

DP
CS 07.

김채리(23.01.16)

DP란?

- **Dynamic Programming (동적 계획법)**

큰 문제를 작은 문제로 나누어 푸는 문제

하나의 큰 문제를 여러 개의 작은 문제로 나누어서 그 결과를 저장하여 다시 큰 문제를 해결할 때 사용하는 것

큰 문제를 작은 문제로 쪼개서 그 답을 저장해두고 재활용

DP 필요성

- 일반적인 재귀를 단순히 사용 시 동일한 작은 문제들이 여러 번 반복되어
비효율적인 계산이 될 수 있음

피보나치 수열

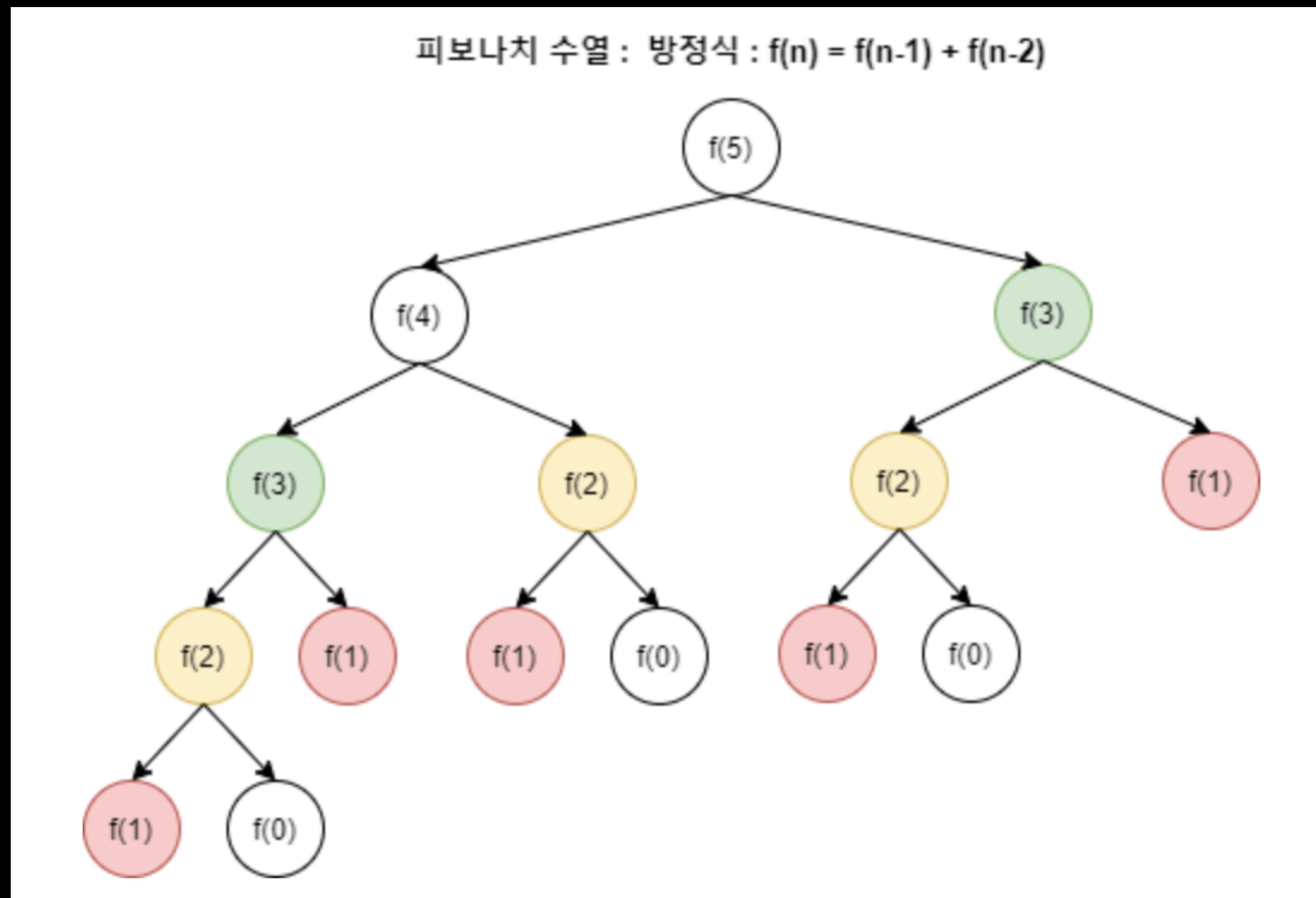
$$f(n) = f(n-1) + f(n-2)$$

DP 사용조건1

Overlapping Subproblems (겹치는 부분 문제)

- 동일한 작은 문제의 반복

반복되지 않는다면 재사용이 불가능해 DP 알고리즘 사용 불가



DP 사용조건2

Optimal Substructure (최적 부분 구조)

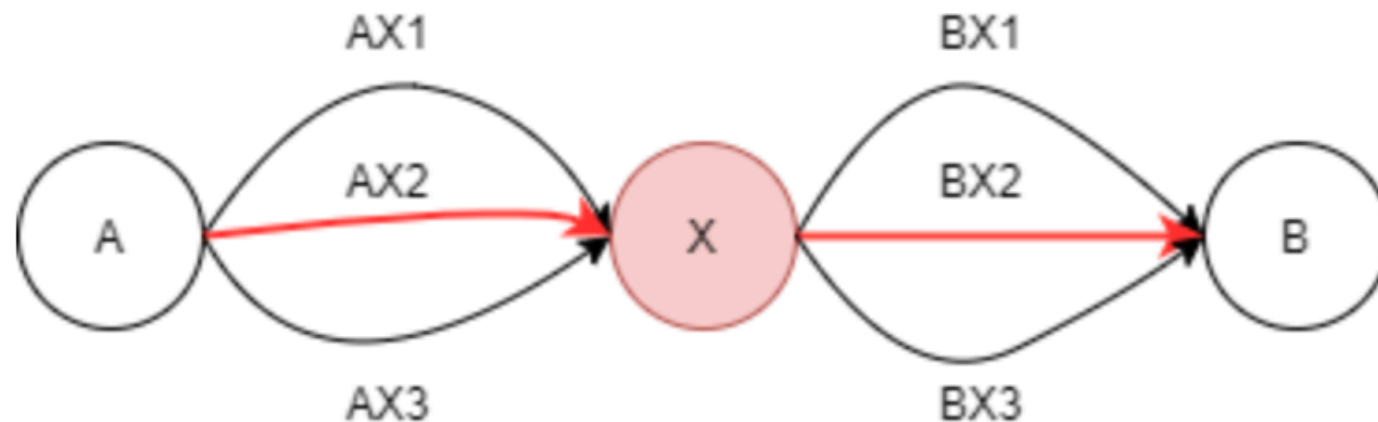
- 부분 문제의 최적 결과 값을 사용해 전체 문제의 최적 결과를 낼 수 있는 경우

그림에서 A - X 사이의 최단 거리는 AX2이고 X - B는 BX2이다. 전체 최단 경로는 AX2 - BX2이다.

다른 경로를 택한다고 해서 전체 최단 경로가 변할 수는 없다.

이와 같이, 부분 문제에서 구한 최적 결과가 전체 문제에서도 동일하게 적용되어 결과가 변하지 않을 때 DP를 사용할 수 있게 된다.

피보나치 수열도 동일하게 이전의 계산 값을 그대로 사용하여 전체 답을 구할 수 있어 최적 부분 구조를 갖고 있다.



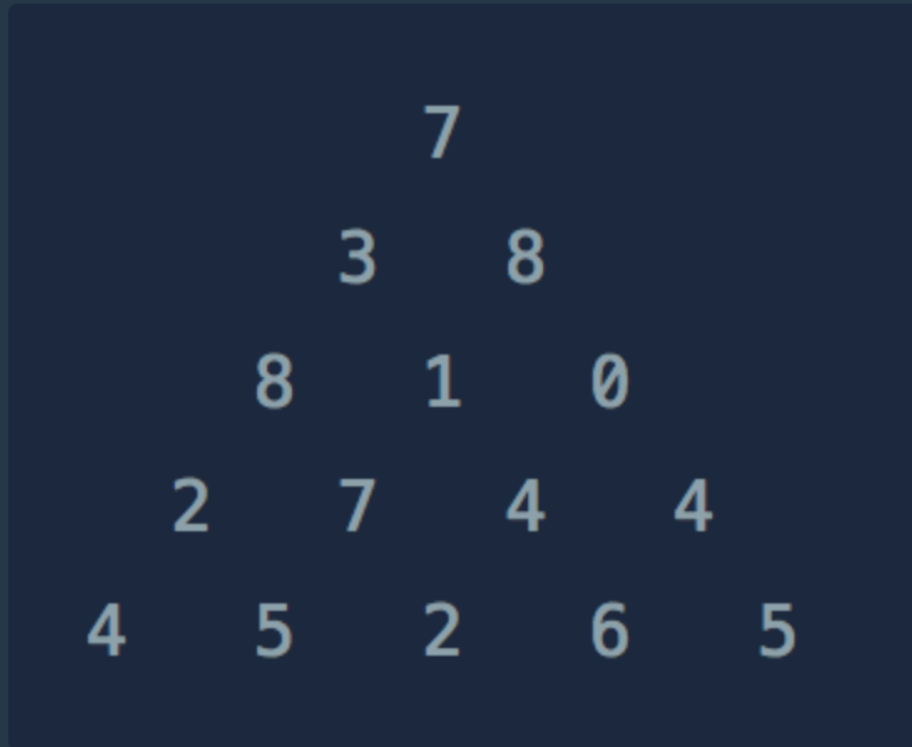
최단 경로 찾기

DP 실전사용

1. DP로 풀 수 있는 문제인지?
2. 문제의 변수 파악
3. 완전탐색 시간 초과 범위 확인
4. 변수간 관계식(점화식) 만들기

Quiz

문제 설명



위와 같은 삼각형의 꼭대기에서 바닥까지 이어지는 경로 중, 거쳐간 숫자의 합이 가장 큰 경우를 찾아보려고 합니다. 아래 칸으로 이동할 때는 대각선 방향으로 한 칸 오른쪽 또는 왼쪽으로만 이동 가능합니다. 예를 들어 3에서는 그 아래칸의 8 또는 1로만 이동이 가능합니다.

삼각형의 정보가 담긴 배열 `triangle`이 매개변수로 주어질 때, 거쳐간 숫자의 최댓값을 `return` 하도록 `solution` 함수를 완성하세요.

프로그래머스 동적계획법문제