

점프

<https://www.acmicpc.net/problem/1890>

- $dp[i][j] = (1, 1) \rightarrow (i, j)$ 로 이동할 수 있는 경로의 개수
- $dp[1][1] = 1$
- $dp[i][j] += dp[i][k] \text{ (} k + a[i][k] == j, 0 \leq k < j \text{)}$
- $dp[i][j] += dp[k][j] \text{ (} k + a[k][j] == i, 0 \leq k < i \text{)}$

모든 좌표를 방문하는 데 $O(N^2)$ 이고 $dp[j][k]/dp[k][j]$ 탐색이 $O(N)$ 이므로 $O(N^3)$ 이다.안

```
dp[1][1] = 1;
for (int i = 1; i <= N; ++i) {
    for (int j = 1; j <= N; ++j) {

        for (int k = 1; k < j; ++k) {
            if (k + A[i][k] == j) {
                dp[i][j] += dp[i][k];
            }
        }

        for (int k = 1; k < i; ++k) {
            if (k + A[k][j] == i) {
                dp[i][j] += dp[k][j];
            }
        }
    }
}
```

점프

<https://www.acmicpc.net/problem/1890>

- $dp[i][j] = (1, 1) \rightarrow (i, j)$ 로 이동할 수 있는 경로의 개수
- $dp[1][1] = 1$
- $dp[i][j + A[i][j]] += dp[i][j]$
- $dp[i + A[i][j]][j] += dp[i][j]$
- $A[i][j] == 0$ 인 부분은 이동 불가능 한 경우로 무시 한다.
- 모든 좌표를 방문하는 데 $O(N^2)$ 이고 한 칸을 채우는 데 $O(1)$ 이므로 총 시간복잡도는 $O(N^2)$

```
dp[1][1] = 1;
for (int i = 1; i <= N; ++i) {
    for (int j = 1; j <= N; ++j) {
        if (A[i][j] == 0) continue;
        if (j + A[i][j] <= N)
            dp[i][j + A[i][j]] += dp[i][j];
        if (i + A[i][j] <= N)
            dp[i + A[i][j]][j] += dp[i][j];
    }
}
```