

# 행렬 곱셈 순서

<https://www.acmicpc.net/problem/11049>

- $dp[i][j]$  =  $i$ 번째 행렬 부터  $j$ 번째 행렬 까지 곱했을 때, 곱셈 연산의 최소값
- 곱할 두 행렬의 범위는  $i \sim k$ ,  $k+1 \sim j$ 로 나눌 수 있다. 즉, 적절한  $k$ 를 정해야 한다.



- $dp[i][k] + dp[k+1][j] + \text{“행렬 곱셈에서 필요한 연산 횟수”}$  를 이용하여 점화식을 구성한다.
- $A[i] = i$ 번째 행렬의 크기( $A[i][0] \times A[i][1]$ )
- $dp[i][j] = \min(dp[i][k] + dp[k+1][j] + A[i][0] * A[k][1] * A[j][1])$
- $i=j$ 가 될 때 까지 탐색한다.

ex)  $(A1) * (A2 \ A3 \ A4 \ A5)$   
       $(A1 \ A2) * (A3 \ A4 \ A5)$   
       $(A1 \ A2 \ A3) * (A4 \ A5)$   
       $(A1 \ A2 \ A3 \ A4) * (A5)$

# 행렬 곱셈 순서

<https://www.acmicpc.net/problem/11049>

```
//dp[i][j] = i번째 행부터 j번째 행 까지 곱했을 때, 곱셈 연산의 최소값
int solve(int i, int j) {

    if (i == j) return 0;
    if (i + 1 == j) return A[i][0] * A[i][1] * A[j][1];

    if (dp[i][j] != -1) return dp[i][j];

    dp[i][j] = inf;
    for (int k = i; k < j; ++k) {
        dp[i][j] = min(dp[i][j], solve(i, k) + solve(k + 1, j) + A[i][0] * A[k][1] * A[j][1]);
    }
    return dp[i][j];
}
```