DP

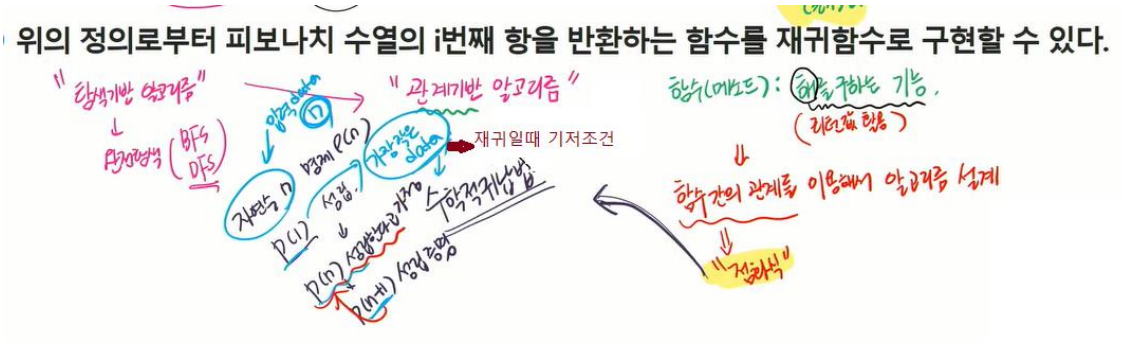
작은 부분 문제들의 해를 구하고 이들을 이용하여 보다 큰 크기의 부분 문제들을 해결하여 최종적으로 원래 주어진 문제를 해결하는 알고리즘 설계 기법

- DP 적용 요건
  - ■ 중복 부분 문제 구조(Overlapping Subproblems)

    큰 문제를 이루는 작은 문제들로 구성되어 있고 작은 문제들의 최적 해를 이용해서 순환적으로 큰 문제를 해결한다. => 점화식 + 메모이제이션(동적 테이블)
  - ■ 최적 부분 문제 구조(Optimal substructure)

    주어진 문제가 최적화 원칙을 만족해야한다.

    ⇨ 최적화 원칙 : 어떤 문제의 대한 해가 최적일 때 그 해를 구성하는 작은 문제들의 해 역시 최적이어야 한다.



DP 문제 풀기

1. DP로 풀수 있는 문제인지 파악

   중복 부분 문제 구조 + 최적 부분 문제 구조
   - ■ 보통 특정 데이터내 최대화/최소화 계산 ,
   - ■ 특정 조건 내에서 데이터를 수를 카운트하거나 확률을 계산하는 경우
2. 문제의 변수(state) 파악

   현재 변수에 따라 그 결과 값을 찾고 그것을 전달하여 재사용하는 것

   Ex)

   피보나치 수열 : n번째 항을 구하는 것 - state는 n

   K-napsack : index와 무게(가치)
3. 변수간 관계식 만들기 (점화식)

4. 메모하기

   변수의 개수에 따라 배열 차원이 1~3차원 등 다양할 수 있다.

5. 기저 상태 파악하기

   가장 작은 문제의 상태를 알아야 한다. ➔ 점화식에서 사용하는 데이터들의 초기 값 설정

   Ex) 피보나치 f(0) =0, f(1)=1 => f(2) = 1

   f(1) =1, f(2)=1 => f(3) = 2

6. 구현 방법

   - Buttom-up : for + Tabulation
   - Top-down : 재귀 + 메모(Memoization)