DP?

다이나믹 프로그래밍.

→ 중복되는 연산을 최소화.

→ 효율적인 풀각.

방식. ┌ 탑-다운. : 큰 문제를 해결하기 위해 작은 문제 호출. → f(5) → f(4) + f(3)
└ 보텀-업 : 작은 문제부터 차근차근 답을 도출. → f(1) = 1, f(2) = ..., ..., f(8) = ...) DP 테이블

메모이제이션

규칙이 여러서 배열된 형태 → 점화식

↳ 재귀함수. (but, 수행시간이 기하급수적)

→ DP로 단축!

[적용조건]

1. 큰 문제는 작은 문제로 나눌수 있다. (= 분할 정복)
2. 작은 문제에서 구한 답은 큰 문제에서도 동일하다. (문제들이 서로 영향)

ex) f(5) = f(4) + f(3)

* 메모이제이션 : 한번 구한 답을 메모리 공간에 저장.
(= 캐싱)

→ 다음번엔 연산 X. : 효율적.