2조

에꼴캡스톤디자인 알고리즘

2022764034	2020675008	2019675028	2022964004	2022764048
이이슬	김규민	서보형	김다빈	최예진

11049

행렬 곱셈 순서

김다빈

11049

행렬 곱셈 문제

문제

크기가 N×M인 행렬 A와 M×K인 B를 곱할 때 필요한 곱셈 연산의 수는 총 N×M×K번이다.
행렬 N개를 곱하는데 필요한 곱셈 연산의 수는 행렬을 곱하는 순서에 따라 달라지게 된다.

예를 들어, A의 크기가 5×3이고, B의 크기가 3×2, C의 크기가 2×6인 경우에 행렬의 곱 ABC를 구하는 경우를 생각해보자.

- AB를 먼저 곱하고 C를 곱하는 경우 (AB)C에 필요한 곱셈 연산의 수는 5×3×2 + 5×2×6 = 30 + 60 = 90번이다.
- BC를 먼저 곱하고 A를 곱하는 경우 A(BC)에 필요한 곱셈 연산의 수는 3×2×6 + 5×3×6 = 36 + 90 = 126번이다.

같은 곱셈이지만, 곱셈을 하는 순서에 따라서 곱셈 연산의 수가 달라진다.
행렬 N개의 크기가 주어졌을 때, 모든 행렬을 곱하는데 필요한 곱셈 연산 횟수의 최솟값을 구하는 프로그램을 작성하시오.
입력으로 주어진 행렬의 순서를 바꾸면 안 된다.

11049

행렬 곱셈 문제

입/출력 조건

첫째 줄에 행렬의 개수 N(1 ≤ N ≤ 500)이 주어진다.
둘째 줄부터 N개 줄에는 행렬의 크기 r과 c가 주어진다. (1 ≤ r, c ≤ 500)
항상 순서대로 곱셈을 할 수 있는 크기만 입력으로 주어진다.

첫째 줄에 입력으로 주어진 행렬을 곱하는데 필요한 곱셈 연산의 최솟값을 출력한다.
정답은 $2^{31}-1$ 보다 작거나 같은 자연수이다. 또한, 최악의 순서로 연산해도 연산 횟수가 $2^{31}-1$보다 작거나 같다.

예제

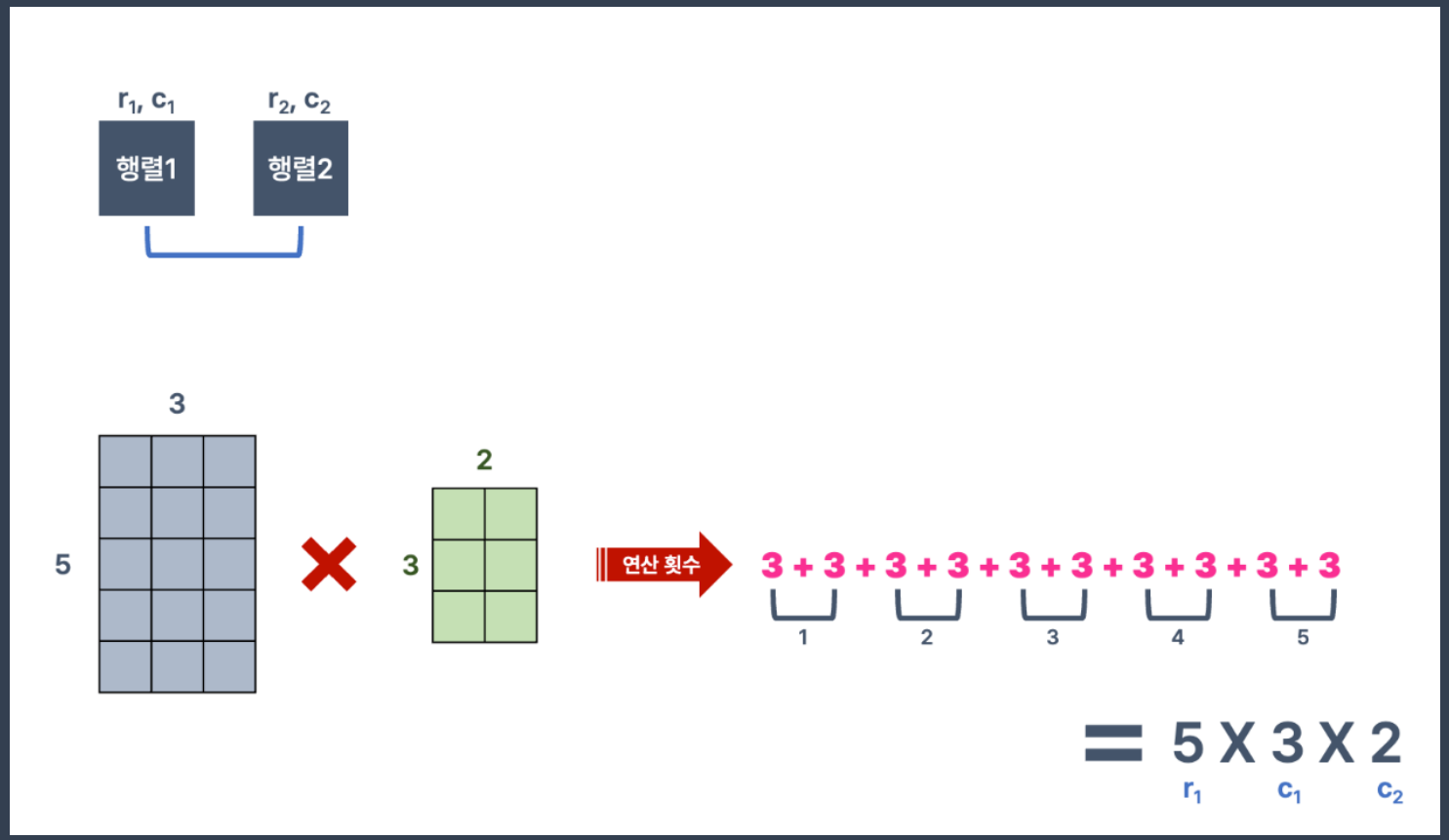
```
3
5 3
3 2
2 6
```

```
90
```

11049

행렬 곱셈 문제

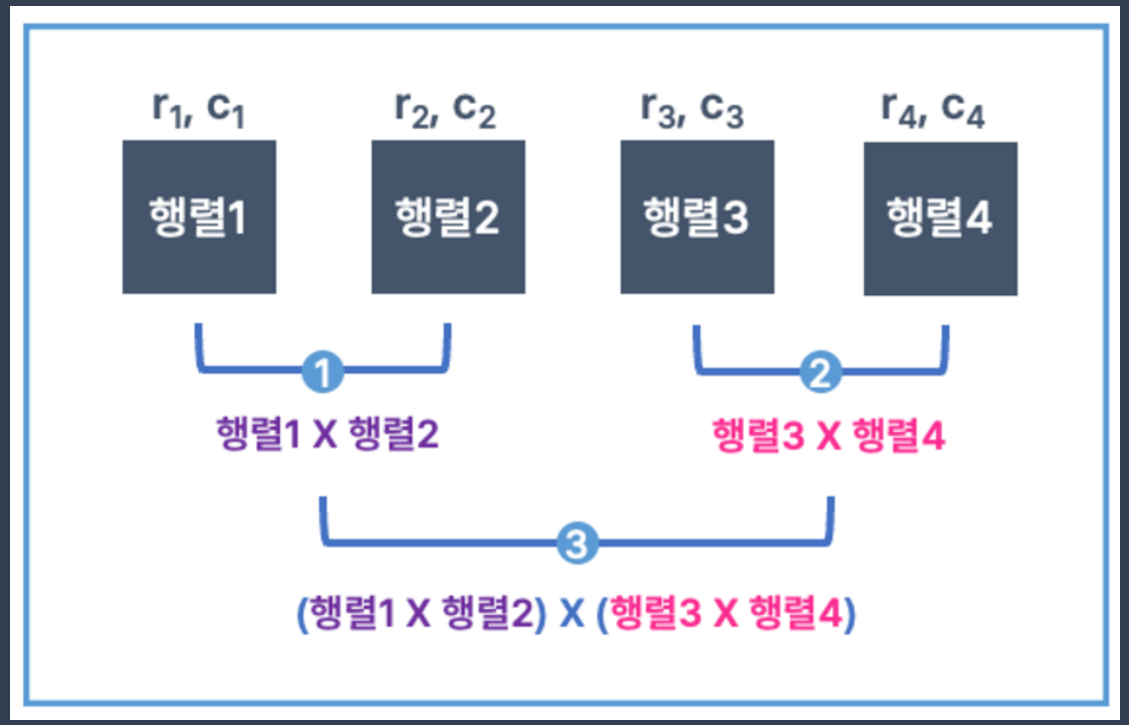
2개 행렬을 곱할 경우



11049

행렬 곱셈 문제

3개 이상의 행렬을 곱할 경우



11049

행렬 곱셈 문제

결과 코드

```python
import sys

# 데이터 입력
n = int(input())
arr = [list(map(int, sys.stdin.readline().split())) for _ in range(n)]

# dp 리스트 선언
dp = [[0]*(n) for _ in range(n)]

for term in range(1, n):
    for i in range(n):  # 첫행렬 : i, 끝행렬: i+term
        if i + term == n:  # 범위를 벗어나면 무시
            break

        dp[i][i+term] = int(1e9)  # 지금 계산할 첫행렬과 끝행렬

        for t in range(i, i+term): # 만들어질 수 있는 각 경우에 대해 모두 비교 연산
            # 연산 횟수를 계산해 더 작은 연산 횟수를 저장
            dp[i][i+term] = min(dp[i][i+term],
            dp[i][t]+dp[t+1][i+term] + arr[i][0] * arr[t][1] * arr[i+term][1])

print(dp[0][n-1])
```