

알 고 리 즈 과 제

ToBig's 9기 이잉걸

Dynamic Programming

기본개념과 연습문제, 심화문제를 풀어보자

Contents

Unit 01 | 개념 설명

Unit 02 | 일반 문제

Unit 03 | 보너스 문제

Unit 01 | 개념 설명

‘동적 계획법’

큰 문제를 작은 문제로 나눠서 푸는 기법.

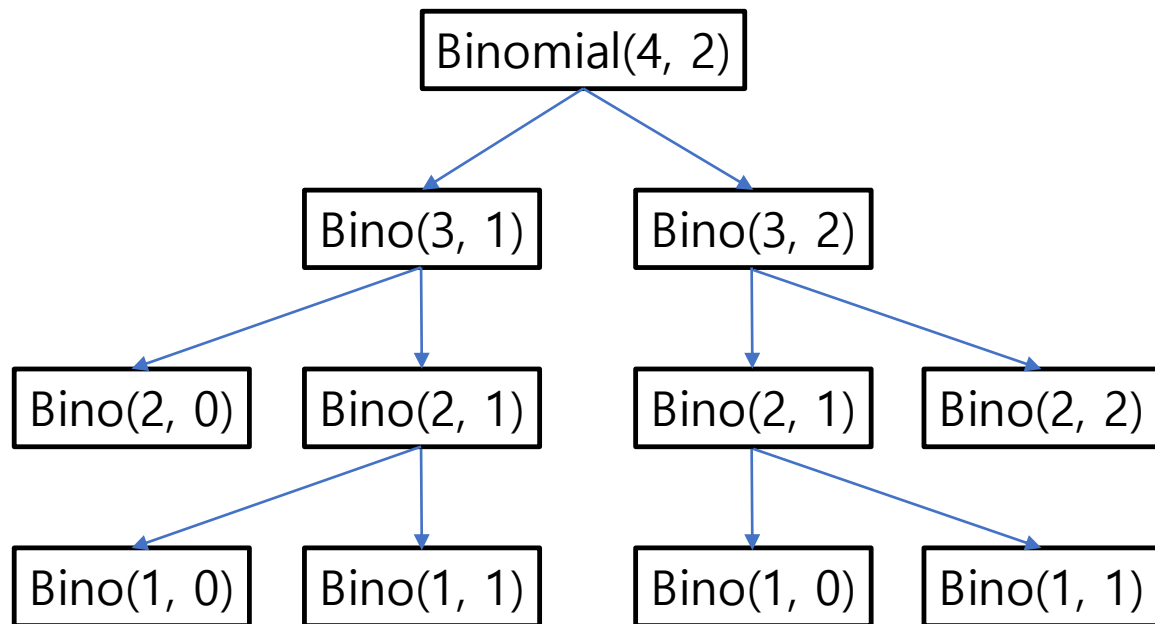
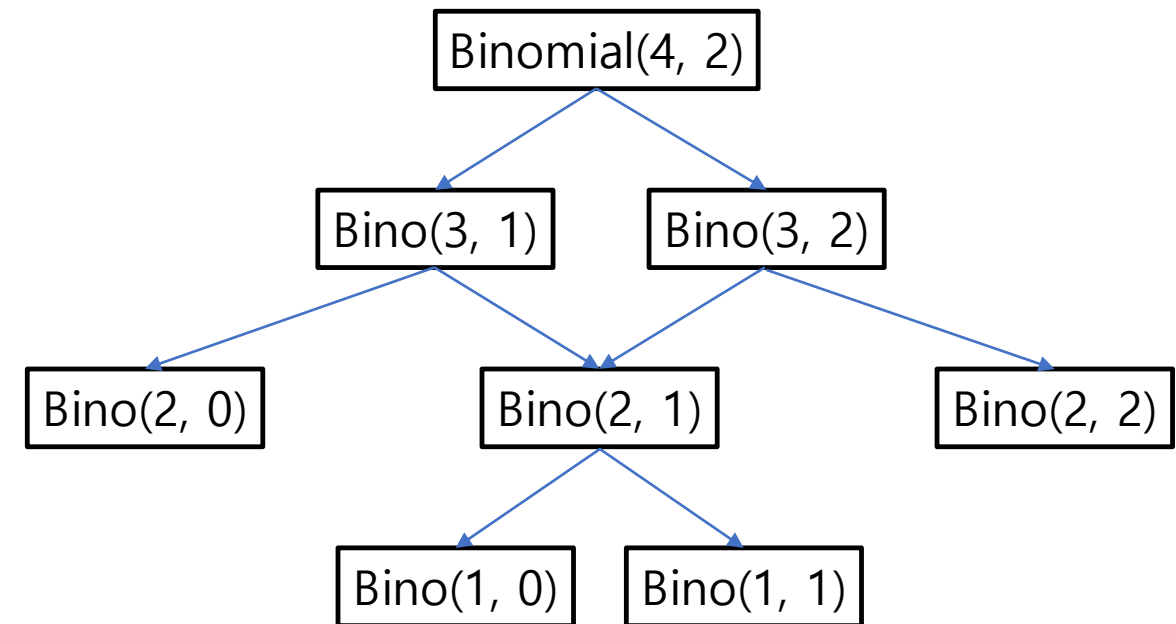
"어떤 문제를 풀기위해 그 문제를 더 작은 문제의 연장선으로 생각하고, 과거에 구한 해를 활용하는" 방식의 알고리즘을 총칭.

주로 Optimal structure라 불리는 것들이 DP를 적용해 좋은 효과를 얻을 수 있는 문제들

이름과는 달리 '다이나믹' 하고도, '프로그래밍' 하고도 큰 연관이 없음. 이광근 교수는 '기억하며 풀기' 로 적절히 번역.

분할 정복 알고리즘(Divide and Conquer)과 비슷하면서도 결정적 차이가 있다.

Unit 01 | 개념 설명

-Divide and Conquer*-DP(Memoization)*

$$\bullet \binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Unit 01 | 개념 설명

TOP-DOWN →

VS

BOTTOM-UP →

```
def fibo(n):  
    if n < 2:  
        return n  
    return fibo(n-1) + fibo(n-2)
```

```
[fibo(i) for i in range(11)]
```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

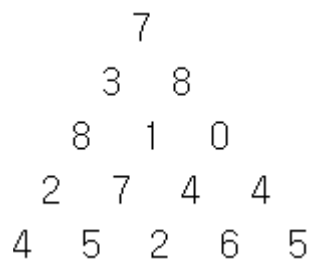
```
memo = {1:1, 2:1}  
def fibo2(n):  
    if n == 0:  
        return 0  
    if n not in memo:  
        for itr in range(3, n+1):  
            memo[itr] = fibo2(itr-2) + fibo2(itr-1)  
    return memo[n]
```

```
[fibo2(i) for i in range(11)]
```

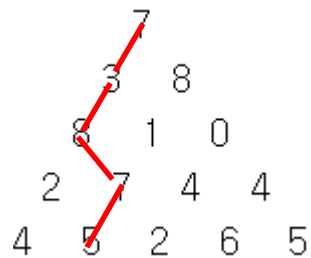
```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

Unit 02 | 일반 문제

Q-1



이 그림은 크기가 5인(5층짜리) 숫자 삼각형이다. 맨 위층의 숫자 7부터 시작해 아래에 있는 수 중 하나를 선택해, 내려올 때 선택되어 왔던 모든 수의 합이 최대가 되는 경로의 합을 구하는 알고리즘을 완성하시오.
아래층에 있는 수는 현재 선택된 수의 대각선 왼쪽 또는 대각선 오른쪽에 있는 것 중에서만 선택할 수 있다.



예제의 경우 최적 경로는 7-3-8-7-5로 그 합은 30이므로 30을 출력해야 한다.
삼각형의 크기(층 수)는 1이상 500이하이고, 삼각형을 이루는 숫자는 모두 정수이며 0이상 99이하이다.

Unit 02 | 일반 문제

Q-2

0과 1로 채워진 행렬이 있다. 행렬 1칸은 1×1 의 정사각형으로 이루어져 있다. 행렬에서 1로 이루어진 가장 큰 정사각형을 찾아 넓이를 반환하는 함수 Large를 완성하시오.

Ex)

Matrix1 $\begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix} \Rightarrow 9$

Matrix2 $\begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix} \Rightarrow 16$

Unit 02 | 일반 문제

Q-3

수열 {10, 20, 40, 25, 30, 15, 50}이 있다. 이 수열의 부분수열 중 모든 원소가 점차 증가하는 수열은 {10, 20, 40, 50}, {25, 30}, {15, 50}이다.
따라서 이 중 길이가 가장 긴 수열은 {10, 20, 40, 50}이고 길이는 4이다.
이렇듯 수열에서 길이가 가장 긴 증가 부분수열의 길이를 반환하는 알고리즘을 완성하시오.

Unit 03 | 심화 문제

Q-4

길이가 서로 다른 젓가락쌍들이 있다고 하자. 만약 옆에서 길이가 맞는 젓가락쌍을 찾는다면 길이가 맞는 젓가락들의 원래의 짝이 하나의 쌍을 이루고 길이가 맞는 젓가락쌍은 짝 곳에 보낸다고 하자. 이 때의 배송비는 새로 만들어진 젓가락쌍과 보내는 젓가락 중 하나의 길이의 곱(원)이다. 우리는 배송비를 최소화하면서 주어진 젓가락을 배송하고자 한다. 예를 들어 네 쌍의 젓가락 A, B, C, D 가 주어진다고 하고 각각의 길이를 $A: 10, 20, B: 20, 5, C: 5, 30, D: 30, 15$ 라고 할 때, 가능한 경우는:

$$* (A \cdot ((B \cdot C) \cdot D)) \Rightarrow (20 \times 5 \times 30) + (20 \times 30 \times 15) + (10 \times 20 \times 15) = 15000$$

$$* (A \cdot (B \cdot (C \cdot D))) \Rightarrow (5 \times 30 \times 15) + (20 \times 5 \times 15) + (10 \times 20 \times 15) = 6750$$

$$* ((A \cdot B)(C \cdot D)) \Rightarrow (10 \times 20 \times 5) + (5 \times 30 \times 15) + (10 \times 5 \times 15) = 4000$$

$$* (((A \cdot B) \cdot C) \cdot D) \Rightarrow (10 \times 20 \times 5) + (10 \times 5 \times 30) + (10 \times 30 \times 15) = 7000$$

$$* ((A \cdot (B \cdot C)) \cdot D) \Rightarrow (20 \times 5 \times 30) + (10 \times 20 \times 30) + (10 \times 30 \times 15) = 13500$$

여기서 $X \cdot Y$ 는 X 와 Y 를 이용해서 새로운 젓가락 쌍을 만든다는 의미이다. 위와 같은 순서로 배송할 수 있고, 이 때의 최소 배송비는 4000원이다. INPUT으로 젓가락 길이들을 받아서 OUTPUT으로 최소의 배송비를 반환하는 함수 chopchop을 구현하라. 예를 들어, 위와 같은 경우에는 INPUT으로 (10, 20, 5, 30, 15)를 받고 이는 젓가락쌍이 (10, 20), (20, 5), (5, 30), (30, 15)이 있음을 의미한다. 또한 세 번째 경우가 최소 배송비이므로 4000을 반환한다.

Unit 03 | 심화 문제

Q-5

A국의 동전 단위는 1, 3, 5원으로 이루어져 있다고 하자. 8원을 지급하는 방법은 $5 + 1 + 1 + 1$, $5 + 3$, $3 + 3 + 1 + 1$, $3 + 1 + 1 + 1 + 1 + 1$, $1 + 1 + 1 + 1 + 1 + 1 + 1 + 1$ 의 다섯 가지가 있다.

다시 말해 입력이 $N=8$, $S=\{1, 3, 5\}$ 라면 5를 출력해야 한다.
이렇게 지급 방법의 경우의 수를 계산하는 알고리즘을 구현하라.

Q & A

풀어주셔서 감사합니다.