DP 연습 **Nov. 15, 2022,** 신찬수

1. 왼쪽 맞춤

a. 입력: 폭 `W` 값과 문자열 (`n`개의 단어로 구성된다고 가정)
   i. 문자열을 공백을 기준으로 나눠 `words` 리스트에 저장 (`words[0], …, words[n-1]`)

b. 한 줄에 오는 단어와 단어 사이의 하나의 공백을 포함해 총 길이가 폭 `W`를 넘지 않아야 한다. 각 줄의 penalty 값은 오른쪽에 남은 공백 개수의 세 제곱으로 정의되고, 하나의 왼쪽 맞춤의 penalty 값은 각 줄의 penalty의 합으로 정의한다. (실제 문서 편집 프로그램에서 사용하는 penalty 값)

c. 출력: 왼쪽 맞춤의 최소 penalty 값

d. 해의 마지막 줄에 초점을 맞춘다. 마지막 줄에 오는 단어는 마지막 몇 개의 단어가 된다. 이 단어들과 인접한 단어 사이의 공백을 하나씩 포함해서 총 길이 W를 넘지만 않으면 된다. 몇 개의 단어가 올지 모르니 가능한 단어의 개수를 모두 고려하면 된다

e. `DP[i]` = `words[0], …, words[i]`를 폭 `W`에 왼쪽 맞춤할 때 최소 penalty 값
f. `penalty(a, b)`는 `words[a], …, words[b]`가 한 줄에 올 때 penalty 값

g. 마지막 줄에 `words[j], …, words[i]`가 온다고 하면, `DP[i] = DP[j-1] + penalty(j, i)`로 정의된다

h. `words[j]`를 모르기 때문에, 한 줄에 올 수 있는 모든 `j`에 대해 모두 계산한 후, 최소 값을 `DP[i]`에 저장하면 된다. 따라서, 점화식은 아래와 같다

$$DP[i] = \min_j (DP[j-1] + penalty(j, i))$$

i. `DP[i]` 값은 `O(n)` 시간이면 계산할 수 있으므로 $O(n^2)$ 시간에 DP 테이블을 채울 수 있다

2. 자리 값의 합

a. 입력: 자리수를 나타내는 `L`, 합을 나타내는 `S`
b. 출력: 자리수가 정확히 `L`인 자연수 중에서 합이 `S`가되는 자연수의 개수
   i. 주의: `L = 3, S = 4`인 경우, `1111, 1201, 2110` 등이 가능하지만, `0121`처럼 실제로 `4`자리가 아닌 자연수는 해당되지 않는다
c. 다시, 가장 오른쪽 자리에 집중하자. 가장 오른쪽 자리에 올 수 있는 값은 총 10가지 (`0, 1, …, 9`)이다. 예를 들어, 마지막 자리에 `3`이 오는 경우라면, 나머지 `L-1` 개의 자리의 합이 `S-3`이 되어야 한다. 결국, 마지막 자리에 수를 모두 넣어보고 경우의 수를 더하면 된다

d. `DP[i][s] = i`개의 자리를 갖는 자연수 중에서 자리 값의 합이 `s`가 되는 자연수 개수로 정의하면 아래 점화식이 성립한다

e. `DP[i][s] = sum (D[i-1][s-k] for k in range(0, min(9, s)+1))`
   i. 단, `DP[1][s] = 1 for s range(1, min(9, S)+1)`로 초기화한다 (`0`은 빠짐!)

f. DP 테이블의 칸의 개수는 `O(LS)`이고, 각 칸은 최대 `0, …, 9`까지만 고려하므로 상수 시간이므로 `O(LS)` 시간에 테이블을 모두 채울 수 있다. 이 시간은 입력의 크기 `L` 이외에 입력 값 `S` 자체가 포함되어 있기 때문에 pseudo polynomial 시간 알고리즘이다