

Dynamic Programming

Presented by Jihee Seon



Dynamic Programming

Overlapping Subproblem 문제 정의 후 작은 문제와의 관계 규명 = 점화식 세우기!
:: 겹치는 부분 문제 :: 문제의 정답을 작은 문제의 정답에서 구할 수 있을 때

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	2	3	5	8	13	21	34	55	89	144	233	377	610

Problem Analysis

1463 1로 만들기

정수 X에 사용할 수 있는 연산은 다음과 같이 3가지이다

1. X가 3으로 나누어 떨어지면, 3으로 나눈다.
2. X가 2로 나누어 떨어지면, 2로 나눈다.
3. 1을 뺀다.

정수 N이 주어졌을 때, 위와 같은 연산 3개를 적절히 사용해서 1을 만들려고 한다. 연산을 사용하는 횟수의 최솟값을 출력하시오.

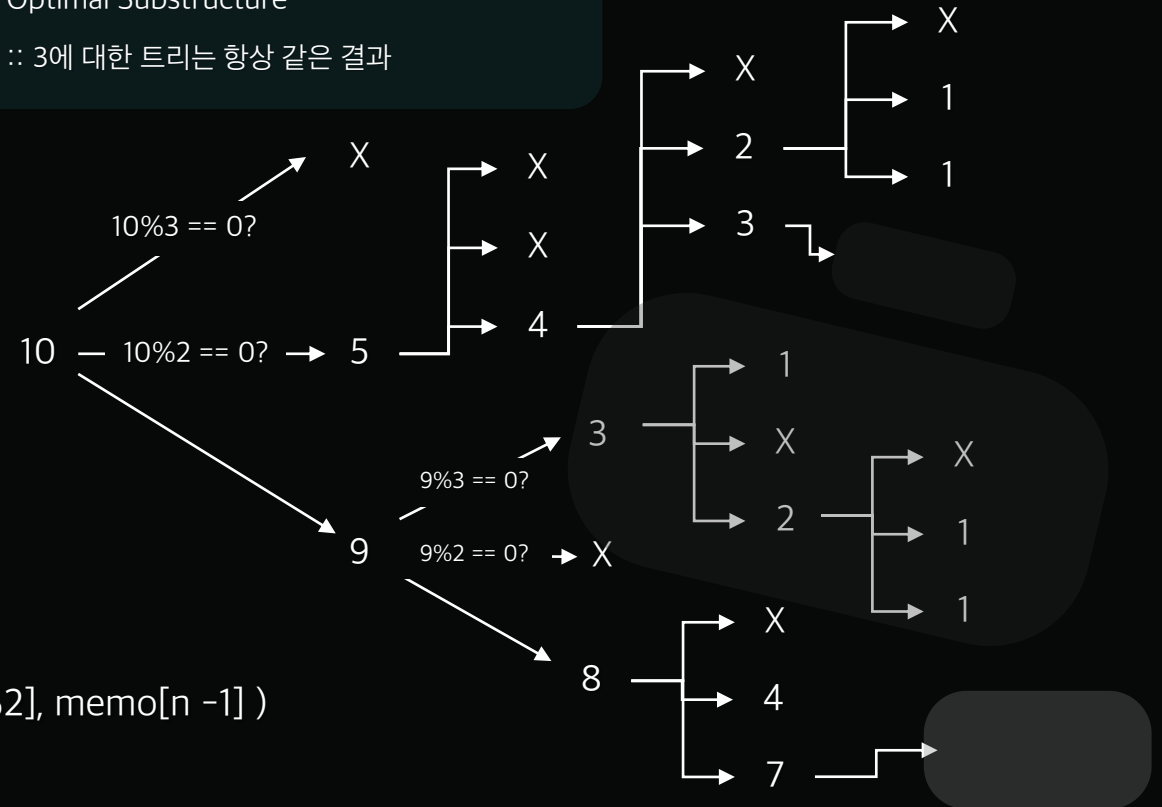
[초기값] $\text{memo}[1] = 0, \text{memo}[2] = 1, \text{memo}[3] = 1$

[점화식] $\text{memo}[n] = 1 + \text{Math.min}(\text{memo}[n\%3], \text{memo}[n\%2], \text{memo}[n-1])$

✓ Overlapping subproblem :: 5, 9 문제와 겹침

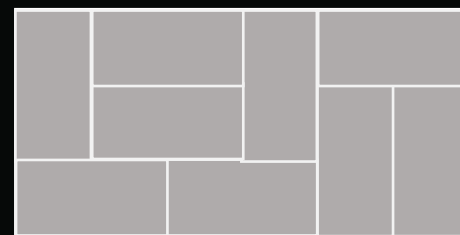
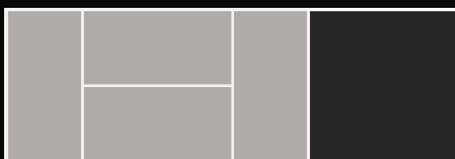
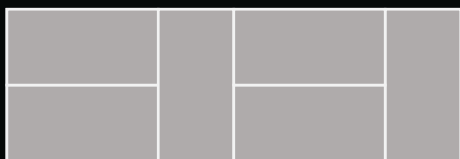
✓ Optimal Substructure

:: 3에 대한 트리는 항상 같은 결과



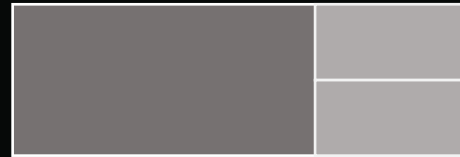
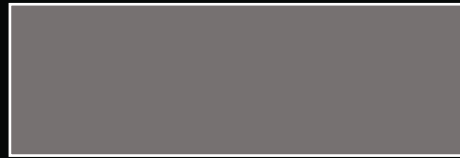
Problem Analysis

타일 채우기 series -- # 11726, # 11727, #2133



Problem Analysis

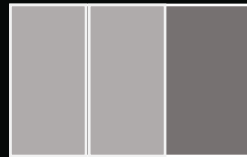
타일 채우기 series -- # 11726, # 11727, #2133



[memo[n]] 2 X N 타일을 채울 수 있는 모든 방법의 수

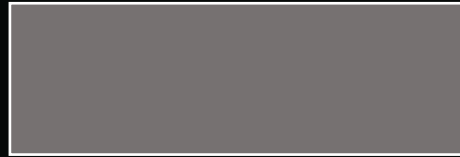
[초기값] Memo[1] = 1 memo[2] = 2

[점화식] memo[n] = memo[n-1] + memo[n-2]



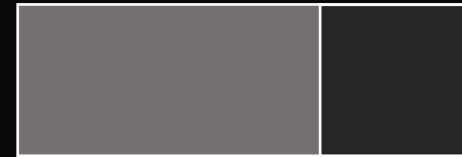
Problem Analysis

타일 채우기 series -- # 11726, # 11727, #2133



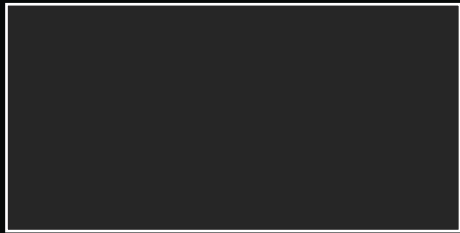
[초기값]

[점화식]



Problem Analysis

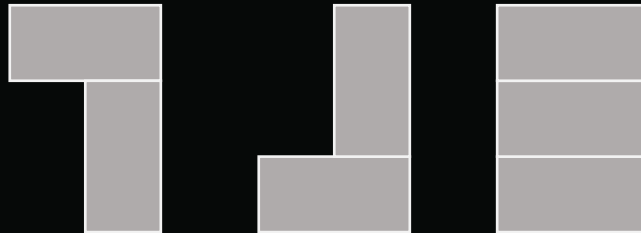
타일 채우기 series -- # 11726, # 11727, #2133



[memo[n]] 3 X N 타일을 채울 수 있는 총 경우의 수

[초기값] memo[0] = 1 memo[1] = 0 memo[2] = 3 memo[3] = 0 memo[4] = 11

[점화식] n 이 짝수일 때 :: memo[n] = 3 X memo[n-2] + 2 X memo[n-4]
n이 홀수일 때 :: memo[n] = 0 (n == 2k + 1), 6k + 3. % 2 == 1 . 따라서 0



Problem Analysis

9095 1,2,3 더하기

정수 n 이 주어졌을 때, n 을 1, 2, 3의 합으로 나타내는 방법의 수

[memo[n]] 1,2,3 의 합으로 N을 나타내는 방법의 수

[초기값] $\text{memo}[1] = 1, \text{memo}[2] = 2, \text{memo}[3] = 4$

[점화식] $\text{memo}[n] = \text{memo}[n-1] + \text{memo}[n-2] + \text{memo}[n-3]$

순서가 다른건 다르게 본다는데 포커스

Problem Analysis

11052 카드 구매하기

카드 팩의 가격이 주어졌을 때,

N개의 카드를 구매하기 위해 민규가 지불해야 하는 금액의 최댓값?

* 카드팩은 1, ..., N 개짜리 들어있는 팩 순으로 입력 받음

Problem Analysis

10844 쉬운 계단수

이 수는 인접한 모든 자리수의 차이가 1이 난다. 이런 수를 계단 수라고 한다.
세준이는 수의 길이가 N 인 계단 수가 총 몇 개?

Problem Analysis

11057 오르막수

오르막 수는 수의 자리가 오름차순을 이루는 수를 말한다.

이때, 인접한 수가 같아도 오름차순으로 치며, 0으로 시작 가능.

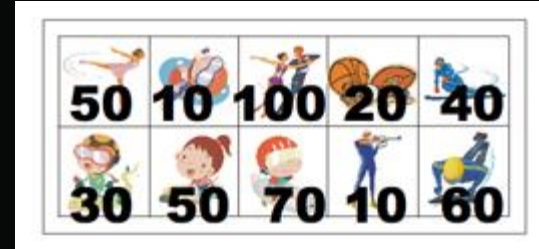
수의 길이 N 이 주어졌을 때, 오르막 수의 개수?

Problem Analysis

9465 스티커

각 스티커마다 점수가 있는데, 스티커를 떼어내면 그 스티커와 변을 공유하는 스티커는 모두 찢어져서 사용할 수 없게 된다. 즉, 떼는 스티커의 왼쪽, 오른쪽, 위, 아래에 있는 스티커는 사용할 수 없게 된다.

이 때 스티커에 대응되는 합이 최대가 될 때의 값?



50	10	100	20	40
30	50	70	10	60

위에만 뗄 때 :: 아래 혹은 둘 다

아래만 뗄 때 :: 위 혹은 둘 다

둘 다 안 뗄 때 :: 위 혹은 아래

1로 만들기 → 조건이 달라져서

Problem Analysis

1912 연속합

n개의 정수로 이루어진 임의의 수열이 주어질 때, 연속된 몇 개의 수를 선택해서 구할 수 있는 합 중 가장 큰 합?

Ex. 10, -4, 3, 1, 5, 6, -35, 12, 21, -1 => 33 (12+21)

지금 내가 연속합의 시작이 되는게?

또는 이전 연속합에 포함되는게?

들어가냐 안 들어가냐

Problem Analysis

1699 제곱수의 합

주어진 자연수 N 을 이렇게 제곱수들의 합으로 표현할 때에 그 항의 최소개수?

$$11 = 3^2 + 1^2 + 1^2$$

Problem Analysis

2156 포도주 시식

테이블 위에 다양한 포도주가 들어있는 포도주 잔이 일렬로 놓여 있었다.

효주는 포도주 시식을 하려고 하는데, 여기에는 다음과 같은 두 가지 규칙이 있다.

1. 포도주 잔을 선택하면 그 잔에 들어있는 포도주는 모두 마셔야 하고, 마신 후에는 원래 위치에 다시 놓아야 한다.
2. 연속으로 놓여 있는 3잔을 모두 마실 수는 없다.

총 포도주 양을 최대한로 마실 수 있을 때의 양?

[memo[n]]

N잔까지 마실 수 있는 포도주의 최대량

[초기값]

[점화식]

$memo[n] = memo[n-1] + memo[n-2] + memo[n-3]$

Problem Analysis

2579 계단 오르기

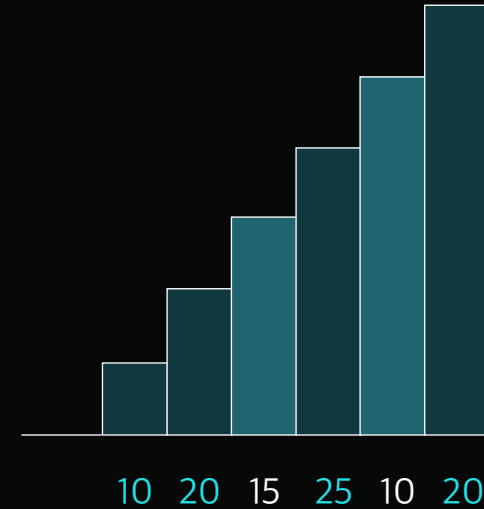
계단 오르는 데는 다음과 같은 규칙이 있다.

1. 계단은 한 번에 한 계단씩 또는 두 계단씩 오를 수 있다.
즉, 한 계단을 밟으면서 이어서 다음 계단이나, 다음 다음 계단으로 오를 수 있다.
2. 연속된 세 개의 계단을 모두 밟아서는 안 된다. 단, 시작점은 계단에 포함되지 않는다.
3. 마지막 도착 계단은 반드시 밟아야 한다.

따라서 첫 번째 계단을 밟고 이어 두 번째 계단이나, 세 번째 계단으로 오를 수 있다.

하지만, 첫 번째 계단을 밟고 이어 네 번째 계단으로 올라가거나, 첫 번째, 두 번째, 세 번째 계단을 연속해서 모두 밟을 수는 없다.

각 계단에 쓰여 있는 점수가 주어질 때 이 게임에서 얻을 수 있는 총 점수의 최댓값?



Problem Analysis

가장 긴 증가하는 부분 수열

수열 A가 주어졌을 때, 가장 긴 증가하는 부분 수열 길이?

{10, 20, 10, 30, 20, 50} 인 경우 → {10, 20, 10, 30, 20, 50} 이고, 길이는 4이다.

Problem Analysis

2225 합분해

0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수?

1. 덧셈의 순서가 바뀐 경우는 다른 경우로 센다($1+2$ 와 $2+1$ 은 서로 다른 경우).
2. 한 개의 수를 여러 번 쓸 수도 있다.

Tips?

문제에 제시되지 않은 제약조건은 정말 확실한 경우 아니면 넣지 않을 것.

- 문제의 비약을 가져올 수 있음.
- DP는 한번 푼 문제의 값을 재사용해서 같은 문제를 반복해서 풀 필요가 없다는 것이지, 연산 횟수를 줄이려는 것이 목표가 아니다.

memo 배열이나 input은 웬만하면 long으로 잡아줄 것.

- 이전 값을 더해나가는 문제 등 축적하는 문제들은 대부분 long이 넘어간다.
- 나머지 연산을 해야 하는 경우 매 연산마다 나머지 연산 실행
- 각 stage마다 $(A+B) \% C$ 를 실행해야 할 경우 $(A+B) \% C = ((A \% C) + (B \% C)) \% C$

Top-down 보다는 Bottom-up

- StackTrace 사용량을 보장할 수 없으며, 시간 사용도 훨씬 많다.

Fin

도같은 디피 안녕~

Classifications

- 내가 n번째에 들어가는 경우와 안 들어가는 경우
- 합할 수 있는 모든 경우 = $1 + (n-1) = 2 + (n-2) = \dots = (n-1) + 1$

Problem type

상태

들어간다 안 들어간다

이중배열

memo 정의하기

인덱스 i, j, l, m

골드는 두번 써야해~ 이중배열 ㄱ~