동적 계획법

동적 계획법이란?

- '다이나믹 프로그래밍'이라고도 부름
- 과거에 구한 값을 이용하여 현재 값을 구함
- 특정 범위까지의 값을 구하기 위해 이전 범위의 값을 활용하여 효율적으로 값을 얻는 기법
- 이전 범위의 값을 저장(Memoization)함으로써 시간적, 공간적 효율을 얻음
- 중복 연산을 피한다.

Memoization (저장)

- 이전에 구해둔 값을 저장해서 중복 계산을 방지
- 이전 범위의 답을 구하면 곧바로 배열에 저장
- 시간과 공간면에서 모두 효율적
- ex) 피보나치 수열값을 배열에 저장한다.

재귀함수 vs. 동적계획법

- 재귀함수 : 보통 $n \leq 20$ 까지만 가능하고 그 이상은 시간초과가 됨
- 동적계획법 : n의 범위가 클 때 활용하면 좋고, 재귀함수보다 훨씬 효율적인 풀이이다.

Top-down 방식 vs. Bottom-up 방식

- Top-down 방식 : n부터 시작해서 구하려하는 문제를 작은 문제로 호출하며 탐색. 재귀함수를 사용해서 시간이 오래 걸린다.
- Bottom-up 방식 : 0부터 시작해서 이미 알고있는 앞의 작은 문제부터 원하는 문제까지 탐색. 함수 사용을 하지 않기 때문에 속도가 빠르다.

동적 계획법을 언제 사용할까?

- 주어진 문제를 부분 문제로 나누었을 때, 부분문제의 답을 통해 전체문제의 답을 도출할 수 있을 때
- 부분문제의 답을 여러번 구해야할 때 >> 즉, 한번 계산한 값을 다시 사용해야 할 때

점화식

- 인접한 항들 사이의 관계식
- 동적계획법 문제를 풀 때, 점화식을 미리 세우고 풀면 좋다.
- 세우는 방법 : 이전 값들을 이용하여 DP(현재의 값)를 정의하면 된다.

  ex) 피보나치 수열 문제의 점화식 : DP[i] = DP[i-1] + DP[i-2]

- DP는 점화식을 세우면 거의 그대로 사용하기 때문에, 점화식만 잘 세운다면 코드구현 자체는 어렵지 않다.