

2293번: 동전 1

2293번 제출 맞은 사람 슛코딩 풀이 풀이 작성 풀이 요청 재채점/수정

문제 추천 채점 현황 내 소스 강의▼ 질문 검색 질문 작성

동전 1 성공 풀이

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	4 MB	11962	4837	3545	39.796%

문제

n가지 종류의 동전이 있다. 각각의 동전이 나타내는 가치는 다르다. 이 동전들을 적당히 사용해서, 그 가치의 합이 k원이 되도록 하고 싶다. 그 경우의 수를 구하시오. (각각의 동전은 몇 개라도 사용할 수 있다.)

입력

문제 유형 :dynamic programming

문제는 기본 문제라고 하지만 , 점화식을 떠올리기가 무척 어려웠다. 처음에 문제를 봤을 때는, n원인 경우를 생각하는 문제에서 n원에서 c원의 동전을 사용하면 그대로 재귀 적으로 n-c원의 문제가 되는 줄 알았다. 하지만 생각해보니, 앞에서 c원을 뺀으나, 뒤에서 c원을 뺀으나 같은 경우의수 문제였다.

이 문제의 핵심 아이디어는 이렇다. c원의 자리 동전이 사용되기 이전에 모든 경우의 수가 계산되어있다고 가정한다.<리커시브적인 생각의 핵심!!!! 무엇인가 계산되어 있다고 가정... 우리는 그 무엇인가를 정의함으로써 재귀 식을 생각해 낼 수 있다.> 그 다음 c원짜리 동전을 사용한다.

그렇다면 c원이 사용 되기 이전 target원을 만드는 경우의수 + c원을 사용해 target원을 만드는 경우의 수로 구현 할 수있다.

이렇게 하면 2점화식 $d[n][k]$ (n번째 동전으로 k원 만들기) = $d[n-1][k-c(n)]$ (n-1 번째 까지의 동전을 사용해서 k-c(n-1)원 만들기) + $d[n-1][k]$ (c 동전 없이 k원 만들기) 의 식을 구할 수있다. c원 짜리 동전이 너무

개각 k원이 1개의 동전(1) + 1개의 동전(2) + 1개의 동전(5)로 만들 수 없는 경우 그 이전 동전 까지의 경우의 수를 사용 하면 된다.

이 점화식을 사용 한 후 2차원 배열을 사용해 프로그램을 작성하면 메모리 초과가 난다. 문제는 2차원이 아닌 1차원을 사용하기 원한다. 문제의 점화식이 조금 복잡하지만, 사용 했던 메모리가 일정한 범위 이후에는 메모이제이션이 필요 없다. 그러므로 그 부분에 계속 업데이트 해주면 된다.

source code

```
numOfCoin , target = map(int,input().split())
coins = [int(input()) for i in range (numOfCoin)]
d = [ 0 for i in range (10001)]
d[0] =1

for i in range (0, numOfCoin):
    for j in range (1,target+1):
        if j >= coins[i] :
            d[j] += d[j-coins[i]]
    #print(d[0:11])
print(d[target])
```

첫째줄에 n, k가 주어진다. ($1 \leq n \leq 100$, $1 \leq k \leq 10,000$) 다음 n개의 줄에는 각각의 동전의 가치가 주어진다.

출력

첫째 줄에 경우의 수를 출력한다. 경우의 수는 2^{31} 보다 작다.

예제 입력

```
3 10
1
2
5
```

예제 출력

10

힌트

알고리즘 분류

- 다이나믹 프로그래밍
- 동전 교환