

Fibonacci Number

$$F(0) = 0, F(1) = 1$$

$$F(n) = F(n-1) + F(n-2)$$

$$F(9) = F(8) + F(7)$$

$$\begin{array}{cc} \parallel & \parallel \\ F(7) + F(6) & F(6) + F(5) \\ \vdots & \vdots \end{array}$$

Memoization

누가 언제 계산할지 모르니 저장해두고 필요할 때 사용.



$F(1)$ $F(2)$ $F(3)$

← 값을 저장해두고 호출하기 전에 확인한다.

따라서 같은 입력으로 호출했을 때 같은 결과가 나오는 것이 보장될 때만 가능!

Dynamic Programming

계산되는 순서를 안다면, 특정 강도를 이전에 계산되었다고 하고, 사용할 수 있다.

Select Working Days

- 연속 근육 분가
- 가장 많이 볼 수 있는 스켈레톤?

각 날짜마다, [그날 일 하는 경우 그날까지 가장 많이 벌수 있는 스케줄로
그날 일 안하는 경우 모두 구한다.

6 3 2 7 10 4 9 2

$$\begin{array}{cccccc} 0 & 6 & 3 & 8 & 13 & 18 \\ x & 0 & 6 & 6 & 8 & 13 \end{array}$$
 x 에 대한 거 \rightarrow (그날 일을 하면 전날에 일을 두 배로 크게 안했어야 하기 때문.)
 \downarrow
 (그날 일을 안하면, 전날은 상관 없음)

Path Counting



• \rightarrow • 여기까지 올 수 있는 path 개수
" "

위쪽 정까지의 path 개수 + 왼쪽 정까지의 path 개수

