

# 동전1

<https://www.acmicpc.net/problem/2293>

- $n$ 가지 종류의 동전이 있을 때, 각각의 동전이 나타내는 가치가 다르다.  
동전들을 적당히 사용하여, 그 가치의 합이  $k$ 원이 되도록 할 경우 경우의 수를 구하는 문제

① 1, 2, 3 더하기(9095번) 문제 : 1+3 과 3+1 을 각각 다른 경우로 간주  
 $a_1 + a_2 + \dots + a_n = N$  일 때,  $a_n$  은 1, 2, 3 중 하나가 올 수 있으므로  
 $dp[i] = dp[i-1] + dp[i-2] + dp[i-3]$

② 동전1 문제 : 1+3과 3+1을 같은 경우로 간주  
동전1 문제에서 1원, 2원 3원 동전이 있을 때  $N$ 을 표현한다면  
 $N = 1 * x + 2 * y + 3 * z$  에서  $(x, y, z)$  쌍의 개수 문제로 표현할 수 있다.

# 동전1

<https://www.acmicpc.net/problem/2293>

- $dp[i][j]$  = A[1] ~ A[i] 까지 동전을 사용하여 j원을 만드는 경우의 수
- j원을 만드는 경우
  - ① A[i]를 사용하는 경우  
A[i]를 사용하면 A[1] ~ A[i-1]까지 동전을 사용하여  $j - A[i]$ 원을 만들어야 한다.
    - $dp[i-1][j - A[i]]$
  - ② A[i]를 사용하지 않는 경우  
아직 사용하지 않았기 때문에 A[1] ~ A[i-1]까지 동전을 사용하여 j원을 만들어야 한다.
    - $dp[i-1][j]$
- $dp[i][j] = dp[i-1][j-A[i]] + dp[i-1][j]$

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

# 동전1

<https://www.acmicpc.net/problem/2293>

- $dp[i]$  = 동전을 사용해서  $i$ 원을 만드는 방법
- 사용할 수 있는 동전  $N$ 가지 ( $A[1], A[2], \dots, A[N]$ )
- 1차원 dp :  $dp[i] += dp[i - A[j]]$  ( $1 \leq j \leq N$ )
- 2차원 dp :  $dp[i][j] = dp[i-1][j-A[i]] + dp[i-1][j]$   
☞ 2차원 dp는 1차원 dp로 축소한 것과 같다.

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

printf("%d\n", dp[K]);
```