3	×	…	×	M_{n-1}	×	M_n
$d_0 \times d_1$		$d_1 \times d_2$		$d_2 \times d_3$				$d_{n-2} \times d_{n-1}$		$d_{n-1} \times d_n$

C(i,j) 1≤i ≤j ≤n

$M_i \times M_{i+1} \times M_{i+2} \times \ldots \times M_{j-1} \times M_j$를 수행하는 데 필요한 곱셈의 최소 횟수

C(1,n) = ?

연쇄 행렬 곱셈 문제의 점화식

$C(i, i) = 0, \ 1 \le i \le n$

$C(i, i+1) = d_{i-1}\, d_i\, d_{i+1}$

결과 행렬의 크기: $d_{i-1} \times d_{i+1}$

$C(i, i+2) = \min\{ M_i(M_{i+1}M_{i+2})$ + 결합비용,

$(M_iM_{i+1})M_{i+2}$ + 결합비용 }

$$= \min\{ C(i, i)+C(i+1, i+2) + d_{i-1}d_id_{i+2} ,$$

$$C(i, i+1)+C(i+2, i+2) + d_{i-1}d_{i+1}d_{i+2} \}$$

연쇄 행렬 곱셈 문제의 점화식

$$
\begin{aligned}
C(i, i+3) = \min\{ & M_i(M_{i+1}M_{i+2}M_{i+3}) + \text{결합비용}, \\
& (M_iM_{i+1})(M_{i+2}M_{i+3}) + \text{결합비용}, \\
& (M_iM_{i+1}M_{i+2})M_{i+3} + \text{결합비용} \} \\
= \min\{ & C(i, i)+C(i+1, i+3) + d_{i-1}d_id_{i+3}, \\
& C(i, i+1)+C(i+2, i+3) + d_{i-1}d_{i+1}d_{i+3}, \\
& C(i, i+2)+C(i+3, i+3) + d_{i-1}d_{i+2}d_{i+3} \}
\end{aligned}
$$

일반화된 점화식

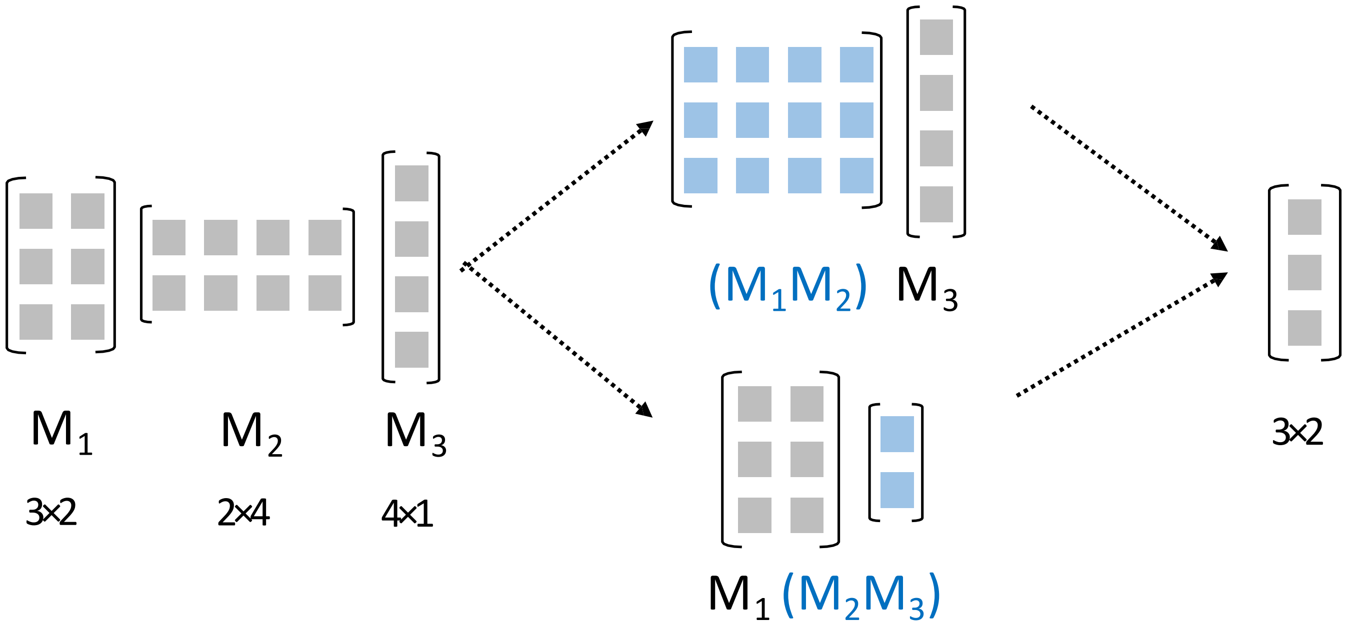
$$
\begin{aligned}
C(i, j) &= \min_{i \le k \le j-1}\{ (M_i \ldots M_k)(M_{k+1} \ldots M_j) + \text{결합비용} \} \\
&= \min_{i \le k \le j-1}\{ C(i, k) + C(k+1, j) + d_{i-1}d_kd_j \}
\end{aligned}
$$

연쇄 행렬 곱셈 문제의 알고리즘

```
MinMatMult (n, d[])
{
   int i, j, k, s;
   int C[n+1][n+1];
   for (i=1; i<=n; i++) C[i][i] =  0;
   for (s=1; s<=n-1; s++)
      for (i=1; i<=n-s; i++) {
          j = i+s;
          C[i][j] = min_{i≤k≤j-1}(C[i][k]+C[k+1][j]+d[i-1]d[k]d[j]);
          P[i][j] = 최소값이 되는 k의 값;
      }
   Return C[1][n];
}
```

입력: 행렬의 개수 n, 행렬의 크기 d[0..n] (i번째 행렬의 크기 d[i-1]×d[i])
출력: C[1][n] : n개의 행렬을 곱하는 데 필요한 곱셈 횟수의 최소값
P[1..n][1..n] : 최적의 곱셈 순서를 구할 수 있는 배열

연쇄 행렬 곱셈 문제의 적용 예_1



연쇄 행렬 곱셈 문제의 적용 예_1

M_1	$\times$	M_2	$\times$	M_3
3×2		2×4		4×1

$d_0=3,\ d_1=2,\ d_2=4,\ d_3=1$

C(i, j)

i \ j	1	2	3
1			C(1,3)
2			
3			

연쇄 행렬 곱셈 문제의 적용 예_1

s=j-i=0 (i=j)

C(i, i) = 0, 1≤i≤3

C(i, j)

i \ j	1	2	3
1	0		
2		0	
3			0

연쇄 행렬 곱셈 문제의 적용 예_1

M_1	×	M_2	×	M_3
3×2		2×4		4×1

$d_0=3,\ d_1=2,\ d_2=4,\ d_3=1$

s=j-i=1

i=1 → $C(1, 2) = \min_{1 \le k \le 1}\{ C(1,k)+C(k+1,2)+d_0 d_k d_2 \}$

$= C(1,1)+C(2,2)+(3\times2\times4) = 24$

i=2 → $C(2, 3) = \min_{2 \le k \le 2}\{ C(2,k)+C(k+1,3)+d_1 d_k d_3 \}$

$= C(2,2)+C(3,3)+(2\times4\times1) = 8$

C(i, j)

i \ j	1	2	3
1	0	24	
2		0	8
3			0

P(i, j)

	1	2	3
1		1	
2			2
3			

연쇄 행렬 곱셈 문제의 적용 예_1

s=j-i=2

$$i=1 \rightarrow C(1,3) = \min_{1\le k\le 2}\{ C(1,k)+C(k+1,3)+d_0 d_k d_3 \}$$
$$= \min \{ C(1,1)+C(2,3)+(3\times 2\times 1), \; C(1,2)+C(3,3)+(3\times 4\times 1) \}$$
$$= \min \{ 14, 36 \} = 14$$

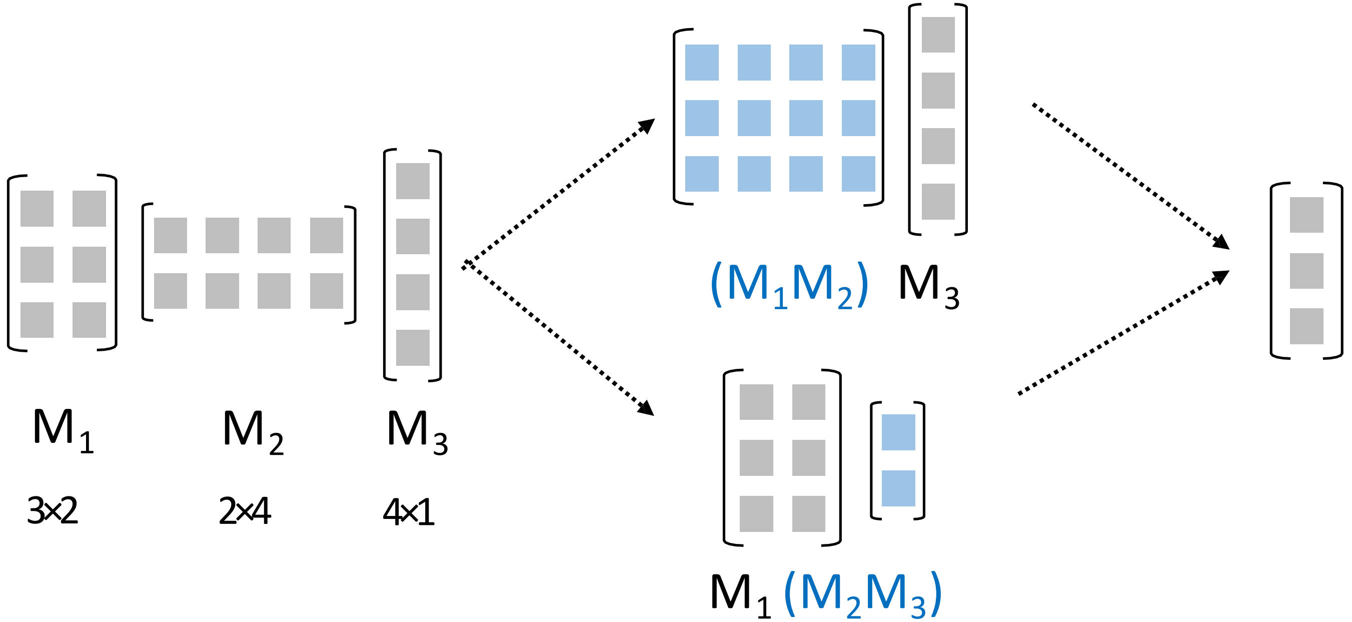
C(i, j)

i \ j	1	2	3
1	0	24	14
2		0	8
3			0

P(i, j)

	1	2	3
1		1	1
2			2
3			

연쇄 행렬 곱셈 문제의 적용 예_1



P(i, j)

	1	2	3
1		1	1
2			2
3			

$M_1 \mid M_2 M_3$

연쇄 행렬 곱셈 문제의 적용 예_2

M_1	×	M_2	×	M_3	×	M_4	×	M_5	×	M_6
5×2		2×3		3×4		4×6		6×7		7×8

$d_0=5,\ d_1=2,\ d_2=3,\ d_3=4,\ d_4=6,\ d_5=7,\ d_6=8$

C(i, j)

i \ j	1	2	3	4	5	6
1						C(1,6)
2						
3						
4						
5						
6						

연쇄 행렬 곱셈 문제의 적용 예_2

s=j-i=0

$C(i, i) = 0, \quad 1 \le i \le 6$

C(i, j)

i \ j	1	2	3	4	5	6
1	0					
2		0				
3			0			
4				0		
5					0	
6						0

연쇄 행렬 곱셈 문제의 적용 예_2

s=j-i=1

$C(1, 2) = \min_{1 \le k \le 1}\{ C(1,k)+C(k+1,2)+d_0 d_k d_2 \}$

$= C(1,1)+C(2,2)+5\times2\times3 = 0 + 0 + 30 = 30$

$C(2, 3) = \min_{2 \le k \le 2}\{ C(2,k)+C(k+1,3)+d_1 d_k d_3 \}$

$= C(2,2)+C(3,3)+2\times3\times4 = 0 + 0 + 24 = 24$

$C(3, 4) = 0 + 0 + 3\times4\times6 = 0 + 0 + 72 = 72$

$C(4, 5) = 0 + 0 + 4\times6\times7 = 0 + 0 + 168 = 168$

$C(5, 6) = 0 + 0 + 6\times7\times8 = 0 + 0 + 336 = 336$

연쇄 행렬 곱셈 문제의 적용 예_2

s=j-i=1

C(i, j)

i \ j	1	2	3	4	5	6
1	0	30				
2		0	24			
3			0	72		
4				0	168	
5					0	336
6						0

연쇄 행렬 곱셈 문제의 적용 예_2

s=j-i=2

$C(1, 3) = \min_{1 \le k \le 2}\{ C(1,k)+C(k+1,3)+d_0 d_k d_3 \}$

$= \min\{ C(1,1)+C(2,3)+ d_0 d_1 d_3, C(1,2)+C(3,3)+ d_0 d_2 d_3 \}$

$= \min\{ 0+24+5\times2\times4, 30+0+5\times3\times4 \} = 64$

$C(2, 4) = \min_{2 \le k \le 3}\{ C(2,k)+C(k+1,4)+d_1 d_k d_4 \}$

$= \min\{ C(2,2)+C(3,4)+ d_1 d_2 d_4, C(2,3)+C(4,4)+ d_1 d_3 d_4 \}$

$= \min\{ 0+72+2\times3\times6, \underline{24+0+2\times4\times6} \} = 72$

$C(3, 5) = \min\{ 0+168+3\times4\times7, \underline{72+0+3\times6\times7} \} = 198$

$C(4, 6) = 392$

연쇄 행렬 곱셈 문제의 적용 예_2

s=j-i=2

C(i, j) i \ j	1	2	3	4	5	6
1	0	30	64			
2		0	24	72		
3			0	72	198	
4				0	168	392
5					0	336
6						0

연쇄 행렬 곱셈 문제의 적용 예_2

s=j-i=3

$C(1, 4) = \min_{1\le k\le 3}\{ C(1,k)+C(k+1,4)+d_0d_kd_4 \}$

$= \min\{ C(1,1)+C(2,4)+ d_0d_1d_4,$

$C(1,2)+C(3,4)+ d_0d_2d_4$

$C(1,3)+C(4,4)+ d_0d_3d_4\}$

$= \min\{ 0+72+5\times2\times6, 30+72+5\times3\times6, 64+0+5\times4\times6 \}$

$= 132$

$C(2, 5) = \min\{ 0+198+2x3x7,\quad 24+168+2x4x7,$

$\underline{72+0+2\times6\times7} \} = 156$

$C(3, 6) = \min\{ 0+392+3x4x8,\quad 72+336+3x6x8,$

$\underline{3198+0+3\times7\times8} \} = 366$

연쇄 행렬 곱셈 문제의 적용 예_2

s=j-i=3

C(i, j)

i \ j	1	2	3	4	5	6
1	0	30	64	132		
2		0	24	72	156	
3			0	72	198	366
4				0	168	392
5					0	336
6						0

연쇄 행렬 곱셈 문제의 적용 예_2

s=j-i=4

$C(1, 5) = \min_{1\le k\le 4}\{ C(1,k)+C(k+1,5)+d_0 d_k d_5 \}$

$= \min\{ C(1,1)+C(2,5)+ d_0 d_1 d_5,\ C(1,2)+C(3,5)+ d_0 d_2 d_5,$

$C(1,3)+C(4,5)+ d_0 d_3 d_5,\ C(1,4)+C(5,5)+ d_0 d_4 d_5 \}$

$= \min\{ 0+156+5\times2\times7,\ 30+198+5\times3\times7,$

$64+168+5\times4\times7,\ 132+0+5\times6\times7 \}$

$= 226$

$C(2, 6) = 268$

연쇄 행렬 곱셈 문제의 적용 예_2

s=j-i=4

C(i, j)

i \ j	1	2	3	4	5	6
1	0	30	64	132	226	
2		0	24	72	156	268
3			0	72	198	366
4				0	168	392
5					0	336
6						0

연쇄 행렬 곱셈 문제의 적용 예_2

s=j-i=5

$$C(1,6) = \min_{1 \le k \le 5}\{ C(1,k)+C(k+1,6)+d_0 d_k d_6 \}$$

$$= \min\{ C(1,1)+C(2,6)+ d_0 d_1 d_6,\ C(1,2)+C(3,6)+ d_0 d_2 d_6,$$

$$C(1,3)+C(4,6)+ d_0 d_3 d_6,\ C(1,4)+C(5,6)+ d_0 d_4 d_6,$$

$$C(1,5)+C(6,6)+ d_0 d_5 d_6 \}$$

$$= \min\{ 0+268+5\times2\times8,\ 30+366+5\times3\times8,$$

$$64+392+5\times4\times8,\ 132+336+5\times6\times8,$$

$$226+0+5\times7\times8 \}$$

$$= 348$$

연쇄 행렬 곱셈 문제의 적용 예_2

s=j-i=5

C(i, j)

i \ j	1	2	3	4	5	6
1	0	30	64	132	226	348
2		0	24	72	156	268
3			0	72	198	366
4				0	168	392
5					0	336
6						0

연쇄 행렬 곱셈 문제의 적용 예_2

- 최적의 곱셈 순서
  - P[i][j]=k → "i부터 j까지 행렬을 곱할 때 최적의 순서로 갈라지는 지점이 k이다"

P(i, j)

i \ j	1	2	3	4	5	6
1		1	1	1	1	1
2			2	3	4	5
3				3	4	5
4					4	5
5						5
6						

P[1][6] = 1 →1을 기점으로 최적의 순서로 갈림

$M_1(M_2M_3M_4M_5M_6)$

P[2][6] = 5 → $M_1((M_2M_3M_4M_5)M_6)$

P[2][5] = 4 → $M_1(((M_2M_3M_4)M_5)M_6)$

P[2][4] = 3 → $M_1((((M_2M_3)M_4)M_5)M_6)$

연쇄 행렬 곱셈 문제의 성능 분석

```
…
for (s=1; s<=n-1; s++)
    for (i=1; i<=n-s; i++)
        C[i][j] = min_{i≤k≤j-1}( …);
…
```

$$O\left(\sum_{s=1}^{n-1}\left[\boxed{(n-s)}\times\boxed{s}\right]\right)=O\left(\frac{n(n-1)(n+1)}{6}\right)=O(n^3)$$

i 루프

k 루프:
(j-1)-i+1 →j=i+s이므로 (i+s-1)-i+1 = s번

과제 안내

연쇄 행렬 곱셈 문제

- M1 : 30×5 M2 : 5×20 M3 : 20×15 M4 : 15×10

$M_1(M_2(M_3M_4))$	$20\times15\times10 + 5\times20\times10 + 30\times5\times10 = 5{,}500$
$(M_1M_2)(M_3M_4)$	$30\times5\times20 + 20\times15\times10 + 30\times20\times10 = 12{,}000$
$M_1((M_2M_3)M_4)$	$5\times20\times15 + 5\times15\times10 + 30\times5\times10 = 3{,}750$
$((M_1M_2)M_3)M_4$	$30\times5\times20 + 30\times20\times15 + 30\times15\times10 = 16{,}500$
$(M_1(M_2M_3))M_4$	$5\times20\times15 + 30\times5\times15 + 30\times15\times10 = 6{,}900$

Microsoft Visual Studio 디버그 콘솔

```
Minimum number of multiplications is 3750
```

5주차 과제

- 연쇄 행렬 곱셈 문제 구현하기
  - e-Class 업로드
- 양식 (한글, 워드,PDF -> 자유)
- 파일명 (이름_학번_전공)
  - 예) 최희석_2014182009_게임공학

- 질의 응답은 e-Class 질의응답 게시판에 남겨 주시길 바랍니다.