파이썬 심화

동적 프로그래밍

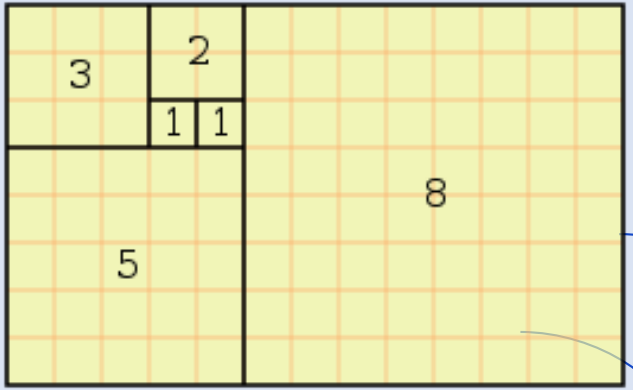
동적 프로그래밍

동적 프로그래밍(Dynamic Programming)이란?

- 분할정복과 마찬가지로 하나의 문제에 여러 개의 작은 문제가 포함되어 해당 문제를 해결하고 그 해를 병합하여 문제를 풀 수 있지만, 중복 수행이 많이 발생하는 경우에 이를 해결하는 알고리즘 기법
  - 작은 문제의 해가 항상 같은 경우
  - 추가적인 메모리 공간 필요

피보나치(Fibonacci) 수열

- 피보나치(L. Fibonacci)가 “산술의 서”라는 책에서 소개한 수열
  - F(0) = 0, F(1) = 1
  - F(n) = F(n-1) + F(n-2)



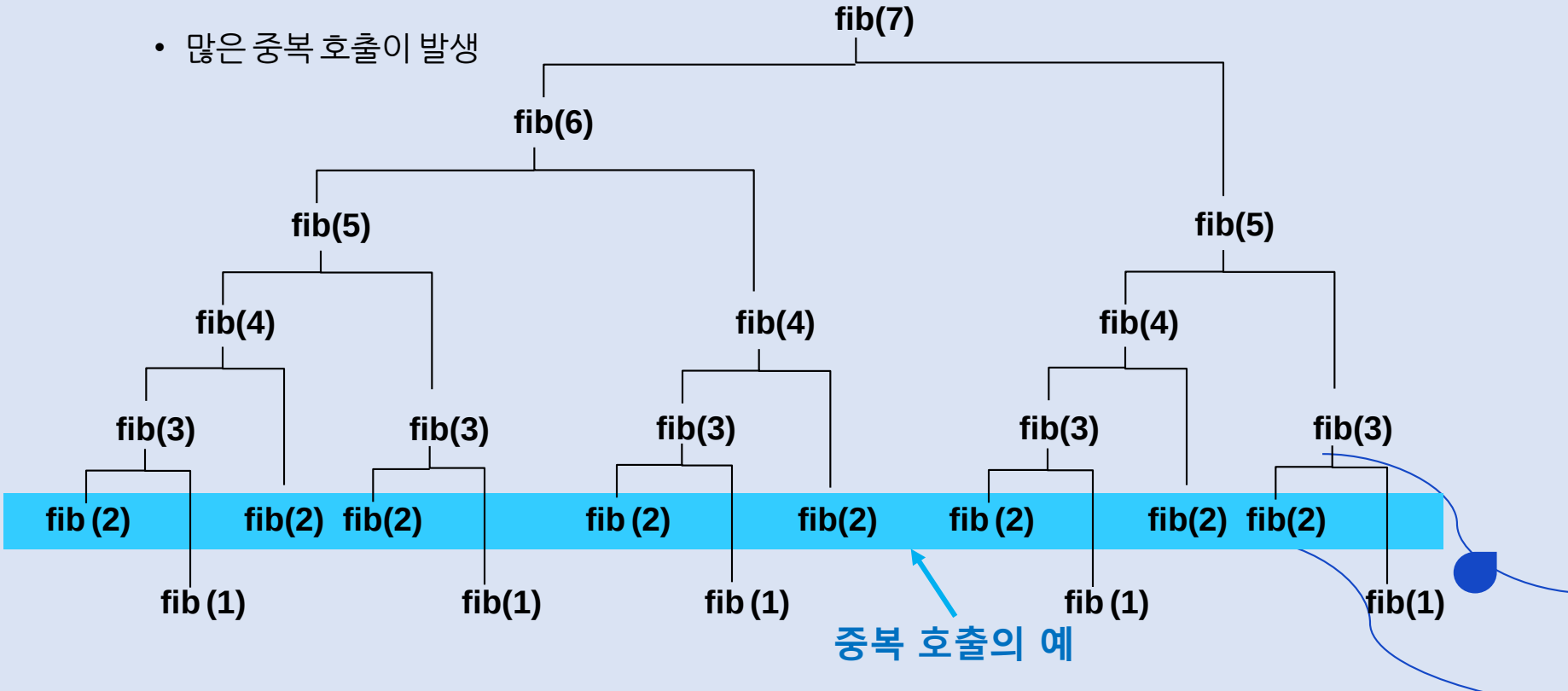
피보나치 수열의 재귀적 구현

- 피보나치 수열의 점화식을 이용한 구현 방법

```
def fib(n):
  if n == 0: return 0
  if n == 1: return 1
  return fib(n-1) + fib(n-2)
```

피보나치 수열의 재귀 프로그램 문제

- 많은 중복 호출이 발생



중복호출을 해결하는 방법

- 일반적으로 사용되는 방법들
  - 비재귀 프로그램(forward)을 사용
  - 메모리에 호출된 결과값을 저장하고 다음 호출때 해당 값을 사용
  - 수학적 접근 방법
  - 행렬식 사용

비재귀 프로그램

- 반복문을 사용한 해결 방법

```
def fib(n):
  f = [ 0, 1 ]
  for _ in range(2, n+1, 2):
    f[0] += f[1]
    f[1] += f[0]
  return f[n%2]
```

동적 프로그래밍을 이용한 재귀 프로그램

- 메모리 공간을 사용한 해결 방법

```
dp = [-1]*(n+1)      #  피보나치 수열을 저장할 공간 확보
dp[0], dp[1] = 0, 1
def fib(n):
  if dp[n] != -1: return dp[n]
  return dp[n] = fib(n-1) + fib(n-2)
```

수학적 접근방법

- 황금비율수를 이용한 접근 방법

```
Sqrt5 = 2.236068
Phi = 1.618034
def fib(n):
  return int((Phi**n - (1-Phi)**n)/Sqrt5 + 0.5)
```

행렬을 이용한 접근방법

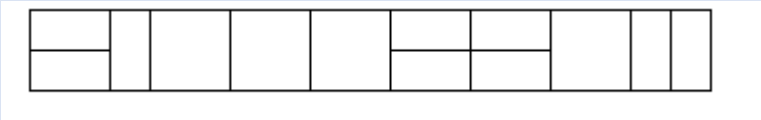
- 두개의 항목을 벡터로 묶어서 행렬 계산

```
def mul(a, b):
    m11, m12 = a[0]*b[0] + a[1]*b[2], a[0]*b[1] + a[1]*b[3]
    m21, m22 = a[2]*b[0] + a[3]*b[2], a[2]*b[1] + a[3]*b[3]
    return (m11, m12, m21, m22)
def fib(n):
    n -= 1
    a, b = (1, 1, 1, 0), (1, 0, 1, 0)
    while n > 0:
        if n%2 == 1: b = mul(a, b)
        a = mul(a, a)
        n //= 2
    return b[2]
```

실습

2xn 타일링

- 2xn 공간에 2x1 직사각형을 채우는 방법의 수를 구하세요
  - 2x17 공간의 예



입력 2 → 출력 3

입력 8 → 출력 171

가장 긴 공통 부분 수열(LCS) 구하기

- 두개의 문자열이 주어지면, 그 문자열의 부분 문자열 중 공통으로 존재하는 부분 문자열 중 가장 긴 문자열의 길이를 구합니다.
  - 예) ACAYKP, CAPCAK 두 문자열이 주어진 경우 ACAK는 두 문자열에 공통으로 있는 부분 문자열이고 이 문자열이 가장 길므로 4를 출력
  - 참조 : https://www.acmicpc.net/problem/9251

동전의 경우의 수 구하기

- N가지 서로 다른 종류의 동전이 주어졌을 때, 해당 동전을 무한히 쓸 수 있다고 가정하였을 때, K 금액을 만드는 방법의 경우의 수를 계산합니다.
  - 예) 동전 종류 : [1, 5, 10] 금액 : 10 → 결과 4
    - (10, 0, 0), (5, 1, 0), (0, 2, 0) (0, 0, 1)
  - 예) 동전 종류 : [1, 5, 10] 금액 : 100 → 결과 121